IMPERIAL

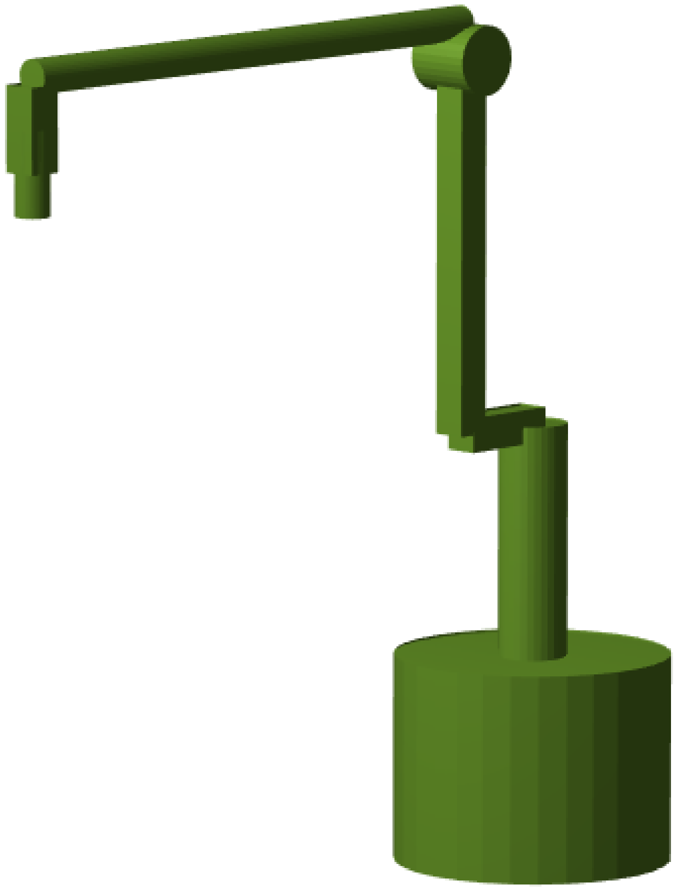
Group Coursework

Medical Robotics and Instrumentation

Group 14
03/12/2024

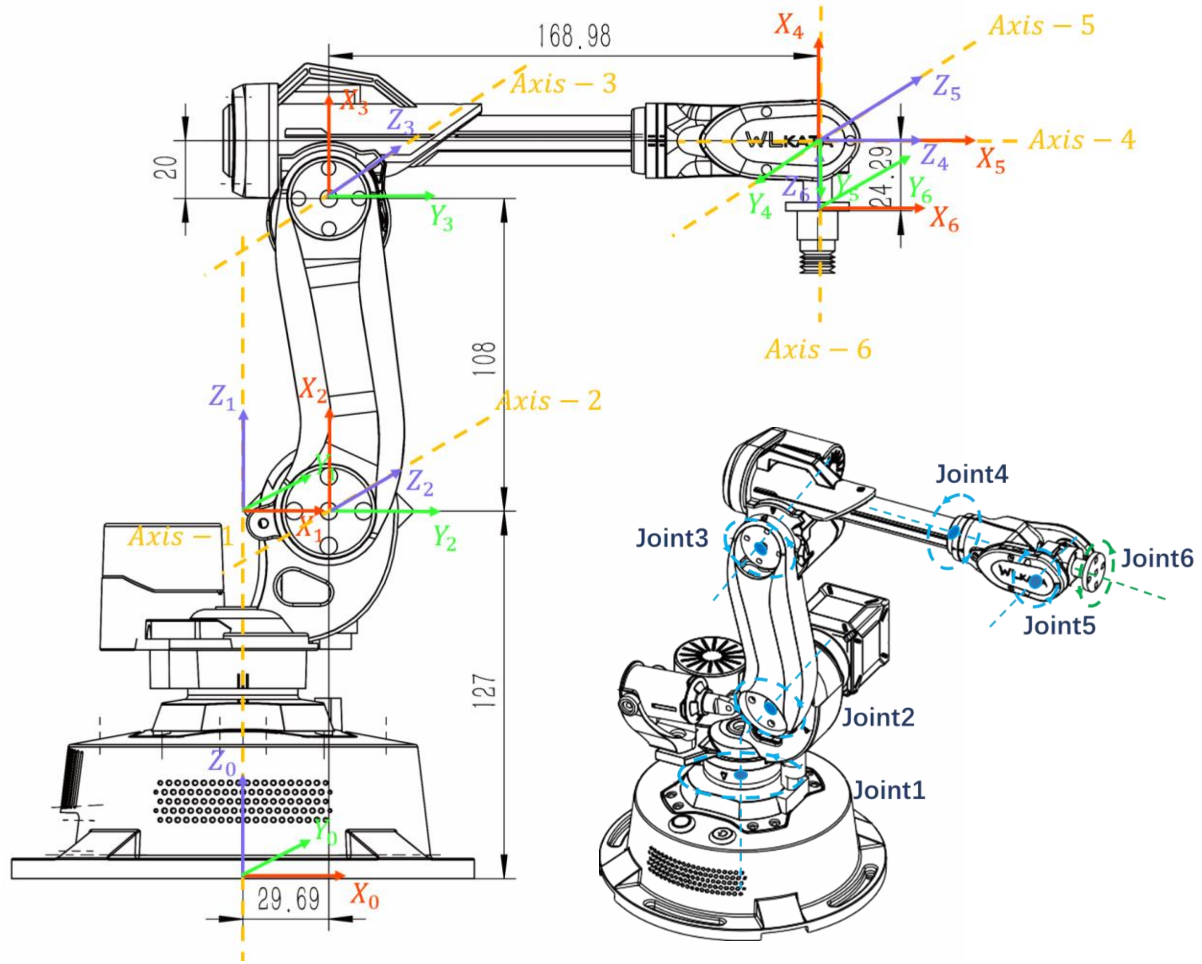
Task I

Robot Modeling



Task I – Robot Modeling

01. Modified Denavit-Hartenberg Tables (without End-effector)



- Identify the Joints, Z-axis and Direction
- $R0 \parallel R1 \mapsto R2 \parallel R3 \mapsto R4 \perp R5 \mapsto R6$
- Identify X-axis and Y-axis
- Identify other variables: a, α, d and θ

	a	α	*d*	θ
Joint 1	0	0	0.12700	0
Joint 2	0.02969	- Pi / 2	0	- Pi / 2
Joint 3	0.10800	0	0	0
Joint 4	0.02000	- Pi / 2	0.16898	0
Joint 5	0	Pi / 2	0	Pi / 2
Joint 6	0	Pi / 2	- 0.02429	0

MDH Table (without End-effector)

Task I – Robot Modeling
02. Modified Denavit-Hartenberg Tables (with End-effector)

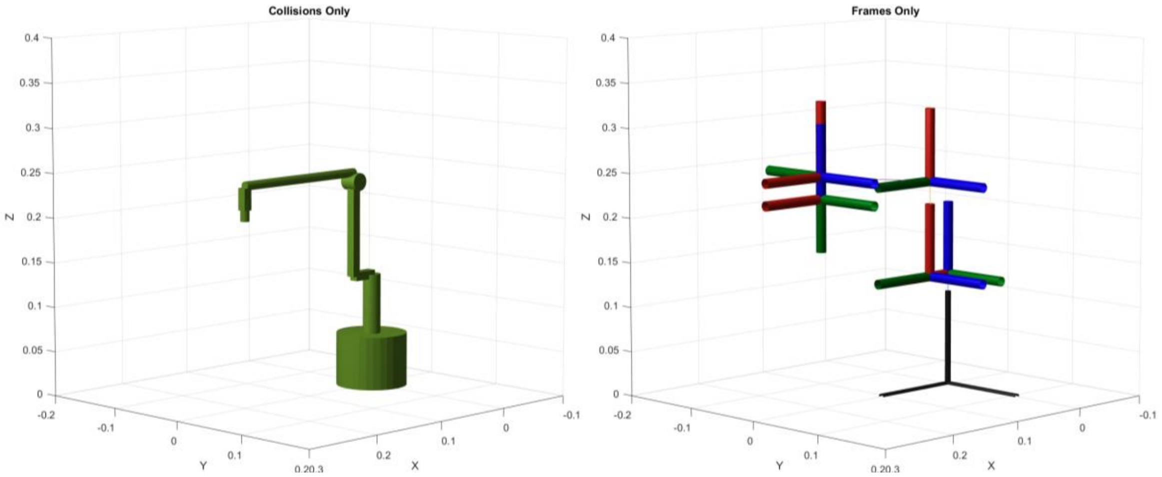
Two-Finger Gripper

	a	α	d	θ
Joint 1	0	0	0.12700	0
Joint 2	0.02969	- Pi / 2	0	- Pi / 2
Joint 3	0.10800	0	0	0
Joint 4	0.02000	- Pi / 2	0.16898	0
Joint 5	0	Pi / 2	0	Pi / 2
Joint 6	0	Pi / 2	- 0.02429	0
End-effector	0	0	-0.07200	0

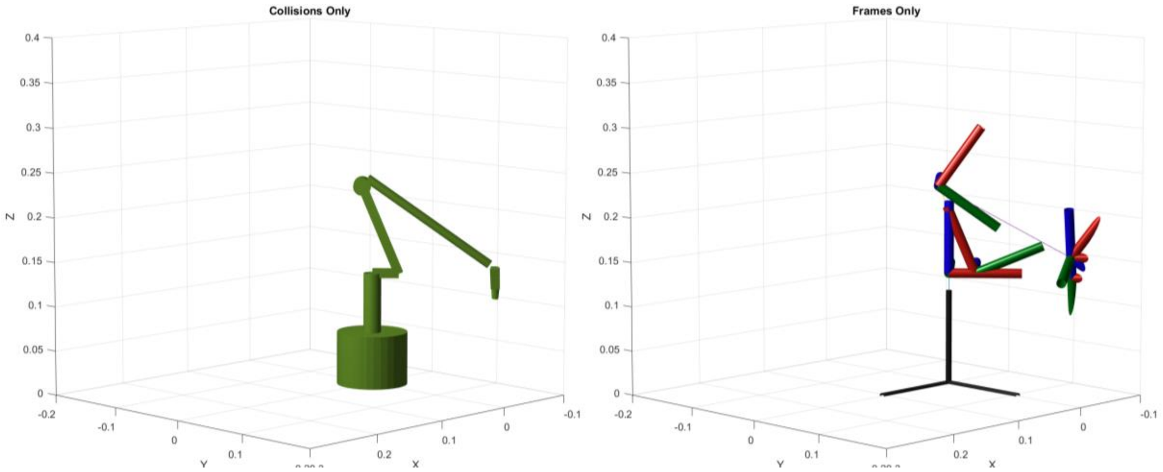
Task I – Robot Modeling
03. Visualization via MATLAB

- Visualization contains 2 parts:
  - **Collision View** ⇒ For Link Boundaries
  - **Frame View** ⇒ For Coordinate Frame Alignment

- Confirmed correctness of robot kinematics and physical modeling
- Ready for motion planning and control tasks.



