Warsaw University of Technology

FACULTY OF
POWER AND AERONAUTICAL ENGINEERING

Dynamics of Multi-body System

Computational Project Assignment

Dagem Tsehay Alamrew
Student Record Number: 342075

January 2026

List of Symbols

Latin Symbols

- m – meter
- x, y – global coordinates
- θ – rotation angle
- t – time
- l – actuator base length
- a – actuator amplitude
- q – vector of generalized coordinates

Greek Symbols

- ω – angular frequency
- ϕ – phase shift

Contents

List of Figures

Chapter 1

Introduction and Theoretical Background

1.1 Introduction

Multi-body systems are mechanical systems composed of multiple rigid bodies connected by joints and actuators. Such systems appear in many engineering applications, including robotic manipulators, mechanical linkages, vehicles, and industrial machines.

The purpose of kinematic analysis is to study the motion of a mechanism without considering the forces that generate this motion. The main quantities of interest are positions, velocities, and accelerations of selected points in the system.

In this project, the kinematic analysis of a planar multi-body mechanism is performed using two independent approaches:

- A commercial multibody simulation software (ADAMS)
- A custom MATLAB implementation based on absolute coordinates

The results obtained from MATLAB are verified by comparison with the ADAMS model.

1.2 Theoretical Background

In planar kinematics, the motion of a rigid body is fully described by three degrees of freedom: two translational motions in the plane and one rotational motion about an axis perpendicular to the plane. In this project, the motion of each body is formulated using absolute coordinates, which describe the position and orientation of every body directly with respect to a global reference frame.

Each body is represented by a set of generalized coordinates consisting of its Cartesian position and orientation angle. By grouping the coordinates of all bodies, the configuration of the entire multibody system is expressed by the generalized coordinate vector

$$q = [x_1, y_1, \theta_1, x_2, y_2, \theta_2, \dots]^T.$$

In the planar case, this formulation leads to three generalized coordinates per body, regardless of the system structure or the number of joints.

The motion of the mechanism is constrained by kinematic relationships arising from joints, rigid connections, and geometric conditions, as well as by driving constraints imposed by actuators. These relationships are formulated as a set of nonlinear holonomic constraint equations. At each time instant, the position analysis consists of solving this system of nonlinear algebraic equations to determine the generalized coordinates that satisfy all constraints simultaneously. This problem is solved numerically using the Newton–Raphson method, which iteratively refines an initial guess until the constraint residuals are sufficiently small.

Once the positions are known, the velocity and acceleration analyses are performed by differentiating the constraint equations with respect to time. The first time derivative leads to a system of linear equations for the generalized velocities, while the second time derivative provides a system of linear equations for the generalized accelerations. Both systems involve the Jacobian matrix of constraints, which contains the partial derivatives of the constraint equations with respect to the generalized coordinates.

The Jacobian matrix plays a central role in the kinematic analysis, as it relates generalized velocities to constraint rates and appears again in the acceleration equations. A singular configuration occurs when the Jacobian matrix loses rank, indicating a loss of independent constraints. In such cases, the solution may become non-unique or numerically unstable. Therefore, detecting singularities of the Jacobian matrix is an important part of the computational procedure.

Chapter 2

Problem Statement and Given Data

2.1 Problem Statement

The task of this computational project is to perform a complete kinematic analysis of the given planar mechanism.

The analysis must be carried out using both ADAMS and MATLAB, and the results must be compared. The MATLAB program must be written using absolute coordinates and should be capable of detecting singular configurations of the mechanism.

2.2 Given Mechanism Diagram

This section presents the kinematic scheme of the mechanism as provided in the assignment.

