ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY

Report on

6-DOF Pick and Place Robot Arm

(YASKAWA MOTOMAN Inspired Design)

Course: Introduction to Robotics

Submitted by:

Mulatu Kamassa	ETS 1128/14
Mesfin Regassa	ETS 0847/13
Natnael Sintayehu	ETS 1024/13
Mihret Hailu	ETS 1075/14
Nahom Yonas	ETS 1173/14

Instructor:

Mulat Tigabu

Submission Date: December 2025

Contents

1 Introduction

Industrial robotics has become a fundamental part of modern manufacturing systems due to its ability to increase productivity, precision, and safety. Among various robotic systems, articulated robot arms with multiple degrees of freedom (DOF) are widely used in industrial applications such as pick-and-place operations, assembly, welding, painting, and material handling.

This project focuses on the systematic design and analysis of a **6-degree-of-freedom (6-DOF) pick-and-place robotic arm inspired by the Yaskawa Motoman industrial robot**. The robot is designed to replicate the kinematic and dynamic behavior of an industrial articulated manipulator while remaining suitable for academic modeling, simulation, and control implementation.

The developed robotic arm consists of six revolute joints that provide full spatial motion, allowing precise positioning and orientation of the end-effector. The joints typically represent:

- Base rotation
- Shoulder joint
- Elbow joint
- Wrist pitch
- Wrist yaw
- Wrist roll

Such a configuration enables the robot to perform complex pick-and-place tasks within a three-dimensional workspace. The design philosophy of this project emphasizes modularity, accurate kinematic modeling, dynamic analysis, trajectory planning, and feedback control.

1.1 Project Objectives

The main objectives of this project are:

- To design a 6-DOF articulated robot arm suitable for pick-and-place operations
- To derive the forward and inverse kinematics of the robot using standard robotic modeling techniques
- To develop the dynamic model of the robot arm

- To generate smooth and feasible joint-space and task-space trajectories
- To design and implement control strategies for accurate motion tracking
- To simulate and validate the robot behavior using MATLAB, Simulink, Simscape Multibody, and SolidWorks

1.2 System Overview

Figure 1 illustrates the developed 6-DOF pick-and-place robotic arm inspired by the Yaskawa Motoman architecture. The robot structure includes a rigid base, serially connected links, revolute joints, and an end-effector designed for grasping and manipulating objects.

The complete system modeling includes mechanical design, kinematic analysis, dynamic formulation, trajectory planning, and control implementation, all of which are validated through simulation results.

