PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research
University of Amar Telidji - Laghouat



Faculty of Technology
Department of Electrical Engineering

Master's Dissertation

Domain: Science and Technology
Field: Automatic
Option: Automatic and systems

Presented by

BENSARKHA Hicham
AIDAOUI Saad

Control of Robotic Manipulator Using ROS

Jury members

Full Name	***Grade***	***Quality***
ABDEREZZAK Hadjer	*MCA*	*Chair*
BOUGRINE Med Djamel Eddine	*MCB*	*Examiner*
RAHMANI Belkacem	*MCA*	*Supervisor*
OUBBATI Khalil	*MCB*	*Co-Supervisor*

Academic Year 2025/2026

Dedication

To those who taught us our first letters and guided our first steps toward knowledge,

To those who instilled in us the values of perseverance, dedication, and hard work,

To our beloved parents and families, whose endless love, sacrifices, and support have been our greatest source of strength,

To our esteemed professors, for their invaluable guidance, encouragement, and generous sharing of knowledge,

To our colleagues and friends, who accompanied us throughout this academic journey, sharing both challenges and successes,

To everyone who offered a kind word, sincere advice, or heartfelt prayer,

We dedicate this humble work with profound gratitude and appreciation.

Acknowledgements

In the name of Allah, the Most Gracious, the Most Merciful. First and foremost, we thank Allah Almighty for granting us the strength, patience, and determination to complete this work.

We would like to express our deepest gratitude to our supervisor, Dr. Belkacem RAHMANI, for his invaluable guidance, continuous support, constructive suggestions, and unwavering encouragement throughout the development of this work. His expertise, patience, and confidence in our abilities have been essential to the successful completion of this dissertation.

Our sincere thanks also go to Dr. OUBBATI Khalil for his valuable assistance, support, and helpful contributions during the realization of this project.

We would like to express our sincere appreciation to all the professors and teachers of the Department of Electrical Engineering at the University of Amar Telidji, Laghouat, for their dedication, teaching, and support throughout our academic journey.

We are deeply grateful to our families for their unconditional love, patience, and encouragement. Their continuous support has always been a great source of strength and motivation throughout our studies.

We also thank our classmates and friends who shared this academic journey with us and provided support, advice, and encouragement whenever needed.

Finally, we would like to thank everyone who contributed, directly or indirectly, to the completion of this work. To all of them, we express our sincere gratitude and appreciation.

ملخص

يتناول هذا البحث نمذجة والتحكم في الذراع الروبوتية التعاونية ذات ست درجات من الحرية تم اشتقاق نماذج الحركية الأمامية والحركية العكسية باستخدام اتفاقية دينافيت-هارتنبرغ المعدّلة، كما تم اشتقاق النموذج الديناميكي الكامل باستخدام صياغة أويلر-لاغرانج. وقد تم تصميم ومقارنة ثلاث استراتيجيات للتحكم في فضاء المفاصل، وهي: المتحكم التناسبي التفاضلي، والمتحكم التناسبي التفاضلي مع تعويض الجاذبية، ومتحكم العزم المحسوب. يعتمد متحكم العزم المحسوب على النموذج الديناميكي الكامل لتعويض التأثيرات غير الخطية وفك الترابط بين المفاصل، مما يحول النظام إلى مجموعة من الأنظمة الخطية المستقلة. تم تنفيذ جميع استراتيجيات التحكم والتحقق من أدائها في بيئة محاكاة نظام تشغيل الروبوتات ومحاكي غازيبو. وأظهرت النتائج أن متحكم العزم المحسوب يحقق دقة أعلى في تتبع المسار وخطأً أقل في الحالة المستقرة مقارنة بالمتحكم التناسبي التفاضلي والمتحكم التناسبي التفاضلي مع تعويض الجاذبية، مما يؤكد فعالية أساليب التحكم المعتمدة على النموذج في تحسين أداء الذراع الروبوتية التعاونية.

الكلمات المفتاحية: الذراع الروبوتية، الذراع الروبوتية التعاونية، يو آر خمسة إي، الحركية الأمامية، الحركية العكسية، النموذج الديناميكي، متحكم العزم المحسوب، التحكم المعتمد على النموذج، نظام تشغيل الروبوتات، محاكي غازيبو، تتبع المسار، المتحكم التناسبي التفاضلي.

Résumé

Ce mémoire traite de la modélisation et de la commande du manipulateur collaboratif UR5e à six degrés de liberté. Les modèles cinématiques direct et inverse sont établis selon la convention de Denavit–Hartenberg modifiée, et le modèle dynamique complet est dérivé par la formulation d'Euler–Lagrange. Trois stratégies de commande dans l'espace articulaire — PD pur, PD avec compensation de la gravité (PD+G) et commande par couple calculé (CTC) — sont conçues et évaluées comparativement. L'approche CTC exploite le modèle dynamique complet pour annuler les non-linéarités et découpler les articulations, ramenant le système à un ensemble d'intégrateurs doubles linéaires indépendants. Les trois commandes sont implémentées et validées dans un environnement de simulation ROS/Gazebo. Les résultats démontrent que la CTC offre une précision de suivi de trajectoire supérieure et une erreur en régime permanent réduite par rapport au PD et au PD+G, confirmant l'efficacité de la commande basée sur modèle pour le manipulateur UR5e.

Mots-clés : Manipulateur robotique, UR5e, Robot collaboratif, Commande par couple calculé, Commande basée sur modèle, ROS, Gazebo, Suivi de trajectoire, Commande PID.

Abstract

This thesis addresses the modeling and control of the Universal Robots UR5e 6-DOF collaborative manipulator. Complete forward and inverse kinematic models are derived using the modified Denavit–Hartenberg convention, alongside a full dynamic model obtained via the Euler–Lagrange formulation. Three joint-space control strategies—pure PD, PD with gravity compensation (PD+G), and Computed Torque Control (CTC)—are designed and comparatively evaluated. The CTC approach leverages the complete dynamic model to cancel nonlinearities and decouple the joints, transforming the system into independent linear double integrators. All controllers are implemented and validated in a ROS/Gazebo simulation environment. Results demonstrate that CTC achieves superior trajectory tracking accuracy and reduced steady-state error compared to both PD and PD+G, confirming the effectiveness of model-based control for the UR5e manipulator.

Keywords: Robotic Manipulator, UR5e, Collaborative Robot, Computed Torque Control, Model-Based Control, ROS, Gazebo, Trajectory Tracking, PID Control.

Table of Contents

1 Modeling and Control of the UR5e Robot **1**

1.1 Context and Motivation 1
1.2 Kinematic Modeling of URe5 4
1.2.1 Coordinate Frames and DH Parameters 4
1.3 URe-Series Cobot Forward Pose Kinematics 6
1.4 Forward pose analytical expressions 6
1.5 Analytical Inverse Pose Kinematics Model (IPK) 8
1.6 Dynamic Modeling 11
1.7 Trajectory Planning 13
1.7.1 Fifth-Degree Polynomial 14
1.8 Controller Design 15
1.8.1 Proportional-Derivative Controller 15
1.8.2 Proportional-Derivative Controller with Gravity Compensation 15
1.8.3 Computed Torque Control Design 16

2 ROS-Based Implementation **19**

2.1 Introduction 19
2.2 Core ROS Concepts 20
2.2.1 ROS Computation Graph 22
2.3 UR5 ROS Environment 22
2.3.1 Required Packages 22
2.3.2 Workspace Configuration 23
2.3.3 Robot Description (URDF/Xacro) 24
2.3.4 Launching the Simulation 24
2.4 Communication Architecture 25
2.4.1 Topics 25
2.4.2 Node Architecture 25
2.4.3 Controller Configuration 26
2.4.4 MATLAB–ROS Communication 27
2.5 Implementation of the Computed Torque Controller 27
2.5.1 Simulink Block Diagram 27
2.5.2 Data Flow Summary 29

2.5.3 Control Loop Timing 30
2.6 Simulation Setup 30

3 Results and Discussion **32**
3.1 Trajectory Tracking Tests 32
3.1.1 Experimental Setup 32
3.1.2 Reference Trajectory 32
3.1.3 Controller Parameters 33
3.2 Control Strategy 1 : Pure PD Controller 34
3.3 Control Strategy 2 : PD with Gravity Compensation (PD+G) 36
3.4 Control Strategy 3 : Computed Torque Controller (CTC) 38
3.5 Discussion of Results 40
3.5.1 Tracking Performance 40
3.5.2 Control Torque Behavior 40
3.5.3 Main Limiting Factors 40
3.6 Conclusion 41

General Conclusion **42**

A Complete Dynamic Model of the UR5e Robot **44**
1.1 Dynamic Parameters 45
1.2 Inertia Matrix 45
1.3 Coriolis and Centrifugal Matrix 50
1.4 Gravity Vector 59

Acronyms and Symbols

a_2, a_3	Link lengths of the UR5e robot arm (m)
d_1, d_4, d_5, d_6	DH link offsets (m)
$e_i(t)$	Tracking error of joint i (rad)
e_{RMS}	Root Mean Square tracking error (rad)
$\mathbf{G}(\mathbf{q})$	Gravity vector (N·m)
$\mathbf{J}(\mathbf{q})$	Jacobian matrix
K_p	Proportional gain matrix
K_d	Derivative gain matrix
K_i	Integral gain matrix
m_i	Mass of link i (kg)
$\mathbf{M}(\mathbf{q})$	Inertia (mass) matrix (kg·m^2)
n	Harmonic gear reduction ratio
N	Number of samples
$\mathbf{p}$	Position vector of the end-effector (m)
$\mathbf{q}$	Joint angle vector (rad)
$\mathbf{q}_A$	Initial joint configuration (rad)
$\mathbf{q}_B$	Target joint configuration (rad)
$\mathbf{q}_d$	Desired joint trajectory (rad)
$\dot{\mathbf{q}}$	Joint velocity vector (rad/s)
$\ddot{\mathbf{q}}$	Joint acceleration vector (rad/s^2)
$\mathbf{R}$	Rotation matrix
t	Time (s)
T_f	Final time of motion (s)
${}^0T_6$	Homogeneous transformation matrix from frame {0} to {6}
$\mathbf{u}$	Control input vector (N·m)
$\mathbf{y}$	Auxiliary control signal
α_i	DH twist angle of link i (rad)

$\boldsymbol{\tau}$	Joint torque vector (N·m)
$\tau_{\max}$	Maximum joint torque (N·m)
θ_i	Joint angle of joint i (rad)
$\Delta\mathbf{q}$	Joint displacement vector, $\mathbf{q}_B - \mathbf{q}_A$ (rad)
λ	Normalized time, $\lambda = t/T_f$

6-DOF	Six Degrees of Freedom
CTC	Computed Torque Control
Cobot	Collaborative Robot
DH	Denavit–Hartenberg
EPS	Encapsulated PostScript
FF	Feedforward
FPK	Forward Pose Kinematics
IPK	Inverse Pose Kinematics
MATLAB	Matrix Laboratory
PD	Proportional-Derivative
PD+G	Proportional-Derivative with Gravity Compensation
RMS	Root Mean Square
ROS	Robot Operating System
UR5e	Universal Robots 5e collaborative manipulator
WSL	Windows Subsystem for Linux

List of Figures

1.1 The UR5e collaborative robotic manipulator. 1
1.2 Human-robot collaboration in an industrial setting. 2
1.3 Six-dof Universal URe-Series Kinematic Diagram with Coordinate Frames . . 5
1.4 Joint-space control strategy. 13

2.1 ROS filesystem organization. 20
2.2 ROS communication architecture diagram. 22
2.3 Simulink implementation of the computed torque controller. 27
2.4 Gazebo simulation environment used for validating the UR5e trajectory tracking controllers. 30

3.1 Joint position tracking under pure PD control. Blue : actual position ; orange dashed : desired position. 34
3.2 Joint tracking errors under pure PD control. 35
3.3 Joint torques under pure PD control. 35
3.4 Joint position tracking under PD with gravity compensation (PD+G). Blue : actual ; orange dashed : desired. 36
3.5 Joint tracking errors under PD with gravity compensation (PD+G). 37
3.6 Joint torques under PD with gravity compensation (PD+G). 37
3.7 Joint position tracking under Computed Torque Control (CTC). Blue : actual ; orange dashed : desired. 38
3.8 Joint tracking errors under Computed Torque Control (CTC). 39
3.9 Joint torques under Computed Torque Control (CTC). 39

Chapter 1

Modeling and Control of the UR5e Robot

1.1 Context and Motivation

Robotic manipulators have become essential tools in modern industrial and manufacturing systems, especially as automation increasingly requires flexible, accurate, and safe interaction with human operators. Collaborative robots, or cobots, address this need by combining industrial manipulation capabilities with safety-oriented operation. This project focuses on the Universal Robots UR5e, a six-degree-of-freedom (6-DOF) collaborative manipulator from the URe-Series. The UR5e has a reach of 850 mm and a payload capacity of 5 kg, making it suitable for tasks such as assembly, pick-and-place, machine tending, and quality inspection [7, 8, 11, 10].



Figure 1.1: The UR5e collaborative robotic manipulator.

Although the UR5e provides a user-friendly programming interface, precise trajectory tracking requires a control strategy that accounts for the nonlinear and coupled nature of its dynamics. Classical independent joint PID controllers are simple to implement, but they treat each joint separately and do not explicitly compensate for inertial, Coriolis, centripetal, and gravitational coupling effects. Consequently, their tracking performance may degrade during fast motions or under varying dynamic conditions [1, 8, 7].

To address these limitations, this work adopts Computed Torque Control (CTC), a model-based control strategy that uses the dynamic model of the robot to compensate nonlinear effects and decouple the joints. This transforms the joint-space control problem into a set of independent linear systems, enabling more accurate and robust trajectory tracking. The controller is implemented and validated in a ROS/Gazebo simulation environment [8, 7, 4].

The main objective of this project is to develop, implement, and validate a model-based control strategy for the UR5e collaborative manipulator within a ROS/Gazebo simulation environment. This requires developing the kinematic and dynamic models of the manipulator using the modified Denavit–Hartenberg (DH) convention and dynamic modeling techniques, designing a Computed Torque Controller based on the derived model, integrating the controller into the ROS/Gazebo simulation environment, and evaluating its trajectory tracking accuracy, actuator torque requirements, transient response, and robustness. The obtained performance is then compared with that of a conventional PID controller [1, 7, 8, 5].

The remainder of this report follows a progressive structure. The modeling and control chapter presents the UR5e structure and specifications, derives the kinematic and dynamic models, and develops the CTC control law. The implementation chapter describes the ROS/Gazebo simulation environment, communication architecture, controller implementation, and integration procedure. The results chapter presents the simulation outcomes, analyzes trajectory tracking performance, compares the controllers, and discusses the observed limitations.

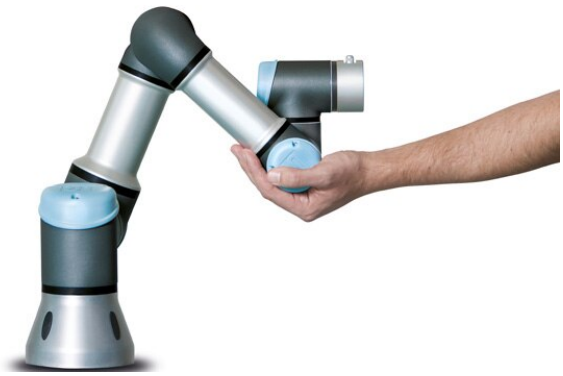


Figure 1.2: Human-robot collaboration in an industrial setting.

The Universal Robots UR5e is a 6-DOF serial collaborative manipulator developed by

Universal Robots. It belongs to the URe-Series, which includes the UR3e, UR5e, UR10e, and UR16e models. These robots share a similar kinematic structure and mainly differ in size, reach, and payload capacity. The UR5e consists of six revolute joints arranged as a waist-shoulder-elbow arm followed by an offset pitch-yaw-roll wrist. This structure provides a spherical workspace with an 850 mm reach and allows the robot to perform a wide range of industrial and collaborative tasks [11, 10, 12].

Table 1.1: Key technical specifications of the Universal Robots UR5e cobot

Parameter	Value (UR5e)
Degrees of freedom	6 (all revolute)
Reach	850 mm
Payload	5 kg
Mass	20.6 kg
Footprint diameter	149 mm
Payload-to-mass ratio	0.24
Power consumption	$\sim$200 W
Positioning repeatability	$\pm$0.03 mm
Link 2 length (a_2)	425 mm
Link 3 length (a_3)	392 mm
d_4 offset	133 mm
d_5 offset	100 mm
L_{TP} tool-plate offset	100 mm
L_{B} base height	163 mm

The UR5e joints are divided into two actuator categories. Joints 1, 2, and 3 use Size 3 modules rated at 150 N m, while Joints 4, 5, and 6 use Size 1 modules rated at 28 N m. Each joint incorporates a harmonic gear with a reduction ratio of $n = 101$, which supports the precision required for collaborative operation. Table 1.2 summarizes the joint naming convention and motion limits [11, 10].

Table 1.2: Six-DOF UR5e joint naming convention and motion limits

Joint	Name	Motion	Limit
1	Base	Waist yaw	$\pm$360°
2	Shoulder	Shoulder pitch	$\pm$360°
3	Elbow	Elbow pitch	$\pm$360°
4	Wrist 1	Wrist pitch	$\pm$360°
5	Wrist 2	Wrist yaw	$\pm$360°
6	Wrist 3	Wrist roll	$\pm$360°

1.2 Kinematic Modeling of URe5

The kinematic model describes the relationship between the robot joint variables and the pose of the end-effector. The first step in kinematic modeling is to define the generalized coordinate variables. For an n-DOF robotic manipulator, the generalized coordinate vector is given by

$$q = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix},$$

while the corresponding velocity and acceleration vectors are

$$\dot{q} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix} \quad \text{and} \quad \ddot{q} = \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \vdots \\ \ddot{q}_n \end{bmatrix}.$$

1.2.1 Coordinate Frames and DH Parameters

Before deriving the Denavit–Hartenberg parameters, coordinate frames must be assigned consistently to each joint and link of the manipulator. This frame assignment defines the geometric relationships between successive links and provides the basis for constructing the homogeneous transformation matrices. The rules for coordinate frame assignment follow the standard robotics convention described in classical robot modeling references [1, 7]:

- The origin of the first and second frames are placed at the same position to simplify the calculation of the DH parameters.
- The axis of rotation of each joint is outward of the page.
- In frames $\{0\}$ and $\{1\}$, the axes y_0 and y_1 point toward the page.
- The axes z of frames $\{2\}$, $\{3\}$, $\{4\}$ and $\{6\}$ are also directed toward the page.
- For frame $\{5\}$, the axis y_5 is vertical and outward of the page.

The frame attachment of this robot is illustrated in Figure 1.3.

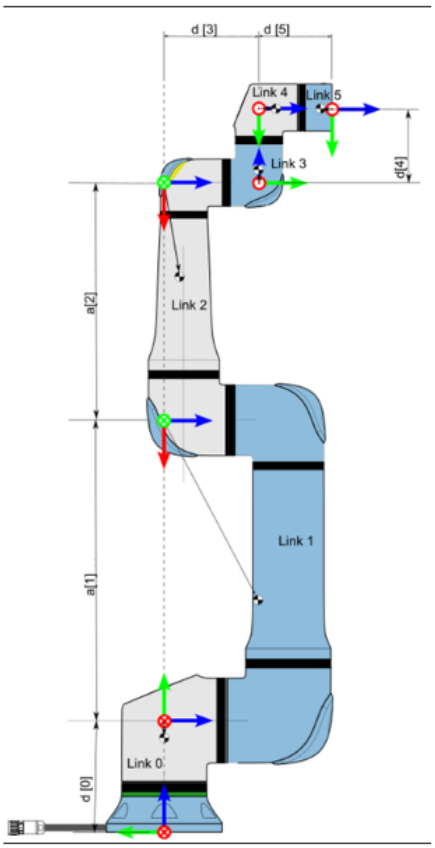


Figure 1.3: Six-dof Universal URe-Series Kinematic Diagram with Coordinate Frames

The standard Denavit–Hartenberg parameters are generally represented as follows:

Table 1.3: Denavit–Hartenberg parameters of the Universal Robots UR5e

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	d_1	θ_1
2	90°	0	0	θ_2
3	0	a_2	0	θ_3
4	0	a_3	d_4	θ_4
5	90°	0	d_5	θ_5
6	−90°	0	d_6	θ_6

The specific length parameters for the UR5e cobot are:

$$d_1 = 0.1625 \text{ m}, \quad a_2 = -0.425 \text{ m}, \quad a_3 = -0.3922 \text{ m}, \quad d_4 = 0.1333 \text{ m}, \quad d_5 = 0.0997 \text{ m}, \quad d_6 = 0.0996 \tag{1.1}$$

1.3 URe-Series Cobot Forward Pose Kinematics

In general, the Forward Pose Kinematics (FPK) problem for a serial-chain robot is stated as follows:

> Given the joint values, calculate the pose (position and orientation) of the end-effector frame of interest.