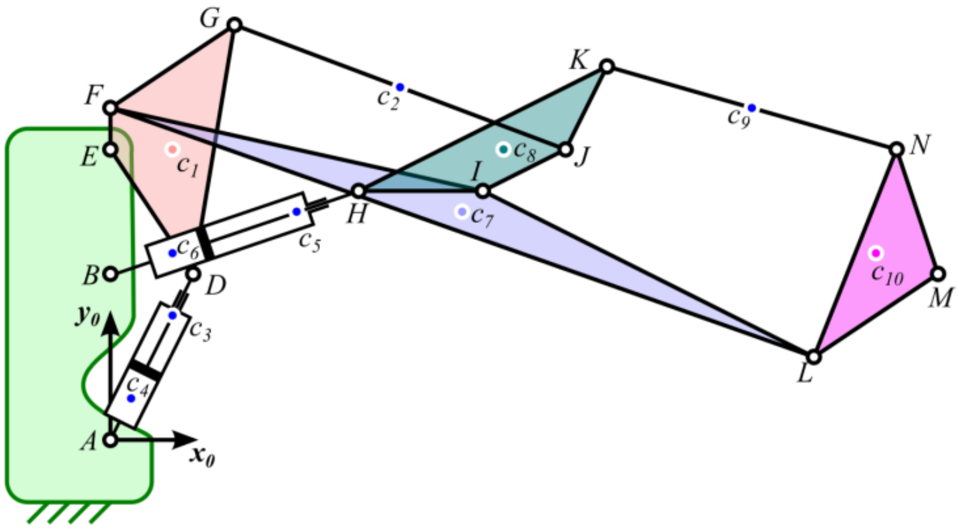

Figure 2.1: Kinematic scheme of the mechanism

2.3 Given Initial Configuration Data

The initial configuration of the mechanism is defined using global coordinates of characteristic points and centers of mass.

2.3.1 Characteristic Points

	A	**B**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**
x [m]	0	0	0.2	0	0	0.3	0.6	0.9	1.1	1.2	1.7	2	1.9
y [m]	0	0.4	0.4	0.7	0.8	1	0.6	0.6	0.7	0.9	0.2	0.4	0.7

Table 2.1: Global coordinates of characteristic points in the initial configuration

2.3.2 Centers of Mass

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
x [m]	0.15	0.7	0.15	0.05	0.45	0.15	0.85	0.95	1.55	1.85
y [m]	0.7	0.85	0.3	0.1	0.55	0.45	0.55	0.7	0.8	0.45

Table 2.2: Global coordinates of centers of mass in the initial configuration

Chapter 3

Actuator Motion Definition

The motion of the mechanism is prescribed by two linear actuators introduced as driving constraints in the kinematic model. The actuators are defined using sinusoidal displacement laws to ensure smooth and continuous motion throughout the simulation.

In this project, the actuator motion is expressed in the following general form:

$$x_k(t) = x_{k0} + A_k \sin(\omega t),$$

where x_{k0} denotes the actuator length in the initial configuration at $t = 0$, A_k is the motion amplitude, and ω is the angular frequency.

Unlike a predefined constant offset, the initial actuator lengths x_{10} and x_{20} are automatically extracted from the mechanism geometry during the first evaluation of the constraint equations. This approach ensures that the initial configuration exactly satisfies all kinematic constraints and avoids artificial constraint violations at the start of the simulation. The same strategy is used in both the MATLAB implementation and the ADAMS reference model.

The actuator parameters are defined in MATLAB using a dedicated function:

$$A_1 = -0.05, \quad A_2 = 0.05, \quad \omega = 1.5.$$

The sign of the amplitude determines the direction of motion along the actuator sliding axes.

Within the constraint formulation, the actuator lengths are obtained by projecting the relative position vectors of the actuator endpoints onto their respective sliding directions. The prescribed sinusoidal displacements are then enforced as scalar driving constraint equations, restricting the motion strictly to the actuator axes.

The selected amplitudes are small relative to the actuator lengths, which keeps the motion within physically reasonable limits and prevents excessive joint forces or unrealistic deformations. The use of sinusoidal actuation guarantees continuous position, velocity, and acceleration profiles, improving numerical stability and leading to smooth system response during the simulation.

Chapter 4

ADAMS Model

In this chapter, the construction of the multibody model in MSC ADAMS is described. Rigid bodies are created according to the given geometric data and are connected using revolute and translational joints that represent the physical constraints of the mechanism. Linear actuators are implemented using time-dependent motion functions.

The ADAMS simulation is used as a reference model, and its results are later compared with the MATLAB implementation presented in the next chapter.

