Modelling, Simulation and Experiment of UR5 Robot

BEng Project Final Report

2022-2023

Author: Junhao TU

Student ID: 20217175

Date of report: 4th May 2023

Supervisor: Prof. David Branson III

Department of Mechanical, Materials and Manufacturing Engineering

University of Nottingham

Summary

Serial manipulators, such as the UR5 robot, have become integral in various robotic systems. Despite its popularity and collaborative design, the relatively low accuracy of the UR5 necessitates the development of novel controllers to enhance its performance. To create these controllers, researchers require precise simulation and mathematical models of the UR5 manipulator. This study aims to develop a comprehensive set of mathematical models for the UR5, including kinematics, dynamics, and dynamics linearization, and implement the corresponding Matlab code to simulate these models. The models' accuracy and utility will be assessed through simulations and experiments. The mathematical models created for the UR5 manipulator could provide a valuable resource for researchers aiming to develop novel controllers to improve the accuracy of the robot. The main contribution of this paper is the development of complete Matlab-based models for the UR5 robot and the evaluation of their accuracy. The report is structured as follows: The modelling section details the derivation process for the UR5's kinematics, dynamics, and dynamics linearization equations. The simulation section evaluates the models' quality, validity, and accuracy through simulation tests. After that, a circular tracking experiment assesses the models' representation of the actual robot. Finally, the results are discussed, and conclusions drawn.

Contents

Nomenclature and Acronyms

Greek Letters

α_i	DH parameter, link twist of link i	rad
θ	Angle of joint	rad
θ_i	DH parameter, joint angle of link i	rad
ρ	Mass density of a body	kgm^{-3}
τ	Joint torque vector	Nm
ω	Angular velocity	$rads^{-1}$
Δt	Change of time	t

Latin Capital Letters

A	State matrix	
B	System matrix	
C	Output matrix	
$C(q,\dot{q})$	Coriolic and centrifugal matrix	kgm^2s^{-1}
D	Input matrix	
F	Force acting on the link	N
$G(q)$	Gravity vector	Nm
I	Motor current	A
I_i	Interia tensor of link i, expressed in frame i	kgm^2
I_{xx}	Mass moments of interia	kgm^2
I_{xy}	Mass products of interia	kgm^2
${}^{C}I$	Interia tensor of a link, expressed in frame attached on the center position of the link.	kgm^2
K_p	Constant for current-torque relationship	
$M(q)$	Interia matrix	kgm^2
M_i	Interia matrix of link i, expressed in frame i	kgm^2
N	Moment	Nm
$O_ix_iy_iz_i$	Cartesian coordinate frame i, attached to link i	
${}^{i-1}P$	Position vector, expressed in frame $i-1$	mm
${}^{i}P_{C_i}$	Position vector, expressed in frame i on the center point of link i	mm

${}^{i-1}_{i}R$	Rotational matrix of $i-1$ frame, expressed in frame i	mm
S	The number of operating points	
${}^{i-1}_{i}T$	Transformation matrix of $i-1$ frame, expressed in frame i	
${}^{i}\hat{Z}_i$	The unit vector of the Z axis of frame i	

Latin Lowercase Letters

a_i	DH parameter, link length of link i	mm
d_i	DH parameter, link offset of link i	mm
f	Force of interaction between the links	N
g	The acceleration of gravity	ms^{-2}
m	Mass of a link	kg
m_{ci}	Vector of the center point of link $i,$ expressed in frame i	m
n	Overall torque acting at joint	Nm
q	Joint angle vector	rad
s_i	Experimental value	
$\hat{s}_i$	Theoretical value	
t	Continuous time	t

Acronyms

AC	Alternating Current
COM	Center of Mass
DH	Denavit-Hartenberg method
DOF	Degree of Freedom
IFR	International Federation of Robotics
LPV	Linear Parameter-Varying
MAE	Mean Absolute Error
PID	Proportional-Integral-Derivative control
POE	Product of Exponentials method
UR5	A robot from Universal Robots, Model 5

1. Introduction

1.1. Background

Serial manipulators are engineered as a series of links connected by joints, extending from the base to the end-effector. Among these, six-degree-of-freedom (DOF) manipulators are widely used in numerous robotic systems due to their ability to manipulate objects in any position and orientation within the robot's workspace. Since the invention of the first 6 DOF manipulator, FAMULUS, in 1973, manipulators have increasingly replaced humans in various industries [1, 2]. Although the current 6 DOF manipulators satisfy most industrial requirements, the development of more advanced robots continues for improved performance. In recent years, the UR5 robot, a manipulator developed by Universal Robots, has garnered significant attention within the robotics research community. As reported by the International Federation of Robotics (IFR), the UR5 has emerged as a leading player in the industrial robot market segment, with its market share rapidly expanding - annual installations increased by 11.8% between 2021 and 2022 [3].

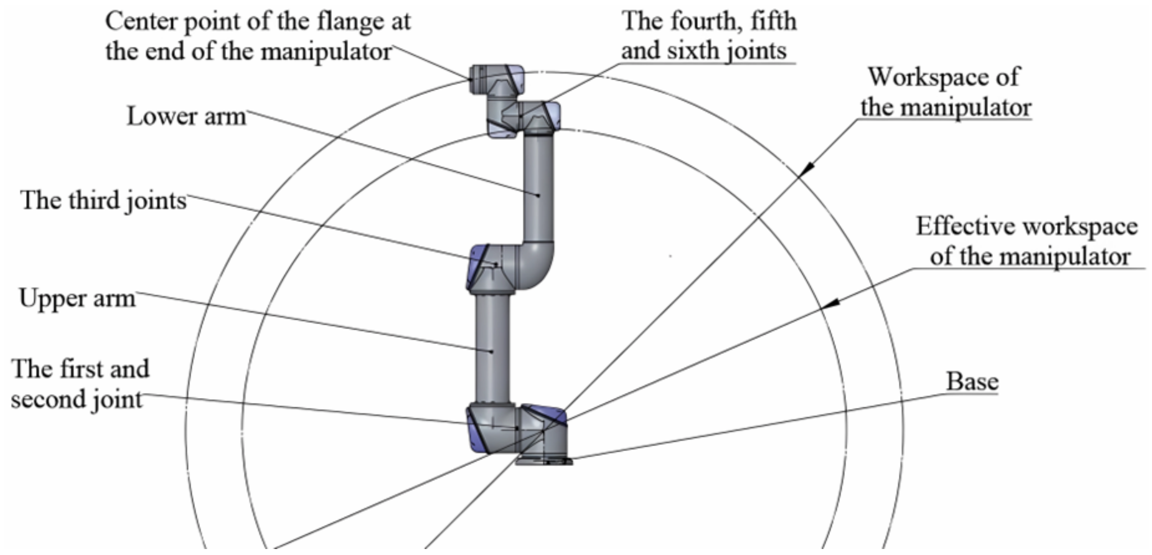
Center point of the flange at the end of the manipulator
The fourth, fifth and sixth joints
Workspace of the manipulator
Lower arm
The third joints
Effective workspace of the manipulator
Upper arm
The first and second joint
Base

Figure 1. The structure and workspace of UR5 manipulator. [4]

The UR5, as shown in Figure 1, features an asymmetrical structure comprising a base, six joints, six links, and an end-effector. Each joint is equipped with a revolute actuator, providing an extensive workspace and flexibility. The arm's lightweight design, crafted from aluminum, facilitates ease of integration and versatility across various applications. Besides, the UR5's collaborative design enables it to operate safely alongside humans, making it a popular choice in both academic and industrial settings [5, 6]. Despite these claimed advantages, the relatively low accuracy of the UR5 manipulator compared to other robotic systems limits its use in certain applications. The UR5 has a repeatability accuracy and absolute accuracy of ±0.1 mm and 3.0 mm, respectively, within its approximately 1 m^3 workspace [7]. In comparison, numerically controlled automation systems can typically achieve absolute positioning with less than 1 mm accuracy within a 1 m^3 working volume [8]. Therefore, numerous approaches have been proposed to improve the accuracy of the UR5 and one of the most effective is the use of a controller [3, 6, 9].

For the development of a novel controller to improve accuracy, researchers require access to precise simulation and mathematical models of the UR5 manipulator. Currently, URSim and ROS are the two main software programs that have open-source built-in models of the UR5 for simulation [9, 10]. However, the detailed parameters of these models are inaccessible, preventing researchers from directly utilizing these software programs to develop controllers. As a result, some researchers have created various models using the Simmechanics and Simulink platforms [5, 6, 11], but these models are either incomplete or only applicable to specific control methods. Furthermore, many modelling studies on various robot manipulators have been also conducted in the existing literature, mainly focusing on well-known 6 DOF manipulators such as ABB and KUKA [1, 12]. The extensive literature on modelling can be attributed to several factors: these models are critical to a wide range of research efforts, and the model parameters are different for each robot, even those with similar structures. In addition, developing these models is challenging because the process is time-consuming and often involves complex and extensive mathematical formulations.

1.2. Aims and Objectives

This study aims to develop a comprehensive set of mathematical models for the UR5 manipulator and implement the corresponding Matlab code to simulate these models. Subsequently, the accuracy and utility of the models will be assessed through simulations and experiments. This paper will specifically present a detailed mathematical model encompassing the kinematics, dynamics, and dynamics linearization of the UR5 manipulator. The kinematic model consists of both the forward and inverse kinematic equations of the robot. The dynamics model covers the closed-form forward and inverse kinetic equations, along with the process of solving specific parameters. The linearization of the dynamics involves obtaining an approximate linear representation of the dynamics model, thereby simplifying its application across various control methods. These models could be readily used by any researcher using the UR5 robot. The main contribution of this paper is the development of complete Matlab-based models for the UR5 robot with evaluating the accuracy of these models. According to the authors' research, this is the most complete mathematical model of the UR5 to date.

