

Faculty of Power and Aeronautical Engineering
Warsaw University of Technology



DYNAMICS OF MULTIBODY SYSTEM TASK 5

Created by:

Yağız Söylegüzel

Supervised by:

Prof. Janusz Fraczek

Contents

1- Introduction	4
2- Theoretical Background	4
2.1- Revolute Joint	4
2.2- Translational Joint.....	5
2.3- Jacobian Matrix	5
2.4- Newton Raphson Method	7
2.5- System's Kinematics.....	7
2.5.1. Position	8
2.5.2. Velocity	8
2.6- Singularity	9
3- Problem Definition	10
4- Solution.....	11
4.1- Comparison of MATLAB and ADAMS Graphs	13
4.1- Singularity Check	16
5- Conclusion	17

Figure 1: Revolute Joint.....	4
Figure 2: Translational Joint	5
Figure 3: Relative Displacement in Translational Joint.....	7
Figure 4: Kinematic scheme of the mechanism.....	10
Figure 5: ADAMS Model	11
Figure 6: Revolute Joint Representation.....	12
Figure 7: Motion Function	12
Figure 8: Point A_X Graphs	13
Figure 9: Point A_Y Graphs.....	14
Figure 10: MATLAB ADAMS Displacement Compare Graph for A	14
Figure 11: Point A Maximum Difference(s) Output from MATLAB	14
Figure 12: Point G_X Graphs	15
Figure 13: Point G_Y Graphs	15
Figure 14: MATLAB ADAMS Displacement Compare Graph for G	16
Figure 15: Point G Maximum Difference(s) Output from MATLAB	16
Figure 16:Result of Singularity Test	16

1- Introduction

The aim of this project is analyzing the motion of a multi-body system by using MATLAB and ADAMS software. The project focuses on studying the kinematic and dynamic behavior of a mechanism composed of multiple rigid bodies connected by revolute and translational joints. MATLAB is utilized for implementing numerical methods, such as Newton-Raphson, to solve constraint equations and calculate velocities and accelerations. ADAMS is employed for simulating the mechanism and validating the MATLAB results.

2- Theoretical Background

In the given system, multiple rigid bodies are connected to each other through various kinematic pairs or joints. These joints can be described in an absolute coordinate system using a set of constraint equations. In this project, a set of distinct kinematic constraint equations was formulated based on the type of joint present in the mechanism. As mentioned earlier, the system consists of two types of joints: prismatic (translational) and revolute joints. These constraints govern the relative motion between the bodies and are essential for analyzing the kinematic behavior of the system.

2.1- Revolute Joint

A revolute joint is a kinematic pair that allows relative rotational motion between two connected bodies around a fixed axis. It restricts all translational motion and ensures the connected points on both bodies remain coincident. Mathematically, the revolute joint is represented by a constraint equation that relates the translational and rotational coordinates of the two bodies. From Lecture 5, we derived the constraint equation for the revolute joint.

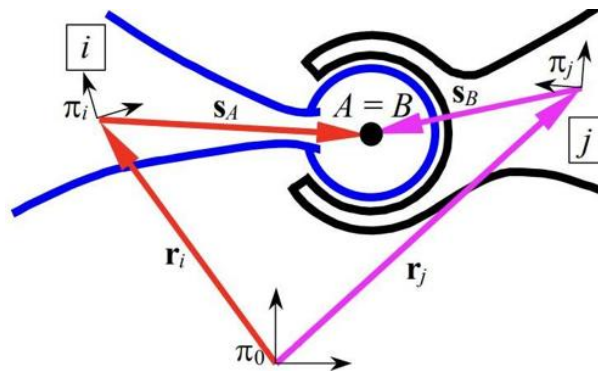


Figure 1: Revolute Joint

The constraint equation can be written in the vector form as:

$$\phi^{K\cdot} \equiv \mathbf{r}_i + \mathbf{R}_i * \mathbf{s}_A^{(i)} - \mathbf{r}_j + \mathbf{R}_j * \mathbf{s}_B^{(j)} = \mathbf{0}$$

2.2- Translational Joint

A translational joint is a kinematic pair that allows relative linear motion between two connected bodies along a specific axis while restricting all other translational and rotational movements. It ensures that the bodies remain aligned along the defined axis during motion.