For serial-chain robots, the FPK problem setup and solution are straightforward. It is based on substituting each line of the Denavit–Hartenberg parameter table into the standard homogeneous transformation formulation, giving the pose of frame $\{i\}$ with respect to its nearest neighboring frame $\{i-1\}$ along the serial chain:

$$ {}^{i-1}_{i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i\, c\alpha_{i-1} & c\theta_i\, c\alpha_{i-1} & -s\alpha_{i-1} & -d_i\, s\alpha_{i-1} \\ s\theta_i\, s\alpha_{i-1} & c\theta_i\, s\alpha_{i-1} & c\alpha_{i-1} & d_i\, c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^{i-1}_{i}R & {}^{i-1}P_i \\ 0 & 1 \end{bmatrix} \tag{1.2} $$

where the following abbreviations are used:

$$ c\theta_i = \cos\theta_i, \quad s\theta_i = \sin\theta_i, \quad c\alpha_i = \cos\alpha_i, \quad s\alpha_i = \sin\alpha_i \tag{1.3} $$

The equation above represents the pose (position and orientation) of frame $\{i\}$ with respect to frame $\{i-1\}$ using a 44 homogeneous transformation matrix. Then homogeneous transformation equations are used to find the pose of the overall end-effector frame with respect to the base reference frame, completing the FPK solution for the serial chain.

1.4 Forward pose analytical expressions

The statement of the FPK problem for the 6-dof serial chain of the Universal Cobot URe-Series is:

$$ \text{Given } (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6), \text{ calculate } {}^{0}_{6}T \tag{1.4} $$

where $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ are the six joint angles. Frame $\{0\}$ is the fixed base frame, while $\{1\}$ through $\{6\}$ are active moving joint frames. Substitute each row of the DH parameters into the equation for ${}^{i-1}_{i}T$ to obtain the six neighboring homogeneous transformation matrices.

$$ {}^{0}_{1}T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.5} $$

$$
{}^{1}_{2}T = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.6}
$$

$$
{}^{2}_{3}T = \begin{bmatrix} c_3 & -s_3 & 0 & a_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.7}
$$

$$
{}^{3}_{4}T = \begin{bmatrix} c_4 & -s_4 & 0 & a_3 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.8}
$$

$$
{}^{4}_{5}T = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & -d_5 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.9}
$$

$$
{}^{5}_{6}T = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ -s_6 & -c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.10}
$$

where:

$$
c_i = \cos\theta_i, \qquad s_i = \sin\theta_i \tag{1.11}
$$

The Universal Cobot FPK solution is:

$$
{}^{0}_{6}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T\,{}^{4}_{5}T\,{}^{5}_{6}T \tag{1.12}
$$

Since coordinate frames {2}, {3} and {4} have parallel Z axes, the grouping becomes:

$$
{}^{0}_{6}T = {}^{0}_{1}T\,{}^{1}_{4}T\,{}^{4}_{6}T \tag{1.13}
$$

with:

$$
c_{23} = \cos(\theta_2 + \theta_3), \quad s_{23} = \sin(\theta_2 + \theta_3) \tag{1.14}
$$

$$
c_{234} = \cos(\theta_2 + \theta_3 + \theta_4), \quad s_{234} = \sin(\theta_2 + \theta_3 + \theta_4) \tag{1.15}
$$

The FPK solution becomes:

$$ {}^{0}_{6}T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x_6 \\ r_{21} & r_{22} & r_{23} & y_6 \\ r_{31} & r_{32} & r_{33} & z_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.16} $$

The translational vector components for the UR5e are:

$$ x_6 = d_4 s_1 + c_1(a_2 c_2 + a_3 c_{23} + d_5 s_{234}) \tag{1.17} $$

$$ y_6 = -d_4 c_1 + s_1(a_2 c_2 + a_3 c_{23} + d_5 s_{234}) \tag{1.18} $$

$$ z_6 = d_1 + a_2 s_2 + a_3 s_{23} - d_5 c_{234} \tag{1.19} $$

The FPK solutions can be evaluated numerically or symbolically, or using a combination of these two methods.

1.5 Analytical Inverse Pose Kinematics Model (IPK)

The Inverse Pose Kinematics (IPK) problem consists of determining the joint variables that produce a desired end-effector pose. For a serial-chain robot, this solution is obtained from the Forward Pose Kinematics (FPK) equations by solving for the joint angles that satisfy the required position and orientation. For the 6-DOF URe-Series cobots, the analytical structure of the wrist and arm can be exploited to obtain closed-form solutions [1, 2, 12]. The IPK problem can be stated compactly as follows: given the constant DH parameters and the desired end-effector pose,

$$ {}^{0}_{6}T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x_6 \\ r_{21} & r_{22} & r_{23} & y_6 \\ r_{31} & r_{32} & r_{33} & z_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.20} $$

Calculate: the joint angles $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ to achieve this pose.

The IPK equations come from the FPK expressions; but with the six joint angles now unknown, coupled nonlinear (transcendental) equations result:

$$ {}^{0}_{6}T = {}^{0}_{1}T(\theta_1)\, {}^{1}_{2}T(\theta_2)\, {}^{2}_{3}T(\theta_3)\, {}^{3}_{4}T(\theta_4)\, {}^{4}_{5}T(\theta_5)\, {}^{5}_{6}T(\theta_6) \tag{1.21} $$

A classic IPK solution approach is to invert some of the consecutive ${}^{i-1}_{i}T$ homogeneous

transformation matrices and multiply them to the given ${}^{0}_{6}T$ on the appropriate sides. For the URe Cobots, use the following equation to separate two unknowns θ_1, θ_6 from the other four unknown joint angles $\theta_2, \theta_3, \theta_4, \theta_5$:

$$\left({}^{0}_{1}T\right)^{-1} \cdot {}^{0}_{6}T \cdot \left({}^{5}_{6}T\right)^{-1} = {}^{1}_{2}T(\theta_2)\,{}^{2}_{3}T(\theta_3)\,{}^{3}_{4}T(\theta_4)\,{}^{4}_{5}T(\theta_5) \tag{1.22}$$

First : solving for θ_1. The $(2,4)$ translational term from the separation equation involves the unknown θ_1:

$$y_6 c_1 - x_6 s_1 = d_4 \tag{1.23}$$

This equation can be solved geometrically [2] as:

$$\theta_1 = \operatorname{atan2}(y_6,\ x_6) \pm \cos^{-1}\left(\frac{d_4}{\sqrt{x_6^2 + y_6^2}}\right) + \frac{\pi}{2} \tag{1.24}$$

Alternatively, rewriting into classical EFG form:

$$E_1 \cos\theta_1 + F_1 \sin\theta_1 + G_1 = 0 \tag{1.25}$$

where:

$$E_1 = y_6, \qquad F_1 = -x_6, \qquad G_1 = -d_4 \tag{1.26}$$

Applying the Tangent Half-Angle Substitution with $t = \tan\left(\frac{\theta_1}{2}\right)$:

$$\cos\theta_1 = \frac{1 - t^2}{1 + t^2}, \qquad \sin\theta_1 = \frac{2t}{1 + t^2} \tag{1.27}$$

yields the quadratic:

$$(G_1 - E_1)\,t^2 + 2F_1\,t + (G_1 + E_1) = 0 \tag{1.28}$$

with solution:

$$t_{1,2} = \frac{-F_1 \pm \sqrt{E_1^2 + F_1^2 - G_1^2}}{G_1 - E_1} \qquad \Longrightarrow \qquad \theta_{1_{1,2}} = 2\arctan(t_{1,2}) \tag{1.29}$$

Second : solving for θ_5. Using the transformation from frame 1 to frame 6, the y-coordinate of the position yields:

$$\theta_5 = \pm\cos^{-1}\left(\frac{x_6 s_1 - y_6 c_1 - d_4}{d_6}\right) \tag{1.30}$$

This provides two solutions for θ_5, corresponding to wrist "in/down" or "out/up" configurations.

Third : solving for θ_6. Using the $6y_1$ coordinate axis and the spherical coordinate relationship:

$$\theta_6 = \text{atan2}\left(\frac{-y_x s_1 + y_y c_1}{s_5},\ \frac{-(-x_x s_1 + x_y c_1)}{s_5}\right) \tag{1.31}$$

where (x_x, x_y, x_z) and (y_x, y_y, y_z) are the components of the x and y axes of frame $\{6\}$ expressed in frame $\{0\}$.

Fourth : solving for θ_2, θ_3, and θ_4. The remaining three joints $(\theta_2, \theta_3, \theta_4)$ form a classical 3R planar arm. The wrist center position ${}^0P_W$ is determined by subtracting the wrist offset from the end-effector position:

$${}^0P_W = \begin{bmatrix} x_6 - d_6 a_x \\ y_6 - d_6 a_y \\ z_6 - d_6 a_z \end{bmatrix} \tag{1.32}$$

where (a_x, a_y, a_z) are the components of the approach vector (third column of the rotation matrix).

The distance from the shoulder to the wrist center is:

$$r = \sqrt{({}^0P_{W,x})^2 + ({}^0P_{W,y})^2 + ({}^0P_{W,z} - d_1)^2} \tag{1.33}$$

The elbow angle θ_3 is found using the law of cosines:

$$\theta_3 = \pm \cos^{-1}\left(\frac{r^2 - a_2^2 - a_3^2}{2a_2 a_3}\right) \tag{1.34}$$

The shoulder angle θ_2 is:

$$\theta_2 = \text{atan2}({}^0P_{W,z} - d_1,\ \sqrt{({}^0P_{W,x})^2 + ({}^0P_{W,y})^2}) - \text{atan2}(a_3 \sin\theta_3,\ a_2 + a_3 \cos\theta_3) \tag{1.35}$$

The wrist angle θ_4 is then:

$$\theta_4 = \theta_{234} - \theta_2 - \theta_3 \tag{1.36}$$

where θ_{234} is obtained from the rotation matrix:

$$\theta_{234} = \text{atan2}(s_{234},\ c_{234}) \tag{1.37}$$

with:

$$s_{234} = -a_z, \qquad c_{234} = \frac{a_x c_1 + a_y s_1}{s_5} \tag{1.38}$$

These equations complete the inverse kinematic formulation used to determine the joint variables corresponding to a desired end-effector pose.

1.6 Dynamic Modeling

The dynamic model describes the relationship between the joint motion and the torques required to produce it. For the UR5e manipulator, the model is obtained using the Lagrange formulation while considering the mass, centre of mass, and inertia tensor of each link. The equations of motion are written in joint space as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \boldsymbol{\tau}, \tag{1.39}$$

where q, $\dot{q}$, and $\ddot{q}$ are the joint position, velocity, and acceleration vectors, respectively. The matrix $M(q)$ is the symmetric positive-definite inertia matrix, $C(q,\dot{q})\dot{q}$ represents the Coriolis and centrifugal torque vector, $G(q)$ is the gravity torque vector, and $\boldsymbol{\tau}$ is the vector of actuator joint torques [8, 7, 3].

Inertia Matrix

The inertia matrix $M(q)$ represents the distribution of mass in joint space and determines how the manipulator resists joint accelerations. It is obtained by summing the translational and rotational kinetic-energy contributions of all links:

$$M(q) = \sum_{i=1}^{n} \left[m_i J_{v_i}^T(q) J_{v_i}(q) + J_i^T(q)\, {}^0R_i(q)\, {}^iI_i\, {}^0R_i^T(q) J_i(q) \right]. \tag{1.40}$$

In this expression, m_i is the mass of link i, $J_{v_i}(q)$ is the translational Jacobian of the centre of mass of link i, $J_i(q)$ is the angular Jacobian of link i, ${}^iI_i$ is the inertia tensor expressed in the local frame of link i, and ${}^0R_i(q)$ is the rotation matrix from link frame i to the base frame. The term ${}^0R_i\, {}^iI_i\, {}^0R_i^T$ expresses the link inertia tensor in the base frame.

The centre-of-mass position and the inertia tensor of link i are represented symbolically as

$${}^ip_{C_i} = \begin{bmatrix} P_{Cx_i} \\ P_{Cy_i} \\ P_{Cz_i} \end{bmatrix}, \qquad {}^iI_i = \begin{bmatrix} I_{xx_i} & I_{xy_i} & I_{xz_i} \\ I_{xy_i} & I_{yy_i} & I_{yz_i} \\ I_{xz_i} & I_{yz_i} & I_{zz_i} \end{bmatrix}. \tag{1.41}$$

Jacobian Matrices

The Jacobian matrices appearing in the inertia matrix are used to express link velocities as functions of joint velocities. For each link i, the translational Jacobian relates $\dot{q}$ to the linear velocity of the link centre of mass, while the angular Jacobian relates $\dot{q}$ to the angular velocity of the link [7, 8]:

$$v_{C_i} = J_{v_i}(q)\dot{q}, \qquad {}_i = J_i(q)\dot{q}. \tag{1.42}$$

The complete geometric Jacobian of link i is defined by stacking these two components:

$$J_i(q) = \begin{bmatrix} J_{v_i}(q) \\ J_{i}(q) \end{bmatrix}. \tag{1.43}$$

The translational Jacobian is obtained by differentiating the centre-of-mass position expressed in the base frame with respect to the joint vector:

$$J_{v_i}(q) = \frac{\partial\, {}^0p_{C_i}}{\partial q}, \tag{1.44}$$

where ${}^0p_{C_i}$ is obtained by transforming the local centre-of-mass vector ${}^ip_{C_i}$ to the base frame. The orientation of link i with respect to the base frame is given by the product of successive rotation matrices:

$${}^0R_i = {}^0R_1\, {}^1R_2 \cdots {}^{i-1}R_i. \tag{1.45}$$

Since all UR5e joints are revolute, the angular Jacobian is formed from the joint axes expressed in the base frame:

$${}^0J_{i} = \begin{bmatrix} {}^0\hat{z}_1 & {}^0\hat{z}_2 & \cdots & {}^0\hat{z}_i & 0 & \cdots & 0 \end{bmatrix}, \tag{1.46}$$

where ${}^0\hat{z}_j$ is the axis of rotation of joint j expressed in the base frame. The zero columns correspond to joints located after link i, which do not affect the motion of that link.

Coriolis and Centrifugal Matrix

The Coriolis and centrifugal matrix is computed from the elements of the inertia matrix. Let $m_{ij}(q)$ denote the (i, j) element of $M(q)$. The elements of $C(q, \dot{q})$ are defined using the Christoffel symbols as

$$c_{ij}(q, \dot{q}) = \frac{1}{2} \sum_{k=1}^{n} \left(\frac{\partial m_{ij}}{\partial q_k} + \frac{\partial m_{ik}}{\partial q_j} - \frac{\partial m_{jk}}{\partial q_i} \right) \dot{q}_k. \tag{1.47}$$

This definition ensures the standard robot-dynamics property that $\dot{M}(q) - 2C(q, \dot{q})$ is skew-symmetric, which is useful for stability analysis. The resulting Coriolis and centrifugal torque contribution is $C(q, \dot{q})\dot{q}$.

Gravity Vector

The gravity vector is obtained from the total potential energy of the manipulator. If $\mathcal{P}$ denotes the total potential energy, then each component of the gravity vector is

$$g_i(q) = \frac{\partial \mathcal{P}}{\partial q_i}. \tag{1.48}$$

The total potential energy is the sum of the potential energies of all links:

$$\mathcal{P} = \sum_{i=1}^{n} \mathcal{P}_i, \tag{1.49}$$

where the potential energy of link i is expressed as

$$\mathcal{P}_i = m_i \begin{bmatrix} 0 & 0 & g \end{bmatrix} {}^0p_{C_i} = m_i g \, {}^0p_{Cz_i}. \tag{1.50}$$

Thus, the complete dynamic model is fully defined by the inertia matrix $M(q)$, the Coriolis and centrifugal matrix $C(q, \dot{q})$, the gravity vector $G(q)$, and the joint torque vector $\boldsymbol{\tau}$.

1.7 Trajectory Planning

Joint-space control is adopted in this work because the inverse pose kinematics provides the desired motion directly in terms of joint variables. The general joint-space control scheme is illustrated in Figure 1.4.

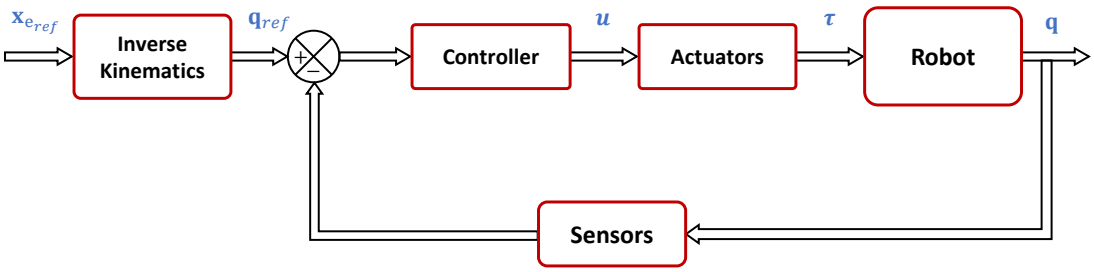


Figure 1.4: Joint-space control strategy.

As described in the inverse pose kinematics (IPK) section, the desired initial and final end-effector poses, represented by the homogeneous transformation matrices T_A and T_B, are first converted into their corresponding joint-space configurations q_A and q_B:

$$T_A \xrightarrow{\text{IPK}} q_A, \qquad T_B \xrightarrow{\text{IPK}} q_B. \tag{1.51}$$

These two joint configurations define the boundary points of the desired motion. A trajectory generation method is then required to produce a smooth transition from q_A to q_B while respecting the kinematic and dynamic constraints of the manipulator.

The generated trajectory must start and end at rest, avoid abrupt changes in velocity and acceleration, and provide continuous bounded profiles for the desired position, velocity, and acceleration. These requirements are expressed as

$$\dot{q}_d(0) = \dot{q}_d(T_f) = 0, \qquad q_d(t),\ \dot{q}_d(t),\ \ddot{q}_d(t) \text{ continuous and bounded.} \tag{1.52}$$

To satisfy these conditions, a fifth-order polynomial interpolation is adopted. In the following derivation, q denotes a single joint coordinate rather than the complete joint vector. Since the same interpolation procedure is applied independently to each robot joint, the resulting expressions are valid for all axes.

1.7.1 Fifth-Degree Polynomial

A fifth-degree polynomial is selected because it can satisfy six boundary conditions: prescribed initial and final positions, zero initial and final velocities, and zero initial and final accelerations. This type of smooth polynomial interpolation is widely used in joint-space robotic trajectory planning [1, 7]. For one joint, these constraints are

$$q_d(0) = q_A, \qquad \dot{q}_d(0) = 0, \qquad \ddot{q}_d(0) = 0, \tag{1.53}$$

$$q_d(T_f) = q_B, \qquad \dot{q}_d(T_f) = 0, \qquad \ddot{q}_d(T_f) = 0, \tag{1.54}$$

where T_f is the total motion time, q_A is the initial joint position, and q_B is the final joint position. To simplify the trajectory expression, the normalized time variable is introduced as

$$\lambda = \frac{t}{T_f}, \qquad 0 \le \lambda \le 1. \tag{1.55}$$

Using this normalized variable, the total joint displacement is defined by

$$\Delta q = q_B - q_A. \tag{1.56}$$

The resulting fifth-degree interpolation profile is then written as

$$q_d(\lambda) = q_A + \Delta q \left(10\lambda^3 - 15\lambda^4 + 6\lambda^5\right). \tag{1.57}$$

Since $\lambda = t/T_f$, differentiation with respect to time gives the desired velocity and acceleration profiles:

$$\dot{q}_d(\lambda) = \frac{\Delta q}{T_f}\left(30\lambda^2 - 60\lambda^3 + 30\lambda^4\right), \tag{1.58}$$

$$\ddot{q}_d(\lambda) = \frac{\Delta q}{T_f^2}\left(60\lambda - 180\lambda^2 + 120\lambda^3\right). \tag{1.59}$$

These expressions generate smooth position, velocity, and acceleration commands for

each joint. In vector form, the same equations are applied component-wise to all six joints of the UR5e manipulator.