The report is structured as follows: Modelling section describes the method of modelling and presents the detailed derivation process of obtaining kinematics, dynamics, and dynamics linearization equations of the UR5. The simulation section will implement the models into Matlab to evaluate the quality of modeling and verify the validity and accuracy through several simulation tests. Subsequently, a circular tracking experiment will be conducted to check if the developed models can be implemented to represent the actual robot. Finally, the results of the simulations and experiments will be discussed, and conclusions will be drawn.

2. Literature Review

This literature review will provide a comprehensive understanding of the kinematics, dynamics, and dynamic linearization modelling of the UR5 manipulator.

2.1. Kinematics

The kinematics of robotic manipulators has been extensively studied, focusing on forward and inverse kinematics, which are crucial for controlling the end-effector.

Forward kinematics involves determining the end-effector's position and orientation given the joint angles of the manipulator. Researchers [1, 13, 14] have investigated the forward kinematics of the manipulators using methods such as the Denavit-Hartenberg (DH) method and product of exponentials (POE) method. The equations derived using these two methods have been shown to be equivalent [14]. However, the derivation of the DH method is standardized to a step-by-step process from which many manipulators are modeled [1, 6]. In this case, the DH method will be used to derive the forward kinematic equations for UR5.

Inverse kinematics is the process of finding the joint angles that achieve a desired end-effector position and orientation. Due to the UR5 having 6 degrees of freedom, the inverse kinematics problem is often redundant, meaning multiple solutions exist [1]. Various methods have been summarized by Siciliano et al. [14] for obtaining multiple solutions, including analytical method, iterative method, and artificial intelligence-based approaches. The analytical solution method is based on geometric analysis or algebra to obtain the solutions, and the form of the resulting solution is closed-form [2, 13]. Many researchers [9, 15] have demonstrated that it is possible to obtain a complete set of inverse kinematics solutions with this method. The numerical method requires inversion and transposition of the Jacobian matrix and can only obtain the relationship between the end-effector position and the joint angle [2, 13]. Kufietia [6] showed how to apply the numerical method to develop an inverse kinematics algorithm. As for artificial intelligence-based methods, they are mainly based on neural networks or machine learning [16], so they are out of the scope of this paper. Since the solutions from the analytical method have higher accuracy and faster computational speed compared with numerical solutions [2, 14], the analytical method is more suitable for modelling.

2.2. Dynamics

The dynamics equation of the manipulator describes the relationship between joint torques and joint motion. Since the UR5 consists of rigid joints and links, the dynamics of the robot can be considered rigid body dynamics [1, 17]. In general, two common methods can be used to derive the equations of motion of a rigid body system, they are the iterative method and the Lagrangian method [1, 12, 13, 18].

The iterative method can be considered a force balance method for solving dynamical problems. Wittenburg [18] proposed to treat a four-axis manipulator as a kinematics chain and obtained the dynamics equations by performing force analysis for each link. Craig [1] summarized this analysis procedure as Newton-Euler iterative method and derived standard equations for solving dynamics of the manipulators. After that, many algorithms [17, 19, 20] based on Newton-Euler iterative method are developed for dealing with dynamics modelling of high DOF manipulators. In contrast, the Lagrangian method is an energy-based dynamical method. Kebria [9] and Guoxian [4] developed a dynamic model of 6 DOF manipulators using Lagrange algorithms on Simmechanics and Simulink platforms, respectively. Kufieta [6] created a Matlab script to derive the dynamics parameters of the UR5 with a simplified Lagrangian method. Besides, Corke and Khatib [13] developed a Matlab toolbox to calculate the dynamics equation of serial manipulators with a Lagrangian algorithm.

For the same manipulator, the dynamics equations obtained by both methods are almost identical [1, 14]. The Newton-Euler iterative algorithm has been used by most scholars because of its wide applicability and ease of programming [17, 19, 20], but the Lagrangian algorithm is much more computationally efficient than the Newton-Euler iterative algorithm for high-degree-of-freedom manipulator arms and is also used by many robotic companies [1, 14]. Since this project is to develop universal models that can be applied to high-speed and high-precision controllers, computational efficiency and universality become more important. Therefore, a Newton-Euler iterative algorithm will be developed to derive a dynamics equation for the UR5.

2.3. Dynamic Linearization

The UR5 manipulator exemplifies a highly nonlinear system, characterized by intricate dynamics equations that complicate the implementation of linear controllers [1, 6, 21]. Linearization is a process that converts the complex nonlinear dynamics of a robotic system into a linear model to facilitate more manageable system analysis and control [1, 14]. At present, three principal linearization techniques address the nonlinear dynamics of robotic systems: local linearization, feedback linearization, and moving linearization [1, 3].

Local linearization approximates the nonlinear equations of motion near a specific operating point using a linear model [14, 22]. De Luca and Oriolo [23] proposed a local linearization-based approach to motion planning for nonholonomic mobile robots. However, this method is unsuitable for controlling manipulators, as they often move over a wide range of the workspace, making it impossible to find a linear model that fits all areas of the workspace [1]. Feedback linearization is a global method that transforms the entire nonlinear system into a linear one by introducing an appropriate feedback control law [24, 25]. Slotine and Li [24] discussed the underlying theory and application of feedback linearization to robotic systems. They suggested that this method demands a deep understanding of nonlinear dynamics and can be computationally complex, making it unsuitable for the linearization of manipulators.

Moving linearization also referred to as linear parameter-varying (LPV) systems, strikes a balance between local linearization's limited applicability and feedback linearization's complexity [1, 26]. This technique enables real-time adjustments of the linearization point as the manipulator moves along its desired trajectory, keeping the linearized model close to the desired position. Apkarian and Adams [27] explored the application of this technique to various uncertain systems, including robotics, demonstrating the potential of moving linearization. After that, Marcos and Balas [26] proposed LPV control techniques for robotic manipulators. Overall, this adaptive approach can be used to facilitate the design of controllers and improve the precision of the end-effector. Therefore, the moving linearization technique has been chosen for system linearization of the UR5.

3. Modelling

As previously stated, a complete collection of Matlab models for the UR5 robot is currently lacking. This section endeavors to address this deficiency by employing the most precise UR5 parameters and measurements accessible to create the models showcased in this paper. The modelling process was divided into three main sections, which are kinematics, dynamics, and dynamic model linearization.

3.1. Kinematic

One of the most notable geometric features of the UR5 manipulator is its asymmetrical layout, which enhances the range of motion and optimizes the workspace [3, 5]. A key characteristic of this feature is that its last three joints do not intersect at a single point. Consequently, all six joints contribute to both the positioning and orientation of the end-effector, unlike other 6 DOF robotic manipulators where the first three joints are responsible for positioning and the last three for orientation [9]. As a result, the kinematic analysis of UR5 is more complex compared to other manipulators.

- **Forward kinematics**

Forward kinematics is the process of calculating the spatial position and orientation of a robotic manipulator's end-effector by incorporating relevant joint angles and link lengths associated with the manipulator's structure. To determine forward kinematics, we first need to assign coordinates to each joint and the end of the UR5 with the DH convention, and then connect each adjacent coordinate with transformation matrices.

In Figure 2a, the structure of UR5 manipulators and allocation of each joint is presented. It has six revolute joints and seven links, and each joint has a single degree of freedom. Based on the position of joints and the definition of DH convention, seven DH frames are assigned to the robot as Figure 2b. Note that the position of joint frame $O_2x_2y_2z_2$ is not on the second link as required by modified DH convention, but on the first link, as this minimizes the number of non-zero DH parameters required and simplifies the subsequent derivation process.

The link dimensions, provided by the manufacturer and verified on the manipulator [28], are used to extract the DH parameters based on the assigned coordinate frames and link lengths. The resulting DH parameters for the specified frames can be found in Table 1.

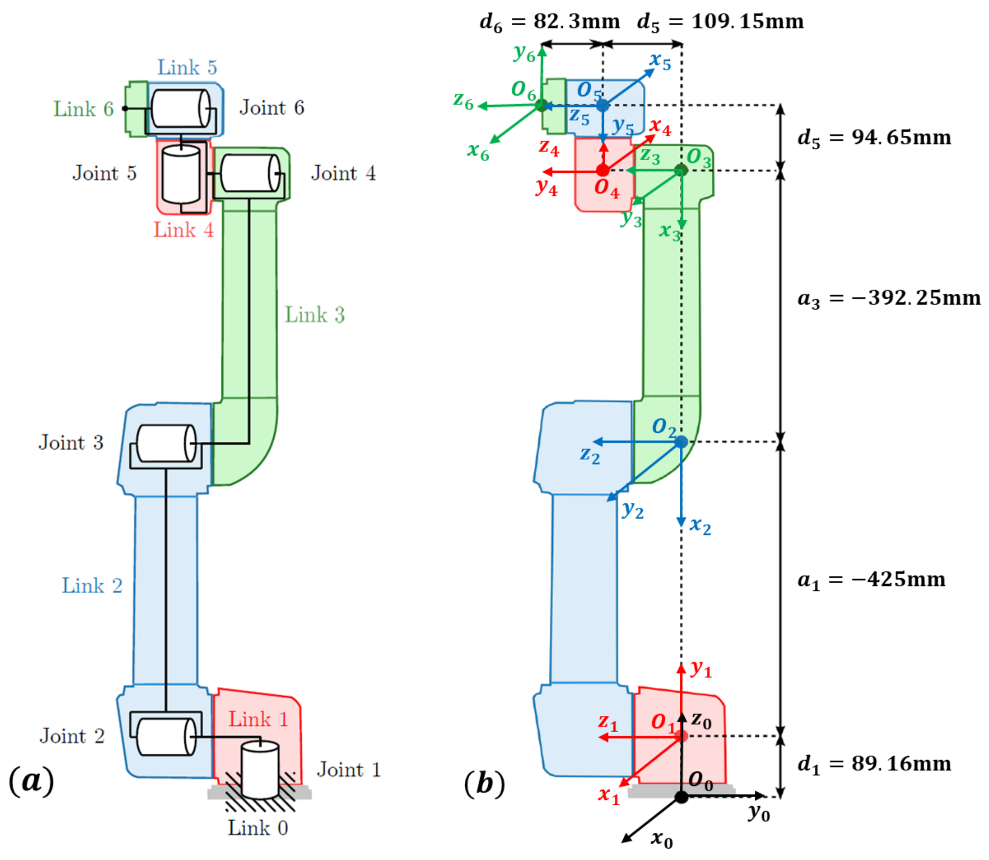

Figure 2. (a) Internal structure of the UR5. [6] (b) DH frames assignment of UR5.