Home Configuration



A Random Generated Joint Angle Set within the Limits

Task II

Joint Space Control

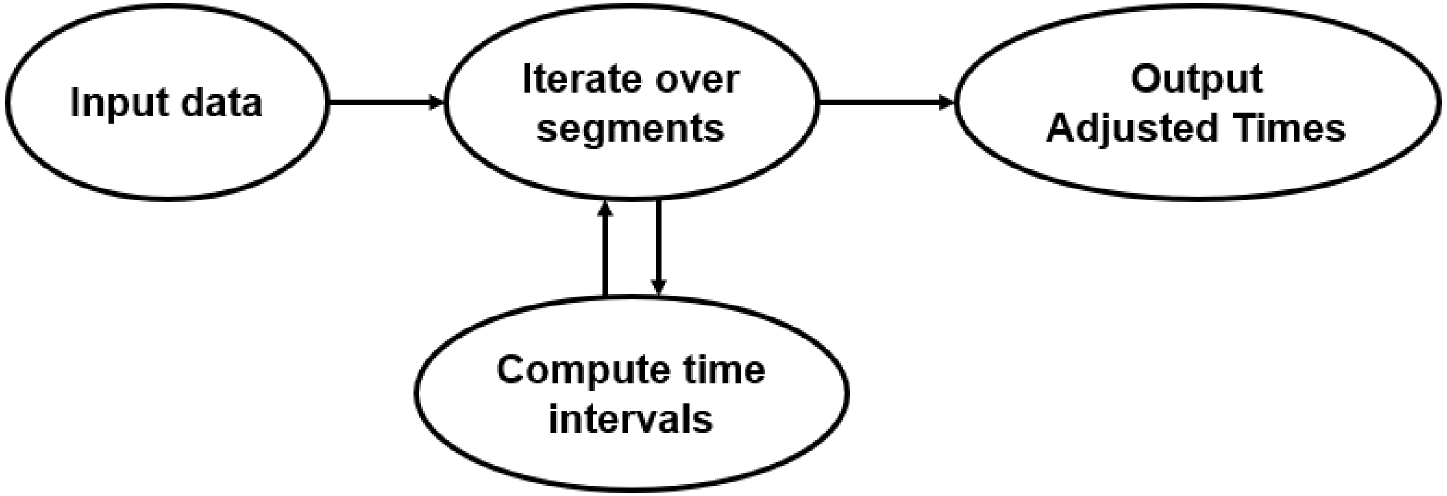
Task II – Joint Space Control

01. Quintic Polynomial Interpolation with Dynamic Time Adjustment

- **Plan the trajectory**
  Generate a smooth quintic polynomial trajectory based on the robot's key joint positions

- **Adjust time dynamically**
  To meet joint velocity constraints, the angular differences between key points are calculated, and the time intervals are dynamically adjusted, ensuring the motion is both smooth and safe.

- **Safety Coefficient (Scaling Factor)**
  Introduced in the calculation of optimal time intervals to ensure joint velocities remain well within limits.



Task II – Joint Space Control
02. Quintic Polynomial joint-space method

Quintic polynomial interpolation → **Forward kinematics** →

Configuration | **Joint Space** | **Cartesian Space**

$Q_{Home}(\theta)$

$Q_{Inter}(\theta)$

$Q_{Final}(\theta)$

Boundary Conditions

$$\theta(t_0) = \theta_0; \dot{\theta}(t_0) = 0; \ddot{\theta}(t_0) = 0$$
$$\theta(t_f) = \theta_f; \dot{\theta}(t_f) = 0; \ddot{\theta}(t_f) = 0$$

Polynomial Coefficient Matrix

$$A \cdot \text{coeffs} = b$$

$$A = \begin{matrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{matrix}, \quad b = \begin{matrix} \theta_0 \\ \dot{\theta}(t_0) = 0 \\ \ddot{\theta}(t_0) = 0 \\ \theta_f \\ \dot{\theta}(t_f) = 0 \\ \ddot{\theta}(t_f) = 0 \end{matrix}$$

Substituting Polynomial

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5$$
$$\dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4$$
$$\ddot{\theta}(t) = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3$$

$Q_{Home}(\theta)$

$Q_{Inter}(\theta)$

$Q_{Final}(\theta)$

General DH transformation

$${}^{i-1}T_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Overall transformation matrix

$${}^0T_6 = \begin{bmatrix} & & & \\ & & & \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Translation vector