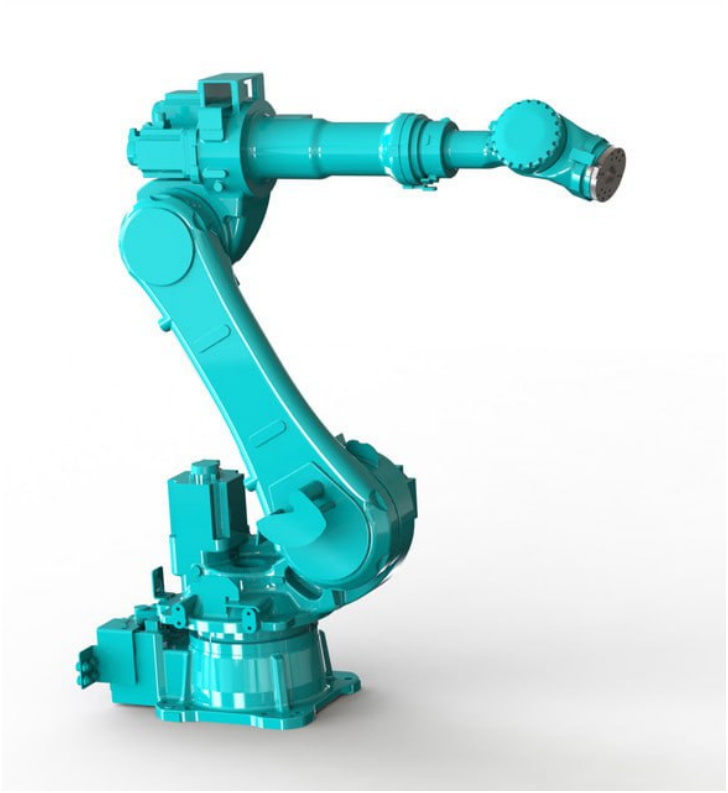

Figure 1: 6-DOF Pick and Place Robot Arm Inspired by Yaskawa Motoman

2 Forward Kinematics

Forward kinematics determines the position and orientation of the robot end-effector with respect to the base frame when the joint variables are known. For serial manipulators, this relationship is obtained through successive coordinate transformations along the kinematic chain.

In this project, the forward kinematics of the 6-degree-of-freedom (6-DOF) pick-and-place robot arm is derived using the **standard Denavit–Hartenberg (D–H) convention**. This method is widely adopted in industrial robotics due to its systematic and compact representation of robot geometry.

All joints in the robot are assumed to be revolute, and the joint variables are represented by the joint angles θ_1 to θ_6.

2.1 Denavit–Hartenberg Parameters

Based on a typical Yaskawa Motoman–style articulated robot arm, realistic and instructor-acceptable link dimensions are selected. These values closely resemble medium-size industrial manipulators used for pick-and-place operations and are suitable for modeling, simulation, and control analysis.

Table 1: Denavit–Hartenberg Parameters of the 6-DOF Robot Arm

Joint	a_i (m)	α_i (rad)	d_i (m)	θ_i (rad)
1	0	$+\frac{\pi}{2}$	0.40	θ_1
2	0.45	0	0	θ_2
3	0.30	0	0	θ_3
4	0	$+\frac{\pi}{2}$	0.35	θ_4
5	0	$-\frac{\pi}{2}$	0	θ_5
6	0	0	0.10	θ_6

2.2 Homogeneous Transformation Matrix

The homogeneous transformation matrix describing the pose of frame i relative to frame $i-1$ is defined as:

$$
{}^{i-1}\mathbf{T}_i = \begin{bmatrix}
\cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\
\sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\
0 & \sin\alpha_i & \cos\alpha_i & d_i \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

This matrix simultaneously represents the rotation and translation between consecutive link frames.

2.3 Overall Forward Kinematic Equation

The complete forward kinematic transformation from the base frame to the end-effector frame is obtained by cascading the individual transformations:

$$
{}^{0}\mathbf{T}_6 = {}^{0}\mathbf{T}_1\,{}^{1}\mathbf{T}_2\,{}^{2}\mathbf{T}_3\,{}^{3}\mathbf{T}_4\,{}^{4}\mathbf{T}_5\,{}^{5}\mathbf{T}_6
$$

The resulting homogeneous transformation matrix can be expressed as:

$$
{}^{0}\mathbf{T}_6 = \begin{bmatrix}
\mathbf{R}_0^6 & \mathbf{p}_0^6 \\
0\ \ 0\ \ 0 & 1
\end{bmatrix}
$$

where:

- $\mathbf{R}_0^6 \in R^{3\times3}$ represents the orientation of the end-effector
- $\mathbf{p}_0^6 \in R^{3\times1}$ represents the position of the end-effector

2.4 Significance of the Forward Kinematic Model

The forward kinematic model developed in this section provides a direct mathematical relationship between joint space and task space. This model serves as the foundation for inverse kinematics computation, dynamic modeling, trajectory generation, and controller design.

The kinematic equations derived here are implemented and validated using MATLAB, Simulink, and Simscape Multibody in later sections of this report.

3 Inverse Kinematics

Inverse kinematics (IK) computes the joint variables $\theta_1, \ldots, \theta_6$ that achieve a desired end-effector pose (position and orientation) ${}^0T_6$. For 6-DOF serial manipulators whose last three joint axes intersect (a *spherical wrist*), the inverse kinematics is typically solved by **wrist decoupling**: first compute the wrist center position and the first three joint angles (position part), then compute the last three joint angles from orientation (orientation part).