Table 1. DH table for the UR5 manipulator

Link i	θ_i [rad]	d_i [mm]	a_i [mm]	α_i [rad]
1	θ_1	89.16	0	$\pi/2$
2	θ_2	0	−425	0
3	θ_3	0	−392.25	0
4	θ_4	109.15	0	$\pi/2$
5	θ_5	94.65	0	$-\pi/2$
6	θ_6	82.3	0	0

According to [1, 29], the transformation matrix between every two adjacent joints can be expressed with Equation (1). Note that $c\theta$ and $s\theta$ represent $cos(\theta)$ and $sin(\theta)$.

$$ {}^{i-1}_{i}T = \begin{bmatrix} {}^{i-1}_{i}R & {}^{i-1}P \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1) $$

Substituting DH parameters in Table 1 into Equation (1), the transformation matrices between each coordinate frame can be obtained respectively:

$$
{}_1^0T = \begin{bmatrix} c\theta_1 & 0 & s\theta_1 & 0 \\ s\theta_1 & 0 & -c\theta_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad
{}_2^1T = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & a_2c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & a_2s\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$
$$
{}_3^2T = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & a_3c\theta_3 \\ s\theta_3 & c\theta_3 & 0 & a_3s\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad
{}_4^3T = \begin{bmatrix} c\theta_4 & 0 & s\theta_4 & 0 \\ s\theta_4 & 0 & -c\theta_4 & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}
$$
$$
{}_5^4T = \begin{bmatrix} c\theta_5 & 0 & -s\theta_5 & 0 \\ s\theta_5 & 0 & c\theta_5 & 0 \\ 0 & -1 & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad
{}_6^5T = \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ s\theta_6 & c\theta_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

The overall transformation matrix for the forward kinematics of the UR5 manipulator is then obtained by multiplying the six transformation matrixes together:

$$
{}_6^0T = {}_1^0T \cdot {}_2^1T \cdot {}_3^2T \cdot {}_4^3T \cdot {}_5^4T \cdot {}_6^5T = \begin{bmatrix} {}_6^0R & {}^0P \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}
$$

Rotational matrix, ${}_6^0R$ and position vector, ${}^0P$ indicate the orientation and position of the end-effector in the base coordinates, respectively. As there is no dedicated tool installed at the end of the last link, the original point O_6 is regarded as the tip of the end-effector. The position vector of O_6 in terms of joint frame $O_6x_6y_6z_6$ can be represented by ${}^6P = [0, 0, 0, 1]^T$. Therefore, the position of this point in the base frame could be calculated as:

$$
{}^0P = {}_6^0T \cdot {}^6P = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}
$$
$$
= \begin{bmatrix} d_5c_1d_{234} + d_4s_1 - d_6c_1c_{234} + a_2c_1d_2 + d_6c_5s_1 + a_3c_1c_2c_3 - a_3c_1s_2s_3 \\ d_5s_1s_{234} - d_4c_1 - d_6s_1c_{234} - d_6c_1c_5 + a_2c_2s_1 + a_3c_2c_3s_1 - a_3s_1s_2s_3 \\ d_1 - d_6s_{234}s_5 + a_3s_{23} + a_2s_2 - d_5c_{234} \\ 1 \end{bmatrix} \tag{4}
$$

Where c_{234} represents the $cos(\theta_2 + \theta_3 + \theta_4)$ and s_{234} stands for the $sin(\theta_2 + \theta_3 + \theta_4)$.

- **Inverse kinematics**

For inverse kinematics, the values of joint variables must be determined based on the given end-effector positions and orientation. Specifically, given a desired ${}_6^0T$ matrix, the joint vector $q = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]^T$ is needed to be calculated. UR5 is a manipulator in which each joint can rotate 360° and the shoulder, elbow, and wrist joints (joints 2, 3, and 4) are parallel to each other. Such a configuration satisfies Pieper's criterion [3, 9], which ensures that its inverse kinematics has a deterministic analytical solution. The inverse kinematics of the UR5 would be solved using Pieper's solution algorithm, which mainly has three steps [1, 15]. The first step is to solve the expression of ${}_{5}^{1}T$ for joints 2, 3, and 4 as a whole. Then the angular expressions of joints 1, 5 and 6 are solved by relating the ${}_{5}^{1}T$ expression to the forward kinematics equations. The second step solves the expression of ${}_{4}^{1}T$ by the obtained joints 1, 5, and 6. Then the angular expressions of joints 2, 3 and 4 are solved by the relation of the complete positive kinematic equation of ${}_{4}^{1}T$. Finally, the singular position of UR5 would be determined by analyzing the valid conditions and boundary cases of the six joint expressions.

- **Inverse kinematics of joints 1, 5 and 6**

Rearranging Equation (3) in terms of ${}_{5}^{1}T$, the transformation matrix can be rewritten as:

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

Equating the term located in the position (3, 4) on both sides of Equation (5):

$$-p_y c_1 + d_6(a_y c_1 - a_x s_1) + p_x s_1 = d_4 \tag{6}$$

Solving Equation (6) for θ_1 with the angle inverse approach proposed in [1]:

$$\begin{gathered}\theta_1 = A\tan2(m_1, n_1) - A\tan2\left(\mathrm{d}_4, \pm\sqrt{m_1^2 + n_1^2 - d_4^2}\right), (m_1^2 + n_1^2 - d_4^2 \geq 0)\\ m_1 \ = d_6 a_y - p_y, n_1 = a_x d_6 - p_x\end{gathered} \tag{7}$$

Then, equating the term located in the position (3, 3) on both sides of Equation (5):

$$a_x s_1 - a_y c_1 = c_5 \tag{8}$$

Applying the inverse cosine function on Equation (8), the θ_5 can be obtained as:

$$\theta_5 = \pm\, cos^{-1}(a_x s_1 - a_y c_1) \tag{9}$$

Lastly, equating the term located in the position (3, 2) on both sides of Equation (5):

$$s_6(o_y c_1 - o_x s_1) - c_6(n_y c_1 - n_x s_1) = s_5 \tag{10}$$

Solving Equation (10) for θ_6 with the angle inverse approach in [1] once again:

$$\begin{gathered}\theta_6 = Atan2(m_2, n_2) - Atan2\left(S_5, \pm\sqrt{m_2^2 + n_2^2 - S_4^2}\right), (\sqrt{m_2^2 + n_2^2 - S_4^2} \geq 0)\\ m_2 = n_x s_1 - n_y c_1, n_2 = o_x s_1 - o_y c_1\end{gathered} \tag{11}$$

➢ **Inverse kinematics of joints 2, 3 and 4**

Expressing Equation (3) in terms of ${}^{1}_{4}T$ as:

$$ {}^{0}_{1}T^{-1} \cdot {}^{0}_{6}T \cdot {}^{5}_{6}T^{-1} \cdot {}^{4}_{6}T^{-1} = {}^{1}_{2}T \cdot {}^{2}_{3}T \cdot {}^{3}_{4}T = {}^{1}_{4}T \quad (12) $$

Based on Equation (12), the following two expressions are obtained by equating the terms located in the position (1, 4), and the position (2, 4) on both sides:

$$ m_3 = a_3 c_{23} + a_2 c_2 \quad (13) $$

$$ n_3 = a_3 s_{23} + a_2 s_2 \quad (14) $$

Where,

$$ m_3 = d_5\left(s_6(n_x c_1 + n_y s_1) + c_6(o_x c_1 + o_y s_1)\right) - d_6(a_x c_1 + a_y s_1) + p_x c_1 + p_y s_1 $$

$$ n_3 = p_z - d_1 - a_z d_6 + d_5(o_z c_6 + n_z s_6) $$

The sum of the squares of Equations (13) and (14) can be expressed as:

$$ {a_2}^2 + {a_3}^2 + 2a_2 a_3 c_3 = m_3^2 + n_3^2 \quad (15) $$

Therefore, the expression of θ_3 can be obtained as:

$$ \theta_3 = \pm cos^{-1}\left(\frac{m_3^2 + n_3^2 - {a_2}^2 - {a_3}^2}{2a_2 a_3}\right), (m_3^2 + n_3^2 \leq (a_2 + a_3)^2) \quad (16) $$