$$Q_t = $$

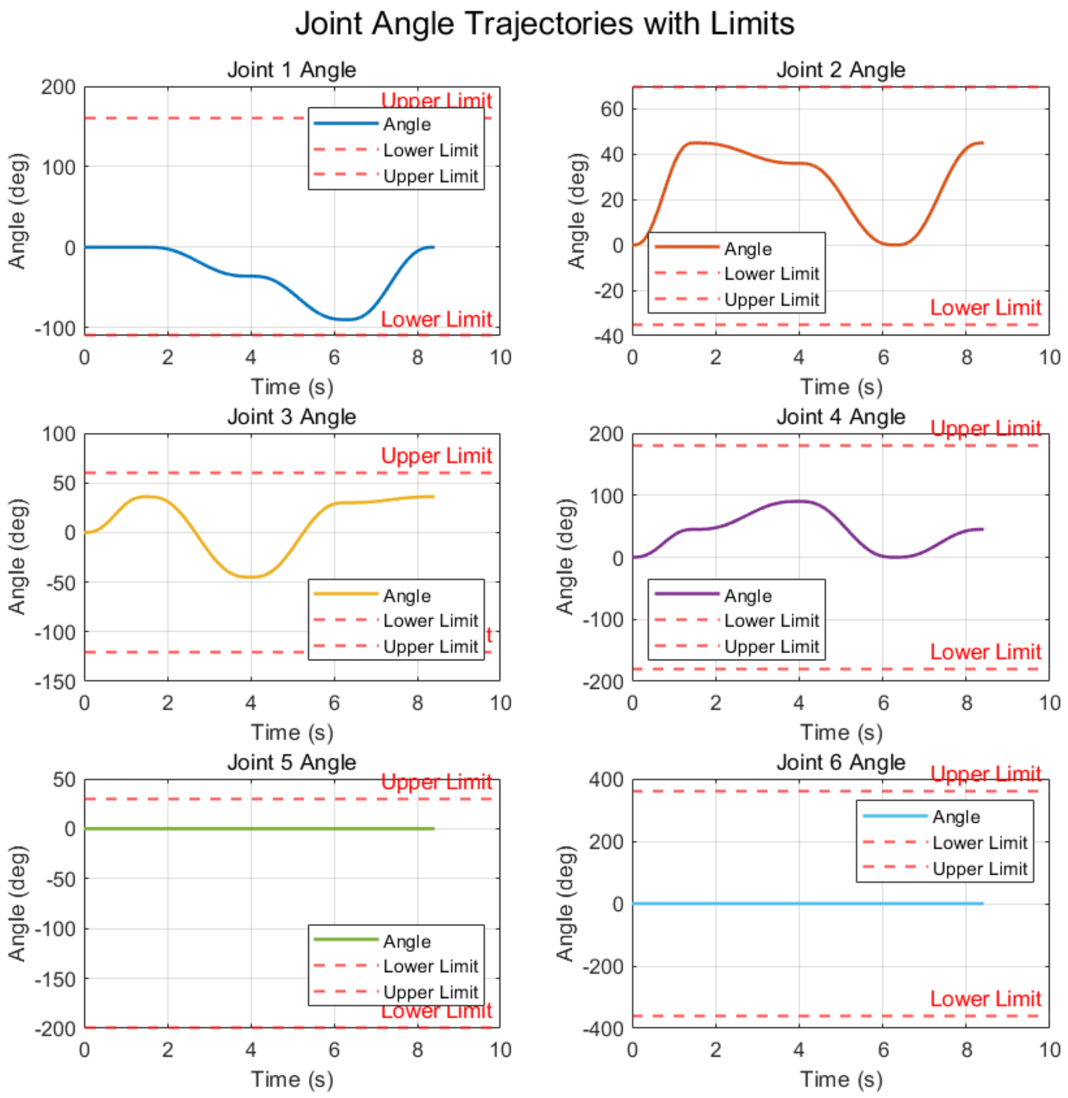
P_{Home}

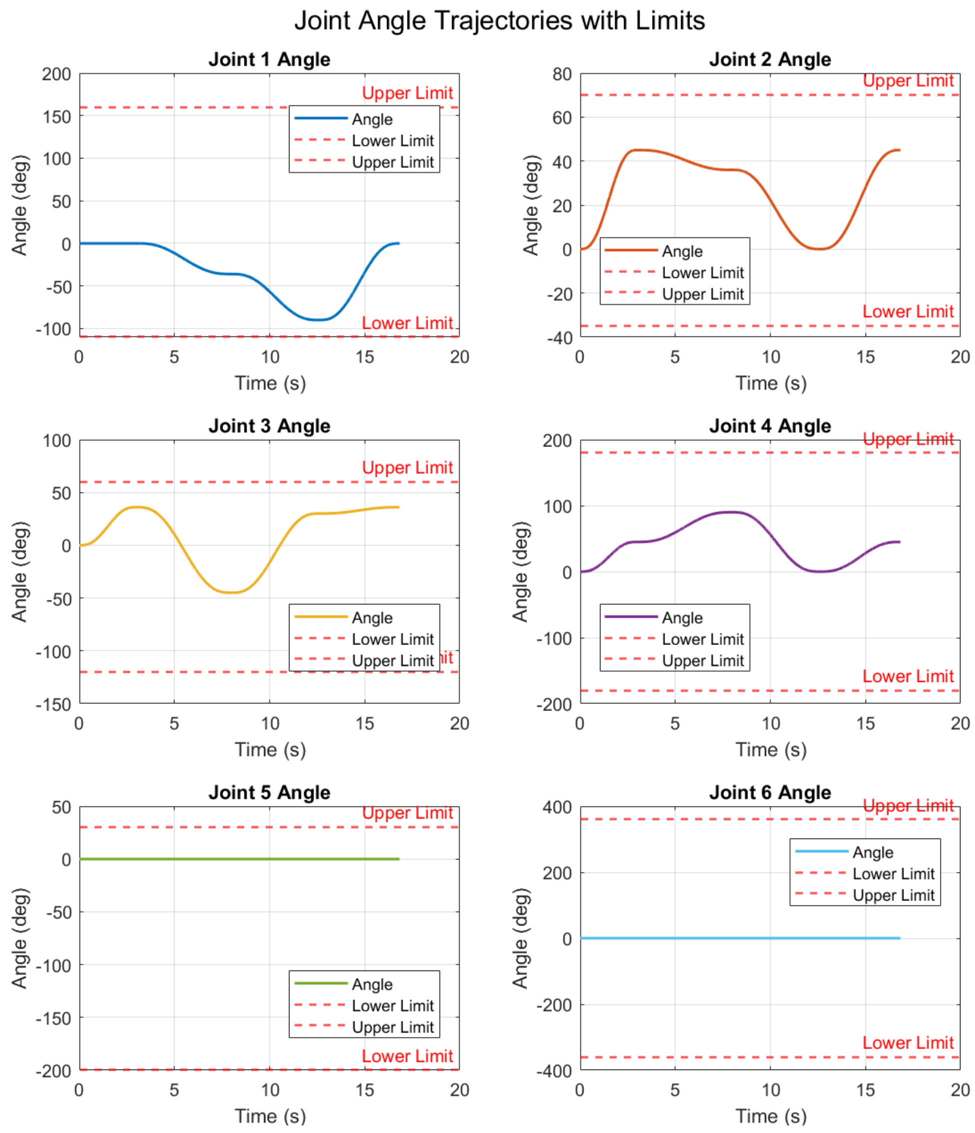
P_{Inter}

P_{Final}

Task II – Joint Space Control

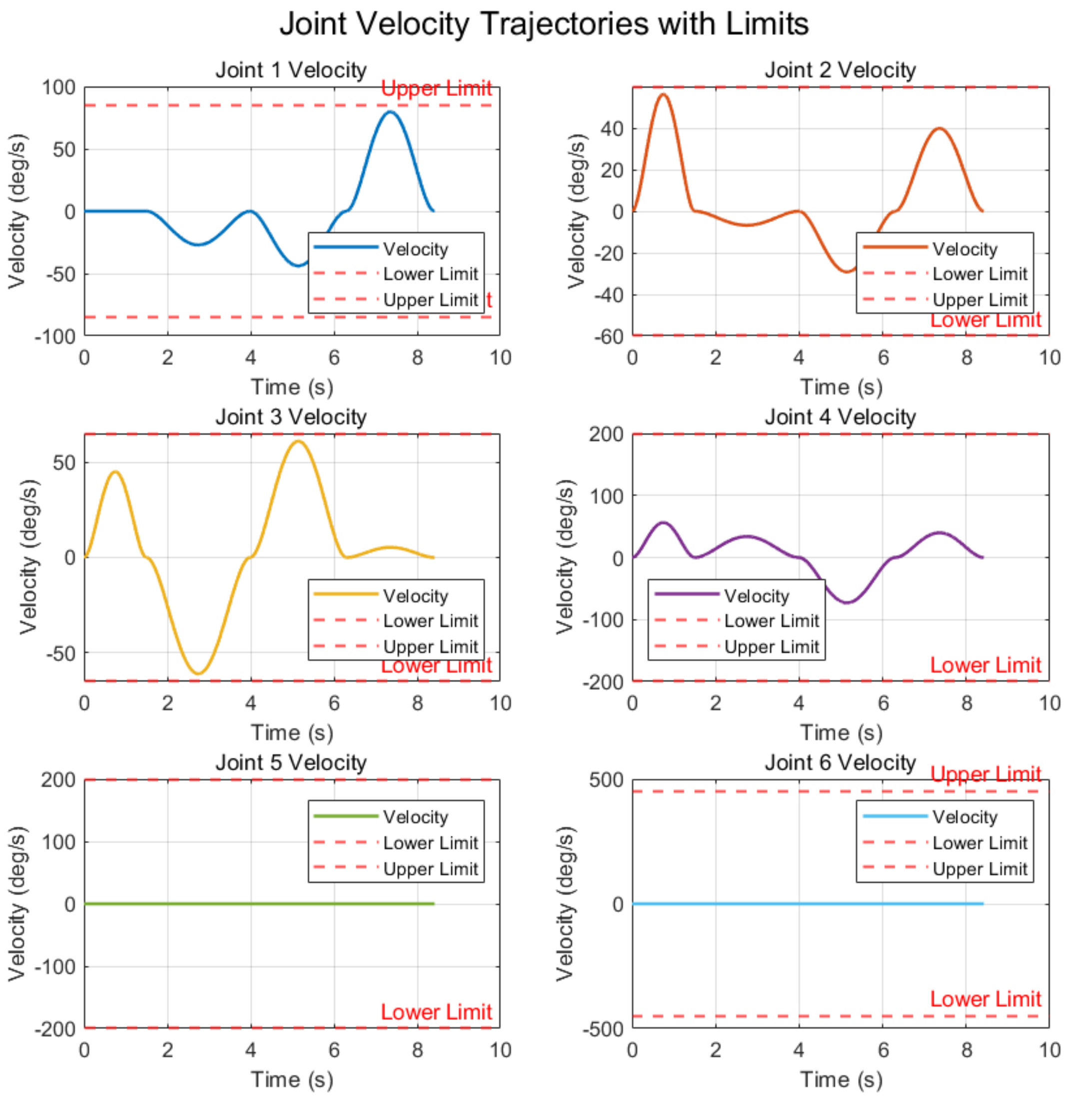
03. Result– Joint position for each joint (L.MATLAB R.CoppeliaSim)

