Faculty of Power and Aeronautical Engineering

Department of Automatic control & Robotics

Multi-body System Project

Problem 4

Prepared by: Beidemaryam Awoke Bishaw

Submitted to – Prof. Janusz Frączek

02 January 2023

Table of Contents

List of Figure

List of Table

1. Introduction

In this project, I aimed to analyse the motion of a multi-body system consisting of 10 bodies connected by 12 revolute and 2 prismatic joints. By utilizing MATLAB and ADAMS software, I was able to simulate and compute the position, velocity, and acceleration of each joint and the centre points of each body in the system. The results of this study provide valuable insight into the kinematics of the given multi-body systems. Finally, I compared the results that I obtained from ADAMS software and MATLAB code.

2. Theoretical Background

In this project I analysed the kinematics of a given multi body system by using absolute coordinate system by taking it as a planar system. Here problem will be formulated by using different kinematic constraint equations. That means the position of the system will be expressed in a holonomic constraint equations imposed on the coordinates. And then velocity and acceleration of the system will be obtained by derivate these constraint equations.

2.1. Constraints

In the given system there are a sort of bodies which are connected to each other by one or more kinematics pair or joints. This joints in the mechanism can be described in absolute coordinate system as a set of equations. In this project I formulated a set of different kinematic constraint equations based on the type of joint. As I mentioned above in the given system there are only two kinds of joints which are prismatic and revolute joints.

Revolute joint (kinematic pair of the 5th class)

From Lecture 5, we obtained a constraint equation which govern a revolute joint. The equation obtained by taking two points that are point A from body ‘i’ and point B from body ‘j’.

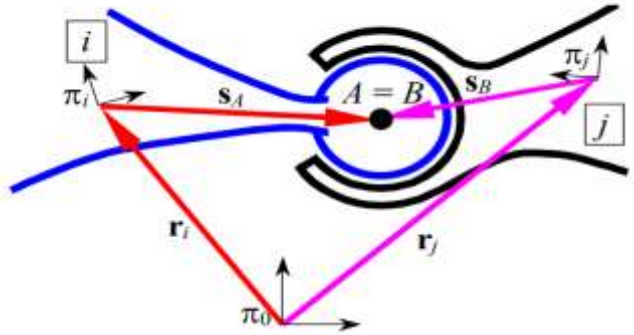

Figure 1 Revolute joint (kinematic pair of the 5th class)

$$\emptyset^{K^\circ} = r_i + R_i * {s_A}^{(i)} - r_j + R_j * {s_B}^{(j)} \quad [1]$$

Translational joint (kinematic pair of the 5th class)

Translational joint eliminates two degrees of freedom, thus it is described by two scalar constraint equations. The first equation expresses the fact that relative orientation of the frame π_J with respect to the frame π_i remains constant:

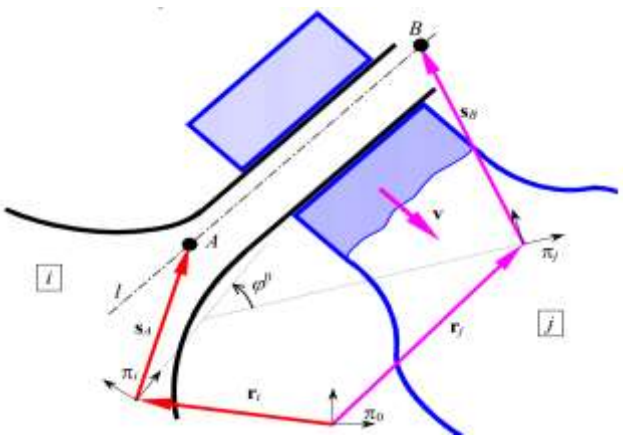

Figure 2 Translational joint (kinematic pair of the 5th class)

$$\emptyset^{K<} \equiv \varphi_i - \varphi_j - \varphi^0 = 0 \quad [2]$$

The second constraint equation represents due to the fact that point B can move with respect to point A only along the axis *l* and there is a vector perpendicular to the length *l*.

$$\emptyset^{K\uparrow} \equiv \left(R_j v^{(j)}\right)^T (r_j - r_i - R_i * {s_A}^{(i)} + \left(v^{(j)}\right)^T {s_B}^{(j)} = 0 \quad [3]$$

The last equation is imposed by a translational joint on the relative motion of two bodies.

$$\emptyset^{K\uparrow} \equiv \left(R_j u^{(j)}\right)^T \left(r_j + R_j * {s_B}^{(j)} - r_i - R_i * {s_A}^{(i)}\right) - f_{AB}(t) = 0 \quad [4]$$

2.2. Jacobian Matrix

The Jacobian matrix in a multi-body system is used to relate the velocities of a point in the system to the velocities of the individual bodies that make up the system. This matrix is important because it allows us to understand how the motion of one part of the system is affected by the motion of other parts of the system (e.g. joints and links in the arm). It is defined as the matrix of partial derivatives of the position vector of the point with respect to the generalized coordinates of the system.

Kinematic constraints for revolute joint

The partial derivative of $\emptyset^{K^\circ}$ which is given in equation [1] with respect of vector of absolute coordinates $q_i = [r_i{}^T + \varphi_i]^T$ $q_j = [r_j{}^T + \varphi_j]^T$ can be written in the matrix form as:

$$\emptyset^{K^\circ}{}_{r_i} = I_{2X2} \quad \text{.....[5]}$$

$$\emptyset^{K^\circ}{}_{\varphi_i} = \Omega R_i s_A{}^{(i)} \quad \text{.....[6]}$$

$$\emptyset^{K^\circ}{}_{r_j} = -I_{2X2} \quad \text{.....[7]}$$

$$\emptyset^{K^\circ}{}_{\varphi_j} = -\Omega R_i s_B{}^{(i)} \quad \text{.....[8]}$$

Kinematic constraints for translational and revolute-translational joints

Potentially non-zero elements of the Jacobian matrix for the constraints $\emptyset^{K<}$ which is given by equation [2].

$$\emptyset^{K<}{}_{r_i} = 0_{2X2} \quad \text{..... [9]}$$

$$\emptyset^{K<}{}_{\varphi_i} = 1 \quad \text{..... [10]}$$

$$\emptyset^{K<}{}_{r_j} = 0_{2X2} \quad \text{..... [11]}$$

$$\emptyset^{K<}{}_{\varphi_j} = -1 \quad \text{..... [12]}$$

Likewise, potentially non-zero elements of the Jacobian matrix for the constraints $\emptyset^{K\uparrow}$ which is given by equation [3].

$$\emptyset^{K\uparrow}{}_{r_i} = -\left(R_j v^{(j)}\right)^T \quad [13]$$

$$\emptyset^{K\uparrow}{}_{\varphi_i} = \left(R_j v^{(j)}\right)^T \Omega R_i s_A{}^{(i)} \quad [14]$$

$$\emptyset^{K\uparrow}{}_{r_j} = \left(R_j v^{(j)}\right)^T \quad [15]$$

$$\emptyset^{K\uparrow}{}_{\varphi_j} = -\left(R_j v^{(j)}\right)^T \Omega(r_j - r_i - R_i * s_A{}^{(i)} \quad [16]$$

Driving constraints for relative translations of points

$$\emptyset^{D^\circ}{}_{r_i} = -R_i{}^T \quad [17]$$

$$\emptyset^{D^\circ}{}_{\varphi_i} = -R_i{}^T \Omega(r_j + R_j * s_B{}^{(j)} - r_i) \quad [18]$$

$$\emptyset^{D^\circ}{}_{r_j} = R_i{}^T \quad [19]$$

$$\emptyset^{D^\circ}{}_{\varphi_j} = R_i{}^T \Omega R_j * s_B{}^{(j)} \quad [20]$$

2.3. Newton-Raphson method

The Newton-Raphson method is a numerical method used to find the roots of a nonlinear equation. It is based on the idea that a nonlinear function can be approximated by a linear function using the tangent line at a specific point on the function. In a multi-body system, the Newton-Raphson method can be used to solve the equations of motion for each body in the system. The method starts with an initial guess for the positions and velocities of each body, and then iteratively updates the guess until the equations of motion are satisfied for all bodies in the system.