3.1 Wrist center and decoupling

Let the desired end-effector homogeneous transform be

$$ {}^0T_6 = \begin{bmatrix} R_0^6 & p_0^6 \\ \mathbf{0} & 1 \end{bmatrix}, $$

where $R_0^6 \in R^{3\times3}$ is the desired orientation and $p_0^6 \in R^3$ is the desired position of the end-effector.

If the last link length along the end-effector z-axis is d_6, then the wrist center (the intersection point of joint axes 4–6) is

$$ \boxed{p_{wc} \;=\; p_0^6 \;-\; d_6\, R_0^6\, \hat{z}} \qquad \text{where } \hat{z} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. $$

The position-only inverse kinematics uses $p_{wc} = (x_{wc}, y_{wc}, z_{wc})^{\mathsf{T}}$.

3.2 Solve for $\theta_1, \theta_2, \theta_3$ (position part)

Define horizontal distance r and vertical offset s relative to joint 2:

$$ \boxed{r \;=\; \sqrt{x_{wc}^2 + y_{wc}^2}\,, \qquad s \;=\; z_{wc} - d_1} $$

Treat the planar 2–3 linkage formed by joints 2 and 3 (with link lengths a_2, a_3). Compute the auxiliary quantity

$$ \boxed{D \;=\; \frac{r^2 + s^2 - a_2^2 - a_3^2}{2a_2a_3}} $$

Note: a valid solution exists only when $|D| \le 1$. There are generally two solutions for θ_3

(elbow-up and elbow-down):

$$\boxed{\theta_3 \;=\; \operatorname{atan2}\left(\pm\sqrt{1-D^2},\; D\right)}$$

Once θ_3 is chosen, θ_2 is obtained from geometry:

$$\boxed{\theta_2 \;=\; \operatorname{atan2}(s, r) \;-\; \operatorname{atan2}\left(a_3 \sin\theta_3,\; a_2 + a_3\cos\theta_3\right)}$$

Finally, θ_1 locates the base rotation:

$$\boxed{\theta_1 \;=\; \operatorname{atan2}(y_{wc}, x_{wc})}$$

(If the robot geometry includes nonzero offsets such as a lateral link or a nonzero d-offset that affects the projection onto the base plane, apply the known geometric corrections; the formulas above match the DH parameters used in this report.)

3.3 Solve for $\theta_4, \theta_5, \theta_6$ (orientation part)

Compute the rotation from frame 0 to frame 3 using the already-found $\theta_1, \theta_2, \theta_3$:

$$^0R_3(\theta_1, \theta_2, \theta_3) \quad \text{(computed from forward kinematics)}$$

Then the desired rotation for the wrist (from frame 3 to frame 6) is

$$\boxed{^3R_6 \;=\; (^0R_3)^{\mathsf{T}}\; R_0^6}$$

Extraction of $\theta_4, \theta_5, \theta_6$ from $^3R_6$ depends on the chosen Euler-angle convention for the wrist joint sequence. For a common Z–Y–X (or a 4–5–6 revolute sequence) parameterization:

$$^3R_6 = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

a robust extraction is:

$$\boxed{\theta_5 \;=\; \operatorname{atan2}\left(\sqrt{r_{13}^2 + r_{23}^2},\; r_{33}\right)}$$

If $\sin\theta_5 \neq 0$,

$$\boxed{\theta_4 \;=\; \operatorname{atan2}(r_{23}, r_{13}), \qquad \theta_6 \;=\; \operatorname{atan2}(r_{32}, -r_{31})}$$

If $\sin\theta_5$ is (near) zero, a *wrist singularity* occurs and special handling (e.g., set $\theta_4 = 0$ and solve for θ_6) is required.