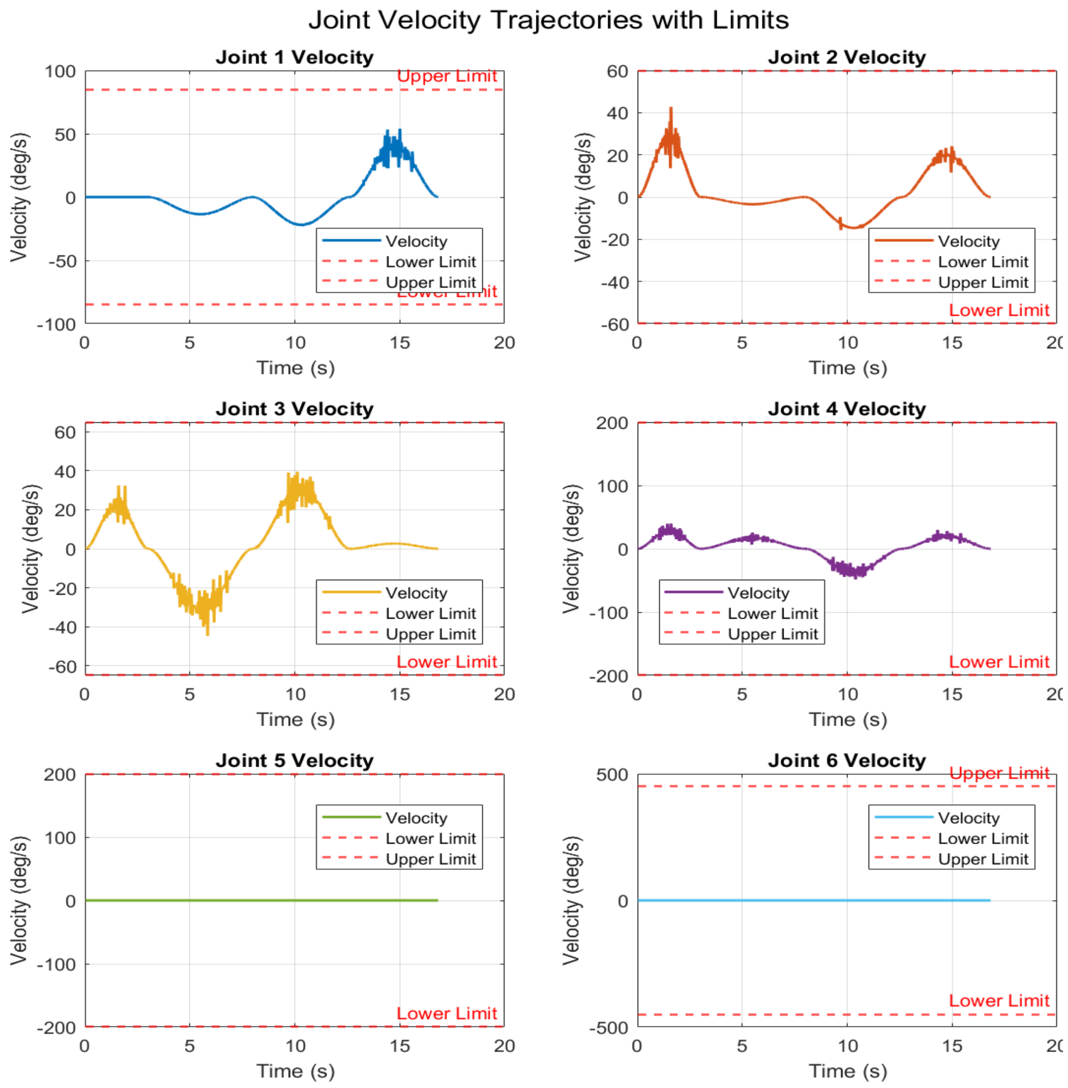




Task II – Joint Space Control

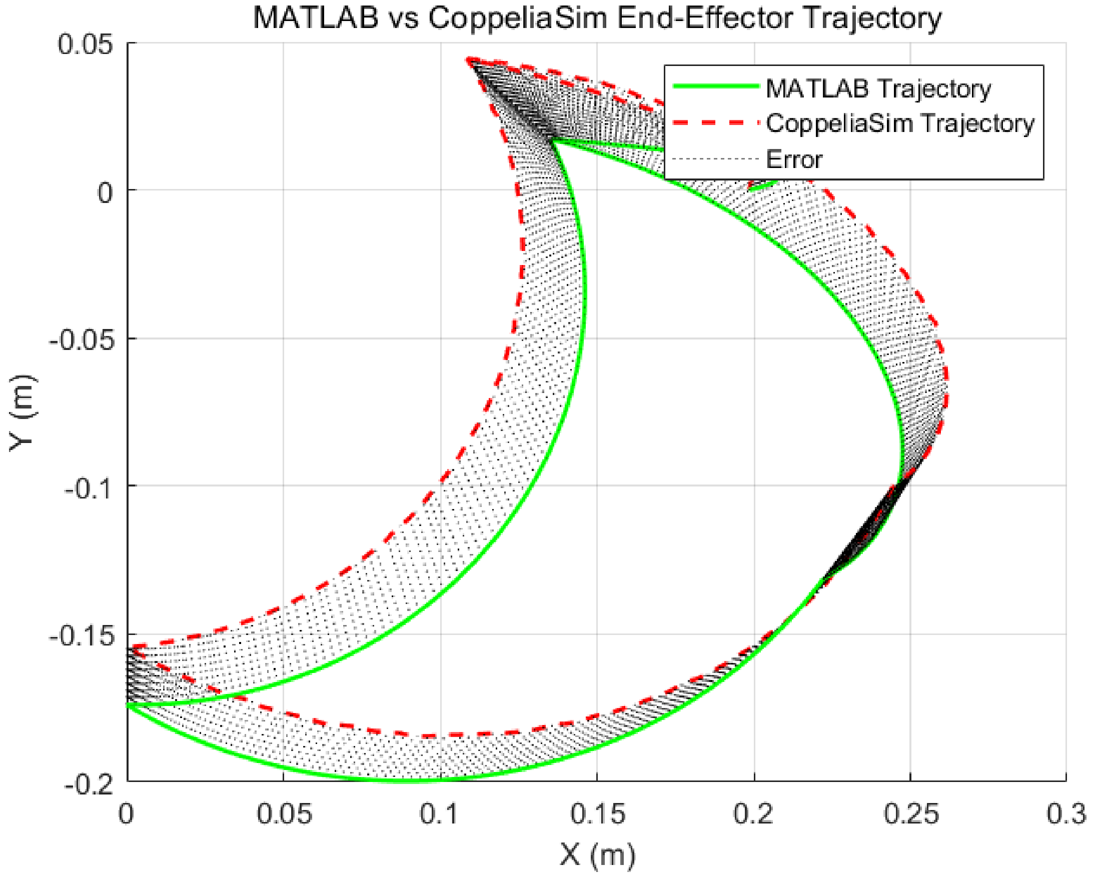
04. Result– Joint velocity for each joint (L.MATLAB R.CoppeliaSim)

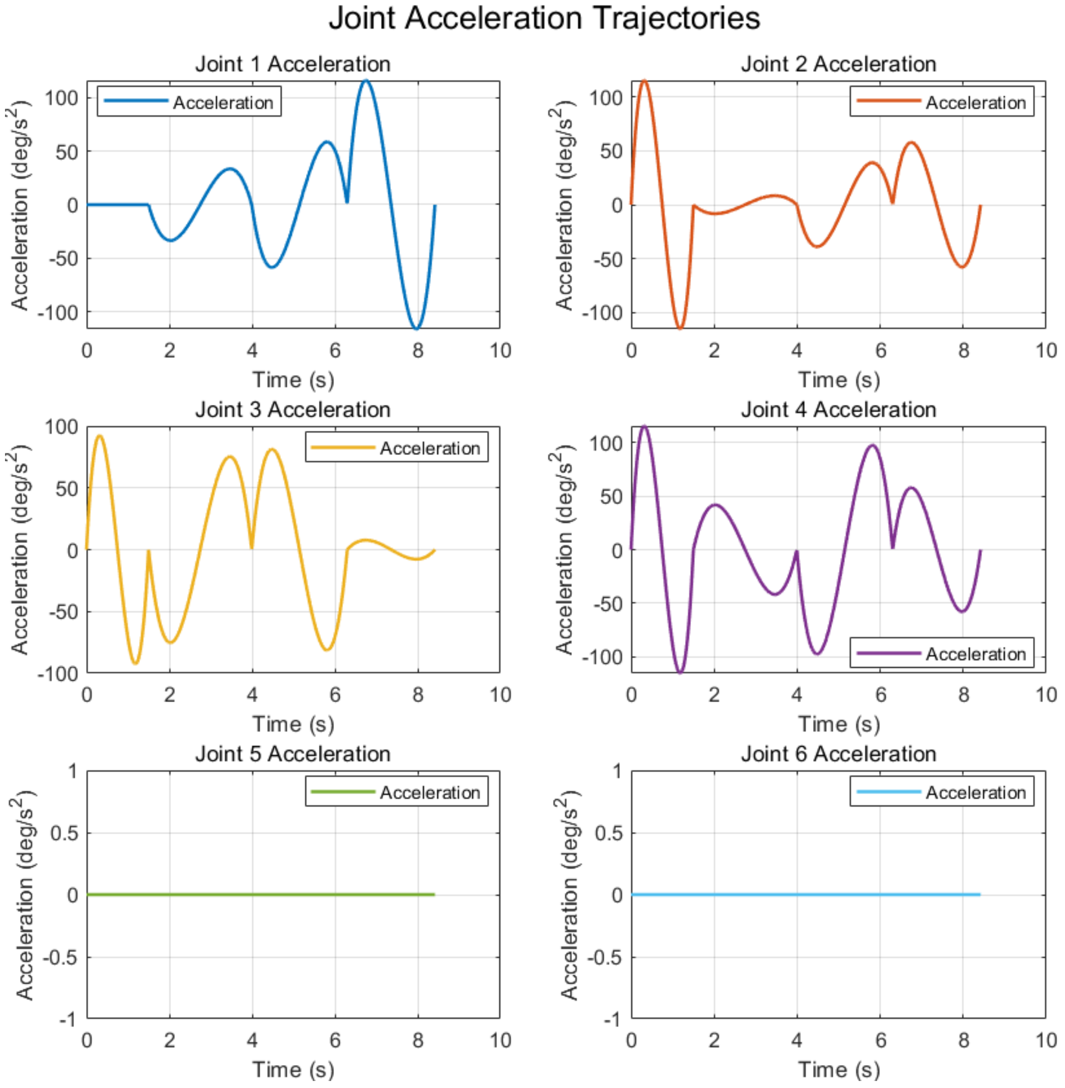




Task II – Joint Space Control

05. Result – Joint acceleration and trajectory of the robot tip





Task II – Joint Space Control

06. Videos on Mirobot (real) and CoppeliaSim (simulation)

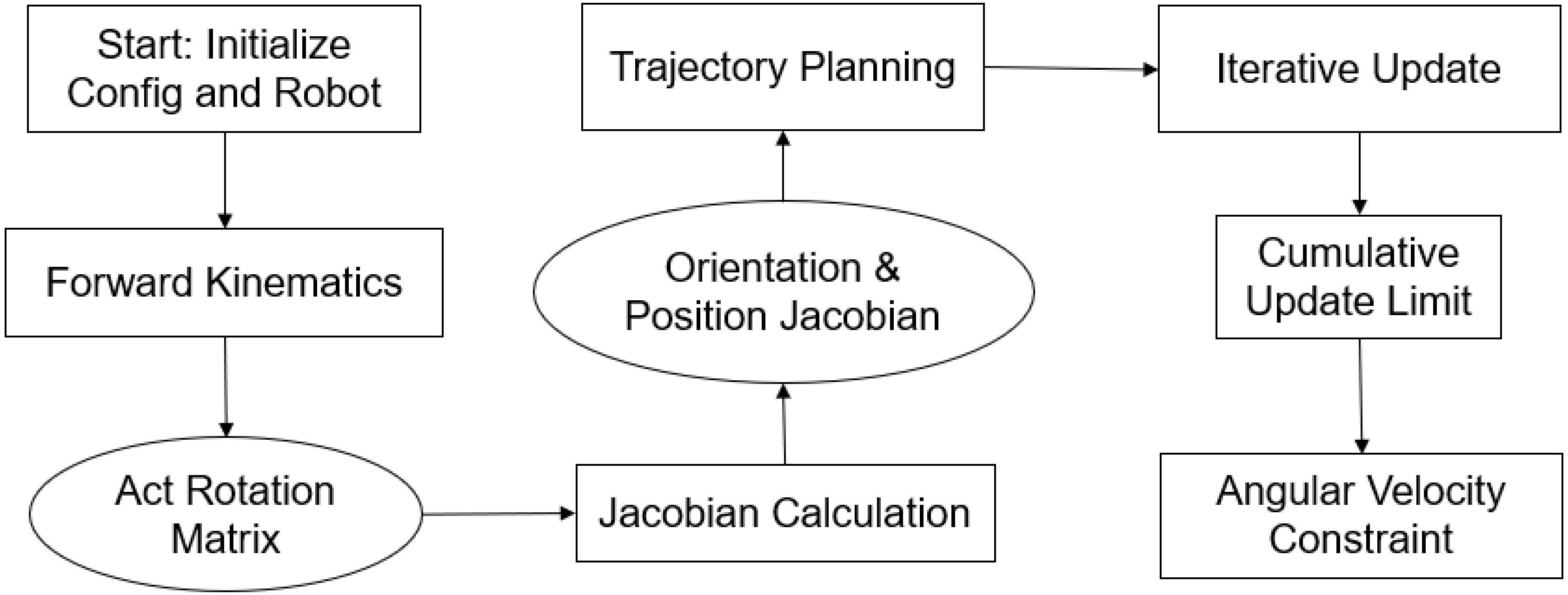
Task III

Cartesian Space Control

Task III – Cartesian Space Control
01. Jacobian matrix method

- **Extraction of Z-Axis Direction Vector**
  The third column of the rotation matrix (i.e., the Z-axis direction vector) is extracted, and the sensitivity of directional changes to joint angle variations (Jacobian matrix) is calculated.