1.8 Controller Design

This section presents the control strategies used for joint-space trajectory tracking of the UR5e manipulator. The desired joint trajectories q_d, $\dot{q}_d$, and $\ddot{q}_d$ are generated by the trajectory planner, while the measured joint states are denoted by q and $\dot{q}$. The control objective is to generate the joint torque vector $\boldsymbol{\tau}$ such that the actual motion follows the desired trajectory with minimum tracking error. Three controllers are considered: a classical Proportional-Derivative controller, a Proportional-Derivative controller with gravity compensation, and a Computed Torque Controller [8, 7].

1.8.1 Proportional-Derivative Controller

The Proportional-Derivative (PD) controller is used as the baseline control strategy because of its simplicity and direct implementation. It generates the torque command from the position and velocity tracking errors only, without explicitly compensating for the nonlinear dynamics of the robot. The PD control law is

$$\boldsymbol{\tau}_{PD} = K_p(q_d - q) + K_d(\dot{q}_d - \dot{q}), \tag{1.60}$$

where K_p and K_d are positive-definite proportional and derivative gain matrices. The proportional term reduces the position error, while the derivative term provides damping and improves transient response. However, because this controller does not include the inertia, Coriolis, centrifugal, or gravity terms of the manipulator, its performance may degrade during fast motion or when the robot configuration changes significantly.

1.8.2 Proportional-Derivative Controller with Gravity Compensation

To improve the classical PD controller, a gravity compensation term can be added. This controller preserves the simple feedback structure of PD control while compensating for the static gravitational torques acting on the robot links. Its control law is expressed as

$$\boldsymbol{\tau}_{PD+G} = K_p(q_d - q) + K_d(\dot{q}_d - \dot{q}) + G(q). \tag{1.61}$$

The term $G(q)$ reduces the steady-state error caused by the weight of the manipulator links, especially for joints that carry large gravitational loads. Compared with pure PD control, PD with gravity compensation generally provides better tracking accuracy at low

and moderate speeds. Nevertheless, it still does not compensate for inertial coupling or Coriolis and centrifugal effects, so its performance remains limited when the manipulator moves rapidly or follows highly dynamic trajectories.

1.8.3 Computed Torque Control Design

The Computed Torque Controller (CTC), also known as inverse dynamics control, is a model-based nonlinear controller. Unlike PD and PD+G control, it uses the complete dynamic model of the robot to compensate for nonlinear and coupled dynamics [8, 7, 4]. Starting from the joint-space dynamic model,

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \boldsymbol{\tau}, \tag{1.62}$$

CTC selects the torque command as

$$\boldsymbol{\tau} = M(q)y + C(q,\dot{q})\dot{q} + G(q), \tag{1.63}$$

where $y \in \mathbb{R}^6$ is an auxiliary acceleration input. Substituting this torque command into the robot dynamic model gives

$$M(q)\ddot{q} = M(q)y. \tag{1.64}$$

Since $M(q)$ is symmetric positive definite and therefore invertible, the closed-loop inner dynamics reduce to

$$\ddot{q} = y. \tag{1.65}$$

Thus, the nonlinear coupled manipulator is transformed into six decoupled double-integrator systems in joint space. The remaining task is to choose y so that the tracking error converges to zero.

Let the tracking error be defined as

$$e = q_d - q, \qquad \dot{e} = \dot{q}_d - \dot{q}. \tag{1.66}$$

The auxiliary input is then selected as

$$y = \ddot{q}_d + K_d(\dot{q}_d - \dot{q}) + K_p(q_d - q). \tag{1.67}$$

Combining the nonlinear compensation term with this stabilizing auxiliary input gives the complete Computed Torque Control law:

$$\boldsymbol{\tau}_{CTC} = M(q)\left[\ddot{q}_d + K_d(\dot{q}_d - \dot{q}) + K_p(q_d - q)\right] + C(q,\dot{q})\dot{q} + G(q) \tag{1.68}$$

Substituting this law into the dynamic model yields the closed-loop tracking error equation

$$\ddot{e} + K_d\dot{e} + K_p e = 0. \tag{1.69}$$

For each joint i, the error dynamics become

$$\ddot{e}_i + k_{d,i}\dot{e}_i + k_{p,i}e_i = 0, \qquad i = 1, \ldots, 6. \tag{1.70}$$

The characteristic polynomial of each joint is therefore

$$s^2 + k_{d,i}s + k_{p,i} = 0. \tag{1.71}$$

Consequently, if K_p and K_d are chosen as positive-definite diagonal matrices, the tracking error of each joint follows a stable second-order response. The main advantage of CTC is that the closed-loop behavior becomes approximately linear and decoupled, with performance determined mainly by the selected gains rather than by the nonlinear robot configuration.

Gain Selection Using Pole Placement

After feedback linearization, each joint behaves as a second-order system whose response is determined by the proportional and derivative gains. Therefore, the gains are selected by assigning the desired closed-loop poles for each joint. For diagonal gain matrices, the proportional and derivative gains are chosen as

$$K_p = \text{diag}\left(^2_{n1}, ^2_{n2}, \ldots, ^2_{n6}\right), \tag{1.72}$$

$$K_d = \text{diag}\left(2\zeta_{1n1}, 2\zeta_{2n2}, \ldots, 2\zeta_{6n6}\right), \tag{1.73}$$

where $_{ni}$ is the desired natural frequency of joint i and ζ_i is its damping ratio. With this choice, the characteristic polynomial of joint i becomes

$$s^2 + 2\zeta_{ini}s + ^2_{ni} = 0, \tag{1.74}$$

and the corresponding closed-loop poles are

$$s_{1,2} = -\zeta_{ini} \pm j_{ni}\sqrt{1-\zeta_i^2}. \tag{1.75}$$

The damping ratio determines the transient behavior of each joint, as summarized in Table 1.4.

In this work, critical damping is selected for all joints in order to obtain fast convergence without overshoot. Thus,

Table 1.4: Effect of damping ratio on closed-loop response

Condition	Response type	Main characteristic
$\zeta_i < 1$	Underdamped	Oscillatory response with overshoot
$\zeta_i = 1$	Critically damped	Fastest response without oscillation
$\zeta_i > 1$	Overdamped	Slow non-oscillatory response

$$\zeta_i = 1, \qquad i = 1, \ldots, 6, \tag{1.76}$$

which gives the simplified gain relation

$$K_d = 2\sqrt{K_p}. \tag{1.77}$$

For a critically damped second-order response, the theoretical settling time using the 2% criterion is approximated by

$$T_s \approx \frac{4}{\zeta_{ini}}. \tag{1.78}$$

This relation provides a practical way to select $_{ni}$ according to the desired settling time of each joint.

Chapter 2

ROS-Based Implementation

2.1 Introduction

The implementation of the proposed control architecture requires a software environment capable of connecting the robot model, the controller, the simulation engine, and the data exchange interfaces. For this reason, the Robot Operating System (ROS) was adopted as the middleware framework for this project. ROS provides a flexible communication layer for robotic applications and allows different software components to run as independent processes while exchanging information in real time [9].

ROS is commonly described as an open-source meta-operating system for robots. It does not replace a conventional operating system such as Linux, but it runs on top of it and provides services that are essential in robotics, including hardware abstraction, low-level device control, message passing, package management, visualization tools, and reusable software libraries. These features make ROS particularly suitable for developing and testing complex robotic systems such as the UR5e manipulator.

A major advantage of ROS is its distributed computation model. In a robotic system, the simulation, robot-state publisher, controllers, visualization tools, and user programs can all run as separate processes. These processes communicate through the ROS network, which improves modularity and makes the system easier to extend, debug, and maintain. This is especially useful in the present work, where Gazebo simulates the UR5e robot while MATLAB/Simulink computes the control torques and exchanges data with the simulation through ROS topics.

The distributed architecture of ROS is based on several core concepts. A node is an executable process that performs a specific function, such as simulating the robot, publishing joint states, managing controllers, or computing torque commands. Nodes exchange data through topics, which are asynchronous communication channels based on a publish/subscribe mechanism. In this project, the most important topics are `/joint_states`, which provides feedback from the robot, and `/joint_group_eff_controller/command`, which

receives the torque commands generated by the controller.

The exchanged data are carried by messages, which are strictly typed data structures. The main message types used in this implementation are `sensor_msgs/JointState` for joint position and velocity feedback and `std_msgs/Float64MultiArray` for torque command transmission. In addition to topics, ROS also provides services for synchronous request-response communication. Services are useful for operations such as switching controllers or resetting the Gazebo simulation.

The ROS master, launched through `roscore`, coordinates the registration of nodes, topics, and services so that all processes can discover and communicate with each other. The parameter server stores shared configuration data, including the robot description, controller parameters, and simulation settings. The TF library manages coordinate transformations between robot frames, allowing the system to query the pose relationship between links during simulation.

At the filesystem level, ROS organizes software into packages. Each package contains the source files, configuration files, launch files, and dependencies required for a specific functionality. This structure supports code reuse and simplifies the integration of external packages such as Gazebo interfaces, robot descriptions, and ROS control plugins. Figure 2.1 illustrates the general organization of the ROS filesystem.

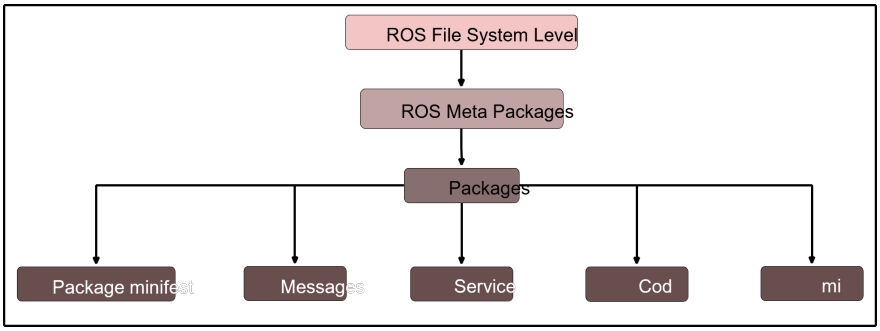


Figure 2.1: ROS filesystem organization.

2.2 Core ROS Concepts

The ROS communication architecture used in this work is summarized in Figure 2.2. The system is organized as a computation graph in which the main processes are connected through topics, services, and the parameter server. Understanding the ROS implementation presented in this chapter requires familiarity with the following fundamental concepts:

Nodes A node is the basic computational unit in ROS — an executable process that performs a specific function. Nodes are written in C++ or Python and communicate with each other through the ROS communication infrastructure. Examples in this

project include the Gazebo simulation node, the `robot_state_publisher` node, the `controller_manager` node, and the MATLAB node.

Topics A topic is a named communication channel through which nodes exchange messages asynchronously using a publish/subscribe pattern. A publisher node writes messages to a topic; any number of subscriber nodes can read from the same topic. Topics are unidirectional and many-to-many. The most important topics in this project are `/joint_states` (robot feedback) and `/joint_group_eff_controller/command` (torque commands).

Messages A message is a strictly-typed data structure used for communication on topics. ROS provides a large library of standard message types (`std_msgs`, `sensor_msgs`, `geometry_msgs`, `trajectory_msgs`) as well as tools for defining custom types. The two message types used in this project are
`sensor_msgs/JointState` and
`std_msgs/Float64MultiArray`.

Services A service provides synchronous request-response communication between nodes. Unlike topics, a service call blocks until the server responds. Services are used in this project for controller switching (`controller_manager/switch_controller`) and simulation reset (`gazebo/reset_simulation`).

ROS Master The ROS master is a central coordination process (`roscore`) that manages the registration of nodes, topics, and services. It must be running before any other ROS node can communicate. The master URI identifies its location on the network.

Parameter Server A shared dictionary accessible by all nodes, used to store configuration parameters. The robot URDF description, controller gains, and simulation parameters are stored on the parameter server.

TF (Transform) Library The TF library manages the tree of coordinate frame transformations throughout the robot system. The `robot_state_publisher` node subscribes to `/joint_states` and uses the URDF to compute and broadcast the complete TF tree, enabling any node to query the relative pose between any two frames at any time.

This architecture allows Gazebo, ROS control, robot-state publishing, and MATLAB/Simulink to operate together as a single integrated simulation platform.

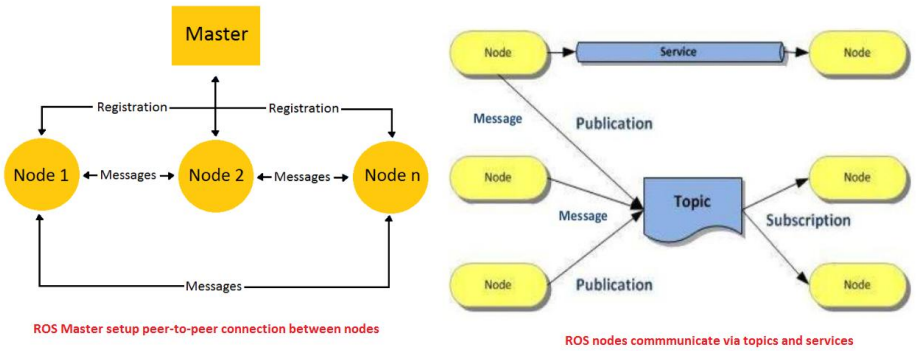


Figure 2.2: ROS communication architecture diagram.

2.2.1 ROS Computation Graph

The computation graph is the peer-to-peer network of ROS nodes connected by topics and services. In this project, the graph includes the Gazebo simulation node, the robot-state publisher, the controller manager, the joint-state controller, the effort controller, and the MATLAB global node. Table 2.1 summarizes the role of each node in the implemented architecture.

Table 2.1: ROS Computation Graph Nodes

Node	Package	Function
`/gazebo`	`gazebo_ros`	Physics simulation of UR5
`/robot_state_publisher`	`robot_state_publisher`	Broadcasts TF from URDF + joint states
`/controller_manager`	`ros_control`	Manages hardware interface and controllers
`/joint_state_controller`	`joint_state_controller`	Publishes `/joint_states` at 125 Hz
`/joint_group_eff_controller`	`effort_controllers`	Receives torque commands, applies to simulation
`/matlab_global_node`	`ROS Toolbox`	MATLAB controller and trajectory planner
`/rosout`	`rosconsole`	Logging aggregator

2.3 UR5 ROS Environment

2.3.1 Required Packages

The ROS environment for the UR5 requires the following packages, installed in the catkin workspace:

`universal_robot` (ROS-Industrial, GitHub: `ros-industrial/universal_robot`)
: This repository is the ROS-Industrial support package for Universal Robots manipulators. It contains the robot description, Gazebo simulation files, kinematic solvers, MoveIt configurations, and supporting messages needed to model and simulate UR manipulators in ROS [6]. The main packages used in this work are:

— `ur_description`: URDF/Xacro robot description files, meshes, joint limits, and transmission definitions for Universal Robots manipulators.
— `ur_gazebo`: Gazebo launch files and simulation configurations for UR robots.
— `ur5_moveit_config`: MoveIt configuration files for the UR5 manipulator, including planning groups and controller interfaces.
— `ur_kinematics`: analytical kinematic solver plugins for Universal Robots arms.
— `ur_msgs`: ROS message definitions used by the Universal Robots support stack.

`ros_control` and `ros_controllers` The standard ROS control framework is used to connect the Gazebo model to the effort controllers required by the computed-torque experiments. The required packages are installed with:

```
sudo apt install ros-noetic-ros-control ros-noetic-ros-controllers
sudo apt install ros-noetic-effort-controllers
sudo apt install ros-noetic-ur-gazebo
```

2.3.2 Workspace Configuration

The catkin workspace is based on the ROS-Industrial `universal_robot` repository and is organized as follows:

```
catkin_ws/
  |-- src/
    |-- universal_robot/
      |-- ur_description/
      |-- ur_gazebo/
      |-- ur5_moveit_config/
      |-- ur_kinematics/
      |-- ur_msgs/
  |-- build/
  |-- devel/
  |-- logs/
```

The repository is cloned into the workspace source folder, then the workspace is built and sourced with:

```
cd ~/catkin_ws/src
git clone https://github.com/ros-industrial/universal_robot.git
cd ~/catkin_ws
catkin_make
source devel/setup.bash
```

2.3.3 Robot Description (URDF/Xacro)

The UR5 robot description is provided by the `ur_description` package. The Xacro model defines the kinematic chain, visual and collision meshes, joint limits, transmissions, and inertial parameters used by Gazebo and `ros_control`. For the UR5 model, the description files are loaded from the `ur_description/urdf` directory and are published to the ROS parameter server as `robot_description` before starting the simulation.

Joint Definitions Six revolute joints are defined with their origin positions and orientations, joint limits, velocity limits, effort limits, and transmission interfaces.

Link Definitions The base and arm links include visual meshes, collision geometry, masses, centers of mass, and inertia tensors consistent with the UR support package data.

Gazebo Integration The `ur_gazebo` package connects the URDF model to Gazebo and the `ros_control` framework so that effort controllers can apply torque commands during simulation.

The UR5 Gazebo model is launched through the ROS-Industrial simulation package; this launch process loads the robot description, starts Gazebo, and spawns the UR5 model with the configured controllers.

2.3.4 Launching the Simulation

The complete simulation environment is launched with:

```
source /opt/ros/noetic/setup.bash
source ~/catkin_ws/devel/setup.bash
roslaunch ur_gazebo ur5_bringup.launch gui:=true
```

This single launch file performs the following actions in sequence:

1. Starts the ROS master (`roscore`) if not already running
2. Loads the UR5 URDF onto the parameter server
3. Starts Gazebo with the default empty world
4. Spawns the UR5 model in Gazebo at the origin
5. Loads the controller configuration from `ur5_controllers.yaml`
6. Starts the `controller_manager` node
7. Activates the `joint_state_controller`
8. Activates the `eff_joint_traj_controller` by default

2.4 Communication Architecture

2.4.1 Topics

During the control experiment, ROS topics form the communication link between Gazebo, the robot state publisher, and the MATLAB/Simulink controller. The main topics are divided into feedback topics, which report the simulated robot state, and command topics, which receive the torque or trajectory references.

Feedback Topics (published by Gazebo/ros_control)

Table 2.2: Feedback Topics

Topic	Type	Publisher	Frequency
`/joint_states`	`sensor_msgs/JointState`	`joint_state_controller`	125 Hz
`/gazebo/model_states`	`gazebo_msgs/ModelStates`	`gazebo`	1000 Hz
`/gazebo/link_states`	`gazebo_msgs/LinkStates`	`gazebo`	1000 Hz
`/clock`	`rosgraph_msgs/Clock`	`gazebo`	1000 Hz
`/tf`	`tf2_msgs/TFMessage`	`robot_state_publisher`	50 Hz

Command Topics (subscribed by ros_control)

Table 2.3: Command Topics

Topic	Type	Subscriber	Description
`/joint_group_eff_controller/command`	`std_msgs/Float64MultiArray`	`joint_group_eff_controller`	Direct torque commands [Nm]
`/eff_joint_traj_controller/command`	`trajectory_msgs/JointTrajectory`	`eff_joint_traj_controller`	Trajectory commands

The most important feedback channel is `/joint_states`.

2.4.2 Node Architecture

The resulting node graph is summarized below. Gazebo publishes the joint state feedback, the `controller_manager` supervises the active controllers, and the MATLAB/Simulink node closes the torque-control loop by publishing commands to the effort controller.

```
[gazebo] --> /joint_states --> [robot_state_publisher]
      |                                |
      |                               /tf
      v
[controller_manager]
      |
[joint_group_eff_controller]
      ^
```

```
      |
/joint_group_eff_controller/command
      ^
      |
[matlab_global_node]  (Simulink controller)
```

2.4.3 Controller Configuration

The `ros_control` framework provides a hardware abstraction layer between the control algorithms and the robot simulation. Three controllers are configured for the UR5:

`joint_state_controller` (`joint_state_controller/JointStateController`) Reads joint positions, velocities, and efforts from the hardware interface and publishes them to `/joint_states`. Always active.

`eff_joint_traj_controller` (`effort_controllers/JointTrajectoryController`) Accepts `JointTrajectory` messages specifying a sequence of waypoints with timing. Internally implements a PID controller to track each waypoint. Active by default at launch but deactivated for computed torque control experiments.

`joint_group_eff_controller` (`effort_controllers/JointGroupEffortController`) Accepts `std_msgs/Float64MultiArray` messages containing six torque values (one per joint) and directly applies them to the simulation. This is the controller used by the computed torque controller. It must be explicitly activated.

For the computed torque experiments, the active controllers are managed from the terminal using the `controller_manager` command-line tool. First, the available controllers are listed. Then the default trajectory controller is stopped, and the direct effort controller is started:

```
rosrun controller_manager controller_manager list
rosrun controller_manager controller_manager stop eff_joint_traj_controller
rosrun controller_manager controller_manager start joint_group_eff_controller
```

After this procedure, `joint_group_eff_controller` becomes the active controller used to receive the torque commands generated by the computed torque controller.