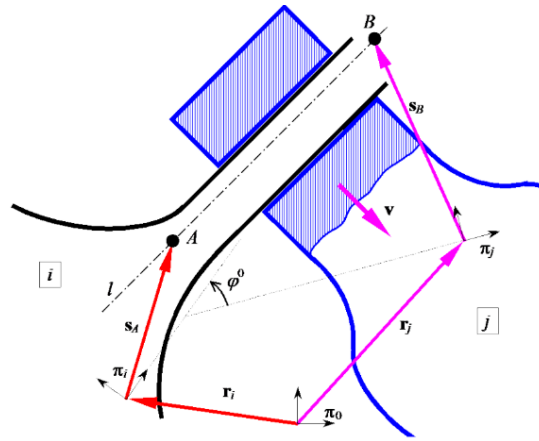


Figure 2: Translational Joint

For translational joints equations 5.18 and 5.21 of Lecture 5 were used. Since the frames were along the line of translation with points “A” and “B” on the local frame origin, simplifications were made such as:

$$\mathbf{s}_A = \mathbf{0}, \mathbf{s}_B = \mathbf{0}$$

Equation 5.18 of Lecture 5:

$$\phi^{K<} \equiv \phi_i - \phi_j - \phi^0 = 0$$

Equation 5.21 of Lecture 5:

$$\phi^{K\uparrow} \equiv (\mathbf{r}_j + \mathbf{R}_j \mathbf{s}_B^{(j)} - \mathbf{r}_i - \mathbf{R}_i \mathbf{s}_A^{(i)})^T \mathbf{R}_j \mathbf{v}^{(j)} \equiv (\mathbf{R}_j \mathbf{v}^{(j)})^T (\mathbf{r}_j + \mathbf{R}_j \mathbf{s}_B^{(j)} - \mathbf{r}_i - \mathbf{R}_i \mathbf{s}_A^{(i)})$$

2.3- 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.

The Jacobian matrix was formulated by taking the partial derivative of each constraint equation with respect to every component of the q-vector, resulting in a 30x30 matrix. For each constraint equation, most of the partial derivatives are zero, except for the terms corresponding to the bodies directly involved in the specific constraint equation.

$$\phi_{r_i}^{K\cdot} = I_{2 \times 2}$$

$$\phi_{\varphi_i}^{K\cdot} = \Omega R_i s_A^{(i)}$$

$$\phi_{r_j}^{K\cdot} = -I_{2 \times 2}$$

$$\phi_{\varphi_j}^{K\cdot} = -\Omega R_j s_B^{(j)}$$

Kinematic Constraints for Translational Joints

Partial derivatives of the translational kinematic joints (5.42 - 5.45 and 5.48 – 5.51 of Lecture 5):

$$\phi_{r_i}^{K\cdot} = \mathbf{0}_{1 \times 2}$$

$$\phi_{\varphi_i}^{K\cdot} = \mathbf{1}$$

$$\phi_{r_j}^{K\cdot} = \mathbf{0}_{1 \times 2}$$

$$\phi_{\varphi_j}^{K\cdot} = -\mathbf{1}$$

$$\phi^{K\uparrow} \equiv (r_j + R_j s_B^{(j)} - r_i - R_i s_A^{(i)})^T R_j v^{(j)} \equiv (R_j v^{(j)})^T (r_j + R_j s_B^{(j)} - r_i - R_i s_A^{(i)})$$

$$\phi_{r_i}^{K\uparrow} = -(R_j v^{(j)})^T$$

$$\phi_{\varphi_i}^{K\uparrow} = -(R_j v^{(j)})^T \Omega R_i s_A^{(i)}$$

$$\phi_{r_j}^{K\uparrow} = (R_j v^{(j)})^T$$

$$\phi_{\varphi_i}^{K\uparrow} = (r_i + R_i * s_A^{(i)})^T \Omega R_j v^{(j)} = (R_j v^{(j)})^T \Omega (r_i + R_i * s_A^{(i)})$$

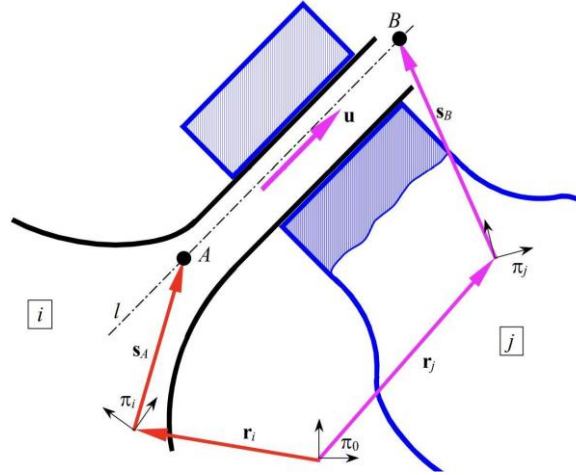


Figure 3: Relative Displacement in Translational Joint

Vector “v” in the equations changed to vector “u” which is a unit vector defined parallel to the motion of translation in the local frame coordinate.

2.4- Newton Raphson Method

The Newton-Raphson method is a numerical technique for finding the roots of nonlinear equations. It approximates a nonlinear function by a linear one using the tangent line at a specific point. In a multi-body system, the Newton-Raphson method is used to solve the constraint equations for the system's generalized coordinates q . Starting with an initial guess for q , the method iteratively updates the guess using the relation:

$$q^{(k+1)} = q^{(k)} - J^{-1}C(q^{(k)})$$

Where:

J is the Jacobian matrix, $C(q)$ is the vector of constraint equations, and k is the iteration index. The process continues until $C(q)$ converges to zero, ensuring that the positions and velocities satisfy all constraints.