- **Angular velocity limitation**
  The cumulative update amount is set, and the joint velocity at each step is limited to ensure that the angular velocity does not exceed the maximum velocity constraints.



Task III – Cartesian Space Control

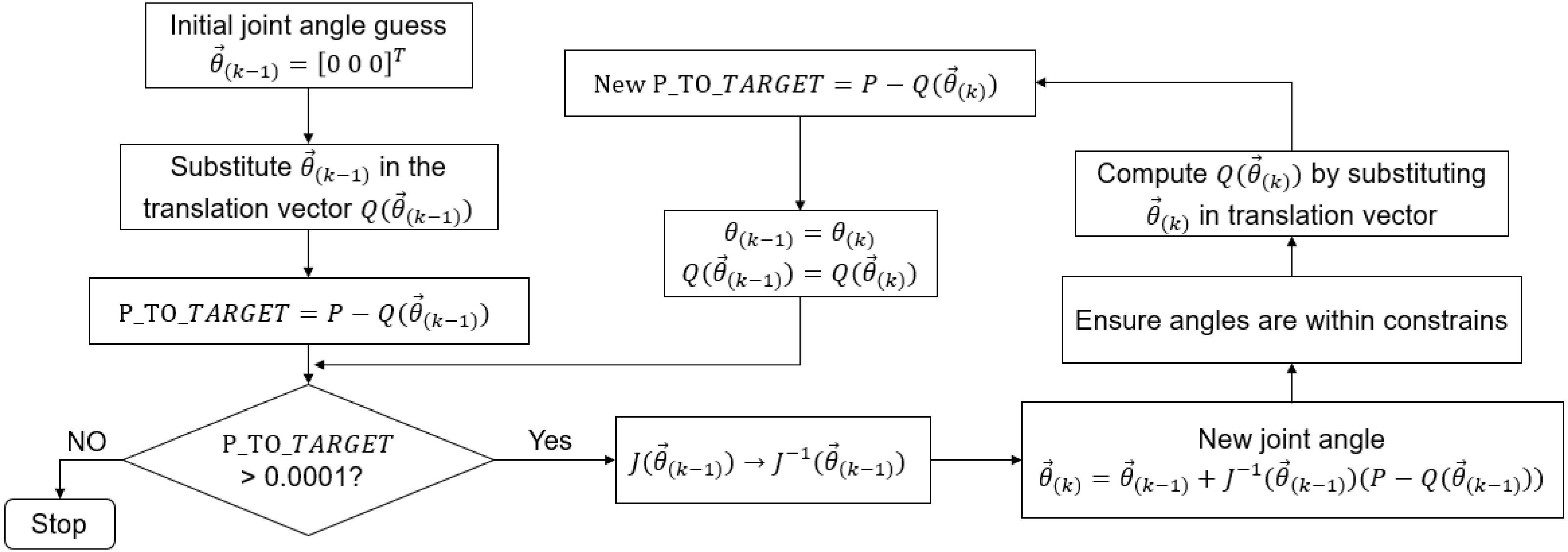
02. Newton-Raphson method

- **Iterative Process**

  The core idea is to update joint angles iteratively using the Jacobian pseudo-inverse and the current error, reducing the error until convergence or the maximum iterations are reached.

- **Iterative Optimization**

  Refines initial guesses to approach accurate solutions, suitable for solving complex nonlinear equations.



Task III – Cartesian Space Control
03. Cartesian Space Control method

Inverse Kinematics → **Forward kinematics**

Cartesian Space | **Joint Space** | **Cartesian Space**

P_{Hime} | $Q_{Home}(\theta)$ | P_{Home}

P_{Inter} | $Q_{Inter}(\theta)$ | P_{Inter}

P_{Final} | $Q_{Final}(\theta)$ | P_{Final}

Jacobian matrix

$$\mathbf{J}(\vec{\theta})=\frac{\partial \mathbf{Q}(\vec{\theta})}{\partial \vec{\theta}}=\begin{bmatrix} \partial q_1(\vec{\theta})/\partial\theta_1 & \partial q_1(\vec{\theta})/\partial\theta_2 & \cdots & \partial q_1(\vec{\theta})/\partial\theta_n \\ \partial q_2(\vec{\theta})/\partial\theta_1 & \partial q_2(\vec{\theta})/\partial\theta_2 & \cdots & \partial q_2(\vec{\theta})/\partial\theta_n \\ \cdots & \cdots & \cdots & \cdots \\ \partial q_N(\vec{\theta})/\partial\theta_1 & \partial q_N(\vec{\theta})/\partial\theta_2 & \cdots & \partial q_N(\vec{\theta})/\partial\theta_n \end{bmatrix}$$

Algebraic equation

$$\vec{\theta}_{(k)}=\vec{\theta}_{(k-1)}+\mathbf{J}^{-1}\left(\vec{\theta}_{(k-1)}\right)\left(\mathbf{P}-\mathbf{Q}\left(\vec{\theta}_{(k-1)}\right)\right)$$

Newton-Raphson method

$$\underbrace{\theta_{(0)} \rightarrow p(\theta_{(0)}) \rightarrow p'(\theta_{(0)})}_{k=0,\ \theta_0 \text{ is a guess}} \rightarrow \underbrace{\theta_{(1)} \rightarrow p(\theta_{(1)}) \rightarrow p'(\theta_{(1)})}_{k=1} \rightarrow \underbrace{\theta_{(2)} \rightarrow p(\theta_{(2)})}_{k=2}, \ldots$$

General DH transformation

$${}^{i-1}T_i=\begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Overall transformation matrix

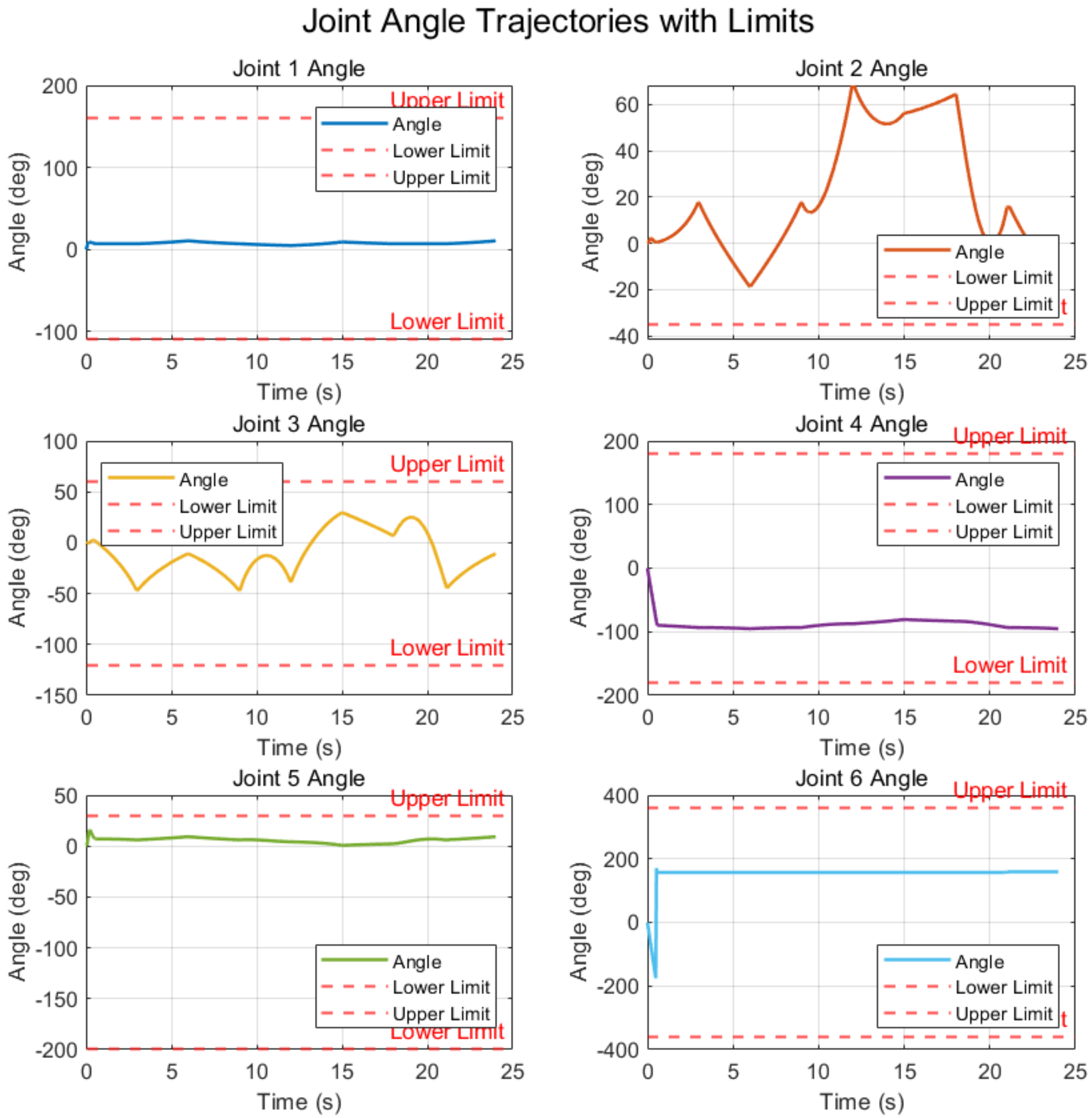
$${}^{0}T_6=\begin{bmatrix} & & & \\ & & & \\ & & & \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