The corresponding YAML configuration defines the effort controller and the six actuated joints:

```
joint_group_eff_controller:
  type: effort_controllers/JointGroupEffortController
  joints:
    - elbow_joint
    - shoulder_lift_joint
    - shoulder_pan_joint
```

```
- wrist_1_joint
- wrist_2_joint
- wrist_3_joint
```

2.4.4 MATLAB–ROS Communication

Before launching the controller, the ROS network variables are configured in the ROS terminal. These variables identify the ROS master and the IP address used by this machine to advertise its ROS nodes:

```
export ROS_MASTER_URI=http://172.22.56.168:11311
export ROS_IP=172.22.56.168
```

The MATLAB Robotics System Toolbox provides the ROS interface used by the Simulink controller. After setting the ROS environment in the terminal, MATLAB is connected to the same ROS master using the following commands:

```
rosinit('http://172.22.56.168:11311', 'NodeHost', '172.22.56.168');
```

Using the same IP address in both the terminal and MATLAB ensures that ROS nodes can discover each other and that subscriber callbacks receive data correctly during the Gazebo–MATLAB/Simulink communication.

2.5 Implementation of the Computed Torque Controller

2.5.1 Simulink Block Diagram

The computed torque controller is implemented as a closed-loop Simulink model connected to Gazebo through ROS. The model receives joint feedback from the simulator, generates the desired fifth-degree polynomial trajectory, evaluates the computed torque law, limits the resulting torque commands, and publishes the final command vector back to the effort controller in Gazebo.

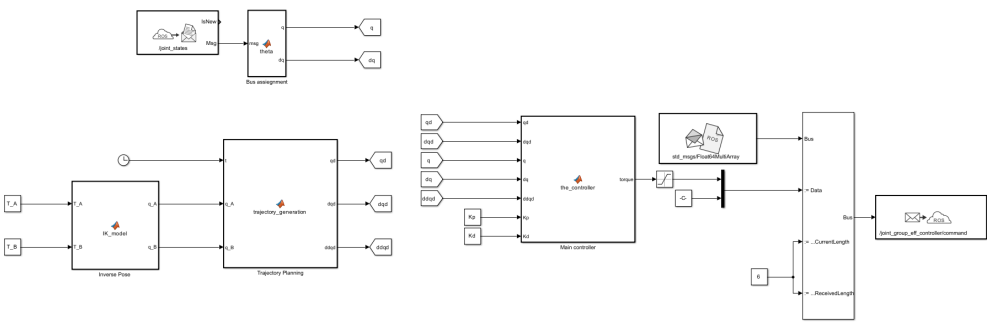


Figure 2.3: Simulink implementation of the computed torque controller.

The main functional blocks of the Simulink model are described below.

Subscribe Block (`robotlib/Subscribe`) This block subscribes to `/joint_states` using the message type `sensor_msgs/JointState`. Its sample time is set to 0.05 s, corresponding to a nominal control frequency of 20 Hz. The output is a message bus containing the joint names, positions, velocities, and efforts.

`theta` Block (MATLAB Function) This block extracts the joint position vector $\mathbf{q}$ and velocity vector $\dot{\mathbf{q}}$ from the ROS feedback message. Since `/joint_states` publishes the joints in alphabetical order, the block reorders the measurements to match the DH model order (`shoulder_pan`, `shoulder_lift`, `elbow`, `wrist_1`, `wrist_2`, `wrist_3`):

```
function [q, dq] = PosVel(msg)
pos = msg.Position(1:6);
vel = msg.Velocity(1:6);
q = [
    pos(3);   % shoulder_pan_joint
    pos(2);   % shoulder_lift_joint
    pos(1);   % elbow_joint
    pos(4);   % wrist_1_joint
    pos(5);   % wrist_2_joint
    pos(6)    % wrist_3_joint
];
dq = [
    vel(3);   % shoulder_pan_joint
    vel(2);   % shoulder_lift_joint
    vel(1);   % elbow_joint
    vel(4);   % wrist_1_joint
    vel(5);   % wrist_2_joint
    vel(6)    % wrist_3_joint
];
end
```

`IK_model` Block (MATLAB Function) This block receives the initial and final end-effector poses, represented by the homogeneous transformation matrices T_A and T_B. It applies the analytical URe-series inverse kinematics solution to compute the corresponding joint configurations $\mathbf{q}_A$ and $\mathbf{q}_B$ [2, 12].

`trajectory_generation` Block (MATLAB Function) This block uses the clock time t and the joint configurations $\mathbf{q}_A$ and $\mathbf{q}_B$ to generate smooth desired position, velocity, and acceleration trajectories. The normalized time variable is $\lambda = t/T_f$, saturated at one when the final time is reached:

```
function [qd, dqd, ddqd] = trajectory_generation(t, q_A, q_B)
t_f     = 30.0;
delta_q = q_B - q_A;
     lambda  = t / t_f;
    qd   = q_A + delta_q * (10*lambda^3 - 15*lambda^4 + 6*lambda^5);
    dqd  = delta_q * ( (30/t_f)*lambda^2 - (60/t_f)*lambda^3 + (30/t_f)*lambda^4 )
    ddqd = delta_q * ( (60/t_f^2)*lambda - (180/t_f^2)*lambda^2 + (120/t_f^2)*lambd
end
```

Sum Block The Sum block computes the joint tracking error $\mathbf{q}_e = \mathbf{q}_d - \mathbf{q}$, which is used by the feedback part of the controller.

`the_controller` **Block (MATLAB Function)** This block evaluates the computed torque control law. It receives $\mathbf{q}_d$, $\dot{\mathbf{q}}_d$, $\ddot{\mathbf{q}}_d$, $\mathbf{q}$, $\dot{\mathbf{q}}$, and the dynamic parameters `cparam`. It computes the torque vector using $M(\mathbf{q})\mathbf{y} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + G(\mathbf{q})$, where M, C, and G are obtained from the Euler–Lagrange dynamic model [8, 7, 3].

Saturation Block The torque output is limited to $[-150, +150]$ Nm. This prevents unrealistic torque commands and keeps the simulation within physically acceptable limits.

Blank Message and Bus Assignment Blocks A blank `std_msgs/Float64MultiArray` message is created, and the saturated torque vector is assigned to its `Data` field.

Publish Block (`robotlib/Publish`) The final torque vector is published to `/joint_group_eff_co` where it is applied to the Gazebo UR5 model through the effort controller.

Scope Blocks Scope blocks are connected to the main signals, including $\mathbf{q}$, $\mathbf{q}_d$, $\mathbf{q}_e$, and the torque vector, to monitor the simulation in real time.

2.5.2 Data Flow Summary

The complete control-loop data flow is summarized as follows:

```
Gazebo --> /joint_states --> Subscribe --> PosVel --> q, dq
                                                 |
Clock --> trajectory_generation <-- IK_model <-- T_A, T_B
                     |
                     v
              qd, dqd, ddqd
                     |
                     v
the_controller(qd, dqd, ddqd, q, dq, cparam) --> torque
                     |
                     v
Saturation --> Bus Assignment --> Publish
```

```
                    |
                    v
/joint_group_eff_controller/command --> Gazebo
```

2.5.3 Control Loop Timing

The Simulink model is configured with a sample time of 0.05 s, giving a nominal control frequency of 20 Hz. This is lower than the 125 Hz update rate of the `ros_control` framework. In practice, the effective control frequency varies between 10 and 20 Hz because of MATLAB–ROS communication overhead and the computational cost of evaluating the dynamic model at each control step.

2.6 Simulation Setup

The Gazebo simulation environment used for the validation tests is shown in Figure 2.4. The UR5e model is placed in an empty world so that the evaluation focuses only on the controller behaviour and not on external interactions with obstacles.

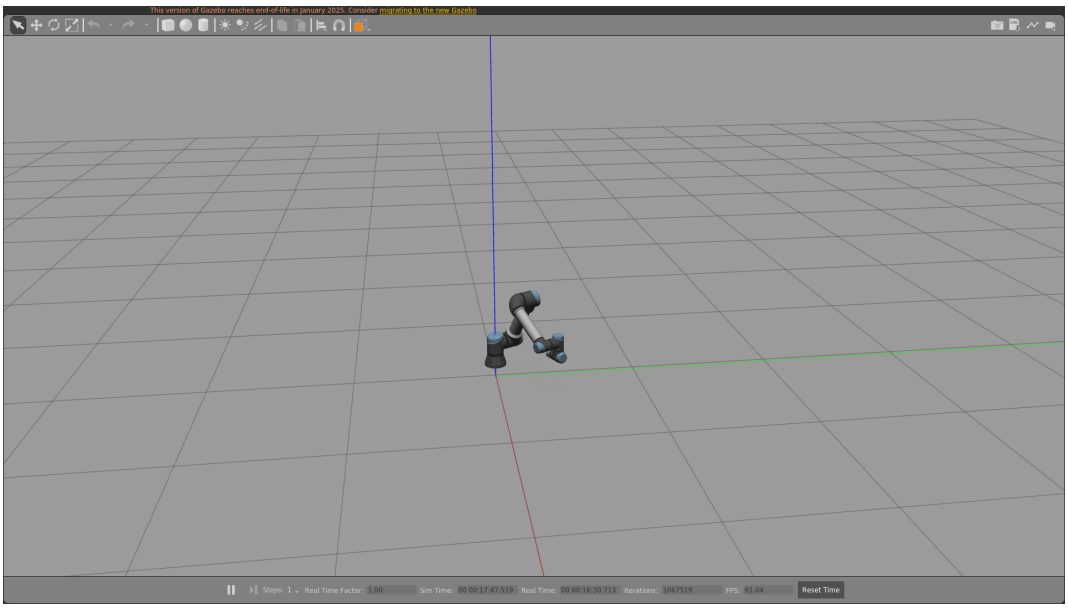


Figure 2.4: Gazebo simulation environment used for validating the UR5e trajectory tracking controllers.

The main simulation parameters associated with this Gazebo setup are summarized in Table 2.4.

Table 2.4: Gazebo Simulation Parameters

Parameter	**Value**
Gazebo version	11.x
Physics engine	ODE (Open Dynamics Engine)
Simulation step size	1 ms (1000 Hz)
Real-time update rate	1000 Hz (target)
Real-time factor	∼0.6–0.9 (achieved)
World	Empty world (no obstacles)
Gravity	$[0,\ 0,\ -9.81]\,\mathrm{m/s^2}$
UR5 initial pose	Home configuration (all joints = 0 rad)

Chapter 3

Results and Discussion

3.1 Trajectory Tracking Tests

3.1.1 Experimental Setup

The validation tests were performed in the Gazebo physics simulator using the ROS–MATLAB/Simulink control architecture described in the previous chapter [9]. At the beginning of each test, the current joint configuration was read from the `/joint_states` topic and used as the initial configuration. This avoided artificial initial offsets and ensured that the tracking error started from the actual simulated robot state.

The `joint_group_eff_controller` was activated through `controller_manager`, allowing the Simulink model to send direct joint torque commands to Gazebo. Three controllers were then evaluated under the same trajectory, the same sampling conditions, and the same gain matrices:

1. **Pure PD controller:** feedback control without model-based dynamic compensation.
2. **PD with gravity compensation (PD+G):** feedback control augmented by the gravity vector $\mathbf{G}(\mathbf{q})$.
3. **Computed Torque Controller (CTC):** model-based inverse dynamics control using $\mathbf{M}(\mathbf{q})$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$, and $\mathbf{G}(\mathbf{q})$.

The effective control frequency observed during the tests was approximately 10–15 Hz. This relatively low rate is an important factor when interpreting the tracking performance, especially for the low-inertia wrist joints.

3.1.2 Reference Trajectory

The desired motion is a point-to-point displacement between two end-effector poses. The initial pose is

$$T_A = \begin{bmatrix} -0.3536 & 0.3536 & 0.8660 & -0.3253 \\ 0.6124 & -0.6124 & 0.5000 & -0.3417 \\ 0.7071 & 0.7071 & 0.0000 & 0.2348 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}. \tag{3.1}$$

The final pose, corresponding to the left-side pose T_3, is

$$T_B = \begin{bmatrix} -0.3706 & 0.6830 & -0.6294 & -0.7054 \\ -0.6294 & -0.6830 & -0.3706 & 0.4173 \\ -0.6830 & 0.2588 & 0.6830 & 0.4037 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}. \tag{3.2}$$

The inverse pose kinematics block converts these two Cartesian poses into the corresponding joint configurations $\mathbf{q}_A$ and $\mathbf{q}_B$ [2, 12]. A fifth-degree polynomial interpolation is then used to generate the desired joint position, velocity, and acceleration profiles over a duration of $T_f = 30\,\mathrm{s}$:

$$\{T_A, T_B\} \xrightarrow{\text{IPK}} \{\mathbf{q}_A, \mathbf{q}_B\} \xrightarrow{5^{\text{th}}\text{-order interpolation}} \{\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d\}. \tag{3.3}$$

The imposed boundary conditions are

$$\mathbf{q}_d(0) = \mathbf{q}_A, \qquad \dot{\mathbf{q}}_d(0) = \mathbf{0}, \qquad \ddot{\mathbf{q}}_d(0) = \mathbf{0} \tag{3.4}$$

$$\mathbf{q}_d(T_f) = \mathbf{q}_B, \qquad \dot{\mathbf{q}}_d(T_f) = \mathbf{0}, \qquad \ddot{\mathbf{q}}_d(T_f) = \mathbf{0}. \tag{3.5}$$

These constraints produce smooth reference trajectories and avoid commanding discontinuous velocities or accelerations at the beginning and end of the motion.

3.1.3 Controller Parameters

The same gains and saturation limits were used for the three controllers so that the comparison focused on the control structure rather than on gain retuning.

Table 3.1: Controller parameters used in the trajectory tracking tests.

Parameter	**Value**
K_P	$\mathrm{diag}(3500, 3500, 3500, 3500, 3500, 3500)$
K_D	$\mathrm{diag}(80, 80, 60, 50, 40, 30)$
Trajectory duration T_f	$30\,\mathrm{s}$
Software torque saturation	$\pm\,(150, 150, 150, 28, 28, 28)\,\mathrm{Nm}$

3.2 Control Strategy 1: Pure PD Controller

The pure PD controller yields markedly heterogeneous tracking performance across the six joints. The proximal joints (1–3) exhibit moderate trajectory following: Joint 1 and Joint 3 remain in the vicinity of the desired path, albeit with appreciable high-frequency noise and phase lag. Joint 2, however, suffers from a persistent tracking offset during the middle phase of the motion. The wrist joints (4–6) display significantly larger deviations; in particular, Joint 6 exhibits severe error spikes that indicate the controller is unable to stabilize the low-inertia wrist dynamics against unmodeled coupling and gravity.

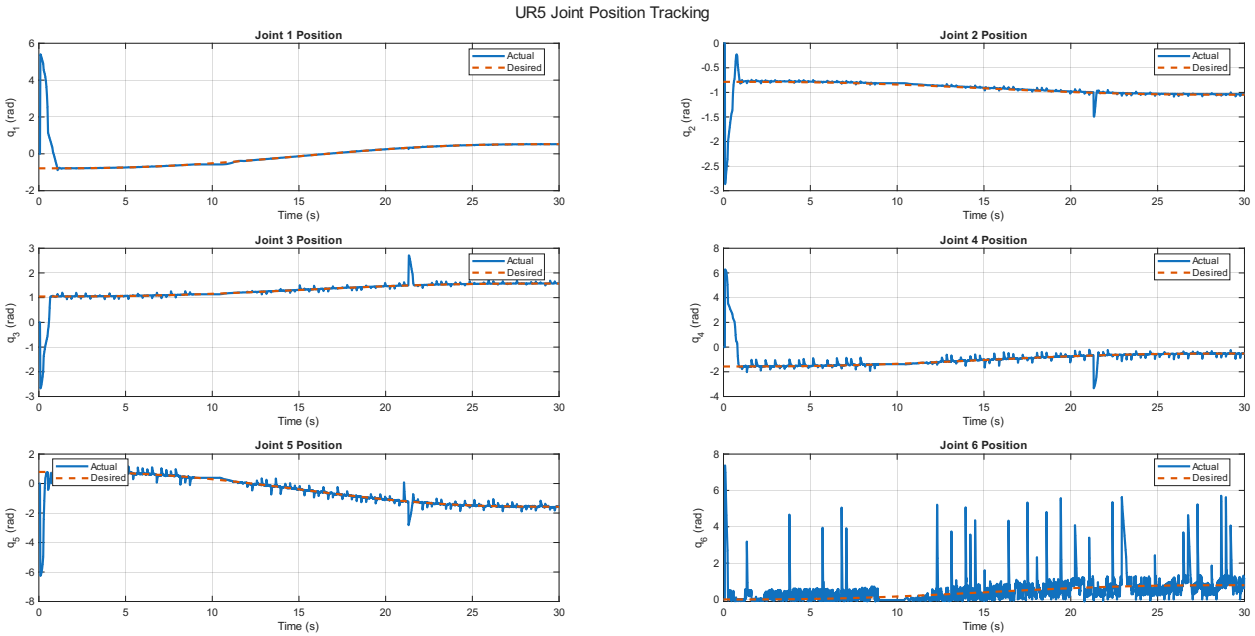


Figure 3.1: Joint position tracking under pure PD control. Blue: actual position; orange dashed: desired position.

The tracking errors in Figure 3.2 confirm that the PD controller fails to achieve uniform convergence. While the initial error on some joints decays partially, the steady-state response remains oscillatory and does not settle to a small bound. Joint 6 is especially problematic, with recurrent error spikes exceeding 5 rad, demonstrating that proportional-derivative action alone is inadequate for the nonlinear, coupled dynamics of the UR5e.

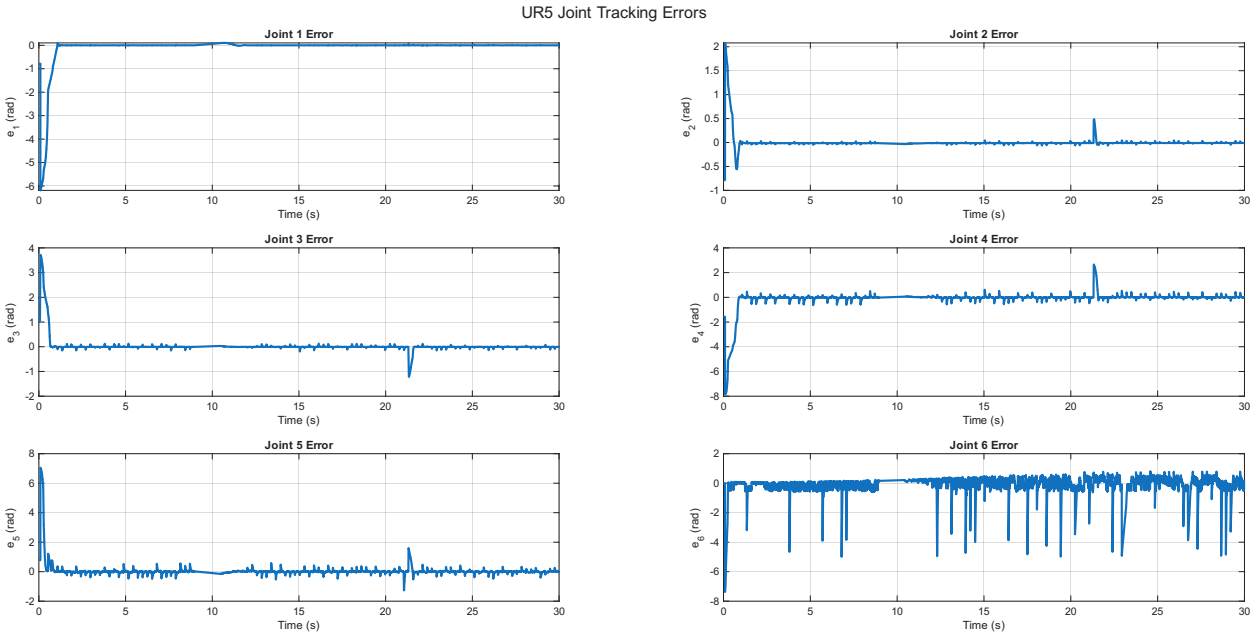


Figure 3.2: Joint tracking errors under pure PD control.

The fundamental weakness of the pure PD strategy is evident in the torque profiles. As shown in Figure 3.3, the commanded torques exhibit high-frequency bang-bang-like switching and repeatedly saturate at the software limits of ± 150 Nm for Joints 1–3 and ± 30 Nm for Joints 4–6. This behavior indicates that the controller attempts to compensate for gravity, inertial coupling, and Coriolis effects entirely through high-gain feedback, thereby demanding excessive actuator effort and risking excitation of unmodeled structural modes.