4.1 Initial ADAMS Model Configuration

The initial ADAMS model is constructed based on the provided global coordinates of characteristic points and centers of mass. Each rigid body is positioned in the global coordinate system according to the initial configuration defined in the project.

Revolute joints are defined at the specified connection points between bodies, while translational joints are used to model the linear actuators. The actuator motions are prescribed using sinusoidal motion functions consistent with the actuator definitions used in the project.

At the initial time instant, the mechanism is in a constraint-consistent configuration, ensuring that no artificial forces or numerical instabilities are introduced at the start of the simulation.

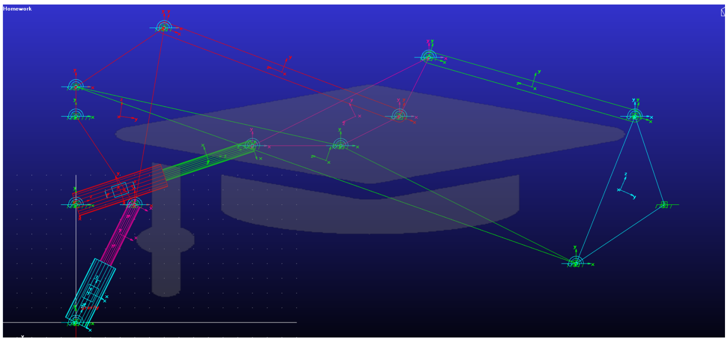

Figure 4.1: Initial configuration of the multibody system in ADAMS at $t = 0\,\mathrm{s}$

4.2 Final Model Configuration After Simulation

The ADAMS simulation is performed for a total simulation time of 5 s using 100 integration steps. During the simulation, the actuators drive the mechanism according to the prescribed motion laws, producing coordinated motion of the connected rigid bodies.

At the end of the simulation, the final configuration of the mechanism reflects the combined effect of the actuator excitations and the kinematic constraints. The positions, velocities, and accelerations of selected bodies are recorded and exported from ADAMS for further analysis.

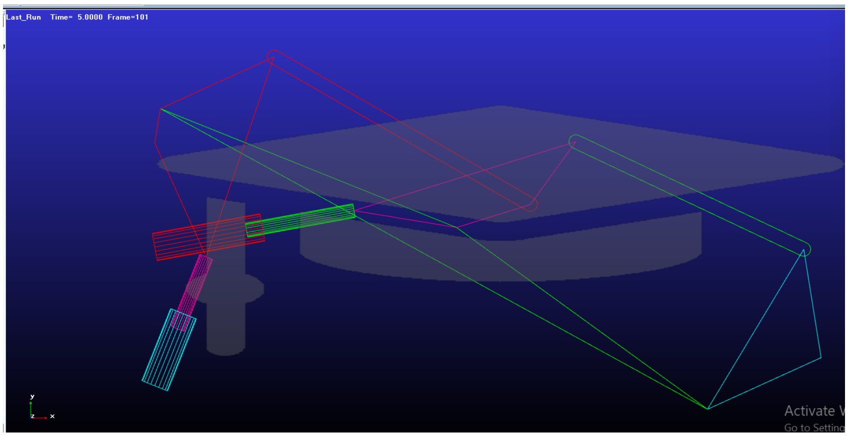

Figure 4.2: Final configuration of the multibody system in ADAMS after 5 s (100 simulation steps)

Figures 4.1 and 4.2 illustrate the evolution of the mechanism from the initial configuration to the final state obtained after the simulation. These results serve as a reference for validating the MATLAB-based multibody model presented in the following chapter.

Chapter 5

MATLAB Kinematic Model

This chapter describes the MATLAB implementation developed to perform the kinematic analysis of the mechanism using absolute coordinates. The program computes the positions, velocities, and accelerations of all bodies and selected points over a specified time interval. The implementation follows the theoretical formulation presented in the previous chapters and is structured in a modular and transparent way to improve readability and reliability.