2.5- System's Kinematics

It examines the position, velocity, and acceleration of the bodies as functions of time. In a multi-body system, each body is described using a reference frame, and its position and orientation relative to a fixed

reference frame are represented by generalized coordinates. The motion of a body is defined in terms of its relative movement with respect to other bodies and the fixed reference frame.

The given system is subject to constraints and non-holonomic conditions, requiring numerical methods and computer simulations for analysis. Techniques such as the Newton-Raphson method are used to solve position, velocity, and acceleration problems efficiently in constrained multi-body systems.

2.5.1. Position

The position of a multi-body system refers to the location and orientation of each body relative to a fixed reference frame. This can be described using various coordinate systems, such as absolute and relative coordinates, and is mathematically represented using vectors and matrices. In this task, the position of each body is determined by solving the constraint equations using numerical techniques like the Newton-Raphson method to ensure the constraints are satisfied.

2.5.2. Velocity

The velocity of a body in a multi-body system is determined by the time derivatives of its generalized coordinates, subject to the constraints imposed by the system. For a constrained system, the relationship between velocities and the constraints is given by:

$$\Phi_q \dot{q} = -\Phi_t$$

Where:

Φ_q is the Jacobian matrix of the constraint equations, \dot{q} is the velocity vector of the system, Φ_t represents the partial derivatives of the constraints with respect to time.

2.5.3. Acceleration

The acceleration of a body in a constrained multi-body system is determined by taking the time derivative of its velocity vector, considering the relationships imposed by the constraints. These constraints introduce additional terms into the equations, linking the accelerations of different bodies. The governing equation for acceleration is:

$$\Phi_q \ddot{q} = \Gamma$$

Where:

\ddot{q} is the acceleration vector, Γ represents the second derivative of the constraints, including terms involving velocities and rotations.

The expression for Γ depends on the type of joint in the system.

For revolute joints:

$$\Gamma = R_i \cdot s_A(i) \cdot \ddot{\phi}^2 - R_j \cdot s_B(j) \cdot \dot{\phi}_j^2$$

For translational joints:

$$\Gamma = (R_i \cdot v(j)^2)(2\Omega(\dot{r}_j - \dot{r}_i)\dot{\phi}_j^2 - R_i \cdot s_A(i)(\dot{\phi}_j - \dot{\phi}_i)^2)$$

Where:

R_i, R_j rotation matrices for the respective bodies, $s_A(i), s_B(j)$ local vectors to the joint points, \dot{r}_j, \dot{r}_i velocities of the bodies, $\dot{\phi}_j, \dot{\phi}_i$ angular velocities of the bodies.

2.6- Singularity

Singularity in a multi-body system occurs when the system's configuration leads to a loss of degrees of freedom or when certain motions become indeterminate. It can be identified using the Jacobian matrix, as this matrix governs the relationships between the system's constraints and its generalized coordinates. Several methods can be used to detect singularities:

Rank of the Jacobian:

A singularity may occur if the rank of the Jacobian matrix reduces by one or more compared to its expected rank. This indicates a loss of independent constraints

Determinant of the Jacobian:

If the determinant of the Jacobian matrix is zero, this suggests that the matrix is singular, leading to a loss of solvability in the equations of motion.

Condition Number:

A high condition number for the Jacobian matrix indicates that the system is near a singular configuration. As the condition number increases, numerical instability may occur, making the solution unreliable.