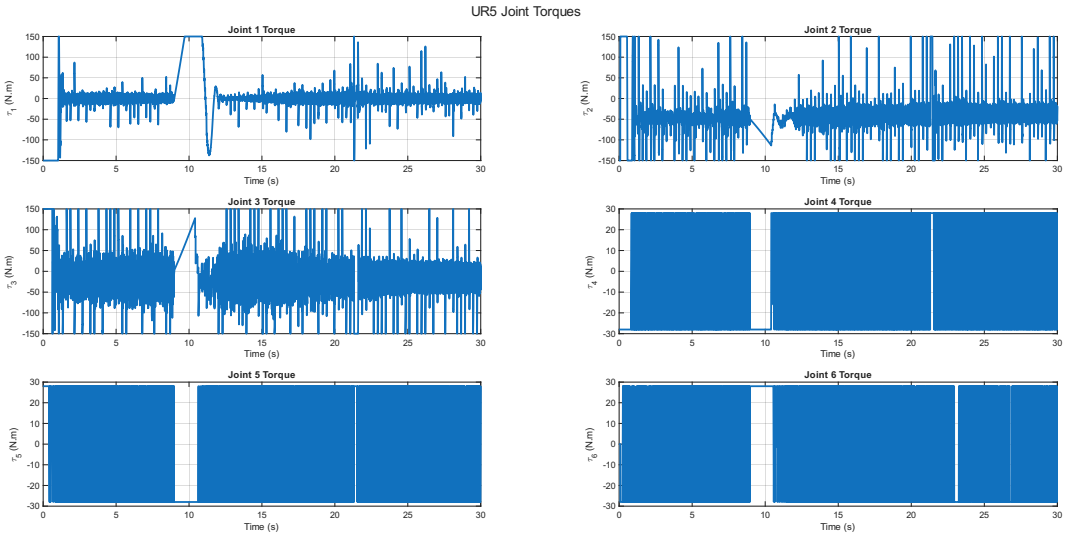


Figure 3.3: Joint torques under pure PD control.

3.3 Control Strategy 2: PD with Gravity Compensation (PD+G)

Incorporating gravity compensation improves the static consistency of the control law, and this is reflected in reduced steady-state bias on the gravity-loaded proximal joints (notably the shoulder and elbow). However, the improvement is not uniform. While Joints 1–3 show some attenuation of low-frequency drift, the wrist joints remain difficult to stabilize because gravity compensation alone does not cancel inertial and Coriolis coupling.

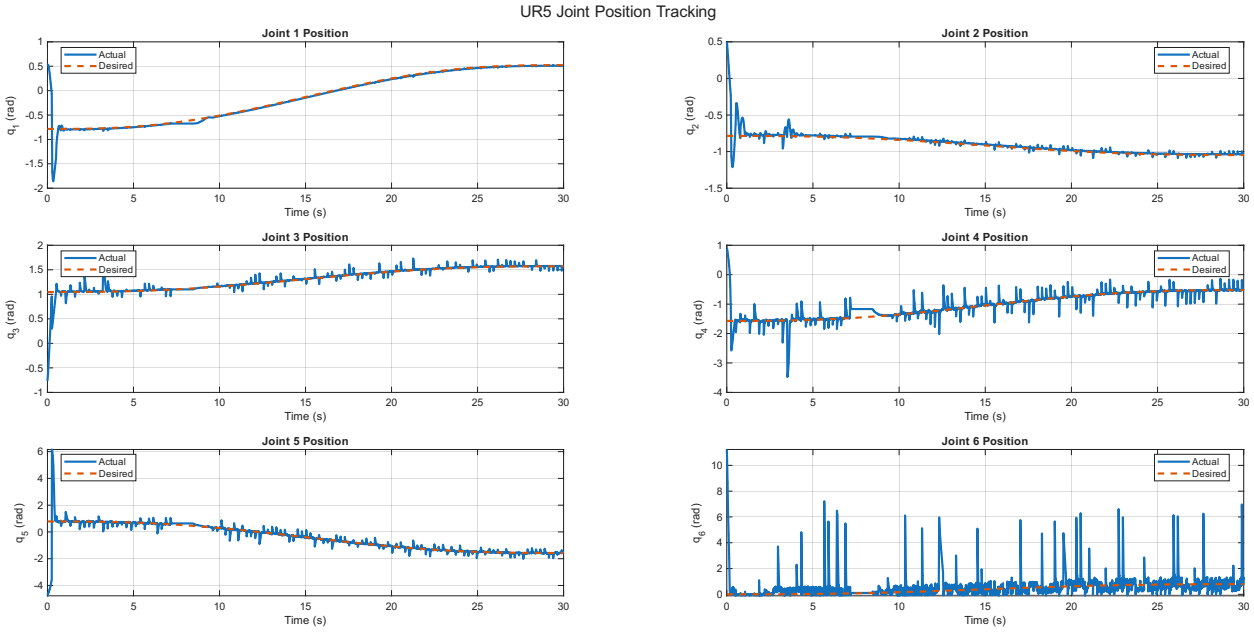


Figure 3.4: Joint position tracking under PD with gravity compensation (PD+G). Blue: actual; orange dashed: desired.

The error curves in Figure 3.5 indicate that gravity compensation alone does not eliminate the transient oscillations or the persistent tracking offsets observed under pure PD. The wrist joints continue to exhibit sensitivity to dynamic coupling and to the limited effective control bandwidth.

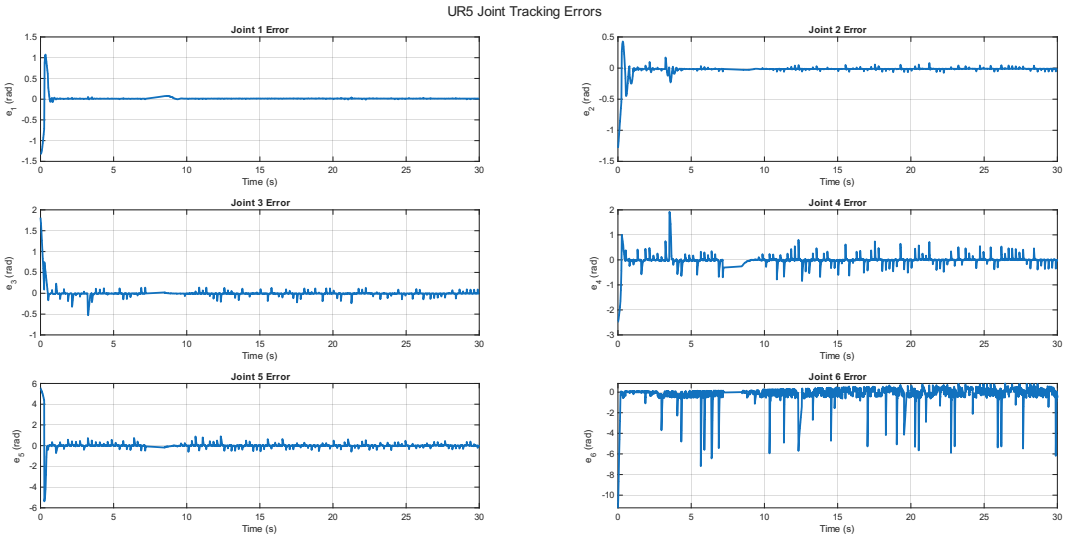


Figure 3.5: Joint tracking errors under PD with gravity compensation (PD+G).

The torque profiles in Figure 3.6 reveal that the PD+G controller continues to demand aggressive actuator effort. Joints 1–3 still display severe chattering and saturation. More critically, Joints 4–6 exhibit sustained saturation at the torque limits for extended intervals, suggesting that the static gravity term is insufficient to prevent the feedback loop from railing the low-torque wrist actuators. Consequently, any tracking improvements on the proximal joints are achieved at the cost of continued high actuator stress and discontinuous command signals on the distal joints.

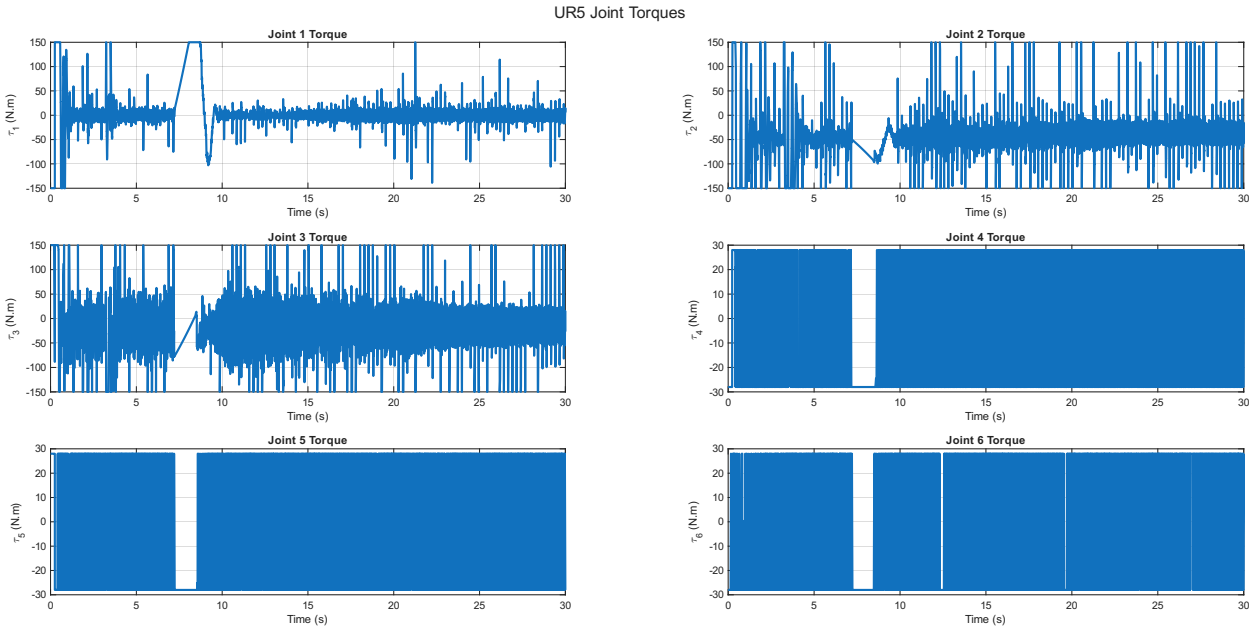


Figure 3.6: Joint torques under PD with gravity compensation (PD+G).

3.4 Control Strategy 3: Computed Torque Controller (CTC)

The computed torque controller delivers a qualitatively different performance. By pre-compensating the nonlinear dynamics via inverse dynamics, the CTC achieves accurate trajectory tracking across *all six joints*. The proximal joints (1–3) follow the desired profiles with small, rapidly decaying deviations. Crucially, the wrist joints (4–6) also track well; their errors converge quickly and remain small, contradicting the expectation that low-inertia distal joints would necessarily perform worse under model-based control.

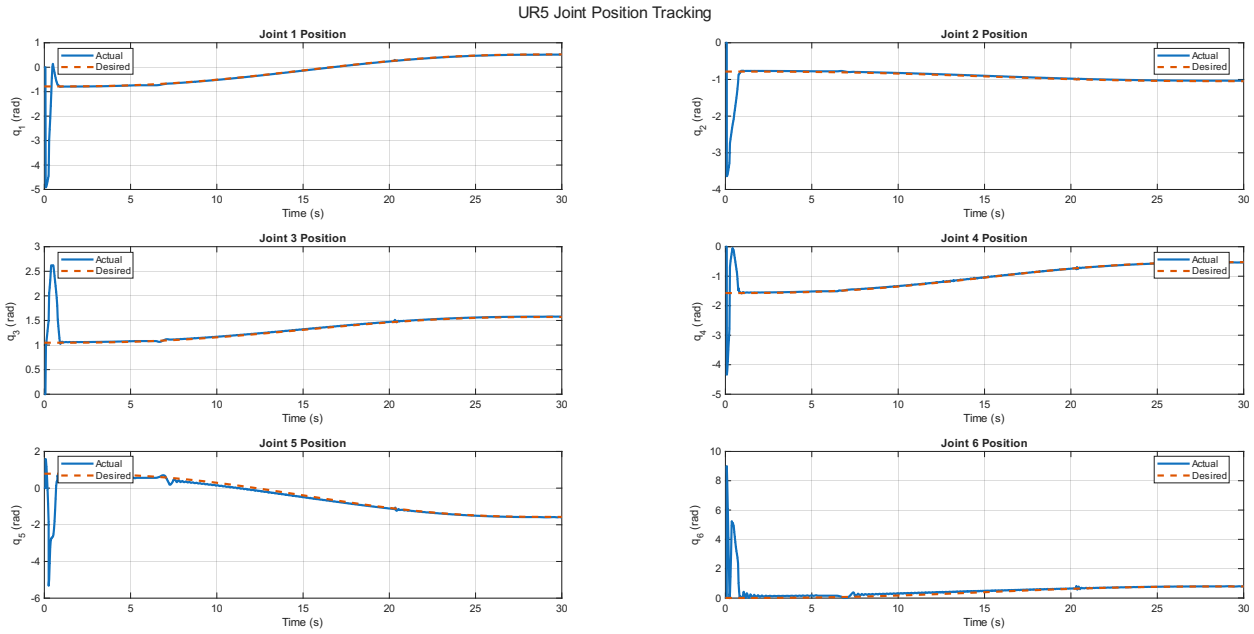


Figure 3.7: Joint position tracking under Computed Torque Control (CTC). Blue: actual; orange dashed: desired.

The error plots in Figure 3.8 highlight the uniform superiority of the CTC. All six joints exhibit rapid transient convergence to a tight neighborhood of zero within approximately one to two seconds, and the residual steady-state errors are negligible compared with those of the PD and PD+G controllers. This confirms that the dynamic decoupling provided by the feedforward term is effective for the full manipulator, including the wrist.

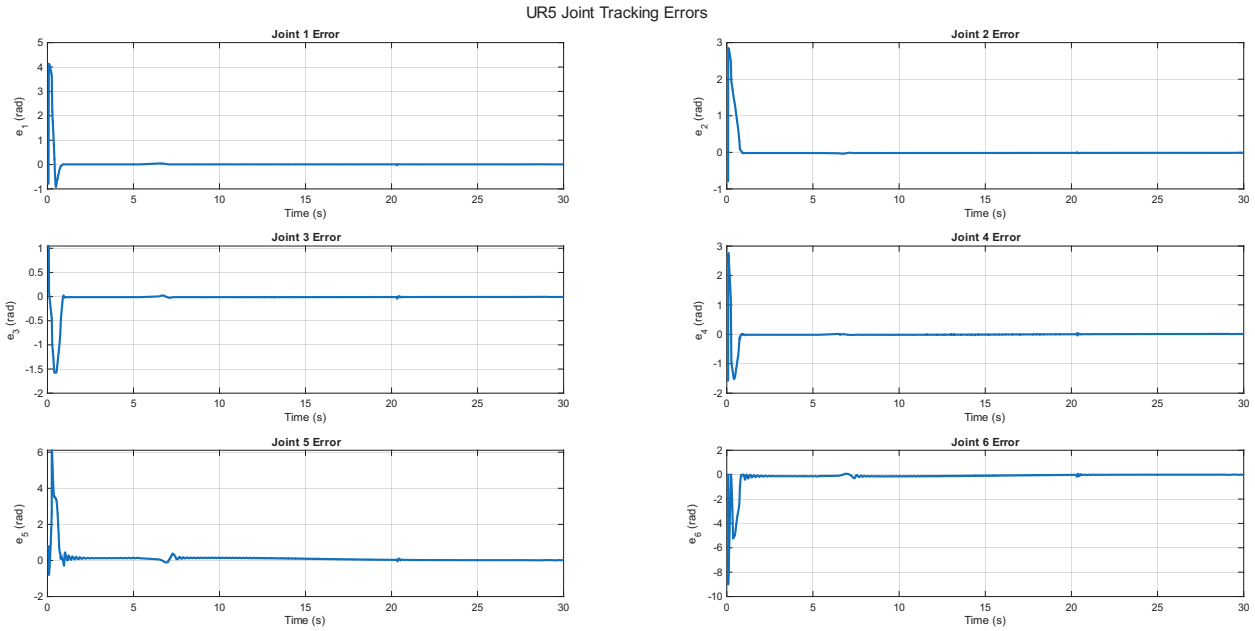


Figure 3.8: Joint tracking errors under Computed Torque Control (CTC).

The principal advantage of the CTC is manifest in the torque response. As shown in Figure 3.9, the control inputs are smooth and physically plausible. After brief initial transients required to synchronize the states, the torques settle to low-amplitude profiles that reflect the nominal dynamics of the trajectory rather than high-gain corrective action. The absence of the aggressive switching and sustained saturation seen in PD and PD+G demonstrates that the inverse dynamics compensation successfully offloads the feedback controller, leaving it to correct only small residual perturbations.

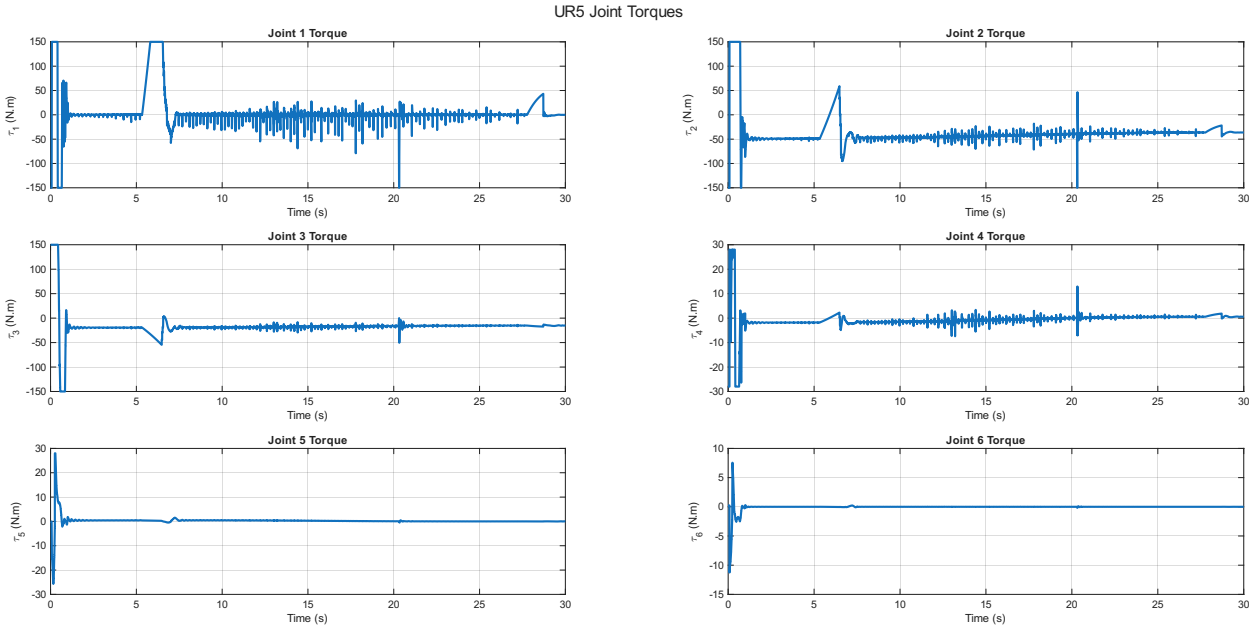


Figure 3.9: Joint torques under Computed Torque Control (CTC).

3.5 Discussion of Results

3.5.1 Tracking Performance

The experimental results establish a clear hierarchy in tracking accuracy. The CTC provides uniformly superior position tracking for all six joints, with rapid error convergence and small residual deviations. In contrast, both PD and PD+G struggle to follow the desired trajectory accurately, particularly on the wrist joints. While PD+G marginally improves the proximal joint behavior by removing static gravity bias, it does not address the fundamental dynamic coupling that degrades wrist performance. The success of the CTC across the entire manipulator indicates that the primary limitation of the feedback-only strategies is not merely parametric uncertainty or gravity, but the lack of dynamic compensation for inertial and Coriolis coupling.

3.5.2 Control Torque Behavior

The torque profiles reveal the most striking difference between the strategies. PD and PD+G produce severely distorted, high-frequency commands that saturate the actuators. This chattering is not merely an actuator inconvenience; it is a symptom of structural controller inadequacy, whereby the feedback loop attempts to invert the plant dynamics through bang-bang action. Such behavior is inefficient, potentially harmful to hardware, and ineffective at improving tracking.