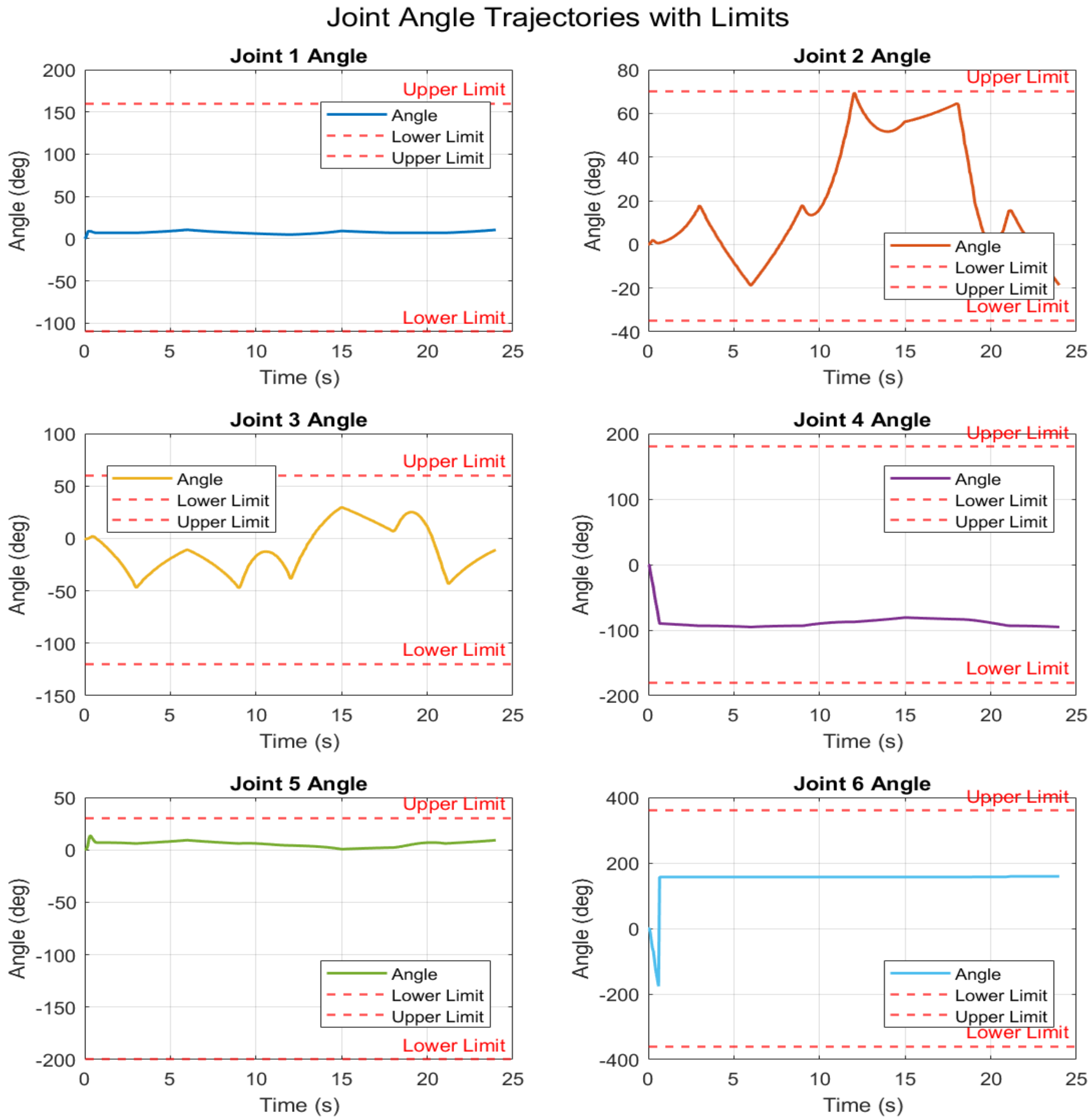
Translation vector

$$Q_t=$$

Task III – Cartesian Space Control

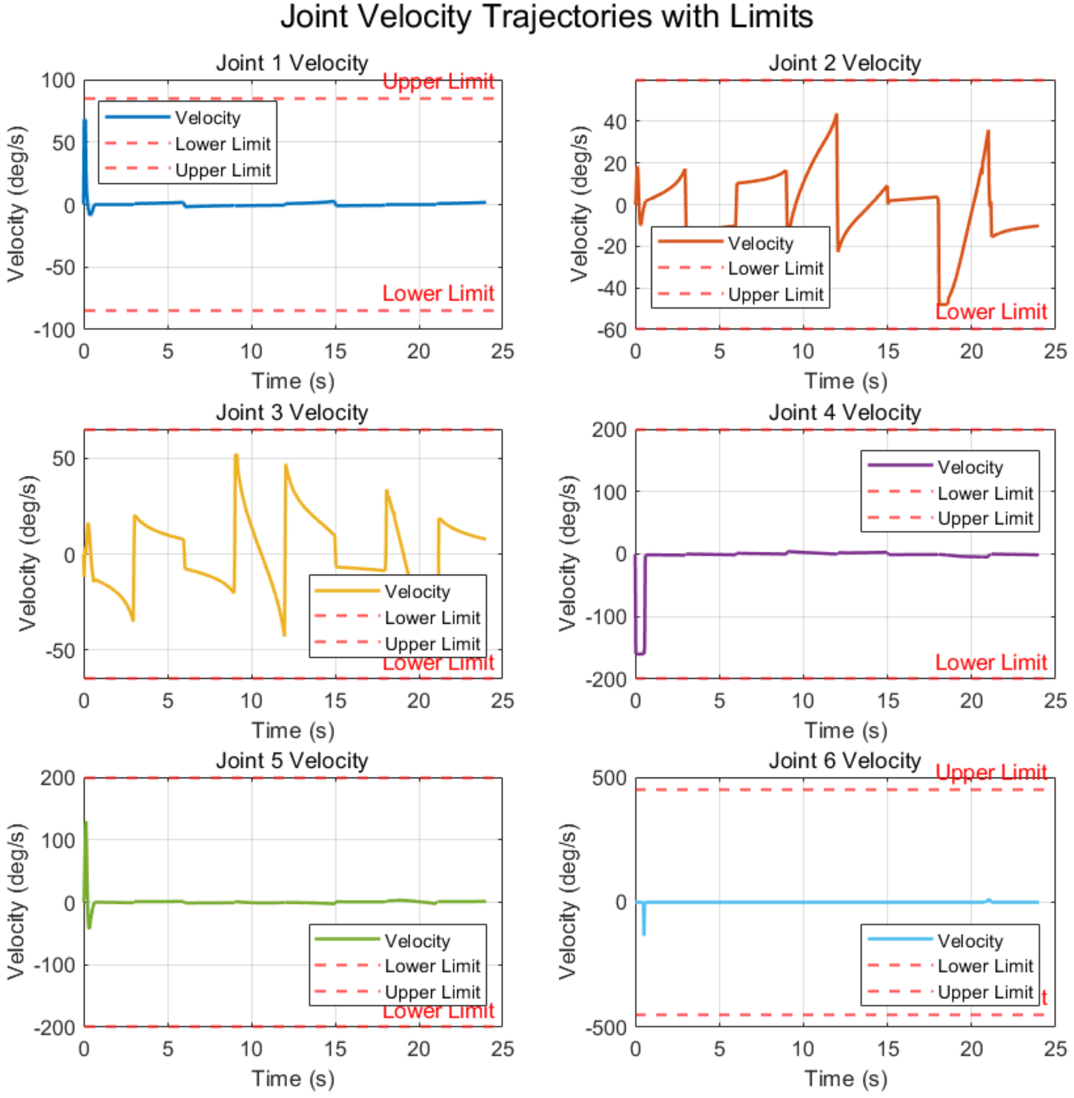
04. Result – Joint position of each joint (L.MATLAB R.CoppeliaSim)