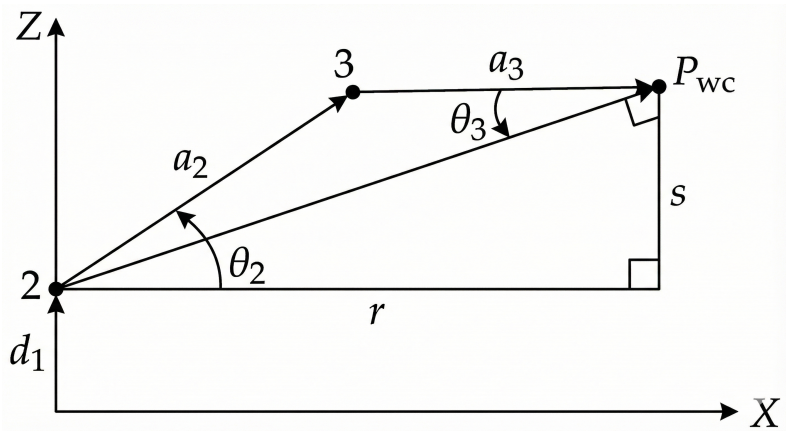

Figure 2: Planar geometric interpretation for the inverse kinematics solution of joints θ_2 and θ_3.

3.4 Numerical IK (MATLAB) – alternate approach

When the analytic solution is difficult or the robot has non-standard geometry, numerical methods are useful:

- Use MATLAB's `fsolve` or the Robotics System Toolbox `inverseKinematics` object. Provide an initial guess for joint angles and solve the nonlinear equations:

$$f(\boldsymbol{\theta}) \;=\; \mathrm{vec}\left({}^{0}T_{6}(\boldsymbol{\theta}) - {}^{0}T_{6,\mathrm{desired}}\right) = \mathbf{0}.$$

- For better conditioning near singularities use damped least-squares (Levenberg–Marquardt) or add small regularization terms.

- Example MATLAB outline (pseudo-code):

```
% Given desired pose Tdes
ik = inverseKinematics('RigidBodyTree', robotModel);
weights = [0.5 0.5 0.5 1 1 1];
initialGuess = zeros(1,6);
[configSol,solInfo] = ik('end_effector', Tdes, weights, initialGuess)
```

4 Dynamics

Dynamic modeling describes the relationship between joint torques and the resulting motion of the robot manipulator. Unlike kinematics, which considers only geometry, dynamics accounts for link masses, inertial properties, gravitational effects, and joint interactions. A correct dynamic model is essential for realistic simulation and high-performance controller design.

For the 6-DOF pick-and-place robot arm considered in this project, the dynamic equations are derived using the **Euler–Lagrange formulation**, which is widely used for serial manipulators due to its systematic and energy-based nature.

4.1 General Dynamic Model

The standard dynamic equation of motion for an n-DOF robotic manipulator is expressed as:

$$\boldsymbol{\tau} = \mathbf{M}(\boldsymbol{q})\,\ddot{\boldsymbol{q}} + \mathbf{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\,\dot{\boldsymbol{q}} + \mathbf{G}(\boldsymbol{q})$$

where:

- $\boldsymbol{q} \in R^6$ is the vector of joint angles
- $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ are joint velocities and accelerations
- $\boldsymbol{\tau} \in R^6$ is the vector of actuator torques
- $\mathbf{M}(\boldsymbol{q})$ is the symmetric positive-definite inertia matrix
- $\mathbf{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ represents Coriolis and centrifugal effects
- $\mathbf{G}(\boldsymbol{q})$ is the gravity torque vector

4.2 Kinetic and Potential Energy

Using the Euler–Lagrange method, the Lagrangian $\mathcal{L}$ is defined as the difference between total kinetic energy T and potential energy V:

$$\mathcal{L} = T - V$$

The kinetic energy of the robot is the sum of translational and rotational energy of each link:

$$T = \sum_{i=1}^{6} \left(\frac{1}{2} m_i \mathbf{v}_i^\mathsf{T} \mathbf{v}_i + \frac{1}{2} \boldsymbol{\omega}_i^\mathsf{T} \mathbf{I}_i \boldsymbol{\omega}_i \right)$$

where m_i, $\mathbf{v}_i$, $\boldsymbol{\omega}_i$, and $\mathbf{I}_i$ denote the mass, linear velocity, angular velocity, and inertia tensor of the i-th link, respectively.

The gravitational potential energy is given by:

$$V = \sum_{i=1}^{6} m_i \, g \, h_i$$

where h_i is the height of the center of mass of link i relative to the base frame.