The CTC, by contrast, generates smooth, bounded torque signals. The feedforward term provides the bulk of the required control effort, while the feedback term handles only small perturbations. Even during trajectory segments with higher acceleration demands, the CTC torques remain free of the persistent saturation observed under the other strategies. This smoothness is a critical practical advantage for safe robot operation and longevity of the actuators.

3.5.3 Main Limiting Factors

The remaining discrepancies between desired and actual trajectories under CTC can be attributed to the following implementation constraints:

Low control frequency The effective control loop of approximately 10–15 Hz introduces discretization effects and delay. While the CTC tolerates this limitation because the feedforward term is largely independent of the current state measurement, the feedback-only controllers are severely affected: their high gains amplify quantization noise and delay-induced phase lag, explaining the observed chattering and instability on the fast wrist dynamics.

Torque saturation The software limits of ±150 Nm (Joints 1–3) and ±30 Nm (Joints 4–6) constrain the achievable acceleration. Under PD and PD+G, these limits are reached continuously, preventing the controller from delivering the necessary corrective action. Under CTC, saturation occurs only briefly during initial transients, indicating that the nominal trajectory demands are compatible with the actuator capabilities when properly pre-compensated.

Communication delay The MATLAB–ROS–Gazebo chain introduces transport and processing delays. For PD and PD+G, this delay reduces phase margin and exacerbates the oscillatory response. The CTC is more robust to this delay because the dominant feedforward signal is computed from the reference trajectory rather than from delayed state measurements.

Model mismatch Minor discrepancies between the symbolic dynamic model and the Gazebo URDF inertial parameters exist. However, the strong performance of the CTC demonstrates that these mismatches are sufficiently small to be handled by the low-gain feedback component; they do not prevent stable, accurate tracking.

In summary, the experimental comparison demonstrates that dynamic model compensation is essential for achieving accurate, smooth, and hardware-friendly control of the UR5e manipulator in a realistic simulation environment. The CTC's combination of feedforward decoupling and low-gain feedback is decisively superior to high-gain feedback strategies that ignore the robot's nonlinear dynamics.

3.6 Conclusion

This chapter compared pure PD, PD+G, and CTC controllers for joint-space trajectory tracking of the UR5e manipulator in Gazebo. The PD and PD+G controllers achieved relatively good tracking for the tested trajectory, but their input torques were aggressive, distorted, and frequently bounded by the saturation limits. This behavior indicates that these controllers rely heavily on corrective feedback effort.

The CTC controller produced a more balanced result. It improved the tracking of the first three joints and generated smooth, stable torque commands. However, the last three joints showed poor tracking performance compared with the first three joints, mainly because of wrist-joint sensitivity, control-loop delay, torque limits, and model mismatch. Therefore, the CTC approach provides the best torque quality, but its tracking performance is limited by the practical simulation and communication constraints.

Overall, the experiments show that controller evaluation must consider both tracking accuracy and control effort. A controller with acceptable tracking may still be undesirable if it requires aggressive saturated torques, while a model-based controller may produce smoother inputs but remain sensitive to real-time implementation limitations.

General Conclusion

This thesis addressed the complete problem of modeling, control design, and ROS-based implementation of the UR5 six-degree-of-freedom serial robotic manipulator. The work was organized into four chapters, each contributing a distinct layer to the overall system.

In Chapter 1, the theoretical foundations were established. The kinematic model of the UR5 was derived using the Denavit-Hartenberg convention, providing a systematic framework for computing the forward kinematics as a product of six homogeneous transformation matrices. The inverse kinematics problem was solved analytically using the URe-series six-DOF IPK solution, which employs the Tangent Half-Angle Substitution to convert the transcendental trigonometric equations into solvable polynomial form. This analytical approach guarantees real-time computation without the convergence issues associated with numerical iterative methods. The complete dynamic model was subsequently derived using the Euler-Lagrange formulation, yielding explicit closed-form expressions for the inertia matrix $\mathbf{M}(\mathbf{q})$, the Coriolis and centrifugal matrix $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$, and the gravity vector $\mathbf{G}(\mathbf{q})$.

In Chapters 2 and 3, the control system was designed and analyzed. The Computed Torque Control strategy was selected as the primary control law due to its theoretical properties of exact feedback linearization and global asymptotic stability [8, 7, 4]. The stability of the closed-loop system was rigorously established using Lyapunov's second method, leveraging the skew-symmetry property of $\dot{\mathbf{M}} - 2\mathbf{C}$ to prove asymptotic convergence of the tracking error to zero. Controller gains were selected using the critically damped pole placement criterion, with natural frequency $_n = 10\,\text{rad/s}$ providing the optimal balance between response speed and robustness. The fifth-degree polynomial trajectory planner was developed to generate smooth reference signals with zero boundary conditions on velocity and acceleration, preventing discontinuous torque commands that could excite mechanical resonances. The complete system was integrated within the ROS Noetic ecosystem, with the MATLAB/Simulink controller communicating with the Gazebo physics simulation through ROS topics.

In Chapter 4, the three control strategies—PD, PD+G, and CTC—were compared through Gazebo simulation experiments. The results showed that PD and PD+G can achieve relatively good tracking on the proximal joints, but their torque commands are aggressive, distorted, and frequently bounded by the saturation limits. In contrast, the CTC controller produced uniformly superior tracking performance across all six joints, including the wrist,

with smooth and stable torque inputs that avoid the persistent saturation observed under the feedback-only strategies. The remaining discrepancies under CTC are minor and attributable to practical implementation constraints such as the low effective control frequency, communication delay, and small model mismatch. These results demonstrate that accurate dynamic compensation is essential for achieving both high tracking accuracy and high-quality control-effort behavior, and that both metrics must be considered when evaluating robot controllers.

Appendix A

Complete Dynamic Model of the UR5e Robot

This appendix presents the complete dynamic model of the UR5e robot used to implement the Computed Torque Controller. The symbolic model was generated using the methodology described in [3], which provides a systematic approach for deriving the kinematic and dynamic equations of serial manipulators. The Denavit–Hartenberg (DH) parameters and dynamic properties, including mass, center of mass, and inertia tensors, were obtained from Universal Robots documentation and related UR kinematic references [10, 12].

The dynamic model is expressed in the standard form:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{A.1}$$

where $M(q) \in \mathbb{R}^{66}$ is the symmetric positive definite inertia matrix, $C(q,\dot{q}) \in \mathbb{R}^{66}$ is the Coriolis and centrifugal matrix, $G(q) \in \mathbb{R}^6$ is the gravity vector, and $\tau \in \mathbb{R}^6$ is the vector of joint torques.

Remark: For simplicity, all dynamic parameters with magnitudes smaller than the order of 10^{-4} have been neglected in the expressions reported in this appendix.

Table 1.1: Modified DH Parameters of the UR5e Robot

Joint i	a_{i-1} [m]	d_i [m]	α_{i-1} [rad]
1	0	0.1625	$\pi/2$
2	-0.425	0	0
3	-0.3922	0	0
4	0	0.1333	$\pi/2$
5	0	0.0997	$-\pi/2$
6	0	0.0996	0

1.1 Dynamic Parameters

The dynamic properties of each link, including mass, center of mass position, and inertia tensor, are provided in Table 1.2. These parameters are based on the official UR5e specifications.

Table 1.2: Dynamic Parameters of the UR5e Robot

Link	**Mass [kg]**	**Center of Mass [m]**		
		x	y	z
1	3.761	0	-0.02561	0.00193
2	8.058	0.2125	0	0.11336
3	2.846	0.15	0	0.0265
4	1.370	0	-0.0018	0.01634
5	1.300	0	0.0018	0.01634
6	0.365	0	0	-0.001159

The inertia tensors for each link (about the center of mass) are:

$$I_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$I_4 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I_5 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I_6 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0.0002 \end{bmatrix}$$

1.2 Inertia Matrix

The inertia matrix $M(q)$ is a symmetric 66 matrix whose elements are functions of the joint positions $q = [q_1, q_2, q_3, q_4, q_5, q_6]^T$:

$$M(q) = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & m_{16} \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} & m_{26} \\ m_{31} & m_{32} & m_{33} & m_{34} & m_{35} & m_{36} \\ m_{41} & m_{42} & m_{43} & m_{44} & m_{45} & m_{46} \\ m_{51} & m_{52} & m_{53} & m_{54} & m_{55} & m_{56} \\ m_{61} & m_{62} & m_{63} & m_{64} & m_{65} & m_{66} \end{bmatrix} \tag{A.2}$$

where

$$m_{11} = 0.50507\cos(q_2) - 0.54069\cos(q_3) - 0.3349\sin(q_3)$$

$$
\begin{aligned}
&- 0.58591\cos(q_2)\cos(q_3) - 0.36291\cos(q_2)\sin(q_3) \\
&- 0.36291\cos(q_3)\sin(q_2) + 0.17312\cos(q_3)\sin(q_4) \\
&+ 0.0018355\cos(q_5)\sin(q_3) + 0.58591\sin(q_2)\sin(q_3) \\
&- 0.032372\cos(q_4)^2 + 0.1876\cos(q_2+q_3)\sin(q_4) \\
&+ 0.001989\cos(q_2+q_3)\cos(q_4)\sin(q_5) \\
&+ 0.001989\cos(q_2)\cos(q_5)\sin(q_3) \\
&+ 0.001989\cos(q_3)\cos(q_5)\sin(q_2) \\
&+ 0.0018355\cos(q_3)\cos(q_4)\sin(q_5) \\
&+ 0.00064403\cos(q_4)\sin(q_4)\sin(q_5) + 4.4684, \qquad \text{(A.3)}
\end{aligned}
$$

$$
\begin{aligned}
m_{12} = {} & 0.25254\cos(q_2) - 0.54069\cos(q_3) - 0.3349\sin(q_3) \\
&- 0.29295\cos(q_2)\cos(q_3) - 0.18145\cos(q_2)\sin(q_3) \\
&- 0.18145\cos(q_3)\sin(q_2) + 0.17312\cos(q_3)\sin(q_4) \\
&+ 0.0018355\cos(q_5)\sin(q_3) + 0.29295\sin(q_2)\sin(q_3) \\
&- 0.032372\cos(q_4)^2 + 0.093801\cos(q_2+q_3)\sin(q_4) \\
&+ 0.0009945\cos(q_2+q_3)\cos(q_4)\sin(q_5) \\
&+ 0.0009945\cos(q_2)\cos(q_5)\sin(q_3) \\
&+ 0.0009945\cos(q_3)\cos(q_5)\sin(q_2) \\
&+ 0.0018355\cos(q_3)\cos(q_4)\sin(q_5) \\
&+ 0.00064403\cos(q_4)\sin(q_4)\sin(q_5) + 1.9416, \qquad \text{(A.4)}
\end{aligned}
$$

$$
\begin{aligned}
m_{13} = {} & -0.27034\cos(q_3) - 0.16745\sin(q_3) \\
&- 0.29295\cos(q_2)\cos(q_3) - 0.18145\cos(q_2)\sin(q_3) \\
&- 0.18145\cos(q_3)\sin(q_2) + 0.086562\cos(q_3)\sin(q_4) \\
&+ 0.00091775\cos(q_5)\sin(q_3) + 0.29295\sin(q_2)\sin(q_3) \\
&- 0.032372\cos(q_4)^2 - 0.093801\sin(q_2)\sin(q_3)\sin(q_4) \\
&+ 0.093801\cos(q_2)\cos(q_3)\sin(q_4) \\
&+ 0.0009945\cos(q_2)\cos(q_5)\sin(q_3) \\
&+ 0.0009945\cos(q_3)\cos(q_5)\sin(q_2) \\
&+ 0.00091775\cos(q_3)\cos(q_4)\sin(q_5) \\
&+ 0.00064403\cos(q_4)\sin(q_4)\sin(q_5) \\
&+ 0.0009945\cos(q_2)\cos(q_3)\cos(q_4)\sin(q_5) \\
&- 0.0009945\cos(q_4)\sin(q_2)\sin(q_3)\sin(q_5) + 0.34066, \qquad \text{(A.5)}
\end{aligned}
$$

$$
\begin{aligned}
m_{14} = {} & -0.029278\cos(q_4) + 0.086561\cos(q_4)\sin(q_3) \\
&+ 0.00031192\sin(q_4)\sin(q_5)
\end{aligned}
$$

$$
\begin{aligned}
&+ 0.093801 \sin(q_2 + q_3) \cos(q_4) \\
&- 0.00091775 \sin(q_3) \sin(q_4) \sin(q_5) \\
&- 0.0009945 \sin(q_2 + q_3) \sin(q_4) \sin(q_5), \qquad \text{(A.6)}
\end{aligned}
$$

$$
\begin{aligned}
m_{15} = {} & 0.0012279 \cos(q_4) - 0.00031192 \cos(q_4) \cos(q_5) \\
&+ 0.00091775 \cos(q_3) \sin(q_5) + 0.00032201 \sin(q_4) \sin(q_5) \\
&- 0.0009945 \sin(q_2) \sin(q_3) \sin(q_5) \\
&+ 0.0009945 \cos(q_2) \cos(q_3) \sin(q_5) \\
&+ 0.00091775 \cos(q_4) \cos(q_5) \sin(q_3) \\
&+ 0.0009945 \cos(q_2) \cos(q_4) \cos(q_5) \sin(q_3) \\
&+ 0.0009945 \cos(q_3) \cos(q_4) \cos(q_5) \sin(q_2), \qquad \text{(A.7)}
\end{aligned}
$$

$$
m_{16} = 0.00025756 \cos(q_4), \qquad \text{(A.8)}
$$

$$
\begin{aligned}
m_{21} = {} & 0.25254 \cos(q_2) - 0.54069 \cos(q_3) - 0.3349 \sin(q_3) \\
&- 0.29295 \cos(q_2) \cos(q_3) - 0.18145 \cos(q_2) \sin(q_3) \\
&- 0.18145 \cos(q_3) \sin(q_2) + 0.17312 \cos(q_3) \sin(q_4) \\
&+ 0.0018355 \cos(q_5) \sin(q_3) + 0.29295 \sin(q_2) \sin(q_3) \\
&- 0.032372 \cos(q_4)^2 + 0.093801 \cos(q_2 + q_3) \sin(q_4) \\
&+ 0.0009945 \cos(q_2 + q_3) \cos(q_4) \sin(q_5) \\
&+ 0.0009945 \cos(q_2) \cos(q_5) \sin(q_3) \\
&+ 0.0009945 \cos(q_3) \cos(q_5) \sin(q_2) \\
&+ 0.0018355 \cos(q_3) \cos(q_4) \sin(q_5) \\
&+ 0.00064403 \cos(q_4) \sin(q_4) \sin(q_5) + 1.9416, \qquad \text{(A.9)}
\end{aligned}
$$

$$
\begin{aligned}
m_{22} = {} & -0.016207 \cos(2q_4 - 0.000174) - 0.54069 \cos(q_3) \\
&- 0.3349 \sin(q_3) + 0.17312 \cos(q_3) \sin(q_4) \\
&+ 0.0018355 \cos(q_5) \sin(q_3) \\
&+ 0.00016102 \cos(2q_4 - 0.013787) \cos(q_5) \\
&- 0.00016102 \cos(2q_4 + 0.013787) \cos(q_5) \\
&+ 0.00016102 \sin(2q_4 - 0.013787) \sin(q_5) \\
&+ 0.00016102 \sin(2q_4 + 0.013787) \sin(q_5) \\
&+ 0.0018355 \cos(q_3) \cos(q_4) \sin(q_5) + 1.9255, \qquad \text{(A.10)}
\end{aligned}
$$

$$
\begin{aligned}
m_{23} = {} & -0.016207 \cos(2q_4 - 0.000174) - 0.27034 \cos(q_3) \\
&- 0.16745 \sin(q_3) + 0.086562 \cos(q_3) \sin(q_4) \\
&+ 0.00091775 \cos(q_5) \sin(q_3)
\end{aligned}
$$

$$
\begin{aligned}
&+ 0.00016102\cos(2q_4 - 0.013787)\cos(q_5) \\
&- 0.00016102\cos(2q_4 + 0.013787)\cos(q_5) \\
&+ 0.00016102\sin(2q_4 - 0.013787)\sin(q_5) \\
&+ 0.00016102\sin(2q_4 + 0.013787)\sin(q_5) \\
&+ 0.00091775\cos(q_3)\cos(q_4)\sin(q_5) + 0.32449, \qquad \text{(A.11)}
\end{aligned}
$$

$$
\begin{aligned}
m_{24} = &-0.029278\cos(q_4) + 0.086561\cos(q_4)\sin(q_3) \\
&+ 0.00031192\sin(q_4)\sin(q_5) \\
&- 0.00091775\sin(q_3)\sin(q_4)\sin(q_5), \qquad \text{(A.12)}
\end{aligned}
$$

$$
\begin{aligned}
m_{25} = &\, 0.0012279\cos(q_4) - 0.00031192\cos(q_4)\cos(q_5) \\
&+ 0.00091775\cos(q_3)\sin(q_5) + 0.00032201\sin(q_4)\sin(q_5) \\
&+ 0.00091775\cos(q_4)\cos(q_5)\sin(q_3), \qquad \text{(A.13)}
\end{aligned}
$$

$$
m_{26} = 0.00025756\cos(q_4), \qquad \text{(A.14)}
$$

$$
\begin{aligned}
m_{31} = &-0.27034\cos(q_3) - 0.16745\sin(q_3) \\
&- 0.29295\cos(q_2)\cos(q_3) - 0.18145\cos(q_2)\sin(q_3) \\
&- 0.18145\cos(q_3)\sin(q_2) + 0.086562\cos(q_3)\sin(q_4) \\
&+ 0.00091775\cos(q_5)\sin(q_3) + 0.29295\sin(q_2)\sin(q_3) \\
&- 0.032372\cos(q_4)^2 - 0.093801\sin(q_2)\sin(q_3)\sin(q_4) \\
&+ 0.093801\cos(q_2)\cos(q_3)\sin(q_4) \\
&+ 0.0009945\cos(q_2)\cos(q_5)\sin(q_3) \\
&+ 0.0009945\cos(q_3)\cos(q_5)\sin(q_2) \\
&+ 0.00091775\cos(q_3)\cos(q_4)\sin(q_5) \\
&+ 0.00064403\cos(q_4)\sin(q_4)\sin(q_5) \\
&+ 0.0009945\cos(q_2)\cos(q_3)\cos(q_4)\sin(q_5) \\
&- 0.0009945\cos(q_4)\sin(q_2)\sin(q_3)\sin(q_5) + 0.34066, \qquad \text{(A.15)}
\end{aligned}
$$

$$
\begin{aligned}
m_{32} = &-0.016207\cos(2q_4 - 0.000174) - 0.27034\cos(q_3) \\
&- 0.16745\sin(q_3) + 0.086562\cos(q_3)\sin(q_4) \\
&+ 0.00091775\cos(q_5)\sin(q_3) \\
&+ 0.00016102\cos(2q_4 - 0.013787)\cos(q_5) \\
&- 0.00016102\cos(2q_4 + 0.013787)\cos(q_5) \\
&+ 0.00016102\sin(2q_4 - 0.013787)\sin(q_5) \\
&+ 0.00016102\sin(2q_4 + 0.013787)\sin(q_5) \\
&+ 0.00091775\cos(q_3)\cos(q_4)\sin(q_5) + 0.32449, \qquad \text{(A.16)}
\end{aligned}
$$

$$
\begin{aligned}
m_{33} = &-0.016207\cos(2q_4 - 0.000174) \\
&+ 0.00016102\cos(q_5 - 2q_4 + 0.013787) \\
&- 0.00016102\cos(2q_4 + q_5 + 0.013787) + 0.32449,
\end{aligned} \tag{A.17}
$$

$$
\begin{aligned}
m_{34} = &-0.029278\cos(q_4 - 0.0001496) \\
&- 0.000020574\cos(q_4 - 2q_5 + 0.066639) \\
&+ 0.000020574\cos(q_4 + 2q_5 - 0.066639) \\
&+ 0.00031698\cos(q_5 - q_4 + 0.0070038),
\end{aligned} \tag{A.18}
$$

$$
m_{35} = -0.00031698\cos(q_4 + q_5 + 0.0070038) + 0.0012279\cos(q_4), \tag{A.19}
$$

$$
m_{36} = 0.00025756\cos(q_4), \tag{A.20}
$$

$$
\begin{aligned}
m_{41} = &-0.029278\cos(q_4) + 0.086561\cos(q_4)\sin(q_3) \\
&+ 0.00031192\sin(q_4)\sin(q_5) \\
&+ 0.093801\sin(q_2 + q_3)\cos(q_4) \\
&- 0.00091775\sin(q_3)\sin(q_4)\sin(q_5) \\
&- 0.0009945\sin(q_2 + q_3)\sin(q_4)\sin(q_5),
\end{aligned} \tag{A.21}
$$