Expanding c_{23} and s_{23} in Equations (13) and (14):

$$ m_4 = (a_3 c_3 + a_2)c_2 - a_3 s_3 s_2 \quad (17) $$

$$ n_4 = a_3 s_3 c_2 + (a_3 c_3 + a_2)s_2 \quad (18) $$

Substituting the expression of θ_3 into Equations (17) and (18):

$$ s_2 = \frac{(a_3 c_3 + a_2)n_4 - a_3 s_3 m_4}{a_2^2 + a_3^2 + 2a_2 a_3 c_3} \quad (19) $$

$$ c_2 = \frac{m_4 + a_3 s_3 s_2}{a_3 c_3 + a_2} \quad (20) $$

Dividing Equation (19) by Equation (20) and using the inverse tangent formula to obtain θ_2:

$$ \theta_2 = Atan\,2(s_2, c_2) \quad (21) $$

Equating the terms in positions (2, 2) and (1, 2) in Equation (12) separately:

$$s_{234} = -s_6(n_x c_1 + n_y s_1) - c_6(o_x c_1 + o_y s_1) \tag{22}$$

$$c_{234} = o_z c_6 + n_z s_6 \tag{23}$$

Dividing Equation (22) by Equation (23) and using the Artan formula to obtain $\theta_2 + \theta_3 + \theta_4$:

$$\theta_2 + \theta_3 + \theta_4 = A\tan2(-s_6(n_x c_1 + n_y s_1) - c_6(o_x c_1 + o_y s_1), o_z c_6 + n_z s_6) \tag{24}$$

As the expressions of θ_2 and θ_3 have been obtained， θ_4 can be simply calculated by subtracting them from Equation (24):

$$\theta_4 = A\tan2(-s_6(n_x c_1 + n_y s_1) - c_6(o_x c_1 + o_y s_1), o_z c_6 + n_z s_6) - \theta_2 - \theta_3 \tag{25}$$

➢ **Singular position analysis**

Singular positions are configurations in which the robot's end-effector loses one or more degrees of freedom [12]. This can result in an inability to perform certain tasks or movements and can also cause the robot to become unstable or even crash [1, 15]. Therefore, it is crucial to identify and avoid singular positions in robot control and motion planning. The UR5 has three joint singularities, including shoulder, elbow and wrist singularities. These singularities are caused by the kinematic structure of the UR5 and can be determined by analyzing the equations of inverse kinematics.

Shoulder singularities occur when the tip of the end-effector O_6 is in the plane formed by the axes z_1 and z_2, and the first joint angle cannot be solved. From Equation (7), the mathematics expression for this condition is:

$$m_1^2 + n_1^2 - d_4^2 = 0 \tag{26}$$

Elbow singularities occur when the first and last three joints are in the same plane, and the arm cannot bend or extend. In this case, the second angle cannot be solved. From Equation (16), the mathematics expression for this condition is:

$$m_3^2 + n_3^2 - (a_2 + a_3)^2 = 0 \tag{27}$$

Wrist singularities occur when axes z_4 and z_6 are parallel and the sixth joint angle cannot be solved. From Equation (11), the mathematics expression for this condition is:

$$m_2^2 + n_2^2 - s_4^2 = 0 \tag{28}$$

By avoiding or managing these quirks, the safe and efficient operation of UR5 across a wide range of tasks and applications can be largely ensured.

3.2. Dynamics

Up until now, the kinematic analysis of the UR5 manipulator has been conducted; however, the forces and torques required to generate its motion have not been examined. This section will derive the equation of motion for the UR5 and explore how the torques produced by joint motors or forces applied to the manipulator cause the manipulator to move. As discussed in the literature review, the Newton-Euler method is utilized to obtain closed-form dynamic equations $\tau = f(q, \dot{q}, \ddot{q})$. The modelling process commences by determining the mass distribution of the links that comprise the manipulator. Subsequently, the detailed derivation process of the equation of motion, through iterative application of the Newton-Euler method, will be demonstrated.

- **Mass distribution**

Consider the links that compose the UR5 manipulator as rigid bodies. During the operation of the UR5, these rigid bodies perform rotational motion around varying axes within a three-dimensional space. In such situations, the inertia tensor is frequently utilized to represent the mass distribution of the links [8]. The inertia tensor can be expressed as:

$$I_i = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \tag{29}$$

Where I_i represents the inertia tensor of link i in the coordinate frame i. The elements I_{xx}, I_{yy} and I_{zz} are called the mass moments of inertia and the other elements with mixed indices are called the mass products of inertia. The elements of inertia tensor can be calculated with the equations as follow:

$$\begin{aligned} I_{xx} &= \iiint (y^2 + z^2)\rho(x,y,z)dxdydz & I_{xy} &= \iiint xy\rho(x,y,z)\mathrm{d}x\mathrm{d}y\mathrm{d}z \\ I_{yy} &= \iiint (x^2 + z^2)\rho(x,y,z)\mathrm{d}x\mathrm{d}y\mathrm{d}z & I_{xz} &= \iiint xz\rho(x,y,z)\mathrm{d}x\mathrm{d}y\mathrm{d}z \\ I_{zz} &= \iiint (x^2 + y^2)\rho(x,y,z)\mathrm{d}x\mathrm{d}y\mathrm{d}z & I_{yz} &= \iiint yz\rho(x,y,z)\mathrm{d}x\mathrm{d}y\mathrm{d}z \end{aligned} \tag{30}$$

Therefore, the mass distribution characteristics of each link in the robotic system can be fully determined by knowing the position of the center of mass and the inertia tensor. The mass properties of the UR5, as provided by the manufacturer [28], are presented in Table 2, which includes the total mass of each link and their corresponding center point vector. Additionally, the official positions of the mass centers are illustrated in Figure 3a.

Table 2. Mass and location of the mass center for each link

Link i	Weight M_i [kg]	Location of mass center point m_{ci} in coordinate frame i [m]
1	3.7	$[0, -0.02561, 0.00193]^T$
2	8.393	$[0.2125, 0, 0.11336]^T$
3	2.33	$[0.15, 0, 0.0265]^T$
4	1.219	$[0, -0.0018, 0.01634]^T$
5	1.219	$[0, 0.0018, 0.01634]^T$
6	0.1879	$[0, 0, -0.001159]^T$

As seen in Figure 3a, the links of UR5 exhibit irregular and asymmetrical structures, with their mass being non-uniformly distributed. This makes it challenging to directly compute their inertia tensors using Equation (30). Therefore, many researchers have proposed various simplified models to address this complexity [6, 9, 21, 30].

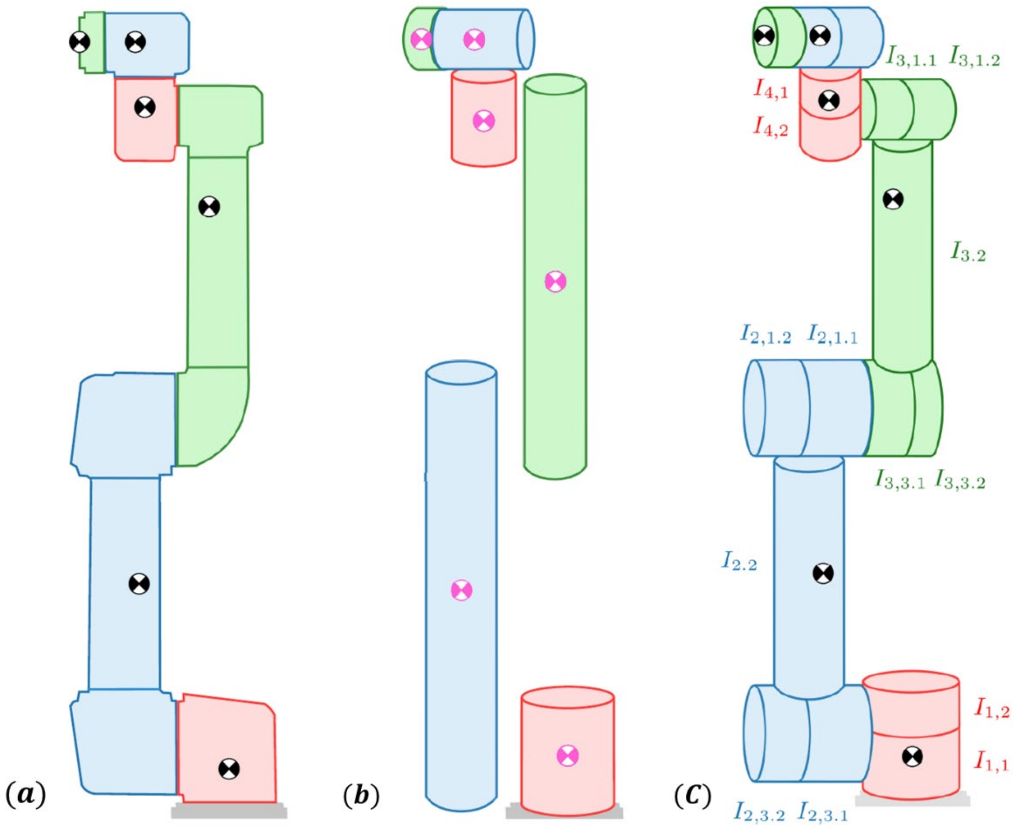

Figure 3. (a) Offical mass center points from the manufacturer. (b) Mass center points from the cylindrical model. (c) Mass center points from Kufieta's model. [6]