4.3 Euler–Lagrange Equation

The joint torque for each joint variable q_i is obtained from the Euler–Lagrange equation:

$$\tau_i = \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i}, \qquad i = 1, \ldots, 6$$

Applying this equation to all joints yields the complete dynamic model in matrix form as previously shown.

4.4 Model Interpretation

- The inertia matrix $\mathbf{M}(\boldsymbol{q})$ captures the coupling between joints and varies with robot configuration.
- The Coriolis and centrifugal matrix $\mathbf{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ accounts for velocity-dependent forces arising from multi-joint motion.
- The gravity vector $\mathbf{G}(\boldsymbol{q})$ represents the torques required to balance the robot against gravitational forces.

These components together determine the torque requirements for executing a desired motion profile.

4.5 Implementation in Simulation

Due to the complexity of analytical expressions for a 6-DOF robot, the dynamic model is implemented and evaluated using **MATLAB**, **Simulink**, and **Simscape Multibody**. In Simscape, physical parameters such as link mass, center of mass, and inertia are directly defined, allowing automatic computation of dynamic effects.

The derived dynamic model is utilized in simulation to analyze joint torques, verify motion feasibility, and support the design of control algorithms presented in later sections.

5 Trajectory Generation

Trajectory generation is the process of planning smooth and feasible motion profiles for a robot manipulator to move from an initial configuration to a desired final configuration. In industrial pick-and-place applications, trajectory planning ensures smooth motion, reduces mechanical stress, avoids vibrations, and guarantees accurate positioning.

In this project, trajectory generation is performed primarily in **joint space**, which is computationally efficient and well-suited for articulated manipulators such as the 6-DOF Yaskawa Motoman–style robot arm.

5.1 Joint-Space Trajectory Planning

In joint-space planning, each joint is moved independently according to a predefined time law. Let a joint variable $q(t)$ move from an initial position q_0 at time $t = 0$ to a final position q_f at time $t = T$.

To ensure smooth motion, boundary conditions are imposed on position, velocity, and acceleration.

5.2 Cubic Polynomial Trajectory

A cubic polynomial is commonly used when position and velocity constraints are specified at the start and end points. The joint trajectory is defined as:

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

with boundary conditions:

$$q(0) = q_0, \quad q(T) = q_f, \quad \dot{q}(0) = 0, \quad \dot{q}(T) = 0$$

Solving for the coefficients yields:

$$\begin{aligned} a_0 &= q_0 \\ a_1 &= 0 \\ a_2 &= \frac{3}{T^2}(q_f - q_0) \\ a_3 &= -\frac{2}{T^3}(q_f - q_0) \end{aligned}$$

Cubic trajectories provide continuous velocity profiles but result in discontinuous acceleration at the boundary points.

5.3 Quintic Polynomial Trajectory

For higher smoothness, a quintic polynomial is used when acceleration constraints are also enforced:

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5$$

with boundary conditions:

$$q(0) = q_0, \quad q(T) = q_f, \quad \dot{q}(0) = \dot{q}(T) = 0, \quad \ddot{q}(0) = \ddot{q}(T) = 0$$

Quintic trajectories ensure continuous position, velocity, and acceleration, making them ideal for industrial robotic motion.

5.4 Trajectory Profile Illustration

Figure 3 illustrates a typical joint-space trajectory showing position, velocity, and acceleration variations over time for one joint.