3- Problem Definition

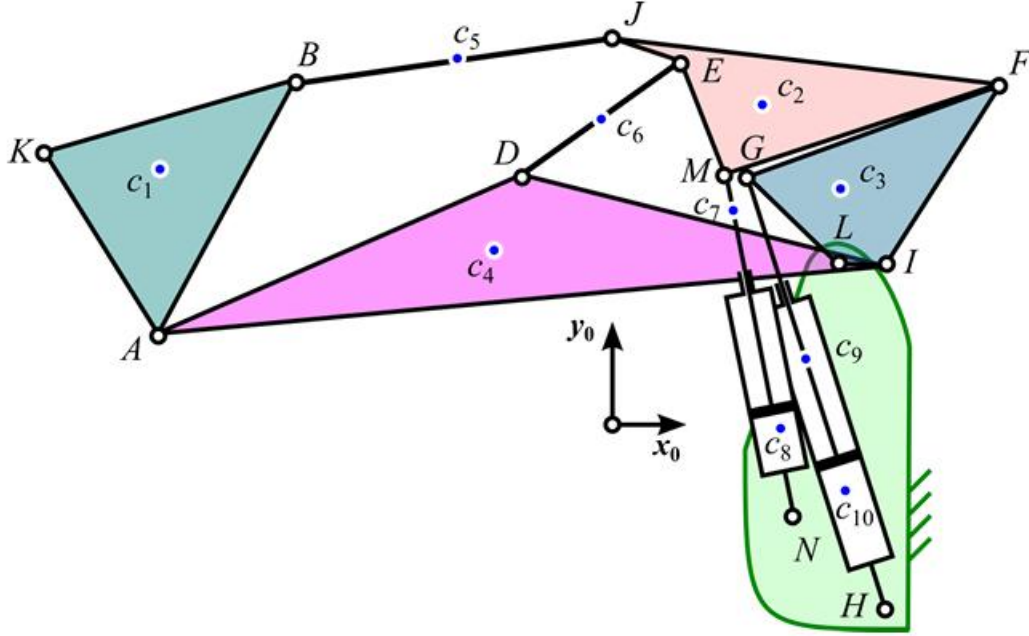


Figure 4: Kinematic scheme of the mechanism

Table 1: Table 1. Global coordinates of characteristic points (initial configuration)

Point	A	B	D	E	F	G	H	I	J	K	L	M	N
X [m]	-1	-0.7	-0.2	0.1	0.9	0.3	0.7	0.7	0	-1.2	0.6	0.2	0.4
Y [m]	0.2	0.8	0.5	0.8	0.7	0.6	-0.4	0.3	0.9	0.6	0.3	0.6	-0.2

Table 2: Global coordinates of centers of masses (initial configuration)

Point	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10
X[m]	-0.95	-0.4	0.6	-0.10	-0.35	-0.05	0.25	0.35	0.4	0.6
Y[m]	0.55	0.75	0.5	0.35	0.85	0.65	0.4	0	0.35	-0.15

Assume that length x_k of the k^{th} actuator is a function of time t :

$$x_k = l_k + a_k \sin(\omega_k t + \varphi_k)$$

The program should be able to calculate and plot positions, velocities, and accelerations of points marked with letters (e.g., ci, A, B, K, etc.) in the defined interval of time, with certain time step h , for the velocities defined above.

The program should be able to detect the singularity of the Jacobian matrix and issue an appropriate warning message.

Comparison of results obtained using ADAMS and MATLAB models (examples of accelerations, velocities, and positions of selected points)

4- Solution

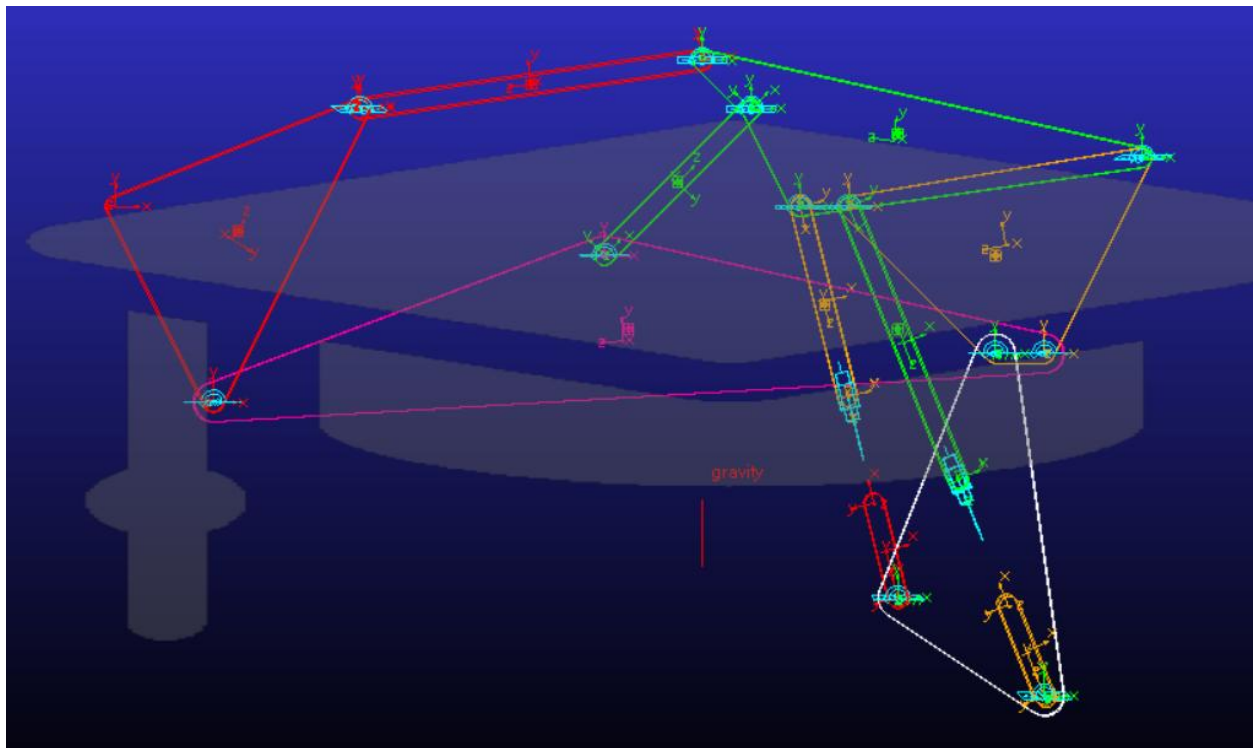


Figure 5: ADAMS Model

Started with modelling in ADAMS. All the local frames for each body determined. With simulations the graphs and data gathered for the comparison with MATLAB outputs.