A common model is to approximate the links as cylinders with homogeneous density [21, 30], as shown in Figure 3b. However, this model's centroids do not align with the mass centers provided by the manufacturer for UR5, resulting in inaccurate dynamic equations. The primary reason is that the UR5 is a manipulator with lightweight design, and the location of the motors significantly influences the mass distribution of the links, thus the links can not be simply regarded as cylinders. To overcome this issue, Kufieta [6] developed an improved model by dividing each link into multiple cylinders with varying homogeneous densities and highlighting the contribution of motors to mass. This approach, shown in Figure 3c, successfully preserves the correct link shape while ensuring the location of the mass center point aligns with the manufacturer's data under the assumption of homogeneous density for indiviual cylinders. With this refined model, the inertia tensors of the six links can be obtained using Equation (30), as well as the mass properties presented in Table 2. The results are as follows:

$$\begin{aligned}
I_1 &= \begin{bmatrix} 0.0067 & 0 & 0 \\ 0 & 0.0064 & 0 \\ 0 & 0 & 0.0067 \end{bmatrix} & I_2 &= \begin{bmatrix} 0.0149 & 0 & 0 \\ 0 & 0.3564 & 0 \\ 0 & 0 & 0.3553 \end{bmatrix} \\
I_3 &= \begin{bmatrix} 0.0025 & 0 & 0.0034 \\ 0 & 0.0551 & 0 \\ 0.0034 & 0 & 0.0546 \end{bmatrix} & I_4 &= \begin{bmatrix} 0.0012 & 0 & 0 \\ 0 & 0.0012 & 0 \\ 0 & 0 & 0.0009 \end{bmatrix} \\
I_5 &= \begin{bmatrix} 0.0012 & 0 & 0 \\ 0 & 0.0012 & 0 \\ 0 & 0 & 0.0009 \end{bmatrix} & I_6 &= \begin{bmatrix} 0.0001 & 0 & 0 \\ 0 & 0.0001 & 0 \\ 0 & 0 & 0.0001 \end{bmatrix}
\end{aligned} \tag{31}$$

This model has been validated by [6, 10], showing that it is the most accurate of the current UR5 mass distribution models. With the inertia tensor from this model and the mass values from the manufacturer, the equations of motion will be derived in the next section.

- **Newton-Euler method for dynamics equation**

The force and torque necessary for the motion of the links within the robotic manipulator is a function of the desired link acceleration and the respective mass parameters [1, 13]. To accurately describe the relationship between force, mass parameters, and acceleration, Newton's equations of motion and Euler's equations, which characterize rotational motion, can be employed.

As illustrated in Figure 4, the mass center of the link moves with acceleration $\dot{v}_C$. According to Newton's equations of motion, the force F acting on the center of mass can be expressed as:

$$F = m\dot{v}_C \tag{32}$$

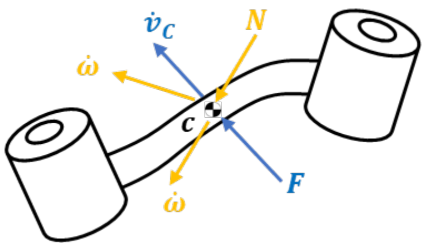

Figure 4. A generalized link under acceleration $\dot{v}_C$ (blue) or under angular velocity ω and angular acceleration $\dot{\omega}$ (yellow).

Similarly, as shown in Figure 4, the link rotates with angular velocity ω and angular acceleration $\dot{\omega}$. Applying Euler's equations, the torque N acting on the rigid body can be expressed as:

$$N = {}^{C}I\dot{\omega} + \omega \times {}^{C}I\omega \tag{33}$$

Where ${}^{C}I$ is the inertia tensor of a frame c whose origin is at the center of mass of the link.

➢ **Outward iteration of velocity and acceleration**

From Equations (32) and (33), it can be inferred that to calculate the forces and torques acting on the center of mass of each link at a given moment, the velocity, acceleration, and angular acceleration of all links at that instant must be obtained. Walker proposed an iterative method to acquire this information [1]. The calculation begins with link 1, then proceeds to the next link, and continues outward until reaching link 6.

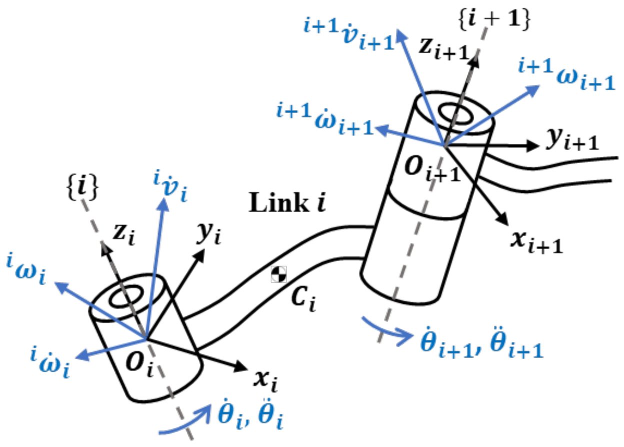

Figure 5. Velocity and acceleration vectors on the link i.

Literature [1, 12, 29] provides a detailed discussion of the propagation of velocity and acceleration between adjacent joints on link i. The all-necessary velocities and accelerations are illustrated on the Figure 5.

For the $(i+1)$th joint, the angular velocity and angular acceleration could be obtained as:

$$^{i+1}\omega_{i+1} = {}^{i+1}_{i}R\ {}^{i}\omega_i + \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} \tag{34}$$

$$^{i+1}\dot{\omega}_{i+1} = {}^{i+1}_{i}R\ {}^{i}\dot{\omega}_i + {}^{i+1}_{i}R\ {}^{i}\omega_i \times \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} + {}^{i+1}\hat{Z}_{i+1} \tag{35}$$

Where ${}^{i+1}_{i}R$ is the rotation matrix of the frame $O_i x_i y_i z_i$ to frame $O_{i+1} x_{i+1} y_{i+1} z_{i+1}$ and ${}^{i+1}\hat{Z}_{i+1}$ is the direction vector of the Z_{i+1} axis.

Also, the velocity at the origin of the $(i+1)$th frame and the velocity at the center of mass of link i can be separately expressed as:

$$^{i+1}\dot{v}_{i+1} =^{i+1}_{i} R\left({}^{i}\omega_i \times {}^{i}P_{i+1} + {}^{i}\omega_i \times \left({}^{i}\omega_i \times {}^{i}P_{i+1}\right) + {}^{i}\dot{v}_i\right) \tag{36}$$

$$^{i}\dot{v}_{C_i} = {}^{i}\dot{\omega}_i \times {}^{i}P_{C_i} + {}^{i}\omega_i \times \left({}^{i}\omega_i + {}^{i}P_{C_i}\right) + {}^{i}\dot{v}_i \tag{37}$$

Where ${}^{i}P_{i+1}$ is the position vector of O_{i+1} to O_i and ${}^{i}P_{C_i}$ is the position vector of C_i to O_i.

After calculating all velocities and accelerations, Equations (38) and (39) can be used to calculate the forces and torques acting on the center of mass of the link $i+1$. The expressions are as follows:

$$F_{i+1} = m\dot{v}_{c_{i+1}} \tag{38}$$

$$N_{i+1} = {}^{C_{i+1}}I\dot{\omega}_{i+1} + \omega_{i+1} \times {}^{C_{i+1}}I\omega_{i+1} \tag{39}$$

In the outward iteration, by evaluating Equations (34) to (39) for each link sequentially, beginning with link 1 and progressing outward from the robot's base, force and torque on the center of mass of each link can be determined. Note that the initial condition for iteration is ${}^{0}\omega = {}^{0}\dot{\omega} = 0$, because the base link is always stationary and installed on the workbench.

- **Inward iteration of force and moment**

After calculating the forces and torques acting on each link, the joint torques can be computed using force balance equations and torque balance equations, which are related to the actual output of joint motors. Figure 6 presents the force diagram for link i of the manipulator under zero gravity. Similarly, other links are also subject to the forces and torques exerted by adjacent links, as well as the forces and torques obtained from the outward iteration process.

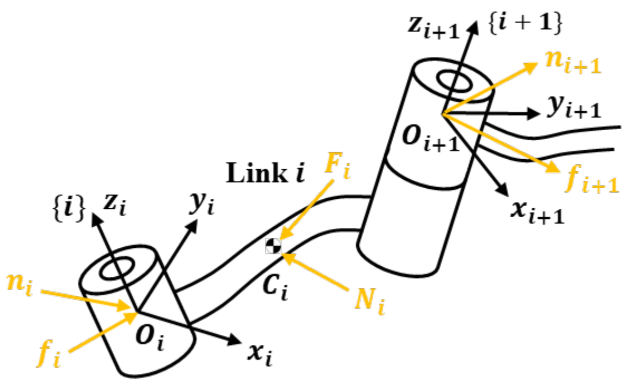

Figure 6. The force balance for a single link i.