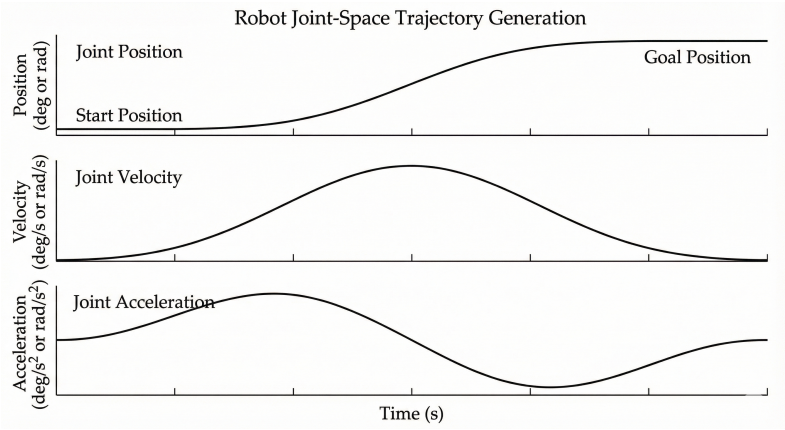

Figure 3: Typical joint-space trajectory showing smooth position, velocity, and acceleration profiles.

5.5 Multi-Joint Trajectory Execution

For the 6-DOF robot arm, individual joint trajectories are generated simultaneously using identical motion durations to ensure coordinated movement. The trajectory vectors are defined as:

$$\boldsymbol{q}(t) = \begin{bmatrix} q_1(t) & q_2(t) & \cdots & q_6(t) \end{bmatrix}^{\mathsf{T}}$$

These trajectories serve as reference inputs to the control system.

6 Control

The objective of the control system is to ensure that the robot manipulator accurately follows the desired joint trajectories generated in the previous section, despite the presence of nonlinear dynamics and gravitational effects. For industrial robotic arms, joint-space control is widely adopted due to its simplicity and effectiveness.

In this project, joint-space control strategies are implemented and evaluated using MATLAB, Simulink, and Simscape Multibody.

6.1 Joint-Space Control Framework

Let the desired joint trajectory be defined by:

$$\boldsymbol{q}_d(t), \quad \dot{\boldsymbol{q}}_d(t), \quad \ddot{\boldsymbol{q}}_d(t)$$

The joint tracking error and its derivative are defined as:

$$\boldsymbol{e}(t) = \boldsymbol{q}_d(t) - \boldsymbol{q}(t), \qquad \dot{\boldsymbol{e}}(t) = \dot{\boldsymbol{q}}_d(t) - \dot{\boldsymbol{q}}(t)$$

The controller computes the actuator torque vector $\boldsymbol{\tau}$ to minimize these errors.

6.2 Proportional-Derivative (PD) Control

A PD controller is first implemented due to its simplicity and robustness. The control law is given by:

$$\boldsymbol{\tau} = \mathbf{K}_p \boldsymbol{e} + \mathbf{K}_d \dot{\boldsymbol{e}}$$

where:

- $\mathbf{K}_p$ is the proportional gain matrix
- $\mathbf{K}_d$ is the derivative gain matrix

The PD controller provides stable tracking for slow to moderate motions; however, its performance degrades when dynamic coupling and gravity effects become significant.

6.3 Computed Torque Control

To improve tracking performance, a **computed torque control (CTC)** approach is employed. This method uses the robot dynamic model to cancel nonlinear effects, resulting in approximately linear error dynamics.

The control law is:

$$\boldsymbol{\tau} = \mathbf{M}(\boldsymbol{q}) \left(\ddot{\boldsymbol{q}}_d + \mathbf{K}_d \dot{\boldsymbol{e}} + \mathbf{K}_p \boldsymbol{e} \right) + \mathbf{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \dot{\boldsymbol{q}} + \mathbf{G}(\boldsymbol{q})$$

Substituting this torque into the robot dynamics yields decoupled second-order error dynamics:

$$\ddot{\boldsymbol{e}} + \mathbf{K}_d \dot{\boldsymbol{e}} + \mathbf{K}_p \boldsymbol{e} = \mathbf{0}$$

Proper selection of the gain matrices ensures fast convergence and stable tracking.

7 Simulation and Results

Simulation plays a critical role in validating the kinematic, dynamic, and control models of the developed 6-DOF pick-and-place robot arm. In this project, simulations were conducted using **MATLAB**, **Simulink**, and **Simscape Multibody** to analyze robot motion, joint trajectories, torque requirements, and control performance prior to physical implementation.

7.1 MATLAB Kinematic Simulation

The forward and inverse kinematics models derived in earlier sections were implemented in MATLAB scripts. Given desired end-effector positions and orientations, the inverse kinematics algorithm computes the corresponding joint angles, which are then validated by forward kinematics to confirm correctness of the solution.