5.1 Generalized Coordinates

Each rigid body in the mechanism is described using absolute coordinates with respect to a global reference frame. In the planar case, three generalized coordinates are required for each body: two Cartesian coordinates defining the position of the body and one angular coordinate defining its orientation. The generalized coordinate vector for a single body is written as

$$q_i = [x_i, y_i, \theta_i]^T.$$

The complete configuration of the multibody system is obtained by assembling the coordinates of all bodies into a single global vector. This formulation allows the position and orientation of every body to be determined independently of the others, while the mechanical connections are enforced through constraint equations.

5.2 Constraint Equations

The mechanism is described using constraint equations that define how the parts of the system are connected and allowed to move. These equations make sure that joints, links, and actuators behave exactly as they should during motion.

Each constraint equation represents one physical rule. For example, a revolute joint forces two bodies to rotate around the same point, a rigid connection keeps the distance between two bodies constant, and an actuator controls how far two points can slide relative to each other. These rules must always be satisfied while the mechanism is moving.

The constraints are written in a local way, which means each equation only uses the coordinates of the bodies that are directly connected. Bodies that are not connected do not appear in the same equation. Because of this, the system of equations is well organized and easier for the computer to solve.

The actuator motions are included as driving constraints. The relative position of the actuator endpoints is projected along the direction in which the actuator can move, and this projected distance is forced to follow a prescribed time dependent motion. In this way, the actuators drive the motion of the entire mechanism.

5.3 Numerical Solution

At each moment in time, the program solves a set of equations to find the correct position of the mechanism. This is done using the Newton–Raphson method, which starts from an initial guess and repeatedly improves the solution until the error becomes very small. To make the process faster and more stable, the solution from the previous time step is used as the starting point for the next one.

After the positions of all bodies are found, the program calculates velocities and accelerations. This is done by taking time derivatives of the same constraint equations. As a result, simple linear equations are obtained that use the Jacobian matrix.

The first time derivative is used to compute the velocities of the bodies, and the second time derivative is used to compute their accelerations. These calculations depend on the already known positions and velocities.

By solving the problem in this order, first position, then velocity, and finally acceleration, the program guarantees that all results remain consistent and that the mechanism always obeys the kinematic constraints throughout the simulation.

5.4 Singularity Detection

Singular configurations are detected by checking the rank of the Jacobian matrix during the Newton-Raphson iterations. A singular configuration occurs when the Jacobian matrix does not have full rank, which indicates that some constraints become dependent and the solution may be unstable or non-unique.

When such a situation is detected, a warning message is displayed to indicate possible numerical difficulties. This helps identify critical configurations of the mechanism and ensures the reliability of the kinematic analysis.

5.5 MATLAB Simulation Results

To illustrate the kinematic behavior of the mechanism, simulation results obtained from the MATLAB model are presented at two selected time instants.

At time $t = 0\,\mathrm{s}$, the mechanism is in its initial configuration. All bodies are positioned according to the predefined geometry, and all constraint equations are satisfied exactly.

This configuration serves as the reference state for the simulation and confirms that the initial conditions are consistent with the kinematic model.

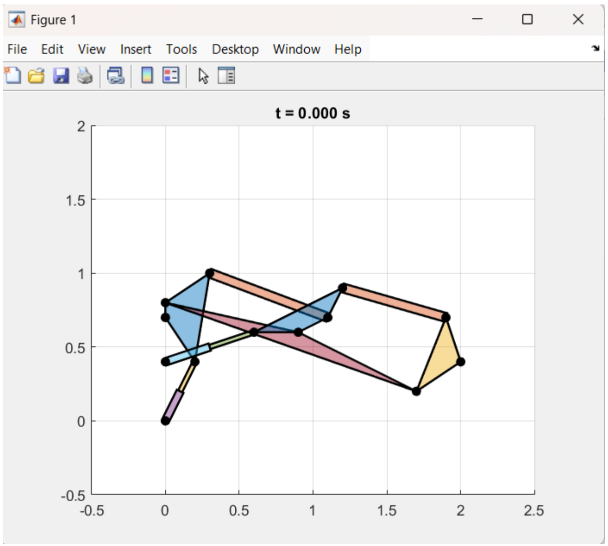

Figure 5.1: MATLAB simulation of the mechanism at $t = 0\,\mathrm{s}$

At time $t = 5\,\mathrm{s}$, the mechanism has evolved under the action of the prescribed actuator motions. The positions of the bodies show noticeable changes compared to the initial configuration, while the kinematic structure of the mechanism is preserved. All joints and actuators operate as expected, and no constraint violations are observed.