Summing the forces acting on link i, the force balance equation is:

$$ {}^{i}F_i = {}^{i}f_i - {}_{i+1}^{i}R \, {}^{i+1}f_{i+1} \tag{40}$$

Rearranging,

$$ {}^{i}f_i = {}^{i}F_i + {}_{i+1}^{i}R \, {}^{i+1}f_{i+1} \tag{41}$$

Summing torques about the center of mass and setting them equal to zero, the torque balance equation is:

$$ {}^{i}N_i = {}^{i}n_i - {}^{i}n_{i+1} + (-{}^{i}P_{C_i}) \times {}^{i}f_i - ({}^{i}P_{i+1} - {}^{i}P_{C_i}) \times {}^{i}f_{i+1} \tag{42}$$

According to [2, 8], substituting Equations (41) into (42) and adding a few rotation matrics, thus overall torque acting at joint i can be obtained as:

$$ {}^{i}n_i = {}^{i}N_i + {}_{i+1}^{i}R \, {}^{i+1}n_{i+1} + {}^{i}P_{C_i} \times {}^{i}F_i + {}^{i}P_{i+1} \times {}_{i+1}^{i}R \, {}^{i+1}f_{i+1} \tag{43}$$

The joint torques supplied by motors can be determined by considering the $\hat{Z}$ component of the torque at joint i applied by one link on its neighboring link:

$$ \tau_i = {}^{i}n_i^T \, {}^{i}\hat{Z}_i \tag{44}$$

In the inward iteration, through evaluating Equations (42), (43) and (44) link by link, starting from link 6 and working inward the base of the robot, the force and torque exerted on link i and link $i-1$ could be obtained. t is important to note that when the UR5 moves in free space, ${}^{6}f_6$ and ${}^{6}n_6$ are set to zero, making the initial application of the equations for link 6 relatively simple. However, when the robot is in contact with the environment, the forces and torques resulting from this contact can be incorporated into the force balance by assigning specific values to ${}^{6}f_6$ and ${}^{6}n_6$.

When the equations in the outward iteration and inward iteration are evaluated symbolically for all links of the UR5 manipulator, a closed-form dynamics equation as below can be yielded:

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) \tag{45}$$

Where τ is the 6×1 joint torque vector supplied by motors of the UR5, while $q, \dot{q}, \ddot{q}$ are 6×1 position, velocity and acceleration vectors, respectively. $M(q)$ denotes the 6×6 inertia matrix of the manipulator, $C(q)$ refers to a 6×6 coriolic and centrifugal matrix, $G(q)$ is the 6×1 gravity vector, with its value depending on the specified gravity value.

According to [1, 14], Equation (45) is called inverse dynamics because joint moments are independent variables in the motion of the operator arm as well as joint variables. If the acceleration term is moved separately to the left side of the equation, a forward dynamics equation can be obtained.

$$\ddot{q} = M^{-1}(q)\big(\tau - C(q,\dot{q})\dot{q} - G(q)\big) \tag{46}$$

In summary, forward dynamics focuses on predicting a robot's motion by considering the input joint torques or forces, whereas inverse dynamics is concerned with computing the necessary joint torques or forces to accomplish a predefined motion or trajectory. The section of simulation will demonstrate how to get the terms in the $M(q)$, $C(q,\dot{q})$ and $G(q)$ with Matlab.

3.3. Dynamic Linearization

The focus of this section is to linearize the closed-form dynamic equations of the UR5 in preparation for controller development. From a control perspective, the UR5 dynamic model is characterized as a nonlinear, strongly coupled, and time-varying system [1, 31]. Although some researchers have proposed methods for designing controllers for such systems directly, traditional control methods, such as PID control or H-infinity control, typically involve analyzing and tuning control parameters based on a linearized dynamic model [31]. Therefore, this section would apply the most appropriate method to linearize the UR5 dynamic equations.

According to [1], the procedure to obtain linearized equations using moving linearization and convert them into transfer functions, which are commonly used in controller design, is as follows:

Step 1: The linearized form of UR5's dynamics equation can be generalized as Equation (47). For a specific operating point, the first-order partial derivative matrices of x and u could be calculated and denoted as A and B, respectively. The moving linearization method essentially obtains the expressions of A and B at various operating points, thereby obtaining the linearized equations of motion for the system at different points.

$$y = Ax + B\tau \tag{47}$$

Where $x = [q, \dot{q}]^T$represents the state vector, $y = [q, \dot{q}]^T$ represents the output vector and τ represents a 6×1 input torque vector.

Step 2: Based on the obtained matrixes A and B, the state-space form of the linearized dynamics equation can be expressed as:

$$\frac{dx}{dt} = Ax + B\tau \tag{48}$$

$$y = Cx + D\tau \tag{49}$$

Where $dx/dt = [\dot{q}, \ddot{q}]^T$, C is the 6×6 output matrix, and D is the 6×6 direct input matrix. If the UR5 manipulator moves in free space without workload, C can be considered as an identity matrix and D is a zero matrix.

Step 3: Substituting matrices A, B, C, and D into the Matlab '$ss2tf$' function to convert the state-space representation into a transfer function.

Step 4: Updating the operating point based on the real-time position and velocity of the UR5 manipulator to achieve moving linearization.

In moving linearization, the most critical step is to obtain the A and B matrices. Given that the positions and velocities of the UR5 manipulator need to be updated at intervals of at least 8ms [28], the first-order partial derivatives of x and u also need to be calculated in near real-time. This requirement can place a considerable computational burden on the controlling computer. Therefore, it is essential to select an appropriate calculation method.

For continuous systems, the Taylor series is a common method for calculating partial derivatives. First, the position vector q and velocity vector q' are expanded to the first-order Taylor series as follows:

$$q(t + \Delta t) = q(t) + \dot{q}(t)\Delta t \tag{50}$$

$$\dot{q}(t + \Delta t) = \dot{q}(t) + \ddot{q}(t)\Delta t \tag{51}$$

By substituting these two series expansions into Equation (45) and neglecting high-order terms, a linearized form can be obtained as:

$$\tau = M(q)\ddot{q}(t) + \frac{\partial M(q)}{\partial q}\dot{q}(t)\Delta t\ddot{q}(t) + C(q,\dot{q}) + \frac{\partial C(q,\dot{q})}{\partial q}\dot{q}(t)\Delta t + \frac{\partial C(q,\dot{q})}{\partial q'}\ddot{q}(t)\Delta t + G(q) + \frac{\partial G(q)}{\partial q}\dot{q}(t)\Delta t \tag{52}$$

Then, performing a partial derivative operation of the Equation (52) in terms of q and q':

$$\frac{\partial\big(M(q)\ddot{q}(t)\big)}{\partial q} \approx -\frac{\partial C(q,\dot{q})}{\partial q} - \frac{\partial g(q)}{\partial q} \tag{53}$$

$$\frac{\partial\big(M(q)\ddot{q}(t)\big)}{\partial \dot{q}} \approx -\frac{\partial C(q,\dot{q})}{\partial q'} \tag{54}$$

Substituting expressions (53) and (54) back into Equation (52) to obtain A and B:

$$A = \frac{\partial\big(M(q)\ddot{q}(t)\big)}{\partial\big(q(t),\dot{q}(t)\big)} \approx \left[0, I;\ -\frac{\partial M\big(q(t)\big)}{\partial t\partial \ddot{q}}, -\frac{\partial M\big(q(t)\big)}{\partial \ddot{q}}\right] \tag{55}$$

$$B = \left[0;\ inv\left(M\big(q(t)\big)\right)\right] \tag{56}$$

Where I is an identity matrix, and $inv(M(q(t)))$ is the inverse matrix of the mass matrix.

However, in practical applications, the feedback data from the UR5 is discrete and cannot be directly applied to the Taylor series method, as shown in Equations (55) and (56). Therefore, a numerical differentiation method called forward difference is employed to enable the use of the Taylor series method in obtaining the partial derivatives of A and B. The forward difference is fundamentally a discrete approximation technique that indirectly utilizes the Taylor series method to approximate nonlinear equations as a series of linear equations [1].

Specifically, small perturbation values are used to make minor changes to the state and input variables, and the corresponding output differences are subsequently calculated to obtain the partial derivatives [32]. Through this process, the A and B matrices are determined, and the linearized model of the system can be established. The equations for A and B using numerical differentiation are as follows:

$$A = \frac{f(q_2,\dot{q}_1,\ddot{q}_1) - \tau_1}{\Delta t} \tag{57}$$

$$B = \frac{f(q_1,\dot{q}_2,\ddot{q}_1) - \tau_1}{\Delta t} \tag{58}$$

Where $f(q\,,\dot{q}\,,\ddot{q}\,)$ is the functional expression of Equation (45), τ_1, q_1, $\dot{q}_1$ and $\ddot{q}_1$ are joint torque, joint angle, joint velocity, and joint acceleration at the initial operating point, respectively. q_2 and $\dot{q}_2$ represent the joint angle and speed at the subsequent operating point. Δt is a small perturbation value, typically set to 1×10^{-6}.

4. Simulations

The complete set of mathematical models of the UR5 has been derived and outlined in the modelling section. The objective of this section is to implement these models in Matlab to test their performance and verify their accuracy in simulations. This section is generally divided into three sections, which are simulations for kinematics, dynamics, and linearization. All scripts can be accessed via this link[1].

4.1. Kinematics Simulations

In the kinematic simulation, a virtual UR5 robot was first created using the DH parameters in Table 1, as shown in Figure 7. Then, scripts invoking forward, and reverse kinematics were implemented to the virtual robot to perform the kinematic simulation. Since the simulations would be performed under ideal conditions, it should be expected that some factors that may affect the robot's motion would be ignored.