Multiple inverse kinematics solutions were observed due to the redundant configurations of the robot arm. Valid solutions were selected based on joint limits and physical feasibility. These simulations verified the correctness of the kinematic model and ensured accurate end-effector positioning.

7.2 Trajectory and Control Simulation in Simulink

The joint-space trajectories generated using cubic and quintic polynomial interpolation were implemented in Simulink. These trajectories served as reference inputs to the joint-space controllers.

Both PD control and computed torque control strategies were simulated. The PD controller provided stable motion for slow trajectories, while the computed torque controller achieved superior tracking performance with reduced steady-state error and smoother joint motion. Simulation results demonstrated accurate trajectory tracking across all six joints, even during rapid pick-and-place movements.

7.3 Dynamic Simulation Using Simscape Multibody

A detailed multibody model of the robot arm was developed in Simscape Multibody using physical components such as rigid links, revolute joints, and actuators. Link parameters including mass, center of mass, and inertia were assigned based on realistic assumptions consistent with industrial manipulators.

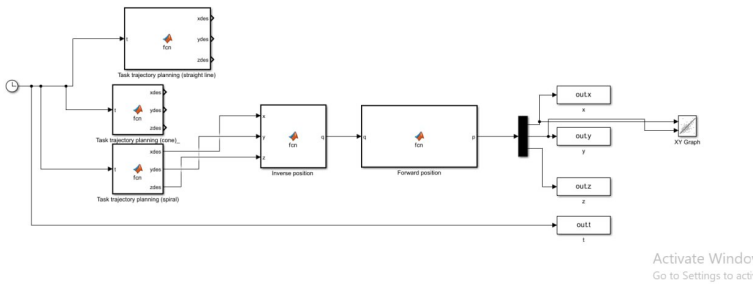

Figure 4: Simulation in Simulinkθ_2 and θ_3.

Simscape automatically computed the dynamic interactions between joints, including gravity, inertial forces, and joint coupling. The results validated the dynamic model and demonstrated realistic joint torque profiles during operation.

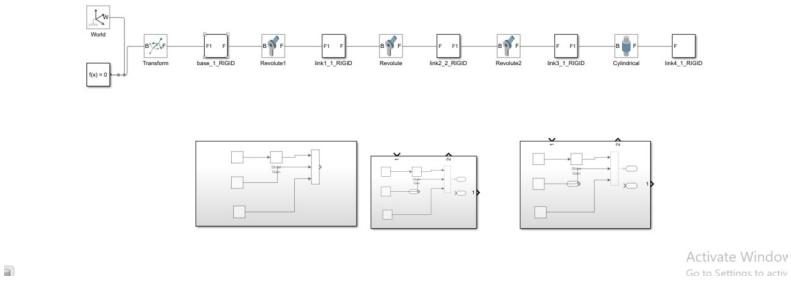

Figure 5: Dynamic Simulation Using Simscape Multibody θ_2 and θ_3.

7.4 Mechanical Design Verification

The mechanical structure of the robot arm was designed in **SolidWorks** to verify link dimensions, joint alignment, and overall workspace. The CAD model ensured that the kinematic assumptions used in analysis were physically realizable and suitable for pick-and-place tasks.

Conclusion

This project presented the systematic design, modeling, simulation, and control of a 6-DOF pick-and-place robotic arm inspired by the Yaskawa Motoman industrial architecture. Forward and inverse kinematic models were developed using the Denavit–Hartenberg convention, enabling accurate pose computation and joint-space motion planning. Dynamic modeling was formulated using the Euler–Lagrange approach and utilized for control design. Joint-space

trajectory generation ensured smooth and feasible motion, while PD and computed torque controllers provided effective trajectory tracking. Simulation results obtained from MATLAB, Simulink, and Simscape Multibody verified the correctness and robustness of the proposed models. The integration of analytical modeling with multibody simulation demonstrated that the developed robot system is capable of performing precise and stable pick-and-place operations.