2.4. Kinematics of the System

The kinematics of a multi-body system is the study of the motion of the individual bodies within the system, without considering the forces or torques that cause the motion. In other words, it is the study of the position, velocity, and acceleration of the bodies in the system as a function of time. In a multi-body system, each body is represented by a reference frame, and

the position and orientation of the body with respect to a fixed reference frame is described by a set of coordinates. The motion of the body can be described in terms of the relative motion between the body and the fixed reference frame. Since the given system is subject to constraints or non-holonomic conditions. Numerical methods and computer simulations are used to study the kinematics of the system. Some common techniques used to solve position problems in multi-body systems like the Newton-Raphson method.

2.4.1. Position

The position of a multi-body system refers to the location and orientation of each individual body within the system relative to a fixed reference point. This can be described using various coordinate systems and can be represented mathematically using vectors and matrices.

2.4.2. Velocity

The velocity of a body in a multi-body system with constraints is determined by its position and the constraints that restrict its motion. The velocity of a body is also related to the time derivatives of the coordinates that define its position, such as the velocities of the joints in a mechanism. The Newton-Raphson method can also be used to solve for the velocities of the bodies in a multi-body system with constraints.

$$\Phi_q \dot{q} = -\Phi_t = \begin{bmatrix} 0_{mx1} \\ -\Phi^D \end{bmatrix}_{3nx1} \quad \text{[21]}$$

2.4.3. Acceleration

In a multi-body system with constraints, the acceleration of each body is still determined by taking the derivative of its velocity vector with respect to time, but the constraints introduce additional relationships between the bodies that must be taken into account. The constraints can be represented mathematically as equations, which can be used to solve for the accelerations of the individual bodies in the system.

$$\Phi_q \ddot{q} = \Gamma = \begin{bmatrix} \Gamma^K \\ \Gamma^D \end{bmatrix} = \begin{bmatrix} -(\Phi_q^K \dot{q})_q \dot{q} \\ -(\Phi_q^D \dot{q})_q q - 2\Phi_{tq}^{\dot{D}} - \Phi_{tt}^D \end{bmatrix}_{3nx1} \quad \text{[22]}$$

Γ is different for different types of joints

$$\Gamma_{2x1}^{K.} = R_i * s_A{}^{(i)} \dot{\varphi}_i^2 - R_j * s_B{}^{(j)} \dot{\varphi}_j^2 \quad \text{[23]}$$

$$\Gamma_{1x1}^{K<} = 0 \quad \text{[24]}$$

$$\Gamma_{1x1}^{K\uparrow} = \left(R_j v^{(j)}\right)^T \left(2\,\Omega(\dot{r}_j - \dot{r}_i)\dot{\varphi}_j^2 - R_i * s_A{}^{(i)} \left(\dot{\varphi}_j - \dot{\varphi}_i\right)^2\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots [25]$$

$$\Gamma_{1x1}^{D^\circ} = (R_i)^T \left(2\,\Omega(\dot{r}_j - \dot{r}_i)\dot{\varphi}_i + (\dot{r}_j - \dot{r}_i)\dot{\varphi}_i^2 - R_j * s_B{}^{(j)} \left(\dot{\varphi}_j - \dot{\varphi}_i\right)^2\right) + (f_{AB})_{tt} \ldots\ldots\ldots\ldots\ldots\ldots\ldots [26]$$

2.5. Singularity

The singularity in a multi body system can be checked using the Jacobian matrix. There are several ways to check for a singularity using the Jacobian matrix, including:

1. Checking the rank of the Jacobian. If the rank of the Jacobian reduces by one or more, this may indicate a singularity.
2. Checking the determinant of the Jacobian approximation. If the determinant of the Jacobian approximation equals zero, this may indicate a singularity.
3. Checking the condition number of the Jacobian matrix. If the condition number of the Jacobian matrix increases, this may indicate a singularity.

$$\text{Rank}(\Phi_q(q,t)) = N = 3n = 3 * 10 = 30 \ldots\ldots\ldots\ldots\ldots\ldots\ldots [27]$$

3. Methodology

In this project I computed the position, velocity and acceleration of a given multi body system by using ADAMS software and MATLAB programming language. The given multi body system is shown in the figure below.

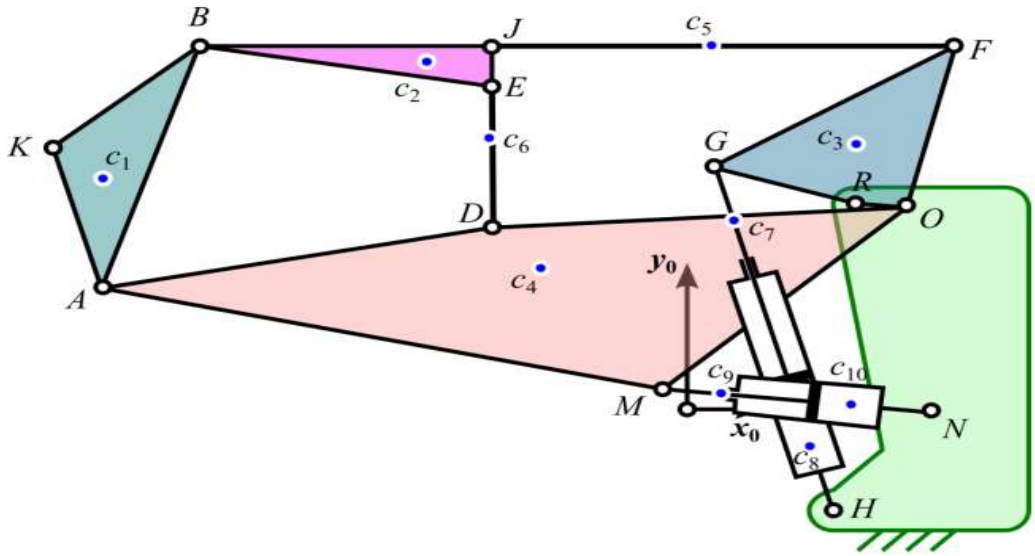

Figure 1 The given multi-body system

Table 1Global coordinates of characteristic points (initial configuration)

	A	B	D	E	F	G	H	J	K	M	N	O	R
X[m]	-1.2	-1.1	-0.4	-0.4	0.6	0.1	0.3	-0.4	-1.3	-0.1	0.5	0.5	0.4
Y[m]	0.3	0.9	0.5	0.8	0.9	0.6	-0.2	0.9	0.7	0.2	0	0.5	0.5

Table 2 Global coordinates of centres of mass (initial configuration)

	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10
X[m]	-1.2	-0.7	0.4	-0.35	0.1	-0.4	0.15	0.25	0.05	0.35
Y[m]	0.6	0.85	0.65	0.4	0.9	0.65	0.4	0	0.15	0.05

3.1. ADAMS Software

ADAMS (Automated Dynamic Analysis of Mechanical Systems) is a multi-disciplinary simulation software that can be used to model and analyse the dynamics of mechanical systems. Based on the given information listed above in Table 1 and Table 2, the multi-body system was modelled in ADAMS STUDENT 2022.