$$
\begin{aligned}
m_{42} = &-0.029278\cos(q_4) + 0.086561\cos(q_4)\sin(q_3) \\
&+ 0.00031192\sin(q_4)\sin(q_5) \\
&- 0.00091775\sin(q_3)\sin(q_4)\sin(q_5),
\end{aligned} \tag{A.22}
$$

$$
\begin{aligned}
m_{43} = &-0.029278\cos(q_4 - 0.0001496) \\
&- 0.000020574\cos(q_4 - 2q_5 + 0.066639) \\
&+ 0.000020574\cos(q_4 + 2q_5 - 0.066639) \\
&+ 0.00031698\cos(q_5 - q_4 + 0.0070038),
\end{aligned} \tag{A.23}
$$

$$
m_{44} = -0.000041147\cos(2q_5 - 0.066639) + 0.034609, \tag{A.24}
$$

$$
m_{45} = 0.00032204\cos(q_5 + 0.013787), \tag{A.25}
$$

$$
m_{46} = 0, \tag{A.26}
$$

$$
\begin{aligned}
m_{51} = &\ 0.0012279\cos(q_4) - 0.00031192\cos(q_4)\cos(q_5) \\
&+ 0.00091775\cos(q_3)\sin(q_5) + 0.00032201\sin(q_4)\sin(q_5) \\
&- 0.0009945\sin(q_2)\sin(q_3)\sin(q_5) \\
&+ 0.0009945\cos(q_2)\cos(q_3)\sin(q_5) \\
&+ 0.00091775\cos(q_4)\cos(q_5)\sin(q_3) \\
&+ 0.0009945\cos(q_2)\cos(q_4)\cos(q_5)\sin(q_3) \\
&+ 0.0009945\cos(q_3)\cos(q_4)\cos(q_5)\sin(q_2),
\end{aligned} \tag{A.27}
$$

$$
\begin{aligned}
m_{52} = {} & 0.0012279\cos(q_4) - 0.00031192\cos(q_4)\cos(q_5) \\
& + 0.00091775\cos(q_3)\sin(q_5) + 0.00032201\sin(q_4)\sin(q_5) \\
& + 0.00091775\cos(q_4)\cos(q_5)\sin(q_3),
\end{aligned} \tag{A.28}
$$

$$
m_{53} = -0.00031698\cos(q_4 + q_5 + 0.0070038) + 0.0012279\cos(q_4), \tag{A.29}
$$

$$
m_{54} = 0.00032204\cos(q_5 + 0.013787), \tag{A.30}
$$

$$
m_{55} = 0.0012279, \tag{A.31}
$$

$$
m_{56} = 0.00025756, \tag{A.32}
$$

$$
m_{61} = 0.00025756\cos(q_4), \tag{A.33}
$$

$$
m_{62} = 0.00025756\cos(q_4), \tag{A.34}
$$

$$
m_{63} = 0.00025756\cos(q_4), \tag{A.35}
$$

$$
m_{64} = 0, \tag{A.36}
$$

$$
m_{65} = 0.00025756, \tag{A.37}
$$

$$
m_{66} = 0.00025756. \tag{A.38}
$$

1.3 Coriolis and Centrifugal Matrix

The Coriolis and centrifugal matrix $C(q, \dot{q})$ is computed using the Christoffel symbols of the first kind:

$$
c_{ij} = \sum_{k=1}^{6} \frac{1}{2} \left(\frac{\partial m_{ij}}{\partial q_k} + \frac{\partial m_{ik}}{\partial q_j} - \frac{\partial m_{kj}}{\partial q_i} \right) \dot{q}_k \tag{A.39}
$$

Due to the length of the resulting expressions, the complete matrix $C(q, \dot{q})$ is omitted here but follows the same structure as $M(q)$ with elements that are functions of both q and $\dot{q}$. The matrix is given by

where

$$
C(q, \dot{q}) = \begin{bmatrix}
c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{16} \\
c_{21} & c_{22} & c_{23} & c_{24} & c_{25} & c_{26} \\
c_{31} & c_{32} & c_{33} & c_{34} & c_{35} & c_{36} \\
c_{41} & c_{42} & c_{43} & c_{44} & c_{45} & c_{46} \\
c_{51} & c_{52} & c_{53} & c_{54} & c_{55} & c_{56} \\
c_{61} & c_{62} & c_{63} & c_{64} & c_{65} & c_{66}
\end{bmatrix} \tag{A.40}
$$

$$
\begin{aligned}
c_{11} = {} & -0.16745\dot{q}_3\cos(q_3) - 0.25254\dot{q}_2\sin(q_2) + 0.27034\dot{q}_3\sin(q_3) \\
& - 0.18145\dot{q}_2\cos(q_2)\cos(q_3) - 0.18145\dot{q}_3\cos(q_2)\cos(q_3) \\
& + 0.00091775\dot{q}_3\cos(q_3)\cos(q_5) + 0.086562\dot{q}_4\cos(q_3)\cos(q_4)
\end{aligned}
$$

$$
\begin{aligned}
&+ 0.29295\dot{q}_2 \cos(q_2)\sin(q_3) + 0.29295\dot{q}_2 \cos(q_3)\sin(q_2) \\
&+ 0.29295\dot{q}_3 \cos(q_2)\sin(q_3) + 0.29295\dot{q}_3 \cos(q_3)\sin(q_2) \\
&+ 0.18145\dot{q}_2 \sin(q_2)\sin(q_3) + 0.18145\dot{q}_3 \sin(q_2)\sin(q_3) \\
&- 0.086562\dot{q}_3 \sin(q_3)\sin(q_4) \\
&+ 0.00064403\dot{q}_4 \cos(q_4)^2 \sin(q_5) \\
&- 0.093801\dot{q}_2 \cos(q_2)\sin(q_3)\sin(q_4) \\
&- 0.093801\dot{q}_2 \cos(q_3)\sin(q_2)\sin(q_4) \\
&- 0.093801\dot{q}_3 \cos(q_2)\sin(q_3)\sin(q_4) \\
&- 0.093801\dot{q}_3 \cos(q_3)\sin(q_2)\sin(q_4) \\
&+ 0.0009945\dot{q}_5 \cos(q_2 + q_3)\cos(q_4)\cos(q_5) \\
&- 0.0009945\dot{q}_4 \cos(q_2 + q_3)\sin(q_4)\sin(q_5) \\
&+ 0.0009945\dot{q}_2 \cos(q_2)\cos(q_3)\cos(q_5) \\
&+ 0.0009945\dot{q}_3 \cos(q_2)\cos(q_3)\cos(q_5) \\
&+ 0.00091775\dot{q}_5 \cos(q_3)\cos(q_4)\cos(q_5) \\
&- 0.00091775\dot{q}_3 \cos(q_4)\sin(q_3)\sin(q_5) \\
&- 0.00091775\dot{q}_4 \cos(q_3)\sin(q_4)\sin(q_5), \qquad \text{(A.41)}
\end{aligned}
$$

$$
\begin{aligned}
c_{12} = &-0.16745\dot{q}_3 \cos(q_3) - 0.25254\dot{q}_1 \sin(q_2) - 0.25254\dot{q}_2 \sin(q_2) \\
&+ 0.27034\dot{q}_3 \sin(q_3) + 0.093801\dot{q}_4 \cos(q_2 + q_3)\cos(q_4) \\
&- 0.18145\dot{q}_1 \cos(q_2)\cos(q_3) - 0.18145\dot{q}_2 \cos(q_2)\cos(q_3) \\
&- 0.18145\dot{q}_3 \cos(q_2)\cos(q_3) \\
&+ 0.00091775\dot{q}_3 \cos(q_3)\cos(q_5) + 0.086562\dot{q}_4 \cos(q_3)\cos(q_4) \\
&+ 0.29295\dot{q}_1 \cos(q_2)\sin(q_3) + 0.29295\dot{q}_1 \cos(q_3)\sin(q_2) \\
&+ 0.29295\dot{q}_2 \cos(q_2)\sin(q_3) + 0.29295\dot{q}_2 \cos(q_3)\sin(q_2) \\
&+ 0.29295\dot{q}_3 \cos(q_2)\sin(q_3) + 0.29295\dot{q}_3 \cos(q_3)\sin(q_2) \\
&+ 0.18145\dot{q}_1 \sin(q_2)\sin(q_3) + 0.18145\dot{q}_2 \sin(q_2)\sin(q_3) \\
&+ 0.18145\dot{q}_3 \sin(q_2)\sin(q_3) - 0.086562\dot{q}_3 \sin(q_3)\sin(q_4) \\
&+ 0.00064403\dot{q}_4 \cos(q_4)^2 \sin(q_5) \\
&+ 0.0009945\dot{q}_5 \cos(q_2 + q_3)\cos(q_4)\cos(q_5) \\
&- 0.0009945\dot{q}_4 \cos(q_2 + q_3)\sin(q_4)\sin(q_5) \\
&+ 0.0009945\dot{q}_1 \cos(q_2)\cos(q_3)\cos(q_5) \\
&+ 0.0009945\dot{q}_2 \cos(q_2)\cos(q_3)\cos(q_5) \\
&+ 0.0009945\dot{q}_3 \cos(q_2)\cos(q_3)\cos(q_5) \\
&+ 0.00091775\dot{q}_5 \cos(q_3)\cos(q_4)\cos(q_5)
\end{aligned}
$$

$$
\begin{aligned}
&- 0.093801\dot{q}_1 \cos(q_2) \sin(q_3) \sin(q_4) \\
&- 0.093801\dot{q}_1 \cos(q_3) \sin(q_2) \sin(q_4) \\
&- 0.093801\dot{q}_2 \cos(q_2) \sin(q_3) \sin(q_4) \\
&- 0.093801\dot{q}_2 \cos(q_3) \sin(q_2) \sin(q_4) \\
&- 0.093801\dot{q}_3 \cos(q_2) \sin(q_3) \sin(q_4) \\
&- 0.093801\dot{q}_3 \cos(q_3) \sin(q_2) \sin(q_4) \\
&- 0.00091775\dot{q}_3 \cos(q_4) \sin(q_3) \sin(q_5) \\
&- 0.00091775\dot{q}_4 \cos(q_3) \sin(q_4) \sin(q_5), \qquad \text{(A.42)}
\end{aligned}
$$

$$
\begin{aligned}
c_{13} = &-0.16745\dot{q}_1 \cos(q_3) - 0.16745\dot{q}_2 \cos(q_3) - 0.16745\dot{q}_3 \cos(q_3) \\
&+ 0.27034\dot{q}_1 \sin(q_3) + 0.27034\dot{q}_2 \sin(q_3) + 0.27034\dot{q}_3 \sin(q_3) \\
&+ 0.093801\dot{q}_4 \cos(q_2 + q_3) \cos(q_4) \\
&- 0.093801\dot{q}_1 \sin(q_2 + q_3) \sin(q_4) \\
&- 0.093801\dot{q}_2 \sin(q_2 + q_3) \sin(q_4) \\
&- 0.093801\dot{q}_3 \sin(q_2 + q_3) \sin(q_4) \\
&- 0.18145\dot{q}_1 \cos(q_2) \cos(q_3) - 0.18145\dot{q}_2 \cos(q_2) \cos(q_3) \\
&- 0.18145\dot{q}_3 \cos(q_2) \cos(q_3) \\
&+ 0.00091775\dot{q}_1 \cos(q_3) \cos(q_5) \\
&+ 0.00091775\dot{q}_2 \cos(q_3) \cos(q_5) \\
&+ 0.00091775\dot{q}_3 \cos(q_3) \cos(q_5) \\
&+ 0.086562\dot{q}_4 \cos(q_3) \cos(q_4) \\
&+ 0.29295\dot{q}_1 \cos(q_2) \sin(q_3) + 0.29295\dot{q}_1 \cos(q_3) \sin(q_2) \\
&+ 0.29295\dot{q}_2 \cos(q_2) \sin(q_3) + 0.29295\dot{q}_2 \cos(q_3) \sin(q_2) \\
&+ 0.29295\dot{q}_3 \cos(q_2) \sin(q_3) + 0.29295\dot{q}_3 \cos(q_3) \sin(q_2) \\
&+ 0.18145\dot{q}_1 \sin(q_2) \sin(q_3) + 0.18145\dot{q}_2 \sin(q_2) \sin(q_3) \\
&+ 0.18145\dot{q}_3 \sin(q_2) \sin(q_3) \\
&- 0.086562\dot{q}_1 \sin(q_3) \sin(q_4) \\
&- 0.086562\dot{q}_2 \sin(q_3) \sin(q_4) \\
&- 0.086562\dot{q}_3 \sin(q_3) \sin(q_4) \\
&+ 0.00032201\dot{q}_4 \cos(q_4)^2 \sin(q_5) \\
&+ 0.0009945\dot{q}_5 \cos(q_2 + q_3) \cos(q_4) \cos(q_5) \\
&- 0.0009945\dot{q}_4 \cos(q_2 + q_3) \sin(q_4) \sin(q_5) \\
&+ 0.0009945\dot{q}_1 \cos(q_2) \cos(q_3) \cos(q_5) \\
&+ 0.0009945\dot{q}_2 \cos(q_2) \cos(q_3) \cos(q_5)
\end{aligned}
$$

$$
\begin{aligned}
&+ 0.0009945\dot{q}_3 \cos(q_2)\cos(q_3)\cos(q_5) \\
&+ 0.00091775\dot{q}_5 \cos(q_3)\cos(q_4)\cos(q_5) \\
&- 0.00091775\dot{q}_1 \cos(q_4)\sin(q_3)\sin(q_5) \\
&- 0.00091775\dot{q}_2 \cos(q_4)\sin(q_3)\sin(q_5) \\
&- 0.00091775\dot{q}_3 \cos(q_4)\sin(q_3)\sin(q_5) \\
&- 0.00091775\dot{q}_4 \cos(q_3)\sin(q_4)\sin(q_5), \qquad \text{(A.43)}
\end{aligned}
$$

$$
\begin{aligned}
c_{14} = {} & 0.029278\dot{q}_4 \sin(q_4) - 0.00057183\dot{q}_5 \sin(q_4) \\
&+ 0.086562\dot{q}_1 \cos(q_3)\cos(q_4) \\
&+ 0.086562\dot{q}_2 \cos(q_3)\cos(q_4) \\
&+ 0.086562\dot{q}_3 \cos(q_3)\cos(q_4) \\
&+ 0.00031192\dot{q}_4 \cos(q_4)\sin(q_5) \\
&- 0.00032201\dot{q}_4 \cos(q_5)\sin(q_4) \\
&+ 0.00031192\dot{q}_5 \cos(q_5)\sin(q_4) \\
&- 0.086562\dot{q}_4 \sin(q_3)\sin(q_4) \\
&+ 0.00064403\dot{q}_1 \cos(q_4)^2 \sin(q_5) \\
&+ 0.00064403\dot{q}_2 \cos(q_4)^2 \sin(q_5) \\
&+ 0.00064403\dot{q}_3 \cos(q_4)^2 \sin(q_5) \\
&+ 0.093801\dot{q}_1 \cos(q_2)\cos(q_3)\cos(q_4) \\
&+ 0.093801\dot{q}_2 \cos(q_2)\cos(q_3)\cos(q_4) \\
&+ 0.093801\dot{q}_3 \cos(q_2)\cos(q_3)\cos(q_4) \\
&- 0.093801\dot{q}_1 \cos(q_4)\sin(q_2)\sin(q_3) \\
&- 0.093801\dot{q}_2 \cos(q_4)\sin(q_2)\sin(q_3) \\
&- 0.093801\dot{q}_3 \cos(q_4)\sin(q_2)\sin(q_3) \\
&- 0.00091775\dot{q}_1 \cos(q_3)\sin(q_4)\sin(q_5) \\
&- 0.00091775\dot{q}_2 \cos(q_3)\sin(q_4)\sin(q_5) \\
&- 0.00091775\dot{q}_3 \cos(q_3)\sin(q_4)\sin(q_5) \\
&- 0.00091775\dot{q}_4 \cos(q_4)\sin(q_3)\sin(q_5) \\
&- 0.00091775\dot{q}_5 \cos(q_5)\sin(q_3)\sin(q_4) \\
&- 0.0009945\dot{q}_1 \cos(q_2)\cos(q_3)\sin(q_4)\sin(q_5) \\
&- 0.0009945\dot{q}_2 \cos(q_2)\cos(q_3)\sin(q_4)\sin(q_5) \\
&- 0.0009945\dot{q}_3 \cos(q_2)\cos(q_3)\sin(q_4)\sin(q_5) \\
&- 0.0009945\dot{q}_4 \cos(q_2)\cos(q_4)\sin(q_3)\sin(q_5) \\
&- 0.0009945\dot{q}_4 \cos(q_3)\cos(q_4)\sin(q_2)\sin(q_5)
\end{aligned}
$$

$$
\begin{aligned}
&- 0.0009945\dot{q}_5 \cos(q_2)\cos(q_5)\sin(q_3)\sin(q_4) \\
&- 0.0009945\dot{q}_5 \cos(q_3)\cos(q_5)\sin(q_2)\sin(q_4),
\end{aligned} \tag{A.44}
$$

$$
\begin{aligned}
c_{15} = &-0.00057183\dot{q}_4 \sin(q_4) \\
&+ 0.00091775\dot{q}_5 \cos(q_3)\cos(q_5) \\
&- 0.00091775\dot{q}_1 \sin(q_3)\sin(q_5) \\
&- 0.00091775\dot{q}_2 \sin(q_3)\sin(q_5) \\
&- 0.00091775\dot{q}_3 \sin(q_3)\sin(q_5) \\
&+ 0.00091775\dot{q}_1 \cos(q_3)\cos(q_4)\cos(q_5) \\
&+ 0.00091775\dot{q}_2 \cos(q_3)\cos(q_4)\cos(q_5) \\
&+ 0.00091775\dot{q}_3 \cos(q_3)\cos(q_4)\cos(q_5) \\
&+ 0.0009945\dot{q}_1 \cos(q_2+q_3)\cos(q_4)\cos(q_5) \\
&+ 0.0009945\dot{q}_2 \cos(q_2+q_3)\cos(q_4)\cos(q_5) \\
&+ 0.0009945\dot{q}_3 \cos(q_2+q_3)\cos(q_4)\cos(q_5) \\
&- 0.0009945\dot{q}_4 \sin(q_2+q_3)\cos(q_5)\sin(q_4) \\
&+ 0.0009945\dot{q}_5 \cos(q_2)\cos(q_3)\cos(q_5) \\
&- 0.0009945\dot{q}_1 \cos(q_2)\sin(q_3)\sin(q_5) \\
&- 0.0009945\dot{q}_1 \cos(q_3)\sin(q_2)\sin(q_5) \\
&- 0.0009945\dot{q}_2 \cos(q_2)\sin(q_3)\sin(q_5) \\
&- 0.0009945\dot{q}_2 \cos(q_3)\sin(q_2)\sin(q_5) \\
&- 0.0009945\dot{q}_3 \cos(q_2)\sin(q_3)\sin(q_5) \\
&- 0.0009945\dot{q}_3 \cos(q_3)\sin(q_2)\sin(q_5) \\
&- 0.0009945\dot{q}_5 \cos(q_5)\sin(q_2)\sin(q_3) \\
&- 0.00091775\dot{q}_4 \cos(q_5)\sin(q_3)\sin(q_4) \\
&- 0.00091775\dot{q}_5 \cos(q_4)\sin(q_3)\sin(q_5),
\end{aligned} \tag{A.45}
$$

$$
c_{16} = 0, \tag{A.46}
$$

$$
\begin{aligned}
c_{21} = &0.25254\dot{q}_1 \sin(q_2) - 0.16745\dot{q}_3 \cos(q_3) + 0.27034\dot{q}_3 \sin(q_3) \\
&+ 0.18145\dot{q}_1 \cos(q_2)\cos(q_3) + 0.00091775\dot{q}_3 \cos(q_3)\cos(q_5) \\
&+ 0.086562\dot{q}_4 \cos(q_3)\cos(q_4) \\
&- 0.29295\dot{q}_1 \cos(q_2)\sin(q_3) - 0.29295\dot{q}_1 \cos(q_3)\sin(q_2) \\
&- 0.18145\dot{q}_1 \sin(q_2)\sin(q_3) - 0.086562\dot{q}_3 \sin(q_3)\sin(q_4) \\
&+ 0.00064403\dot{q}_4 \cos(q_4)^2 \sin(q_5) \\
&+ 0.0009945\dot{q}_1 \sin(q_2+q_3)\cos(q_4)\sin(q_5) \\
&- 0.0009945\dot{q}_1 \cos(q_2)\cos(q_3)\cos(q_5)
\end{aligned}
$$