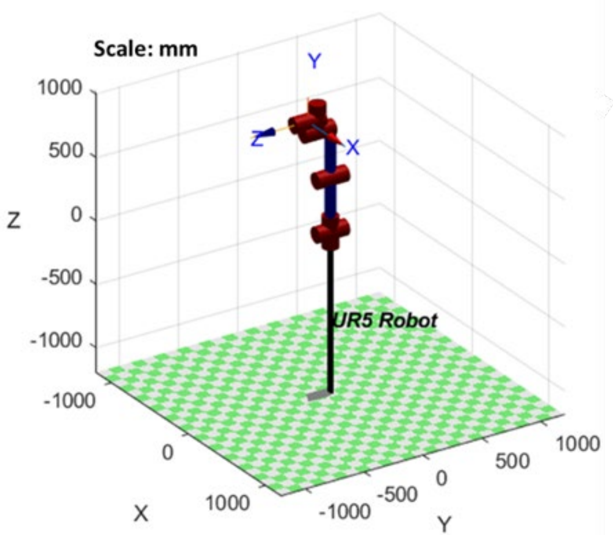

Figure 7. Virtual UR5 manipulator.

- **Forward kinematics verification**

The workspace of the manipulator, also known as the work envelope, represents the set of all possible positions that the end-effector can reach within its range of motion. Workspace

[1] https://github.com/Junhao2000/UR5-final-year-project

analysis studies these reachable positions, providing a means to verify forward kinematics equations by comparing calculated positions to the robot's actual workspace. If the calculated positions and orientations align with the actual workspace, it confirms the accuracy of the forward kinematics equations. Figure 1 shows the workspace of UR5 provided by the manufacturer. This workspace has a quasi-spherical structure, i.e., a spherical shape with an overall course of 1000 mm in diameter, but with a 200 mm diameter cylinder missing from its center [4]. In the simulation, a set of 20,000 random joint variables were entered into Equation (4) to generate a cloud chart of the arrival position of the end-effector, which is illustrated in Figure 8.

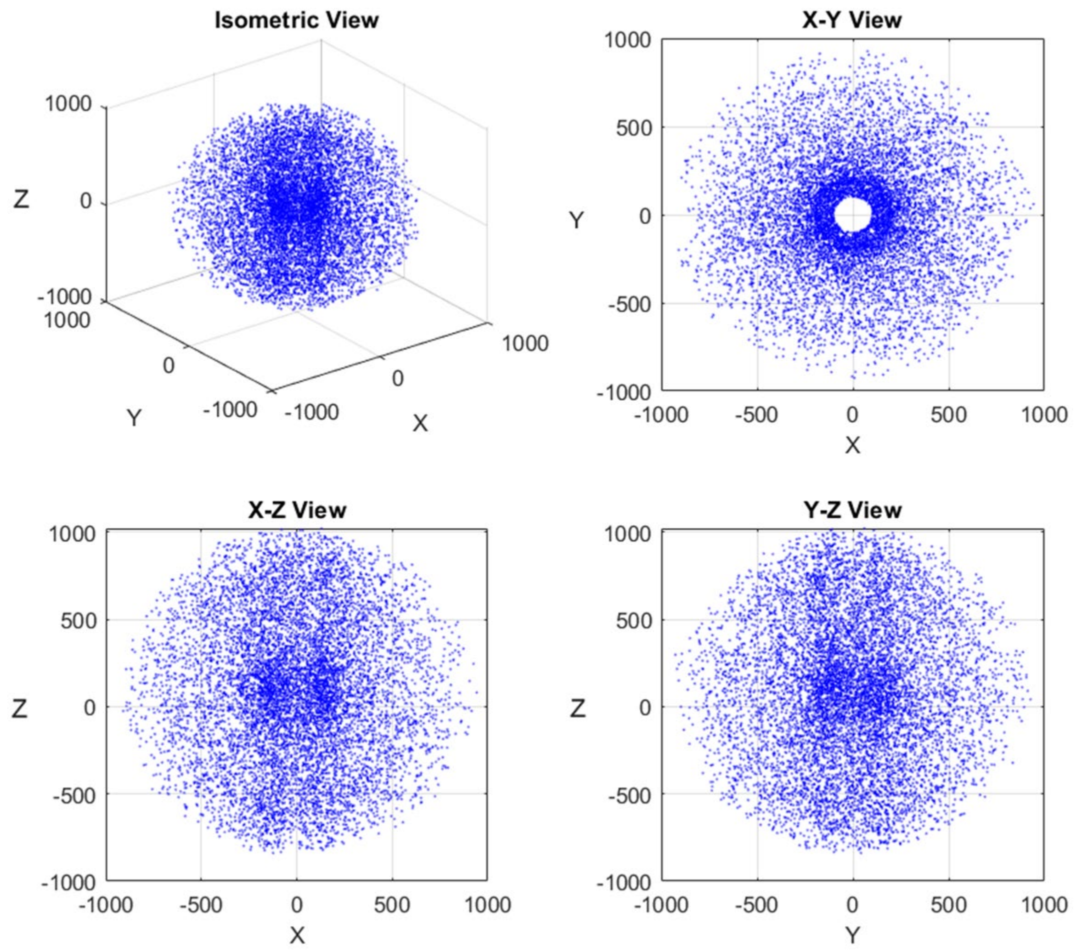

Figure 8. Cloud chart of the workspace of the UR5.

The generated workspace generally matched the actual workspace in size and shape. This confirmed the accuracy of the forward kinematics equation, demonstrating its effectiveness in representing the UR5 manipulator's forward motion characteristics and applicability throughout the entire workspace.

- **Inverse kinematics verification**

Another simulation of kinematics was to apply circular trajectory tracking to verify the inverse kinematic equations. In the simulation, given a circle with a center point of (-400mm, -300mm, 336mm) and a radius of 100mm and parallel to the ground, the virtual UR5 was programmed to track this circle using the kinematic equations. The set of joint variables corresponding to many points uniformly distributed on the circular was first obtained from the inverse kinematic equations. This set of joint variables was then entered into the already evaluated forward kinematic equation, causing the end effector of the virtual robot to track a given circular trajectory in space. The accuracy of the inverse kinematics was evaluated by comparing the deviation of the actual circle with the generated circle. Note that the circle trajectory was away from the singularity positions of the UR5 to avoid the inverse kinematics from being unsolvable. The simulation results are shown in Figure 9 and the trajectories of the two circles are almost identical.

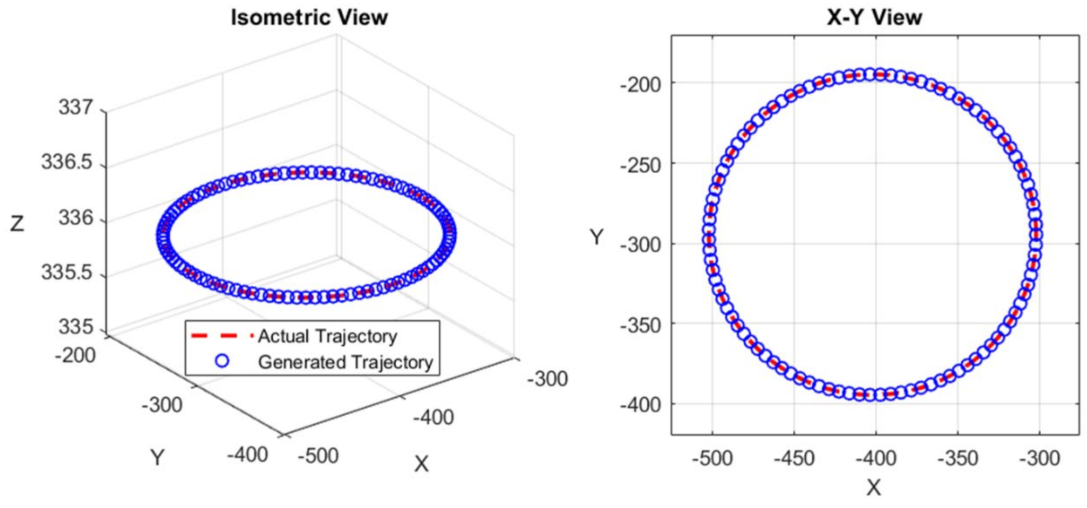

Figure 9. Circular trajectory tracking: actual and generated trajectories.

In order to numerically calculate the deviations of the two circles, the average absolute errors of the 50 operating points in the x, y, and z directions, respectively, were also calculated by Equation (59). Due to $\mathrm{MAE}(\hat{x}_i) = \mathrm{MAE}(\hat{y}_i) = \mathrm{MAE}(\hat{y}_i) = 0$, it can be concluded that the inverse kinematic equation can accurately obtain the joint variables for a specific end-effector position in an ideal environment.

$$\mathrm{MAE}(\hat{s}_i) = \frac{1}{S}\sum_{i=1}^{S} |s_i - \hat{s}_i| \tag{59}$$

Where s_i is the value of the given value, while $\hat{s}_i$ is the value of the generated value. S is the number of operating points.

4.2. Dynamics Simulations

According to the dynamics modelling, the form of the dynamics equation for the UR5 should be $\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q)$. The section on dynamics modelling demonstrated the iterative process of obtaining $M(q), C(q,\dot{q})$ and $G(q)$ in the equation with the Newton-Euler method. Therefore, a dynamics algorithm was developed to implement outward iteration and inward iteration to calculate the three matrices. This algorithm includes the effect of gravity by equating the acceleration of the base link, ${}^{0}\dot{v}_0$ equals to $-g$. However, forces or moments other than rigid body mechanics, such as friction, were not included in this algorithm. Under the common initial conditions ${}^{0}\omega = {}^{0}\dot{\omega} = 0$ and ${}^{6}f_6 = {}^{6}n_6 = 0$, the matrices $M(q), C(q,\dot{q})$ and $G(q)$ can be expressed analytically, and then the dynamics equation was fully determined. The detailed expressions for the equation can be found in the provided link, as they involve an extensive set of parameters.