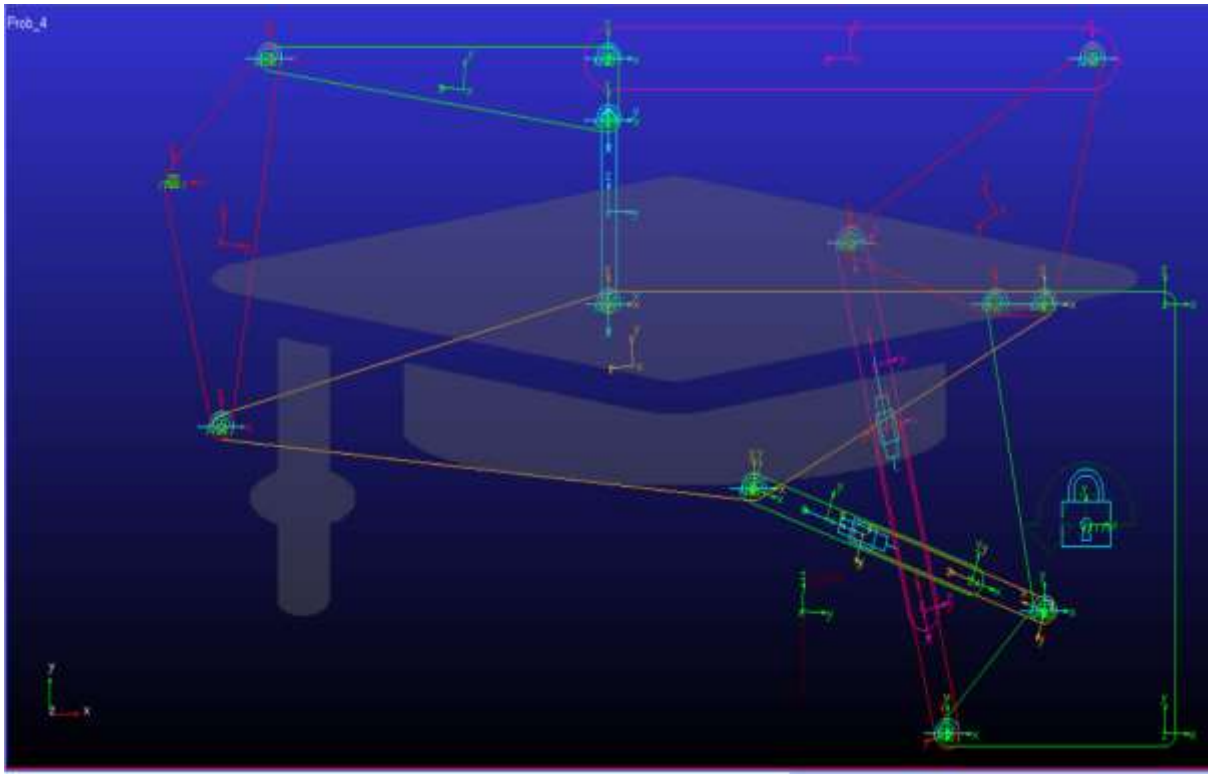

Figure 2 ADAMS model of the system

Steps

1. Open ADAMS software and create a new model.
2. Select the "Multi-Body" option from the main menu and choose "Create Multi-Body System."
3. Click on the "Add Body" button and select "Point."
4. Enter the coordinates of the first point in the system, and click "OK" to add it to the model.
5. Repeat steps 3 and 4 for each additional point in the system.
6. Connect the points by using the "Connect" button and selecting the appropriate joints (e.g. revolute, prismatic, ground etc.).
7. Assign driving constraints by selecting the motion button.
8. Use the "Simulation" option to run simulations and assign the simulation parameters like step size and duration of simulation to analyse the behaviour of the multi-body system.