$$
\begin{aligned}
&+ 0.00091775\dot{q}_5 \cos(q_3)\cos(q_4)\cos(q_5) \\
&+ 0.0009945\dot{q}_1 \cos(q_5)\sin(q_2)\sin(q_3) \\
&- 0.00091775\dot{q}_3 \cos(q_4)\sin(q_3)\sin(q_5) \\
&- 0.00091775\dot{q}_4 \cos(q_3)\sin(q_4)\sin(q_5), \qquad \text{(A.47)}
\end{aligned}
$$

$$
\begin{aligned}
c_{22} = &-0.16745\dot{q}_3 \cos(q_3) + 0.27034\dot{q}_3 \sin(q_3) \\
&+ 0.00091775\dot{q}_3 \cos(q_3)\cos(q_5) + 0.086562\dot{q}_4 \cos(q_3)\cos(q_4) \\
&- 0.086562\dot{q}_3 \sin(q_3)\sin(q_4) \\
&+ 0.00091775\dot{q}_5 \cos(q_3)\cos(q_4)\cos(q_5) \\
&- 0.00091775\dot{q}_3 \cos(q_4)\sin(q_3)\sin(q_5) \\
&- 0.00091775\dot{q}_4 \cos(q_3)\sin(q_4)\sin(q_5), \qquad \text{(A.48)}
\end{aligned}
$$

$$
\begin{aligned}
c_{23} = &-0.16745\dot{q}_1 \cos(q_3) - 0.16745\dot{q}_2 \cos(q_3) - 0.16745\dot{q}_3 \cos(q_3) \\
&+ 0.27034\dot{q}_1 \sin(q_3) + 0.27034\dot{q}_2 \sin(q_3) + 0.27034\dot{q}_3 \sin(q_3) \\
&+ 0.00091775\dot{q}_1 \cos(q_3)\cos(q_5) \\
&+ 0.00091775\dot{q}_2 \cos(q_3)\cos(q_5) \\
&+ 0.00091775\dot{q}_3 \cos(q_3)\cos(q_5) \\
&+ 0.086562\dot{q}_4 \cos(q_3)\cos(q_4) \\
&- 0.086562\dot{q}_1 \sin(q_3)\sin(q_4) \\
&- 0.086562\dot{q}_2 \sin(q_3)\sin(q_4) \\
&- 0.086562\dot{q}_3 \sin(q_3)\sin(q_4) \\
&+ 0.00064403\dot{q}_4 \cos(q_4)^2 \sin(q_5) \\
&+ 0.00091775\dot{q}_5 \cos(q_3)\cos(q_4)\cos(q_5) \\
&- 0.00091775\dot{q}_1 \cos(q_4)\sin(q_3)\sin(q_5) \\
&- 0.00091775\dot{q}_2 \cos(q_4)\sin(q_3)\sin(q_5) \\
&- 0.00091775\dot{q}_3 \cos(q_4)\sin(q_3)\sin(q_5) \\
&- 0.00091775\dot{q}_4 \cos(q_3)\sin(q_4)\sin(q_5), \qquad \text{(A.49)}
\end{aligned}
$$

$$
\begin{aligned}
c_{24} = &-0.00057183\dot{q}_5 \sin(q_4) + 0.029278\dot{q}_4 \sin(q_4) \\
&+ 0.086562\dot{q}_1 \cos(q_3)\cos(q_4) \\
&+ 0.086562\dot{q}_2 \cos(q_3)\cos(q_4) \\
&+ 0.00031192\dot{q}_4 \cos(q_4)\sin(q_5) \\
&- 0.00032201\dot{q}_4 \cos(q_5)\sin(q_4) \\
&+ 0.00031192\dot{q}_5 \cos(q_5)\sin(q_4) \\
&- 0.086562\dot{q}_4 \sin(q_3)\sin(q_4) \\
&+ 0.00064403\dot{q}_1 \cos(q_4)^2 \sin(q_5)
\end{aligned}
$$

$$
\begin{aligned}
&+ 0.00064403\dot{q}_2 \cos(q_4)^2 \sin(q_5) \\
&- 0.00091775\dot{q}_1 \cos(q_3) \sin(q_4) \sin(q_5) \\
&- 0.00091775\dot{q}_2 \cos(q_3) \sin(q_4) \sin(q_5) \\
&- 0.00091775\dot{q}_4 \cos(q_4) \sin(q_3) \sin(q_5) \\
&- 0.00091775\dot{q}_5 \cos(q_5) \sin(q_3) \sin(q_4), \qquad \text{(A.50)}
\end{aligned}
$$

$$
\begin{aligned}
c_{25} = &-0.00057183\dot{q}_4 \sin(q_4) \\
&+ 0.00091775\dot{q}_5 \cos(q_3) \cos(q_5) \\
&- 0.00091775\dot{q}_1 \sin(q_3) \sin(q_5) \\
&- 0.00091775\dot{q}_2 \sin(q_3) \sin(q_5) \\
&+ 0.00091775\dot{q}_1 \cos(q_3) \cos(q_4) \cos(q_5) \\
&+ 0.00091775\dot{q}_2 \cos(q_3) \cos(q_4) \cos(q_5) \\
&- 0.00091775\dot{q}_4 \cos(q_5) \sin(q_3) \sin(q_4) \\
&- 0.00091775\dot{q}_5 \cos(q_4) \sin(q_3) \sin(q_5), \qquad \text{(A.51)}
\end{aligned}
$$

$$
c_{26} = 0, \qquad \text{(A.52)}
$$

$$
\begin{aligned}
c_{31} = &\, 0.16745\dot{q}_1 \cos(q_3) + 0.16745\dot{q}_2 \cos(q_3) \\
&- 0.27034\dot{q}_1 \sin(q_3) - 0.27034\dot{q}_2 \sin(q_3) \\
&+ 0.18145\dot{q}_1 \cos(q_2) \cos(q_3) \\
&- 0.00091775\dot{q}_1 \cos(q_3) \cos(q_5) \\
&- 0.00091775\dot{q}_2 \cos(q_3) \cos(q_5) \\
&- 0.29295\dot{q}_1 \cos(q_2) \sin(q_3) \\
&- 0.29295\dot{q}_1 \cos(q_3) \sin(q_2) \\
&- 0.18145\dot{q}_1 \sin(q_2) \sin(q_3) \\
&+ 0.086562\dot{q}_1 \sin(q_3) \sin(q_4) \\
&+ 0.086562\dot{q}_2 \sin(q_3) \sin(q_4) \\
&+ 0.093801\dot{q}_1 \cos(q_2) \sin(q_3) \sin(q_4) \\
&+ 0.093801\dot{q}_1 \cos(q_3) \sin(q_2) \sin(q_4) \\
&+ 0.0009945\dot{q}_1 \cos(q_5) \sin(q_2) \sin(q_3) \\
&+ 0.00091775\dot{q}_1 \cos(q_4) \sin(q_3) \sin(q_5) \\
&+ 0.00091775\dot{q}_2 \cos(q_4) \sin(q_3) \sin(q_5) \\
&- 0.0009945\dot{q}_1 \cos(q_2) \cos(q_3) \cos(q_5), \qquad \text{(A.53)}
\end{aligned}
$$

$$
\begin{aligned}
c_{32} = &\, 0.16745\dot{q}_1 \cos(q_3) + 0.16745\dot{q}_2 \cos(q_3) \\
&- 0.27034\dot{q}_1 \sin(q_3) - 0.27034\dot{q}_2 \sin(q_3) \\
&- 0.00091775\dot{q}_1 \cos(q_3) \cos(q_5)
\end{aligned}
$$

$$
\begin{aligned}
&- 0.00091775\dot{q}_2 \cos(q_3)\cos(q_5) \\
&+ 0.086561\dot{q}_1 \sin(q_3)\sin(q_4) \\
&+ 0.086561\dot{q}_2 \sin(q_3)\sin(q_4) \\
&+ 0.00091775\dot{q}_1 \cos(q_4)\sin(q_3)\sin(q_5) \\
&+ 0.00091775\dot{q}_2 \cos(q_4)\sin(q_3)\sin(q_5),
\end{aligned} \tag{A.54}
$$

$$
c_{33} = 0, \tag{A.55}
$$

$$
\begin{aligned}
c_{34} = &-0.00061396\dot{q}_5 \sin(q_4) + 0.029278\dot{q}_4 \sin(q_4) \\
&+ 0.00031192\dot{q}_4 \cos(q_4)\sin(q_5) \\
&- 0.00032201\dot{q}_4 \cos(q_5)\sin(q_4) \\
&+ 0.00031192\dot{q}_5 \cos(q_5)\sin(q_4),
\end{aligned} \tag{A.56}
$$

$$
\begin{aligned}
c_{35} = &-0.00061396\dot{q}_4 \sin(q_4) \\
&+ 0.00031192\dot{q}_1 \sin(q_5) + 0.00031192\dot{q}_2 \sin(q_5) + 0.00031192\dot{q}_3 \sin(q_5) \\
&+ 0.00031192\dot{q}_4 \cos(q_5)\sin(q_4) \\
&+ 0.00031192\dot{q}_5 \cos(q_4)\sin(q_5) \\
&+ 0.00032201\dot{q}_5 \cos(q_5)\sin(q_4),
\end{aligned} \tag{A.57}
$$

$$
c_{36} = 0, \tag{A.58}
$$

$$
\begin{aligned}
c_{41} = &-0.016207\dot{q}_1 \sin(2q_4) - 0.016207\dot{q}_2 \sin(2q_4) - 0.016207\dot{q}_3 \sin(2q_4) \\
&- 0.093801\dot{q}_1 \cos(q_2 + q_3)\cos(q_4) \\
&- 0.086562\dot{q}_1 \cos(q_3)\cos(q_4) \\
&- 0.086562\dot{q}_2 \cos(q_3)\cos(q_4) \\
&+ 0.00091775\dot{q}_1 \cos(q_3)\sin(q_4)\sin(q_5) \\
&+ 0.00091775\dot{q}_2 \cos(q_3)\sin(q_4)\sin(q_5) \\
&+ 0.0009945\dot{q}_1 \cos(q_2 + q_3)\sin(q_4)\sin(q_5),
\end{aligned} \tag{A.59}
$$

$$
\begin{aligned}
c_{42} = &-0.016186\dot{q}_1 \sin(2q_4) - 0.016186\dot{q}_2 \sin(2q_4) - 0.016186\dot{q}_3 \sin(2q_4) \\
&- 0.086562\dot{q}_1 \cos(q_3)\cos(q_4) \\
&- 0.086562\dot{q}_2 \cos(q_3)\cos(q_4) \\
&+ 0.00091775\dot{q}_1 \cos(q_3)\sin(q_4)\sin(q_5) \\
&+ 0.00091775\dot{q}_2 \cos(q_3)\sin(q_4)\sin(q_5),
\end{aligned} \tag{A.60}
$$

$$
c_{43} = -0.016207\dot{q}_1 \sin(2q_4) - 0.016207\dot{q}_2 \sin(2q_4) - 0.016207\dot{q}_3 \sin(2q_4), \tag{A.61}
$$

$$
c_{44} = 0, \tag{A.62}
$$

$$
\begin{aligned}
c_{45} = &-0.00032201\dot{q}_5 \sin(q_5) - 0.00032201\dot{q}_1 \cos(q_4)\sin(q_5) \\
&- 0.00032201\dot{q}_2 \cos(q_4)\sin(q_5)
\end{aligned}
$$

$$- 0.00032201\dot{q}_3 \cos(q_4) \sin(q_5), \tag{A.63}$$

$$c_{46} = 0, \tag{A.64}$$

$$\begin{aligned} c_{51} = {} & 0.00091775\dot{q}_1 \sin(q_3) \sin(q_5) \\ & + 0.00091775\dot{q}_2 \sin(q_3) \sin(q_5) \\ & - 0.00091775\dot{q}_1 \cos(q_3) \cos(q_4) \cos(q_5) \\ & - 0.00091775\dot{q}_2 \cos(q_3) \cos(q_4) \cos(q_5) \\ & - 0.00032201\dot{q}_1 \cos(q_4) \cos(q_5) \sin(q_4) \\ & - 0.00032201\dot{q}_2 \cos(q_4) \cos(q_5) \sin(q_4) \\ & - 0.00032201\dot{q}_3 \cos(q_4) \cos(q_5) \sin(q_4) \\ & + 0.0009945\dot{q}_1 \cos(q_2) \sin(q_3) \sin(q_5) \\ & + 0.0009945\dot{q}_1 \cos(q_3) \sin(q_2) \sin(q_5) \\ & - 0.0009945\dot{q}_1 \cos(q_2) \cos(q_3) \cos(q_4) \cos(q_5) \\ & + 0.0009945\dot{q}_1 \cos(q_4) \cos(q_5) \sin(q_2) \sin(q_3), \end{aligned} \tag{A.65}$$

$$\begin{aligned} c_{52} = {} & -0.00091775\dot{q}_1 \cos(q_3) \cos(q_4) \cos(q_5) \\ & - 0.00091775\dot{q}_2 \cos(q_3) \cos(q_4) \cos(q_5), \end{aligned} \tag{A.66}$$

$$\begin{aligned} c_{53} = {} & -0.00091775\dot{q}_1 \cos(q_3) \cos(q_4) \cos(q_5) \\ & - 0.00091775\dot{q}_2 \cos(q_3) \cos(q_4) \cos(q_5), \end{aligned} \tag{A.67}$$

$$\begin{aligned} c_{54} = {} & -0.00032201\dot{q}_1 \cos(q_4) \sin(q_5) \\ & - 0.00032201\dot{q}_2 \cos(q_4) \sin(q_5) \\ & - 0.00032201\dot{q}_3 \cos(q_4) \sin(q_5), \end{aligned} \tag{A.68}$$

$$c_{55} = 0, \tag{A.69}$$

$$c_{56} = 0, \tag{A.70}$$

$$c_{61} = -0.00012878\dot{q}_4 \sin(q_4), \tag{A.71}$$

$$c_{62} = -0.00012878\dot{q}_4 \sin(q_4), \tag{A.72}$$

$$c_{63} = -0.00012878\dot{q}_4 \sin(q_4), \tag{A.73}$$

$$c_{64} = -0.00012878\dot{q}_1 \sin(q_4) - 0.00012878\dot{q}_2 \sin(q_4) - 0.00012878\dot{q}_3 \sin(q_4), \tag{A.74}$$

$$c_{65} = 0, \tag{A.75}$$

$$c_{66} = 0. \tag{A.76}$$

1.4 Gravity Vector

The gravity vector $G(q)$ is derived from the potential energy of the system:

$$G_i(q) = \frac{\partial \mathcal{P}}{\partial q_i}, \qquad \mathcal{P} = \sum_{i=1}^{6} m_i g^T {}^0p_{C_i} \tag{A.77}$$

where $g = [0, 0, -9.81]^T$ m/s^2 is the gravity vector and ${}^0p_{C_i}$ is the position of the center of mass of link i expressed in the base frame. The vector is expressed as

$$G(q) = \begin{bmatrix} g_1 1 \\ g_1 2 \\ g_1 3 \\ g_1 4 \\ g_1 5 \\ g_1 6 \end{bmatrix} \tag{A.78}$$

where

$$\begin{aligned} g_1 &= -5924.0\, g \cos(q_1) + 0.81082\, g \cos(q_1 + q_2 - q_3 - 0.55456) \\ &\quad - 0.2207\, g \cos(q_1 + q_2 + q_3) \sin(q_4) \\ &\quad - 0.00234\, g \sin(q_1 + q_2 + q_3) \cos(q_5) \\ &\quad - 0.00234\, g \cos(q_1 + q_2 + q_3) \cos(q_4) \sin(q_5) \\ &= -58114.4 \cos(q_1) + 7.954 \cos(q_1 + q_2 - q_3 - 0.55456) \\ &\quad - 2.165 \cos(q_1 + q_2 + q_3) \sin(q_4) \\ &\quad - 0.02295 \sin(q_1 + q_2 + q_3) \cos(q_5) \\ &\quad - 0.02295 \cos(q_1 + q_2 + q_3) \cos(q_4) \sin(q_5), \end{aligned} \tag{A.79}$$

$$\begin{aligned} g_2 &= -0.5942\, g \cos(q_1 + q_2) + 0.81082\, g \cos(q_1 + q_2 - q_3 - 0.55456) \\ &\quad - 0.22071\, g \cos(q_1 + q_2 + q_3) \sin(q_4) \\ &\quad - 0.00234\, g \sin(q_1 + q_2 + q_3) \cos(q_5) \\ &\quad - 0.00234\, g \cos(q_1 + q_2 + q_3) \cos(q_4) \sin(q_5) \\ &= -5.829 \cos(q_1 + q_2) + 7.954 \cos(q_1 + q_2 - q_3 - 0.55456) \\ &\quad - 2.165 \cos(q_1 + q_2 + q_3) \sin(q_4) \\ &\quad - 0.02295 \sin(q_1 + q_2 + q_3) \cos(q_5) \\ &\quad - 0.02295 \cos(q_1 + q_2 + q_3) \cos(q_4) \sin(q_5), \end{aligned} \tag{A.80}$$

$$\begin{aligned} g_3 &= 0.81082\, g \cos(q_1 + q_2 - q_3 - 0.55456) \\ &\quad - 0.22071\, g \cos(q_1 + q_2 + q_3) \sin(q_4) \end{aligned}$$

$$
\begin{aligned}
&- 0.00234\, g \sin(q_1 + q_2 + q_3) \cos(q_5) \\
&- 0.00234\, g \cos(q_1 + q_2 + q_3) \cos(q_4) \sin(q_5) \\
&= 7.954 \cos(q_1 + q_2 - q_3 - 0.55456) \\
&- 2.165 \cos(q_1 + q_2 + q_3) \sin(q_4) \\
&- 0.02295 \sin(q_1 + q_2 + q_3) \cos(q_5) \\
&- 0.02295 \cos(q_1 + q_2 + q_3) \cos(q_4) \sin(q_5), \qquad \text{(A.81)}
\end{aligned}
$$

$$
\begin{aligned}
g_4 &= -0.220707\, g \sin(q_1 + q_2 + q_3) \cos(q_4) \\
&\quad + 0.00234\, g \sin(q_1 + q_2 + q_3) \sin(q_4) \sin(q_5) \\
&= -2.165 \sin(q_1 + q_2 + q_3) \cos(q_4) \\
&\quad + 0.02295 \sin(q_1 + q_2 + q_3) \sin(q_4) \sin(q_5), \qquad \text{(A.82)}
\end{aligned}
$$

$$
\begin{aligned}
g_5 &= -0.00234\, g \cos(q_1 + q_2 + q_3) \sin(q_5) \\
&\quad - 0.00234\, g \sin(q_1 + q_2 + q_3) \cos(q_4) \cos(q_5) \\
&= -0.02295 \cos(q_1 + q_2 + q_3) \sin(q_5) \\
&\quad - 0.02295 \sin(q_1 + q_2 + q_3) \cos(q_4) \cos(q_5), \qquad \text{(A.83)}
\end{aligned}
$$

$$
g_6 = 0. \qquad \text{(A.84)}
$$

Bibliography

[1] John J. Craig. *Introduction to Robotics: Mechanics and Control*. Pearson Prentice Hall, 3 edition, 2005.

[2] Kelsey P. Hawkins. Analytic inverse kinematics for the universal robots ur-5/ur-10 arms. Technical report, Georgia Institute of Technology, 2013. Technical report.

[3] Parham M. Kebria, Saba Al-Wais, Hamid Abdi, and Saeid Nahavandi. Kinematic and dynamic modelling of ur5 manipulator. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 004229–004234. IEEE, 2016. Project upload: kebria2016.pdf.

[4] Hassan K. Khalil. *Nonlinear Systems*. Prentice Hall, 3 edition, 2002.

[5] Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[6] ROS-Industrial. Ros-industrial universal robot support, 2026. GitHub repository: https://github.com/ros-industrial/universal_robot.git.

[7] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: Modelling, Planning and Control*. Springer, 2010.

[8] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley & Sons, 2006.

[9] Universal Robots. *Universal Robots ROS Driver Documentation*, 2024. Official ROS driver documentation for Universal Robots manipulators.

[10] Universal Robots A/S. *UR5e Technical Specifications*, 2024. Technical specifications and product data for the UR5e collaborative robot.

[11] Universal Robots A/S. *UR5e User Manual*, 2025. e-Series user manual, SW 5.23.

[12] Robert L. II Williams. Universal robot ure-series cobot kinematics and dynamics, 2024. Project upload: UniversalRobotKinematics.pdf.