Joint Motion

For the driving constraint, equation 5.23 from Lecture 5 was used. The equation of motion implemented in MATLAB differs slightly from that in ADAMS. In ADAMS, the starting distance is automatically determined

Equation of motion used in MATLAB:

$$f_{AB}(t) = L_{AB} + 0.05 \sin(1.5t) [m]$$

Where:

$$L_{AB} = \sqrt{(r_A(x) - r_B(x))^2 + (r_A(y) - r_B(y))^2}$$

The equation of motion for velocity:

$$\frac{df_{AB}(t)}{dt} = 1.5 \times 0.05 \sin(1.5t) [m/s]$$

The equation of motion for acceleration:

$$\frac{d^2 f_{AB}(t)}{dt^2} = (1.5)^2 \times 0.05 \sin(1.5t) [m/s^2]$$

Driving constraint equation for acceleration:

$$F_{1 \times 1}^D = (R_j u)^T \left(2\Omega \left(\frac{dr_j}{dt} - \frac{dr_i}{dt} \right) \frac{d\phi_j}{dt} + (r_j - r_i) \left(\frac{d\phi_j}{dt} \right)^2 - R_i S_A \left(\frac{d\phi_j}{dt} - \frac{d\phi_i}{dt} \right)^2 \right) - \frac{d^2 f_{AB}(t)}{dt^2}$$

4.1- Comparison of MATLAB and ADAMS Graphs

For Point A displacement, velocity and acceleration for X and Y results:

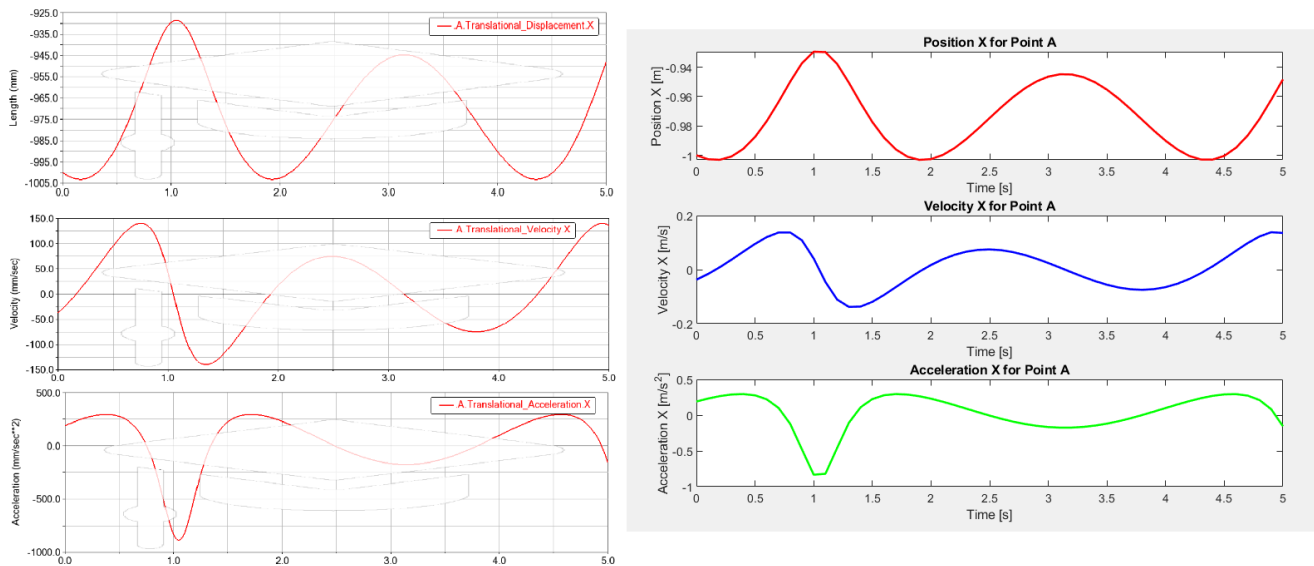


Figure 8: Point A_X Graphs

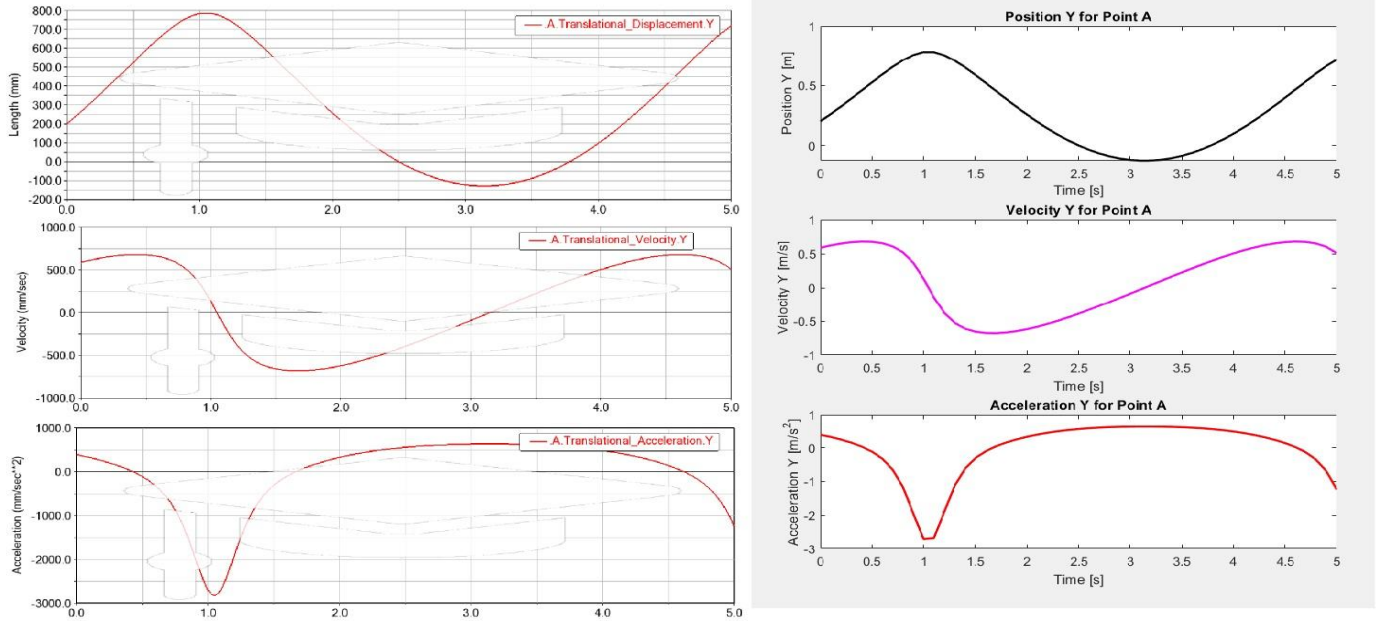


Figure 9: Point A_Y Graphs

It's visible that the graphs are matching but to be sure of the result I compared MATLAB values with using exported data from ADAMS for displacement:

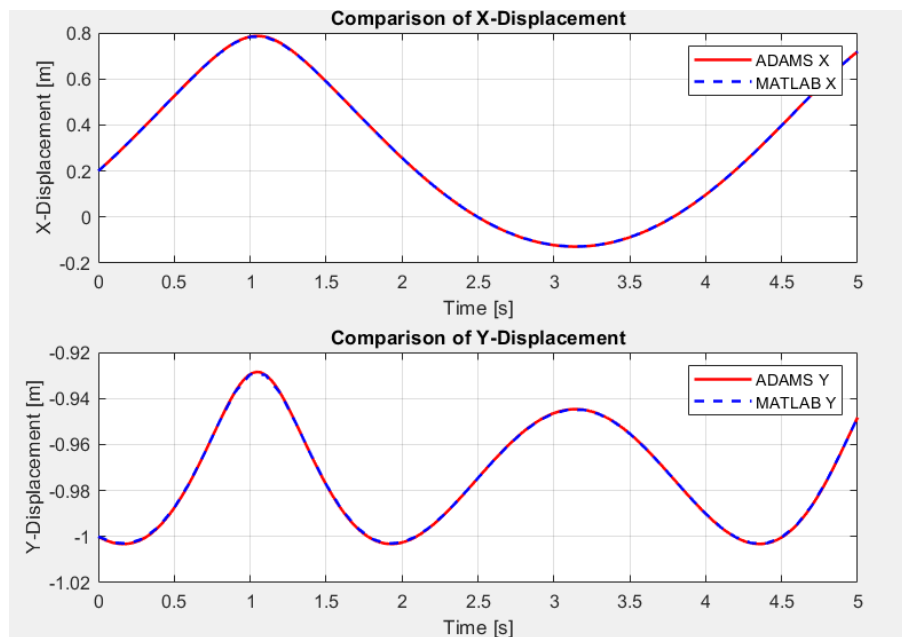


Figure 10: MATLAB ADAMS Displacement Compare Graph for A

Maximum difference in X: 0.003489 m

Maximum difference in Y: 0.001096 m

Figure 11: Point A Maximum Difference(s) Output from MATLAB

For Point G displacement, velocity and acceleration for X and Y results:

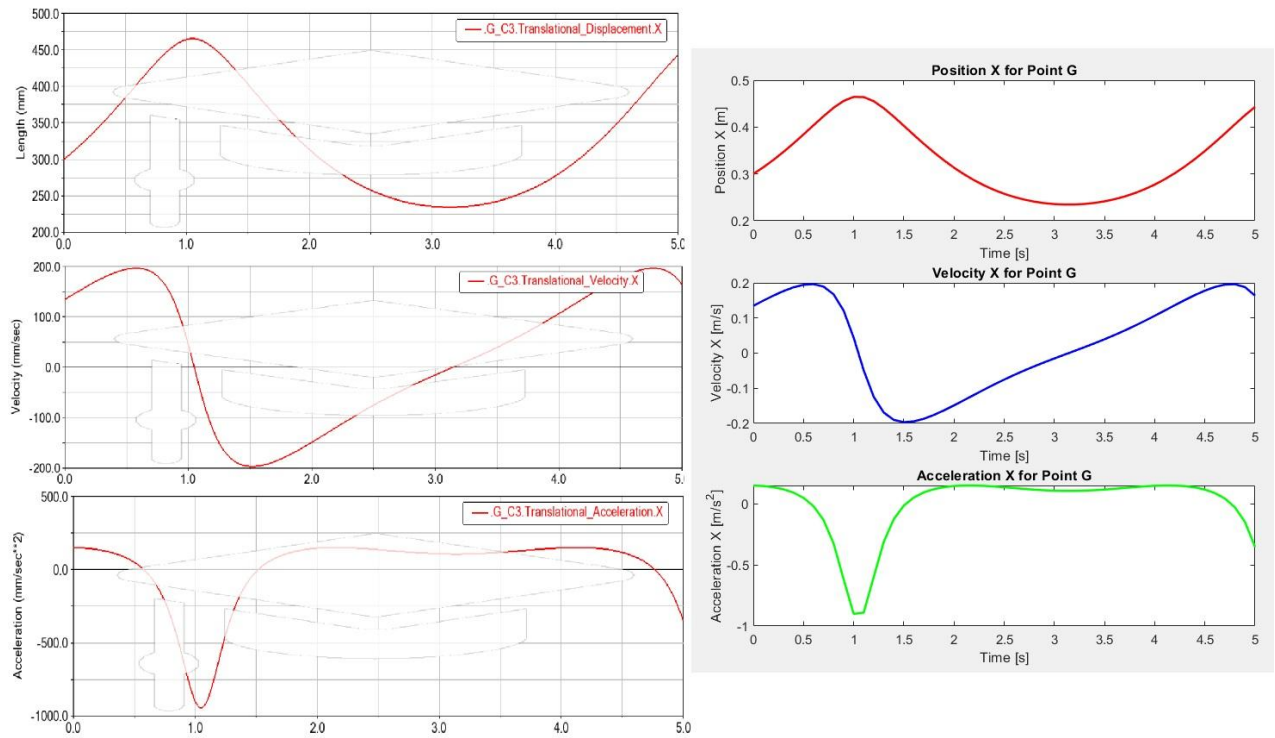


Figure 12: Point G_X Graphs

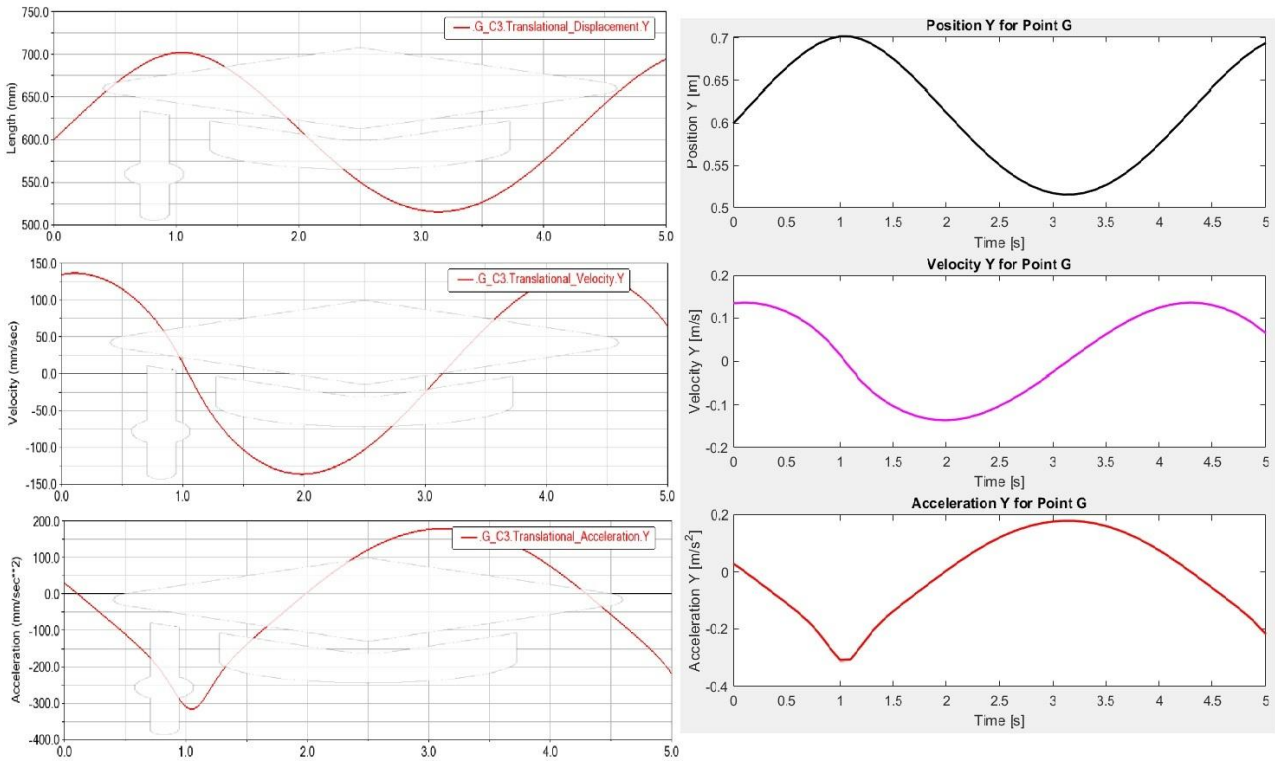


Figure 13: Point G_Y Graphs

And we are making the same comparison for G displacements and seeing the results of our maximum differences:

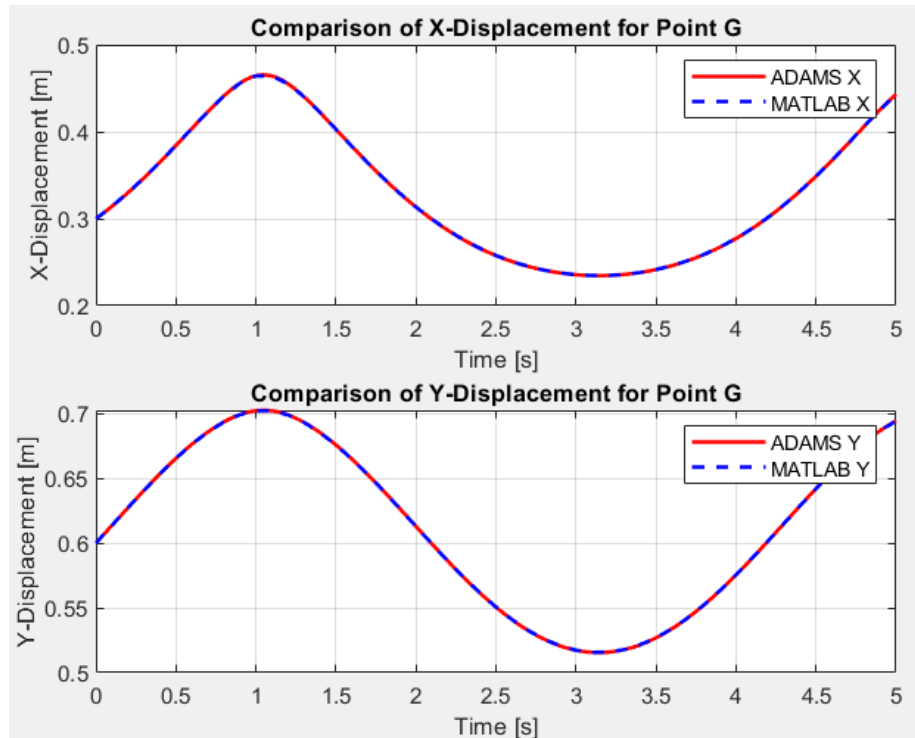


Figure 14: MATLAB ADAMS Displacement Compare Graph for G

Maximum difference in X for Point G: 0.001170 m
Maximum difference in Y for Point G: 0.000393 m

Figure 15: Point G Maximum Difference(s) Output from MATLAB

4.1- Singularity Check

Determinant is small but not zero, indicating that the Jacobian is not singular. However not the most optimal value either.

```
>> Singular
Calculated Determinant: -1.343628e-02
Final norm(F): 2.573946e-16
```

Figure 16: Result of Singularity Test

Final Norm of Constraints shows the constraints are satisfied to a high precision. This confirms that your Newton-Raphson solver is converging correctly.

5- Conclusion

The project's aim is with using kinematic scheme of the mechanism, global coordinates of characteristic points, and global coordinates of centers of mass creating MATLAB and ADAMS simulations and then comparing their displacement, velocity, acceleration results with each other. With methods like Jacobian and Newton Raphson Method we will be able to create simulations and see that both programs' results providing very close results with each other.