Sample result from ADAMS for centre point 1 is given below in the figure:-

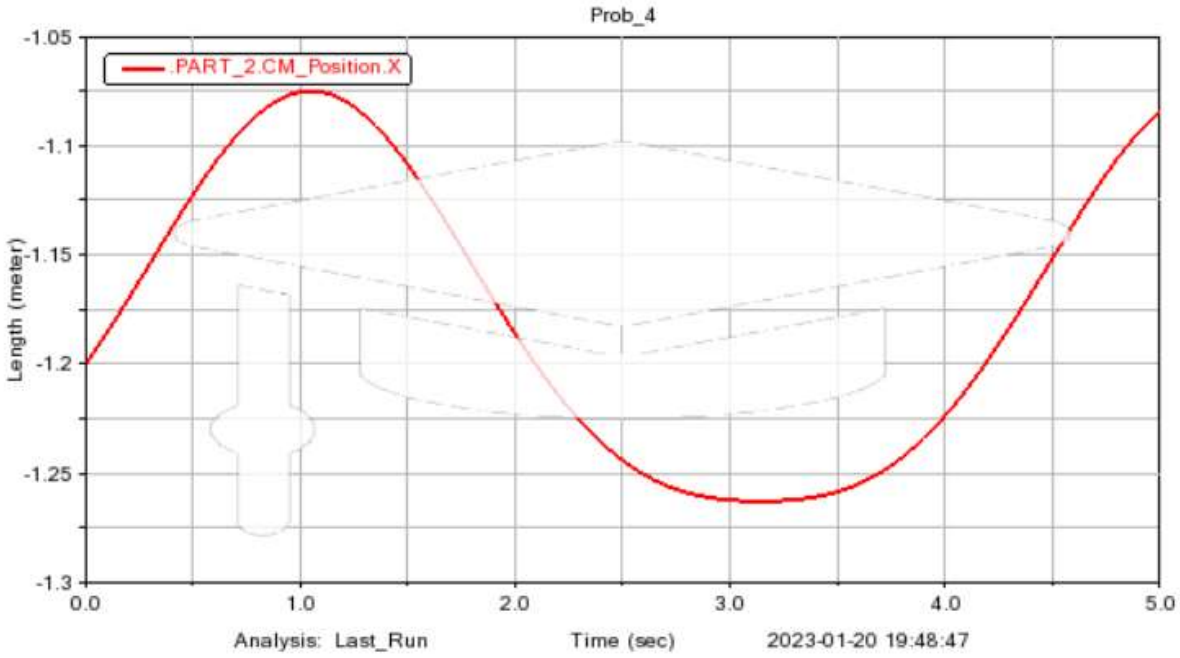

Figure 3 Sample displacement-time graph along x for Centre point 1

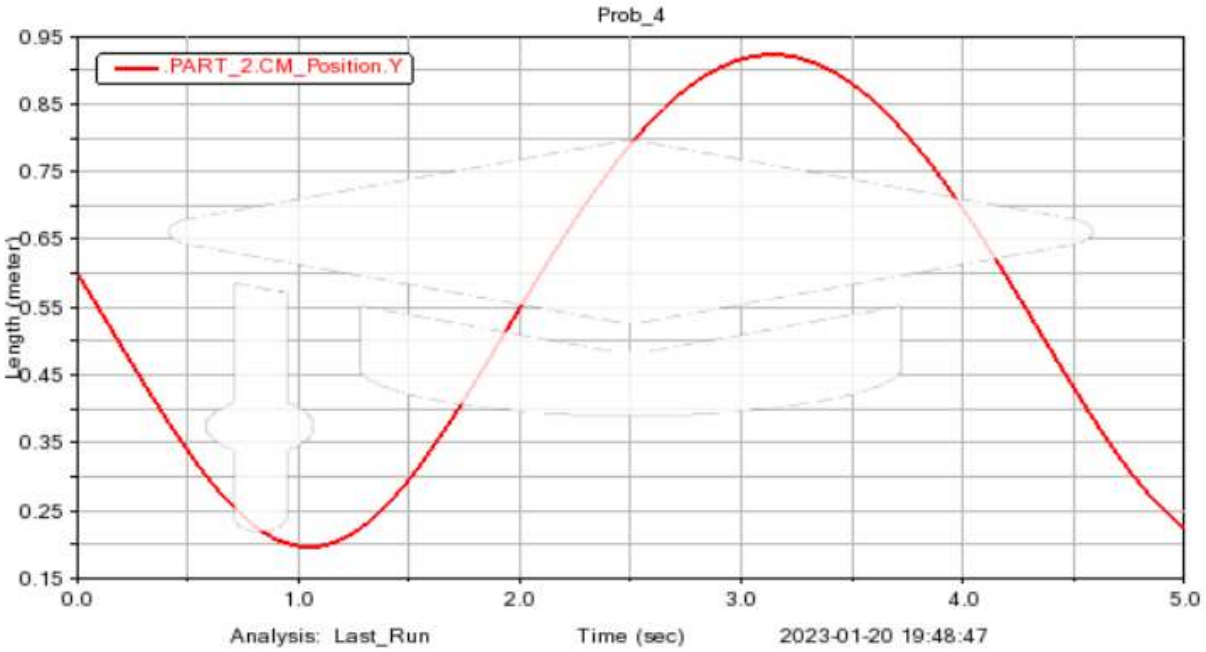

Figure 4 Sample displacement-time graph along y for Centre point 1

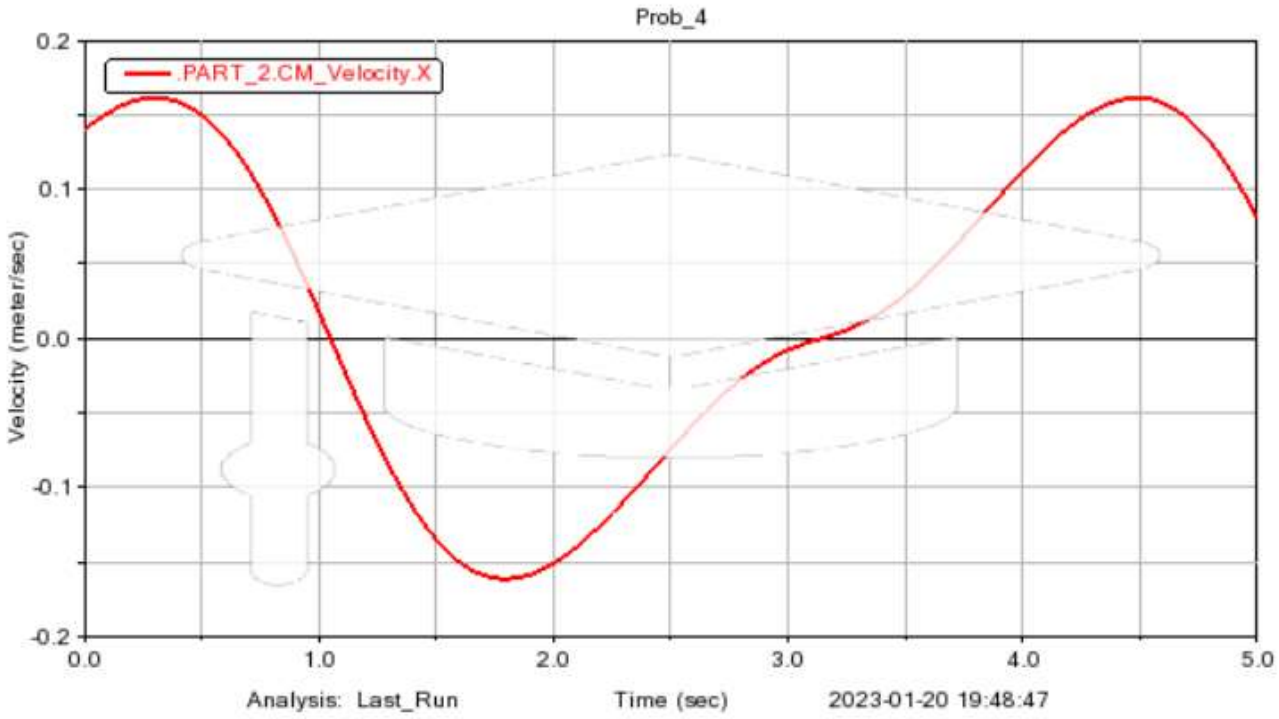

Figure 5 Sample velocity-time graph along x for Centre point 1

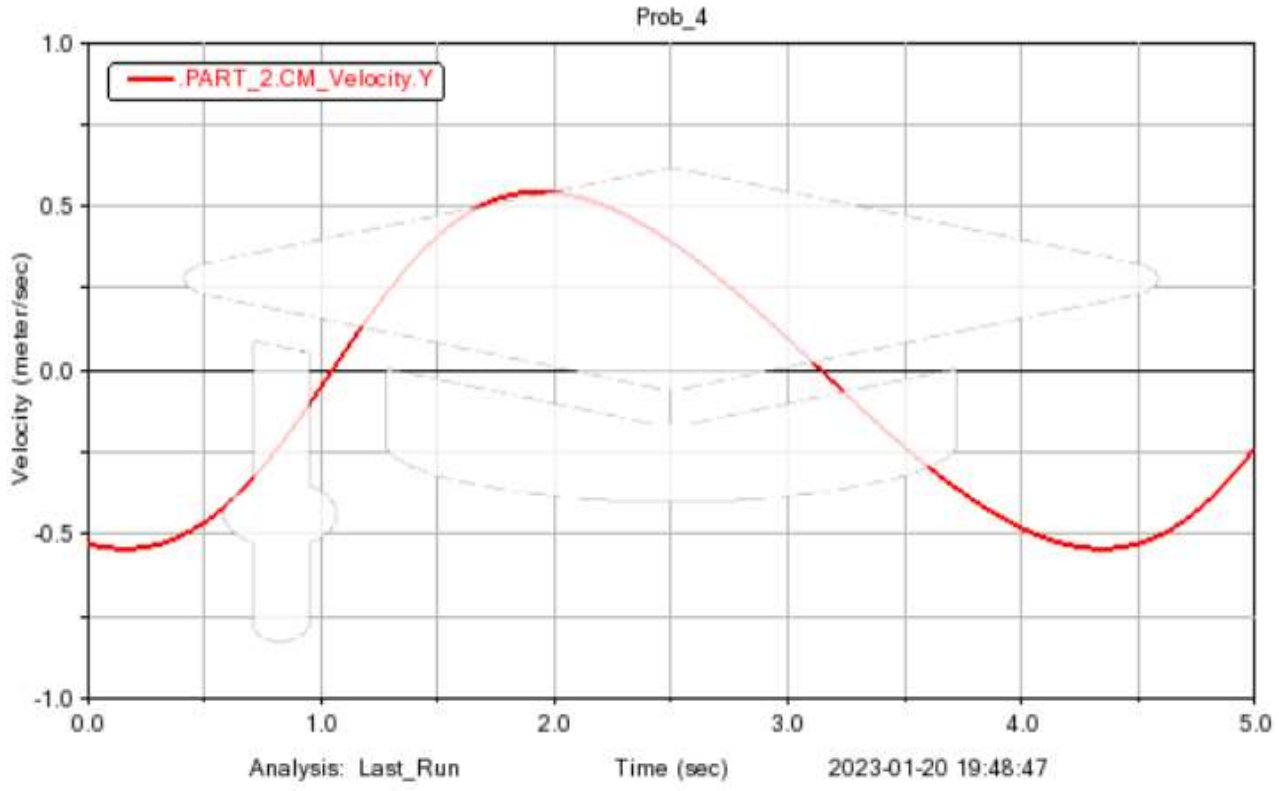

Figure 6 Sample velocity-time graph along y for Centre point 1

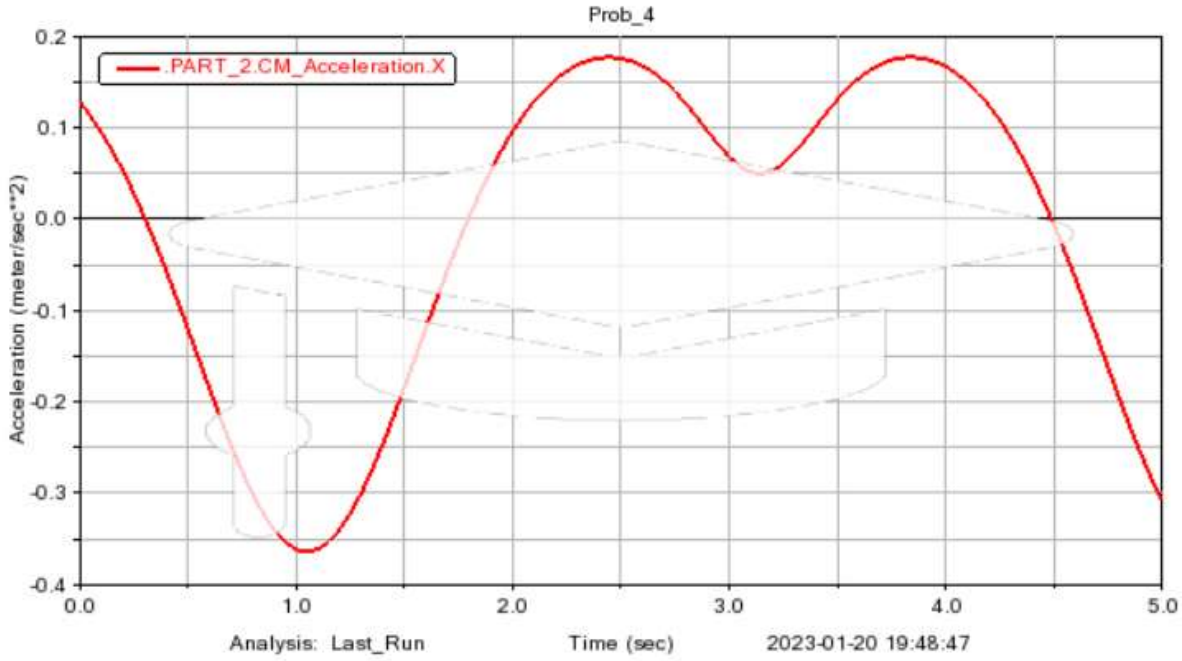

Figure 7 Sample acceleration-time along x for Centre point 1

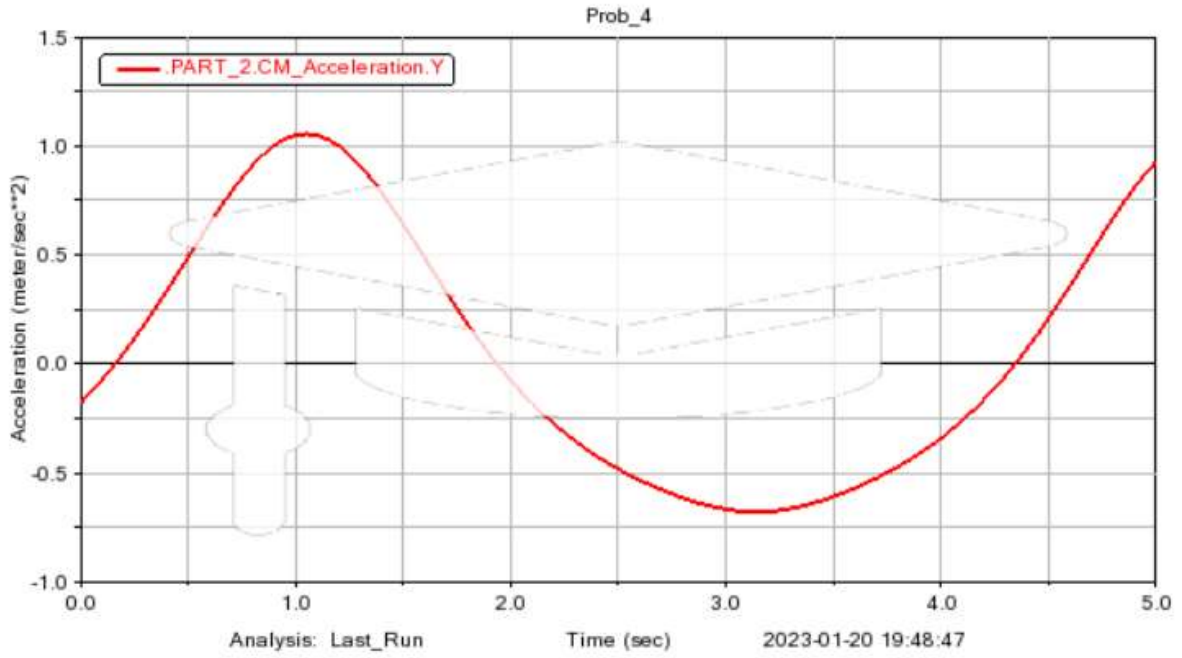

Figure 8 Sample acceleration-time along y for Centre point 1

And the position, velocity and acceleration results from ADAMS will be compared with MATLAB result. Here for the time being I showed only the ADAMS result. In the next section they will be compared.

3.2. MATLAB

MATLAB (short for "Matrix Laboratory") is a numerical computing environment and programming language developed by MathWorks. It is used for data analysis, algorithm development, modelling, simulation, and prototyping, as well as for scientific and engineering graphics.

Steps

1. I started by building the program with the Rot procedure, because it is the lowest in the hierarchy (no refer to subsequent procedures). The Rot procedure has one input argument – angle φ; as a result of its operation, the matrix R (φ) is calculated.
2. The next step is to write a constraint function that calculates non- linear constraint equations according to the formulas in equation 1 and 2. The function will take two arguments which are absolute coordinate q and time t and return a vector F.
3. Jacobian function will compute the constraint Jacobian. This function take the vector q as an input and return vector Fq as an output. This function then used by Newton-Rapson method to update the approximation of vector q.
4. Then I built a Newton-Rapson function which takes q(initial guss) and t as an input. This function use constraints and Jacobian function to iteratively update the approximation of the solution until it satisfies the nonlinear equation within a certain tolerance or until the maximum number of iteration is reached. Beside it also shows as whether the Jacobian matrix is singular or not.
5. Then I created a velocity function which used to compute the time derivative of absolute coordinate of the system. The function takes the absolute coordinate vector (q) and time (t) as an input. And it has only one output which is the time derivative of absolute coordinate (dq).