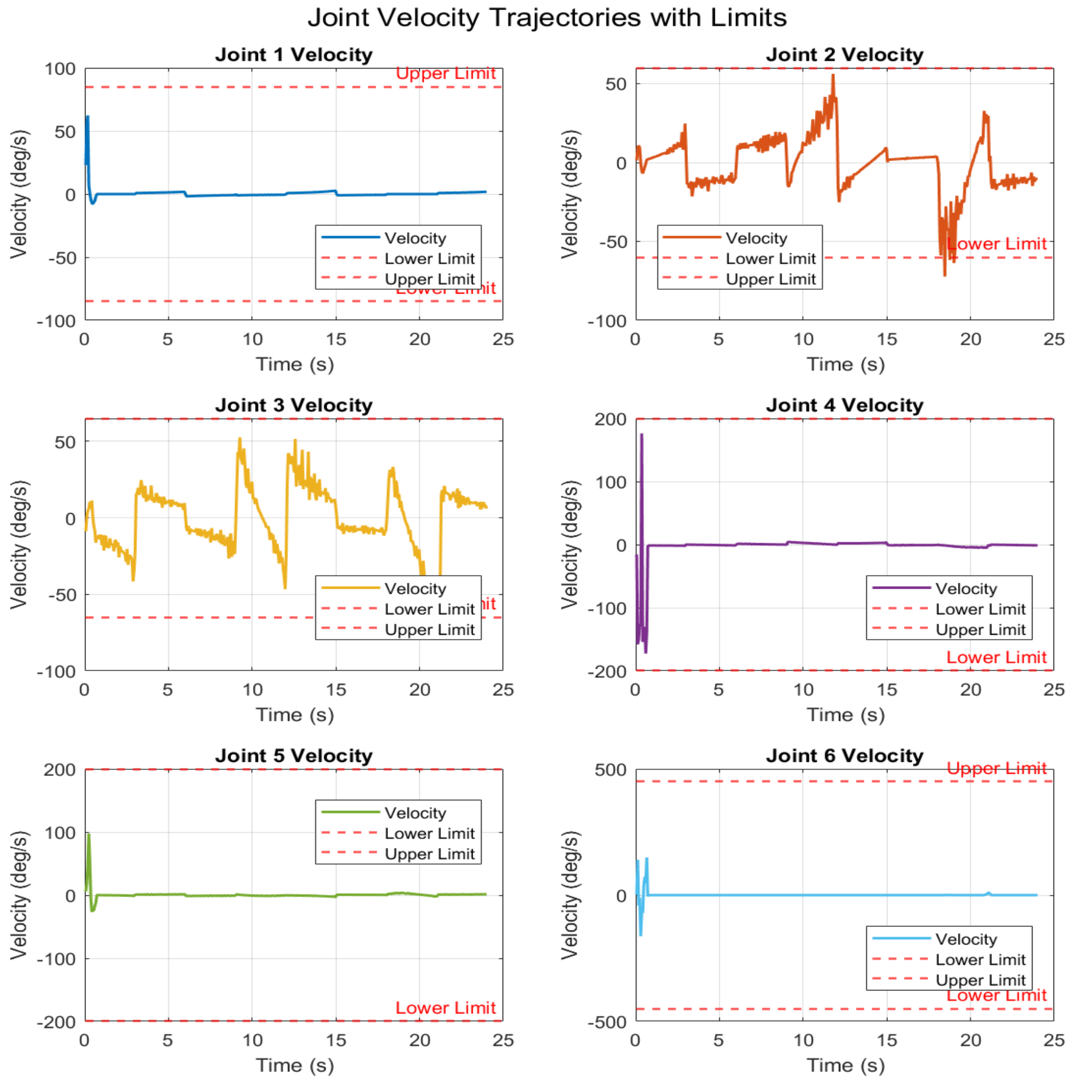




Task III – Cartesian Space Control

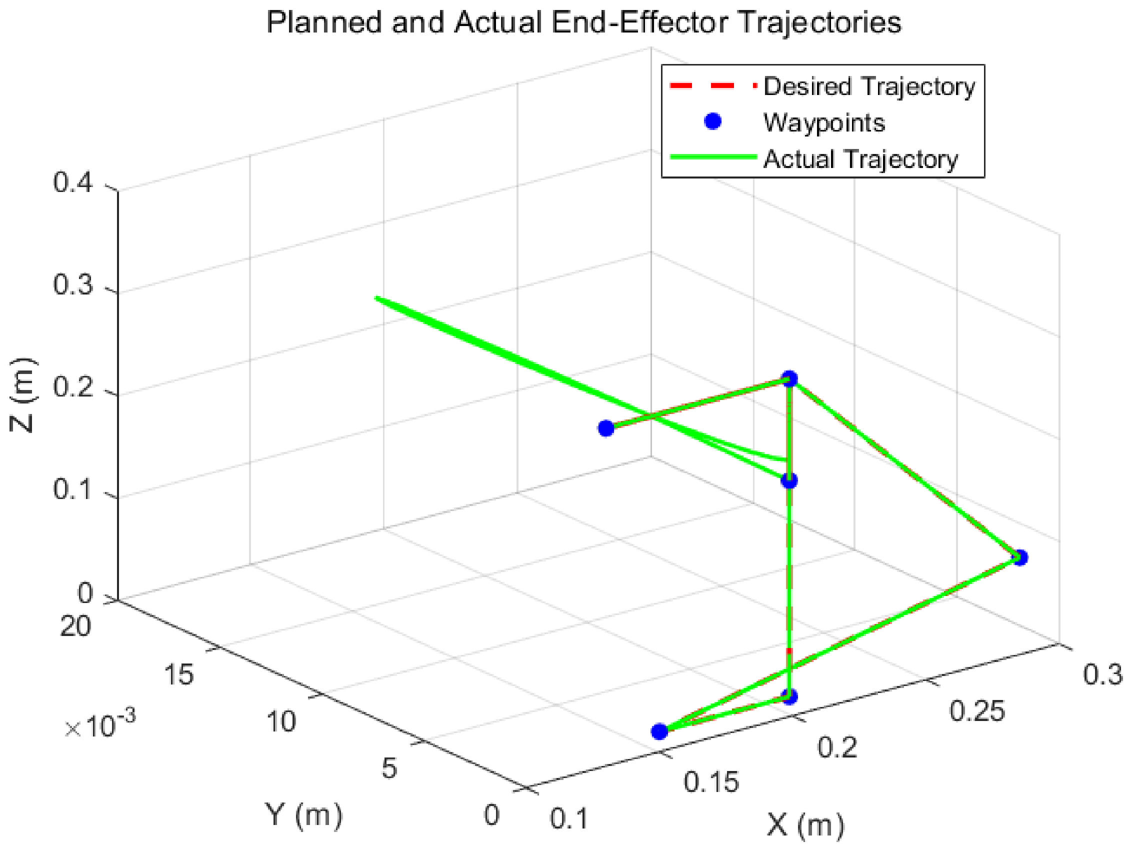
05. Result – Joint speed of each joint (L.MATLAB R.CoppeliaSim)

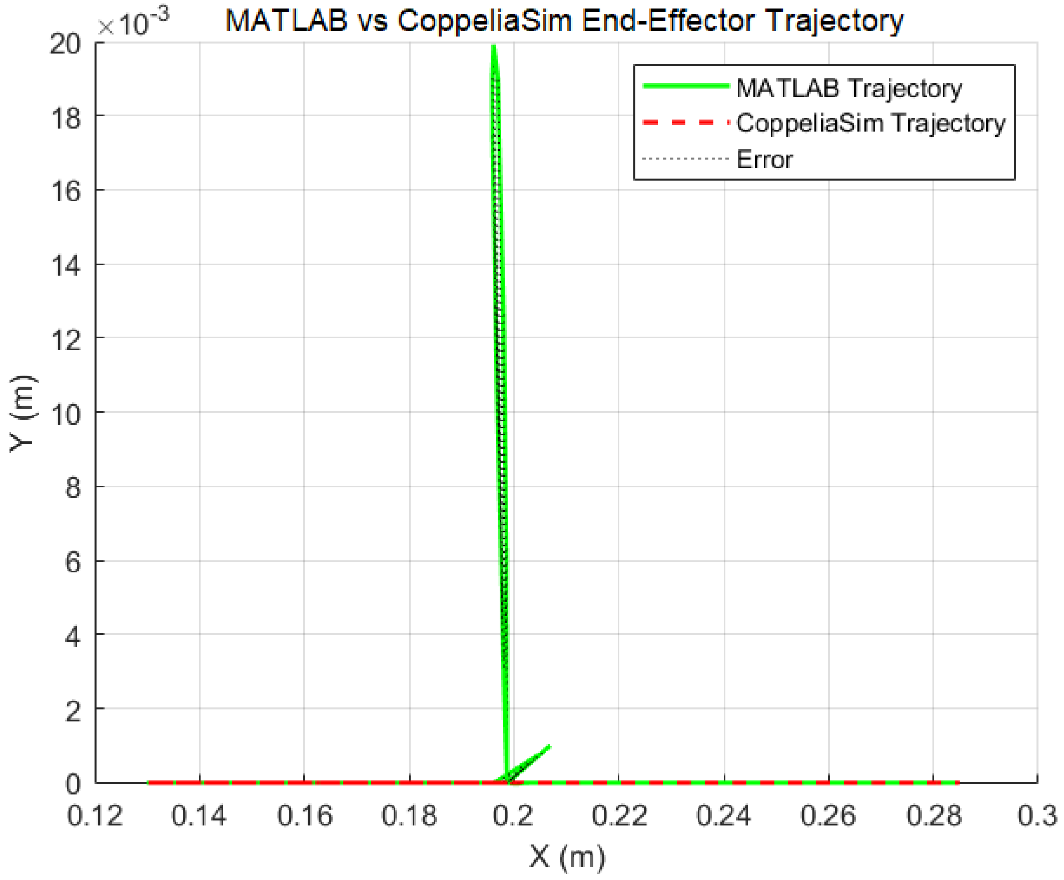




Task III – Cartesian Space Control

06. Result – Planned and actual robot tip trajectories





Task III – Cartesian Space Control

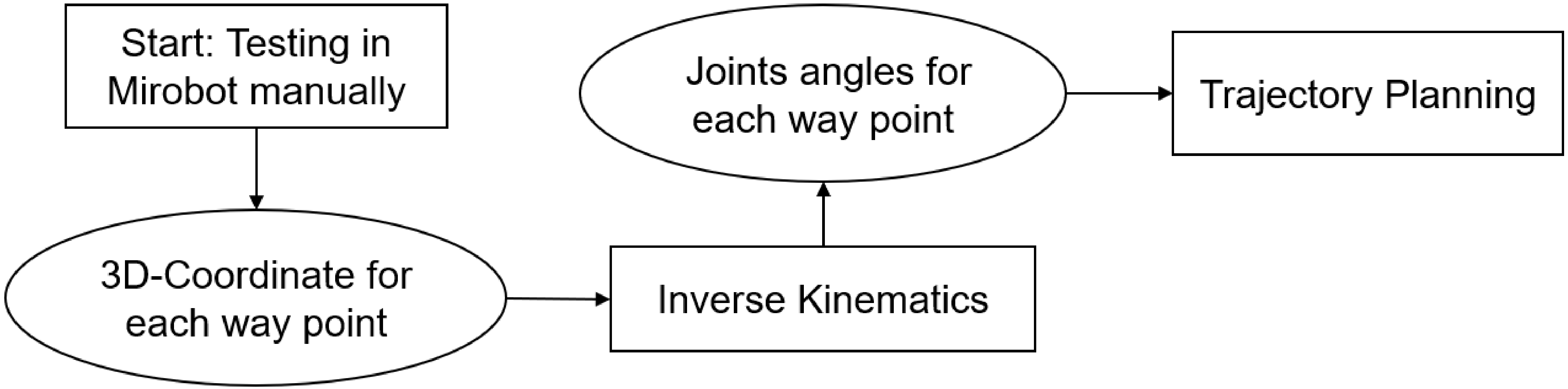
06. Videos on Mirobot (real) and CoppeliaSim (simulation)