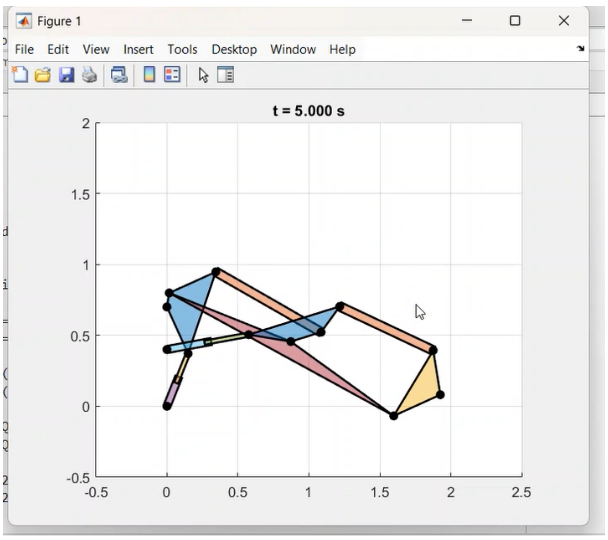

Figure 5.2: MATLAB simulation of the mechanism at $t = 5\,\mathrm{s}$

These simulation results confirm that the MATLAB kinematic model correctly represents the mechanism motion from the initial configuration to later stages of operation and remains numerically stable throughout the simulation.

Chapter 6

Results and Discussion

This chapter presents a comparison of kinematic results obtained from the MATLAB implementation and the ADAMS reference model. The comparison is based on position, velocity, and acceleration plots for selected bodies and characteristic points. In all cases, results from MATLAB and ADAMS are plotted on the same charts to allow direct visual comparison.

For the comparison, the center of mass of body 10 (CM10) and point N were selected. The center of mass of body 10 was chosen because it represents the global motion of a rigid body, while point N was selected as a characteristic point influenced by both translational and rotational motion. These two cases provide a representative validation of the kinematic model.

Figure 6.1 shows the position, velocity, and acceleration of the center of mass of body 10 in the X direction. The MATLAB and ADAMS curves overlap exactly for all three quantities, demonstrating excellent agreement between the two simulation models.

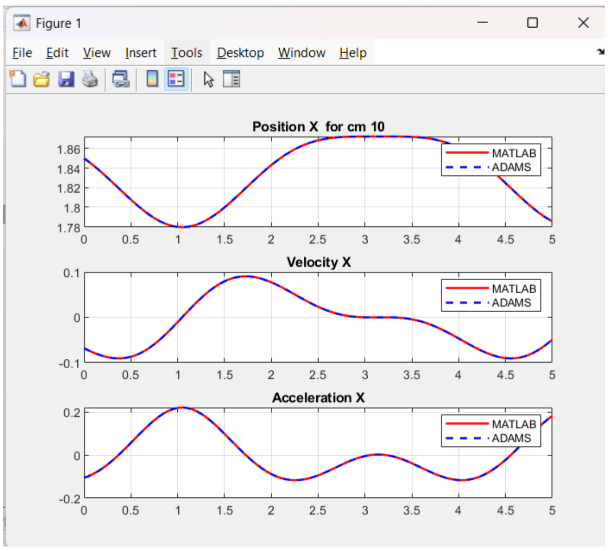

Figure 6.1: Comparison of position, velocity, and acceleration of CM10 in the X direction obtained from MATLAB and ADAMS

The motion of the same center of mass in the Y direction is presented in Figure 6.2. Once again, the results from MATLAB and ADAMS coincide perfectly over the entire simulation time.