6. Next, I created an acceleration function which used to compute the second time derivative of absolute coordinate of the mechanical system. This function takes the vector of time derivatives of absolute coordinates (dq), the vector of absolute coordinates (q)and the current time instant(t) as input.
7. Eventually, I created the main function which used to plot position, velocity and acceleration results by taking the vector of second time derivatives of absolute coordinates (d2q), the vector of time derivatives of absolute coordinates (dq),the vector of absolute coordinates (q) and the current time instant(t) as input.

4. Results

In your project, I used both MATLAB and ADAMS to study the kinematics of a planar multi-body system and obtained approximate results from both software. To evaluate the difference between the results, I compared them graphically and numerically, using the Root Mean Square Error (RMS) as a measure of numerical comparison.

The RMS error is a commonly used measure of the difference between two sets of data, and it gives an indication of how similar the two sets of results are. By comparing the RMS error between the results from MATLAB and ADAMS, I was able to quantify the difference between the two sets of results.

$$RMSE = \sqrt{(\frac{1}{N}(\sum_{i=0}^{t} (Resuslt_{Adam}^{i} - Resuslt_{Matlab}^{i})^2)}$$

Table 3 Root Mean Square Error between ADAMS and MATLAB Result

	PossitionX	Position Y	Velocity X	Velocity Y	Acceleration X	Acceleration Y
c1	3.026e-07	2.861e-08	2.2646e-08	2.8224e-08	2.6456e-08	5.3706e- 08
c2	2.834e-08	1.500e-07	2.6590e-08	2.4895e-08	2.6101e-08	2.6023e-09
c3	2.939e-08	2.891e-08	2.7607e-09	2.3427e-09	1.6868e-08	2.5994e-08
c4	2.736e-08	3.019e-08	2.5282e-09	2.5967e-08	7.8706e-09	2.5874e-08
c5	2.235e-08	2.815e-08	2.8706e-08	2.3528e-08	2.4746e-08	2.8298e-08
c6	2.975e-08	2.583e-08	2.3326e-08	2.7264e-08	2.3269e-08	2.7851e-08
c7	2.677e-08	2.738e-08	2.1782e-09	2.3412e-08	9.7747e-09	2.5284e-08
c8	3.080e-08	2.626e-10	2.3520e-09	2.7756e-10	2.5905e-09	2.9554e-10
c9	1.393e-08	2.046e-08	1.0399e-08	2.1288e-08	2.1323e-08	2.5796e-08
c10	2.754e-08	2.855e-09	1.8918e-09	2.6888e-09	2.6172e-09	9.1867e-09

Graphical comparison of results can also be useful in understanding the differences between the results obtained from MATLAB and ADAMS. By visualizing the results, I can quickly identify any differences in the shape or magnitude of the results. This can help me to understand the underlying causes of the differences between the results and to identify any potential sources of error in the simulation.