Task IV

Preparation of Christmas Tree Gifts

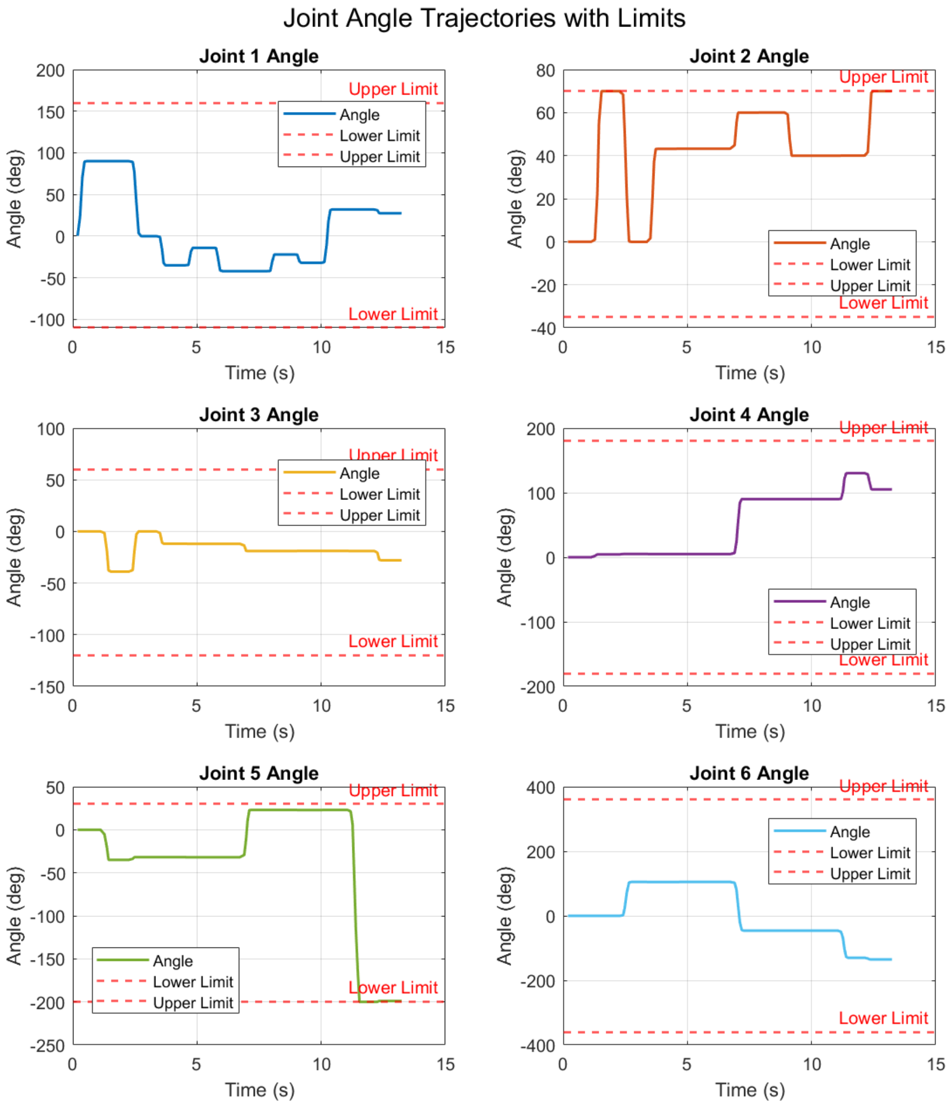
Task IV – Preparation of Christmas Tree Gifts
01. Numerical Iterative Methods

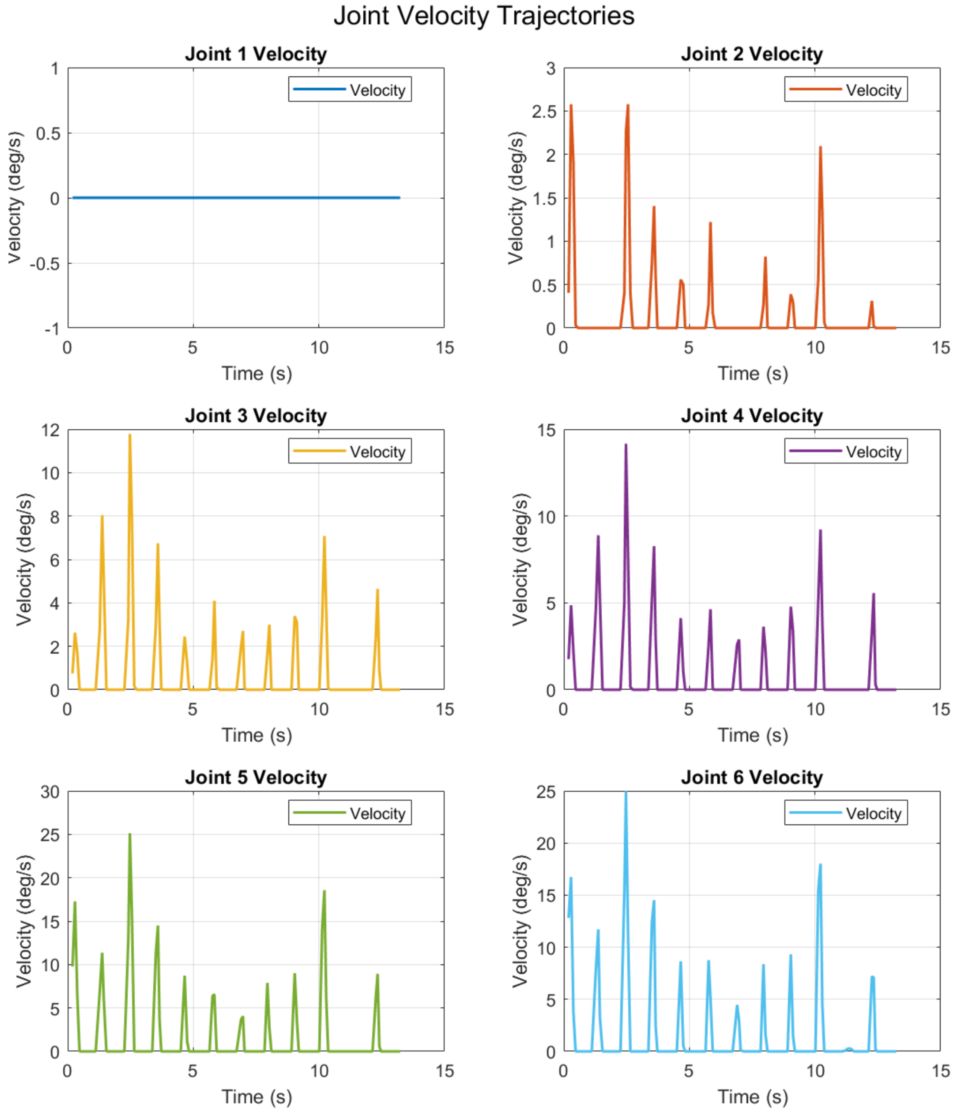
- **Inverse Kinematics (IK)**
  Inverse kinematics (IK) calculates the joint configurations (angles or positions) of a robotic manipulator to reach a specific target point defined by its position and orientation.



Task IV – Preparation of Christmas Tree Gifts

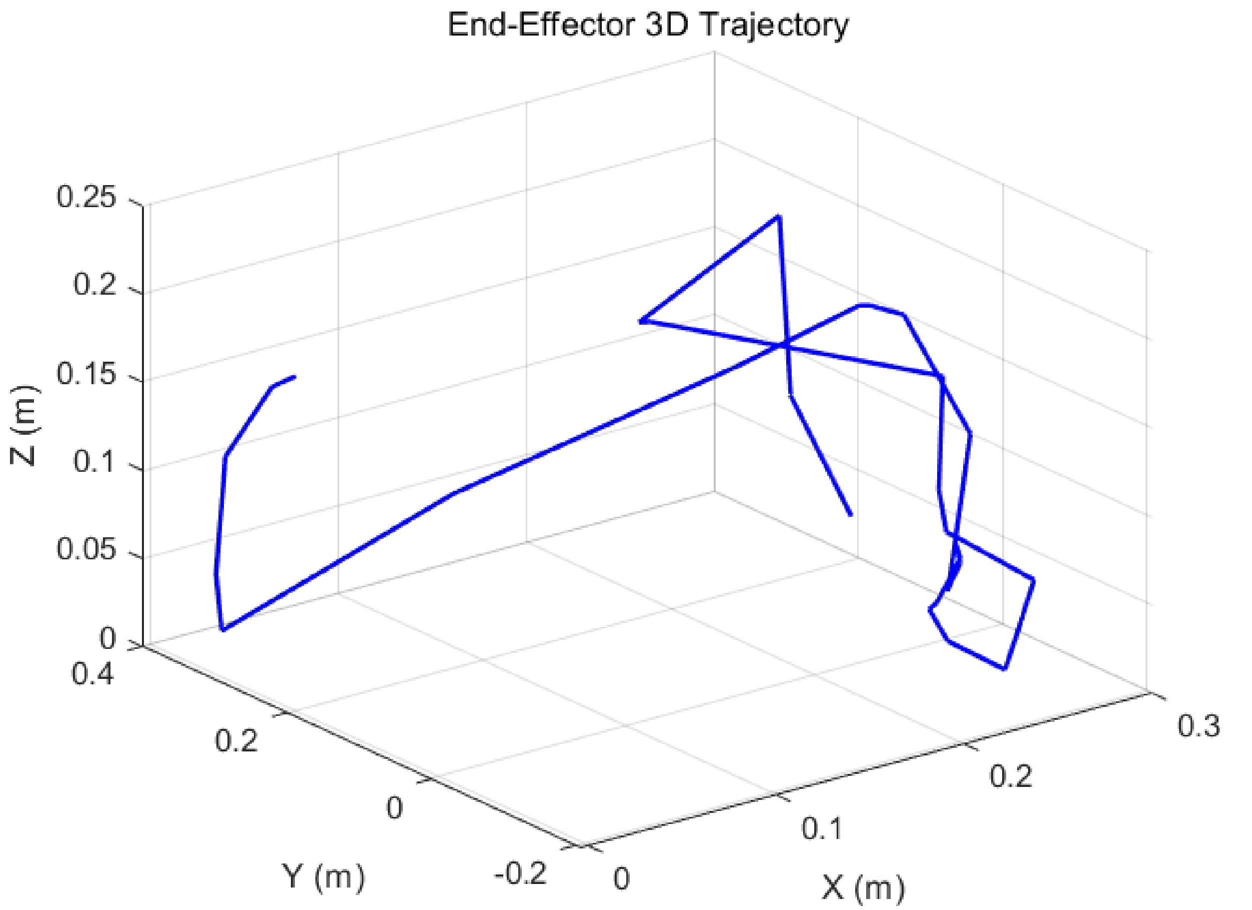
02. Result – Joint position and speed of each joint





Task IV – Preparation of Christmas Tree Gifts

03. Result – Trajectory of the robot tip in 3D space



Task IV – Preparation of Christmas Tree Gifts
04. Videos on Mirobot (real) and CoppeliaSim (simulation)

IMPERIAL

Thank you

Group coursework
06/12/2024

IMPERIAL