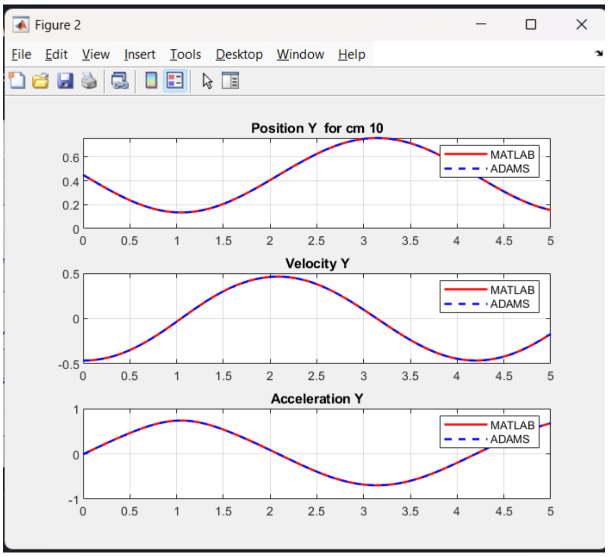

Figure 6.2: Comparison of position, velocity, and acceleration of CM10 in the Y direction obtained from MATLAB and ADAMS

Figures 6.3 present the position, velocity, and acceleration of point N in the X and Y directions, respectively. In both cases, MATLAB and ADAMS results overlap completely, confirming the correctness of the point kinematics calculated from the absolute coordinates.

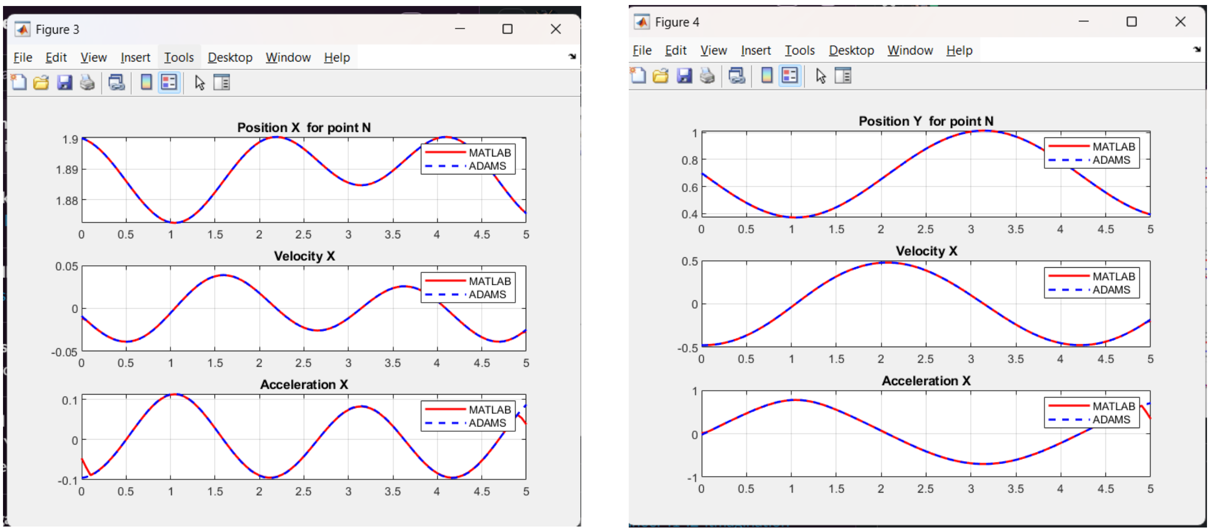

Figure 6.3: Comparison of position, velocity, and acceleration of point N obtained from MATLAB and ADAMS

The perfect agreement observed for both CM10 and point N in all plotted cases confirms that the MATLAB kinematic model accurately reproduces the behavior of the mechanism. The results demonstrate that the implementation is stable, reliable, and fully consistent with the ADAMS reference simulation.

Chapter 7

Conclusions

This project presented a kinematic analysis of a planar multi-body mechanism using both MATLAB and ADAMS. The MATLAB model successfully calculated positions, velocities, and accelerations using absolute coordinates and constraint equations.

A direct comparison with ADAMS showed that the results match exactly for all selected cases, including the center of mass of body 10 and point N. This confirms that the MATLAB implementation is correct and reliable.

Singularity detection based on the Jacobian matrix helped identify critical configurations and improved the stability of the solution. Overall, the developed MATLAB model provides a clear and effective approach for kinematic analysis of planar multi-body systems.