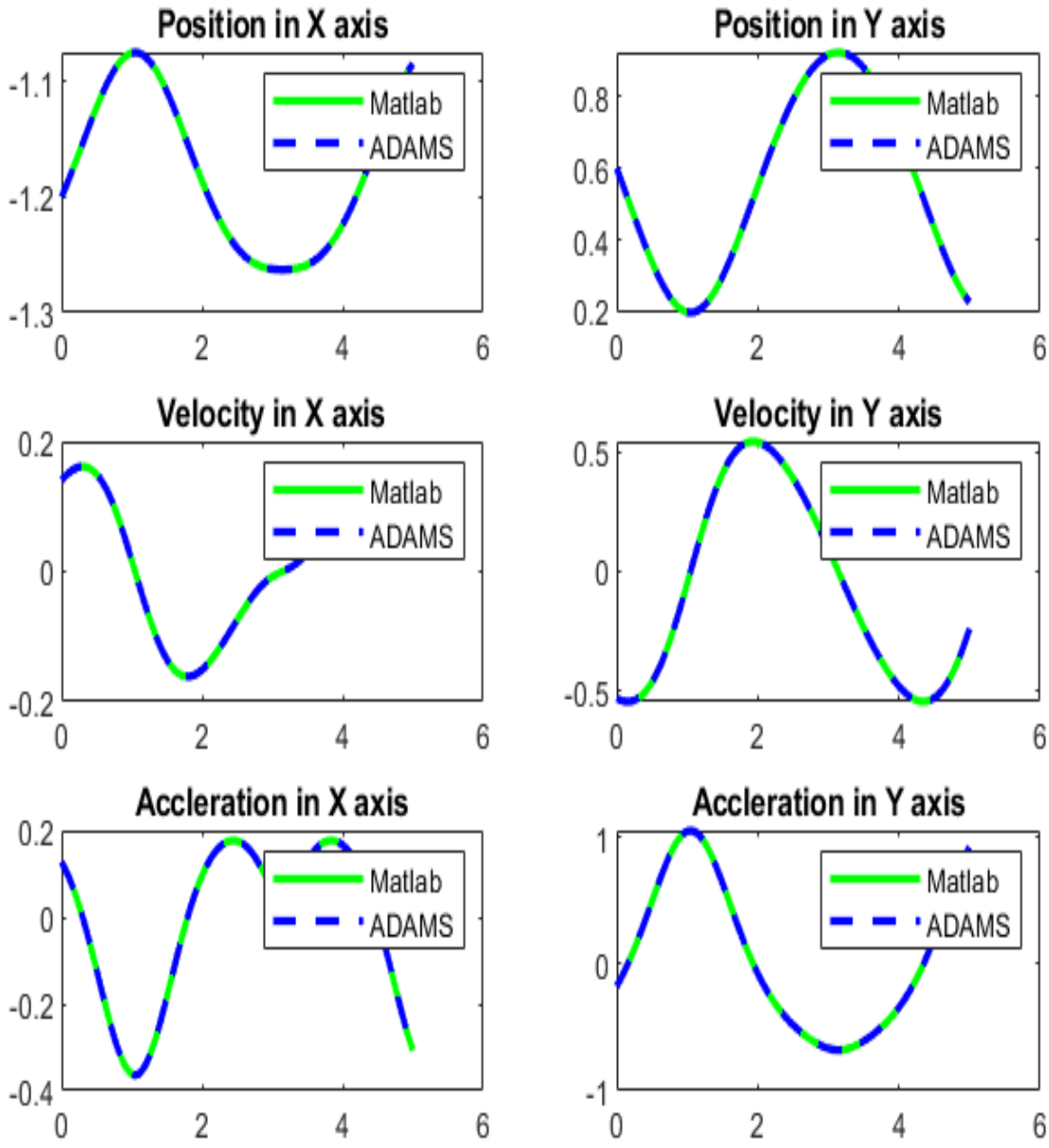

Figure 9 Comparison of ADAMS and MATLAB for Centre point 1

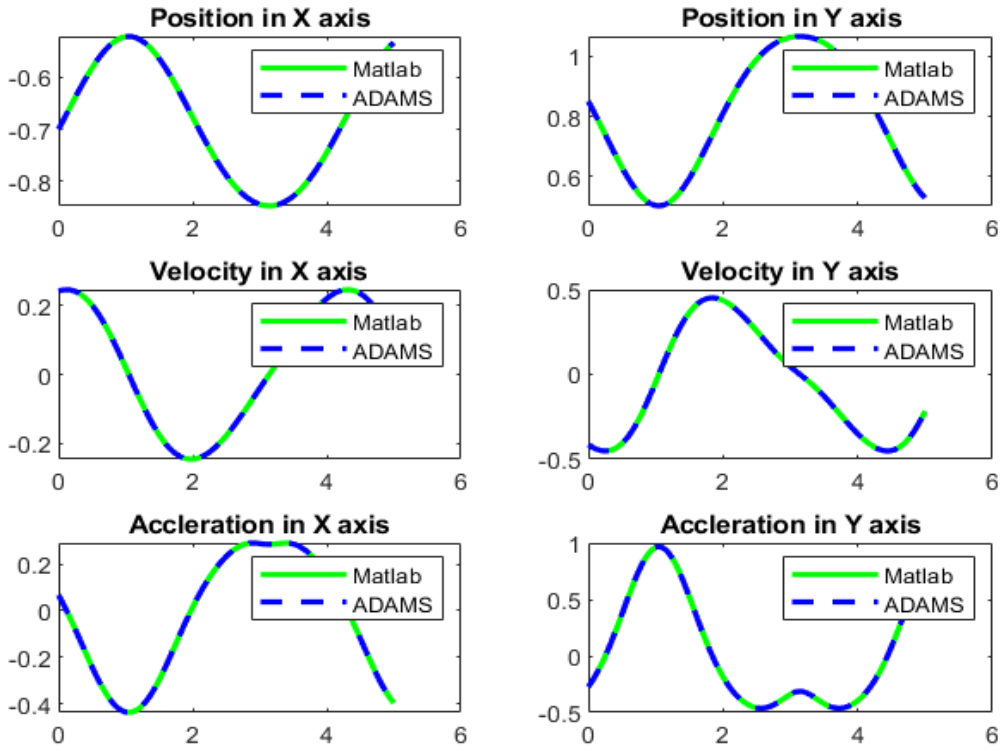

Figure 10 Comparison of ADAMS and MATLAB for Centre point 2

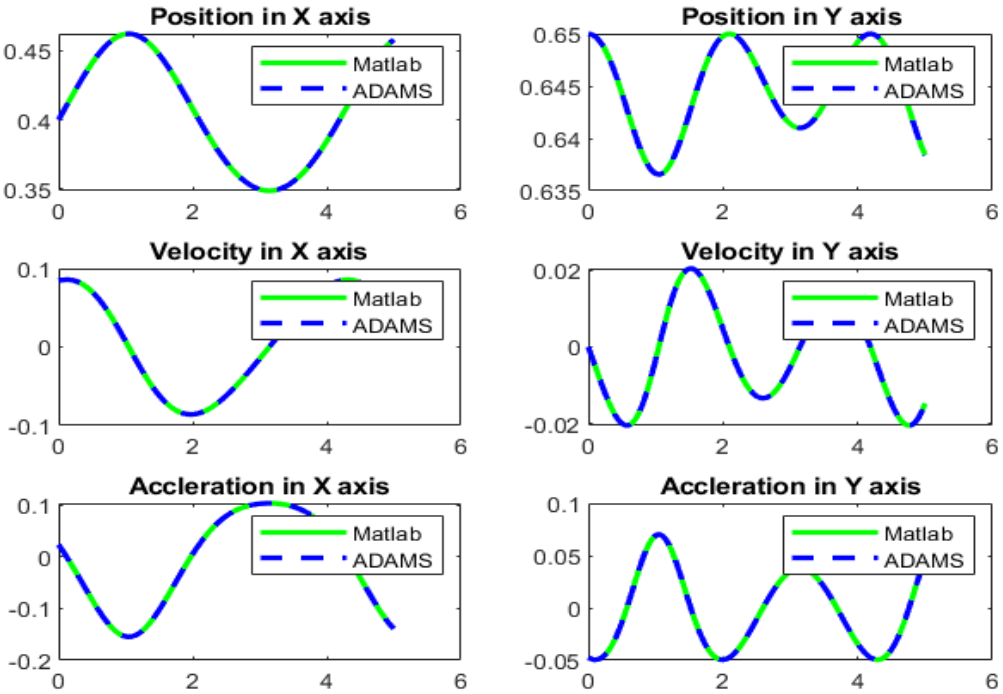

Figure 11 Comparison of ADAMS and MATLAB for Centre point 3

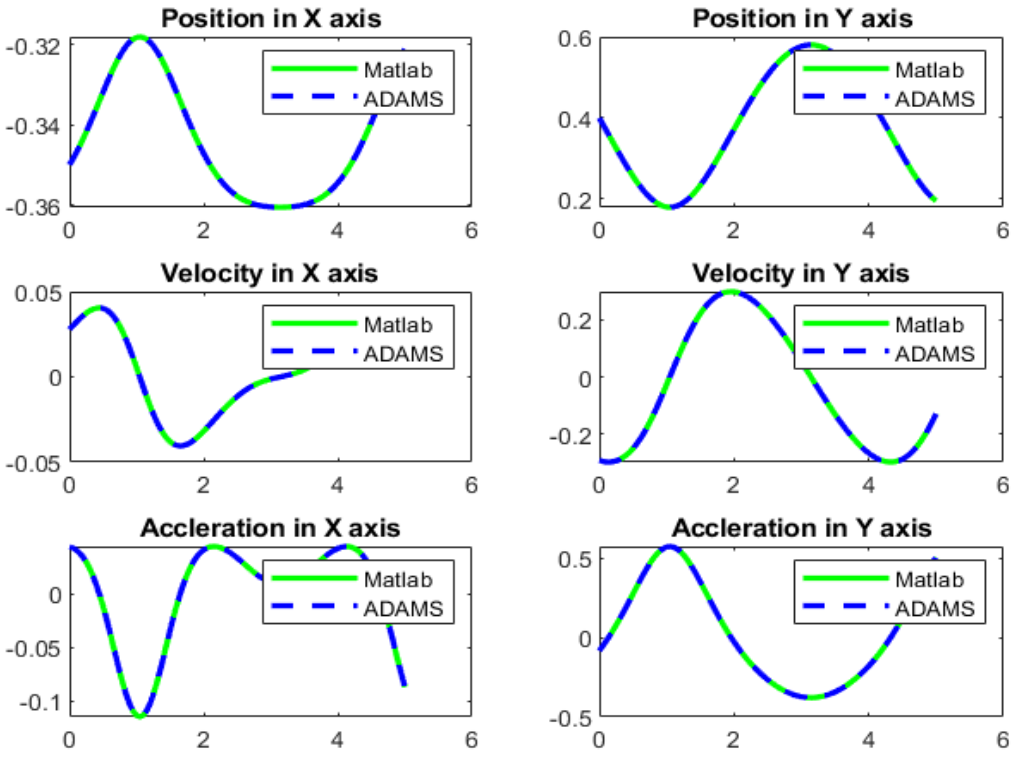

Figure 12 Comparison of ADAMS and MATLAB for Centre point 4

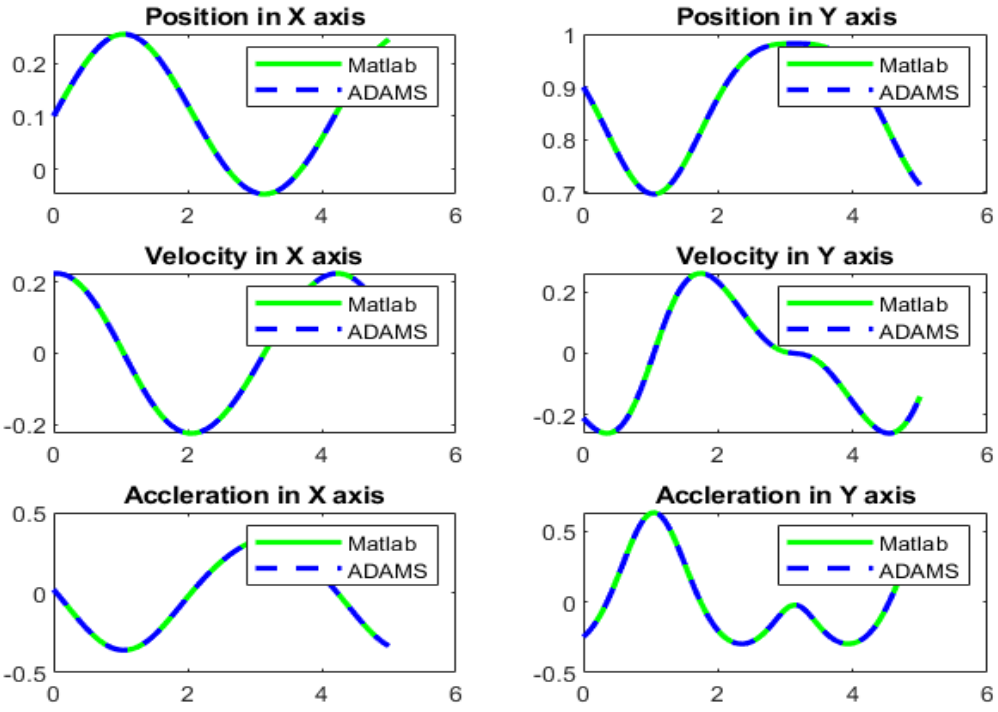

Figure 13 Comparison of ADAMS and MATLAB for Centre point 5

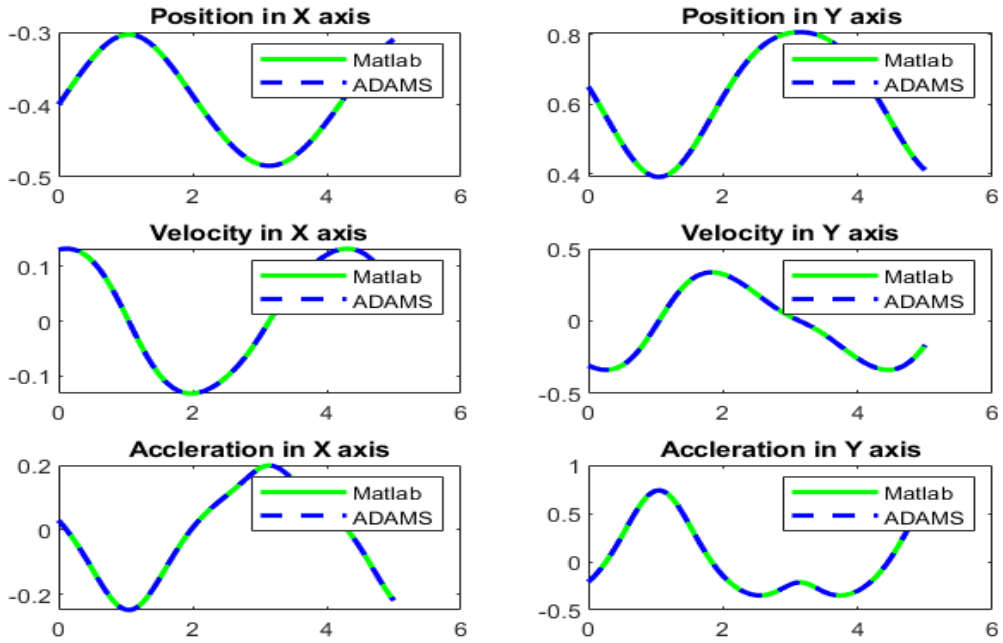

Figure 14 Comparison of ADAMS and MATLAB for Centre point 6

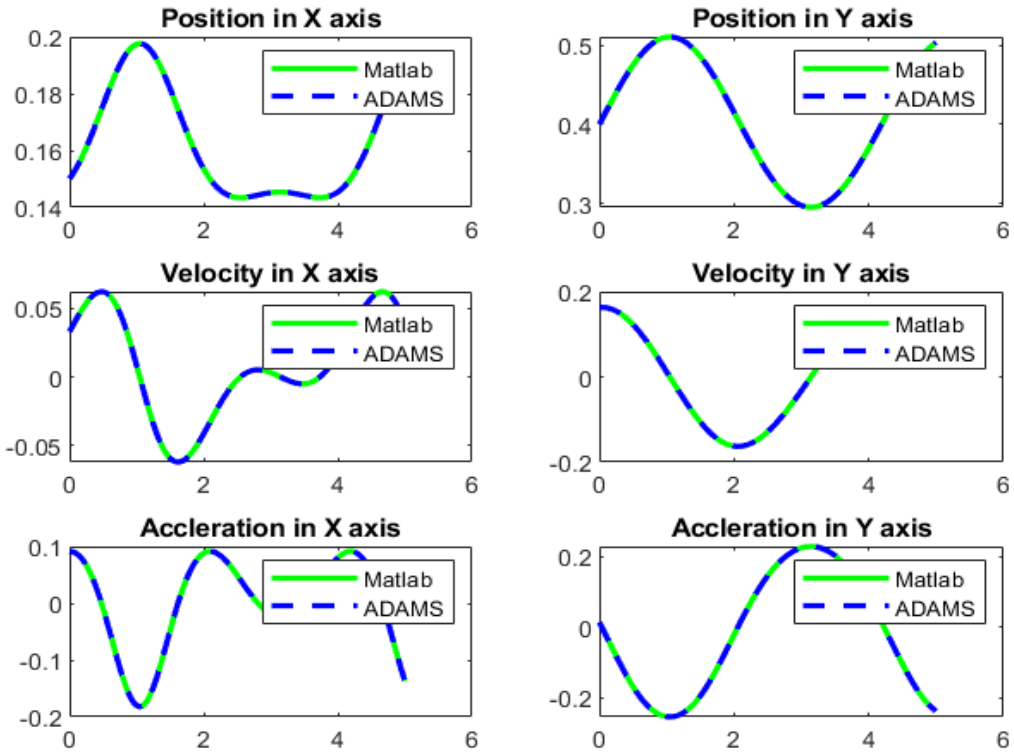

Figure 15 Comparison of ADAMS and MATLAB for Centre point 7

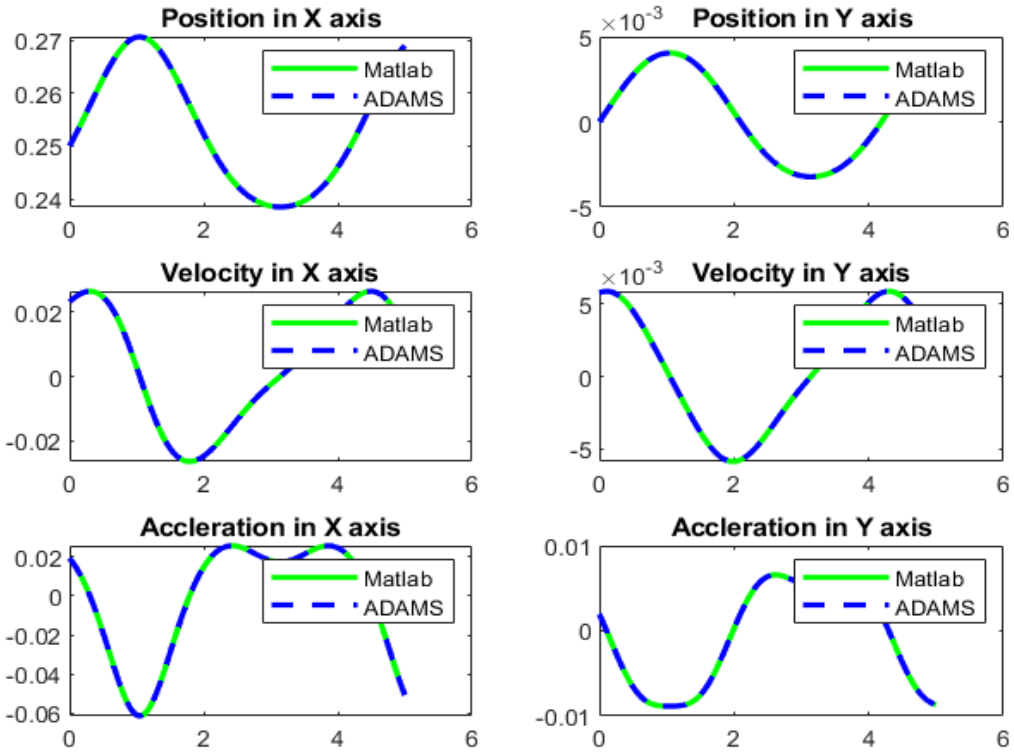

Figure 16 Comparison of ADAMS and MATLAB for Centre point 8

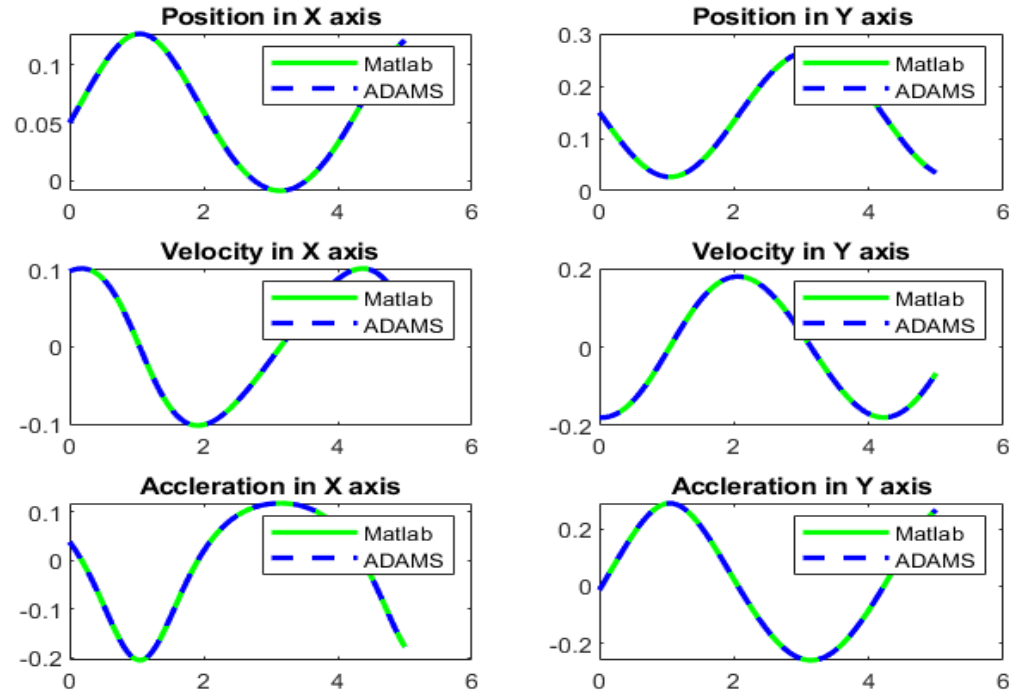

Figure 17 Comparison of ADAMS and MATLAB for Centre point 9

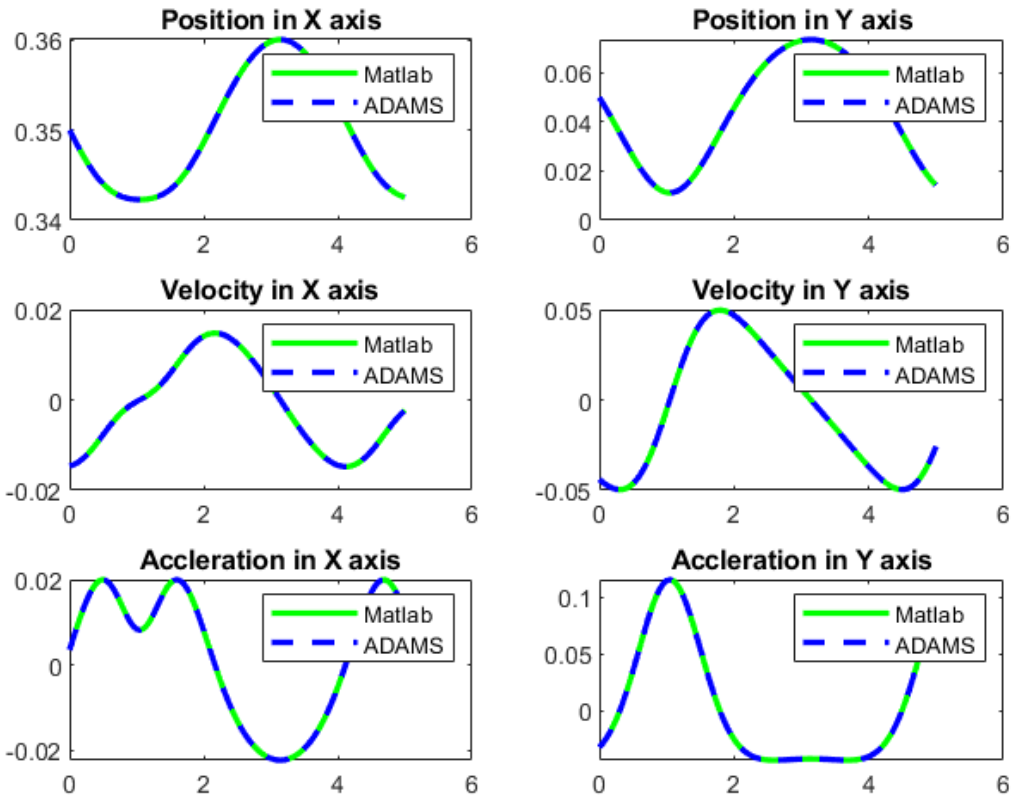

Figure 18 Comparison of ADAMS and MATLAB for Centre point 10

5. Conclusions

The project used ADAMS and MATLAB to analyze the position, velocity, and acceleration of a body using the Newton-Raphson method. The results were compared using root mean squared error and were also presented graphically. The Newton-Raphson method was found to be accurate, with a low root mean squared error, and the graphical comparison showed good agreement between the predicted and actual kinematics of the body.