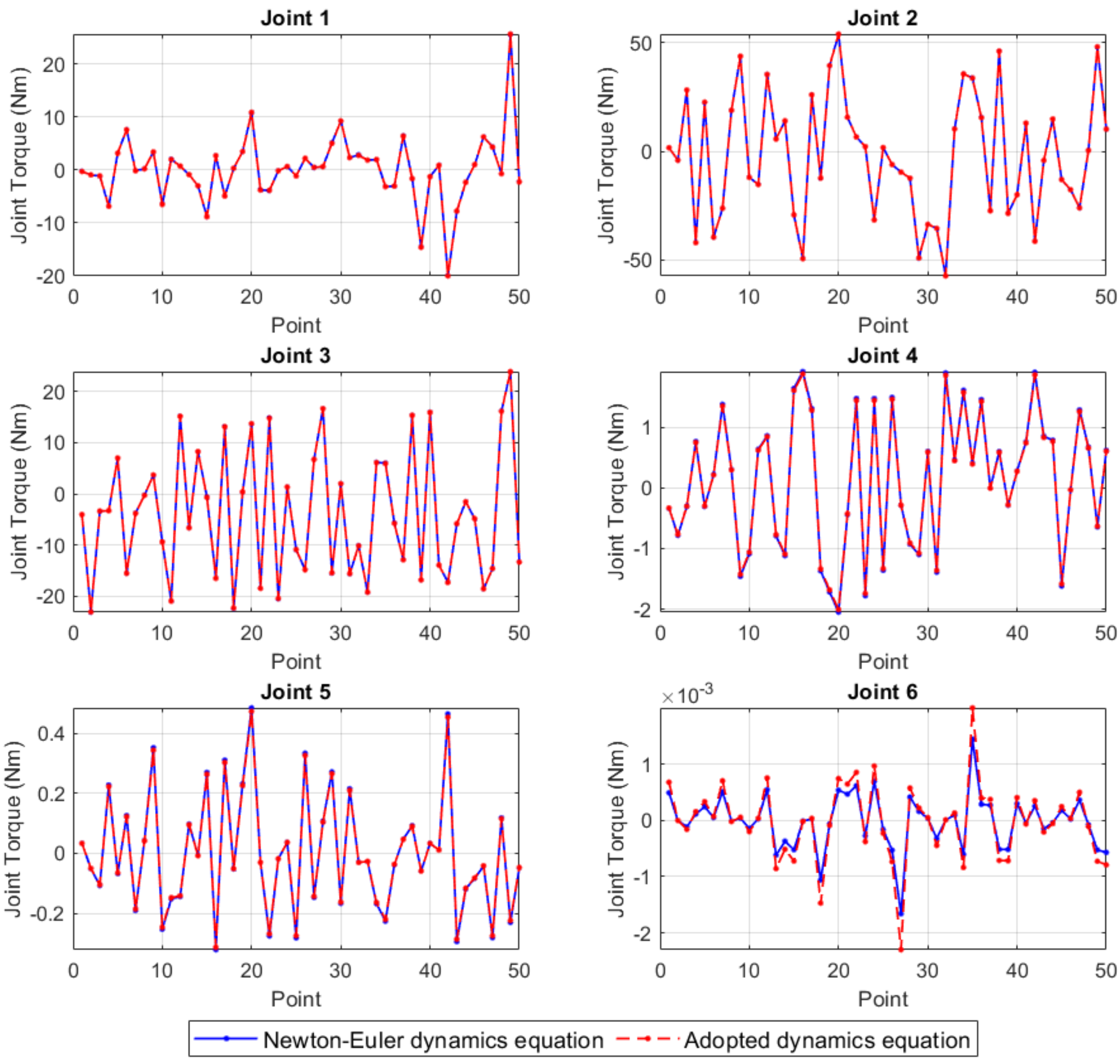

Figure 10. Joint torques from Newton-Euler equation and adopted equation.

After that, a commonly adopted dynamics equation of the UR5 from Kufieta [5, 9, 21] was applied to compare with this equation to perform verification. In the simulation, 50 operating points were input into the adopted and Newton-Euler dynamics equations to calculate the joint torques respectively. Each operating point consisted of a random set of rotation angles, velocity, and acceleration. Figure 10 shows six subplots, each corresponding to each joint of the UR5 robot, and the subplots show the joint torques obtained at each of the 50 operating points using two equations. The joint moments obtained from the two equations were almost the same in the first five joints, while there were some deviations in the sixth joint moment.

To show the moment deviation more visually for each joint, the average absolute error for each joint at all points was calculated using Equation (59), and the results are plotted in Figure 11. A quantitative measure of the agreement between the equations for the UR5 robot dynamics is provided. From the initial joint to the final joint, there was a gradual increase in torque deviation, exhibiting a trend reminiscent of a quadratic function.

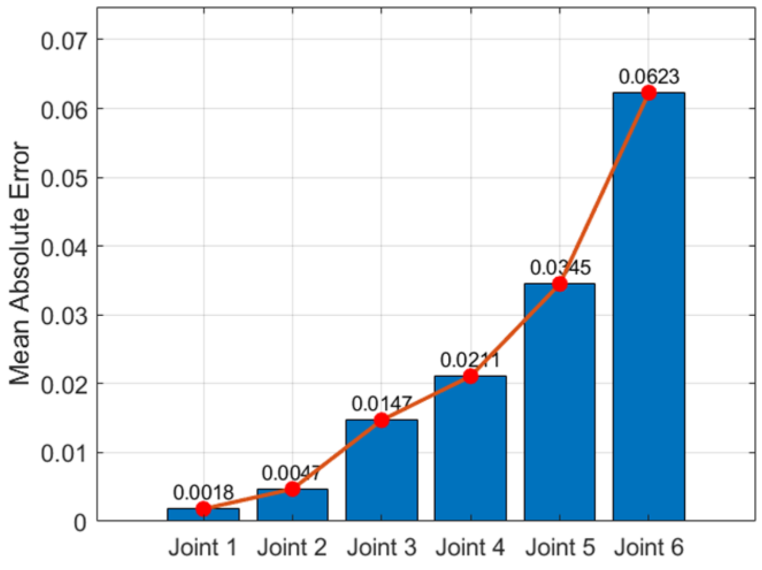

Figure 11. Mean absolute errors of joint torques for each joint.

4.3. Linearization Simulations

In the dynamic model linearization, the original equation of motion of the UR5 has been linearized by moving linearization. According to the process presented in that section, an algorithm could be also developed to compute the matrices A and B at the different operating points. The purpose of this section is to assess whether the linearized kinetic equations maintain the similarity of the original kinetic equations and to calculate the variation errors induced by the linearization. Furthermore, the linearized system will be converted to transfer functions to obtain more information for the development of the controller.

- **Stability analysis**

Several researchers [21, 33] have suggested that one of the most common methods for assessing the linearization quality of dynamical systems is stability analysis because a stable system can be used as a reference for controller design. Therefore, it is critical to evaluate the stability of the linearized system to ensure that it does not diverge or oscillate unexpectedly during operation.

For the stability analysis, the 50 operating points used in dynamics simulation were entered into the linearization algorithm to calculate *A* matrices. The maximum real part eigenvalues of the *A* matrices were calculated, and Figure 12 illustrates the values of these eigenvalues for the operating points. If the maximum real part eigenvalue is less than zero, the linearized system is considered stable at that point; otherwise, it is unstable. If all the real parts of the eigenvalues are negative, the linearized system is considered asymptotically stable, indicating that the linearization process is effective [1, 14].

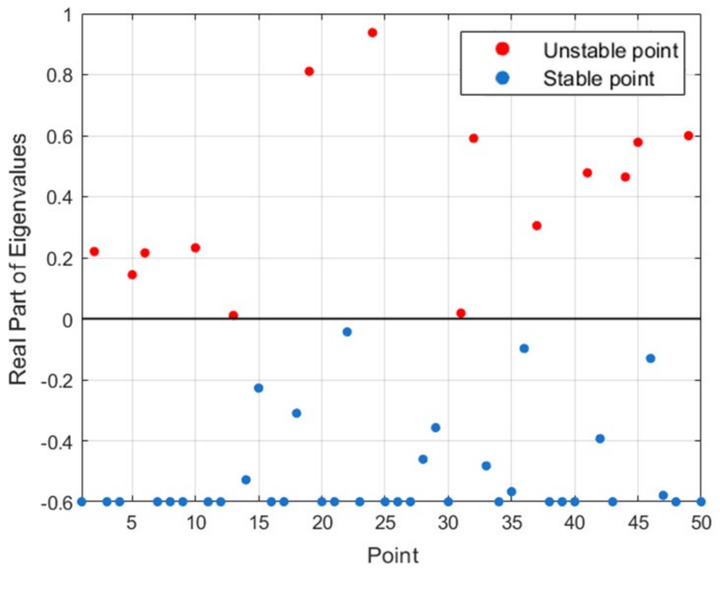

Figure 12. Stability analysis over 50 operating points.

As shown in Figure 12, the real part eigenvalues of 36 of these points were all less than zero, so the linearization could be considered valid for 72% of the operating points. Due to the operating points were chosen randomly, the probability that the linearized kinetic equations are stable over the entire workspace can be assumed to be 72%. In addition, nearly 50% of the real part eigenvalues are around -0.6 at the operating points, which means that nearly half of the operating points in workspace are highly stable after linearization.

- **Absolute error analysis**

Upon stability analysis, joint torques from linearized dynamics equations were compared with those from nonlinear dynamics equations to assess linearization accuracy. The same 50 operating points were used to calculate A and B matrices for the linearization equations. With the same operating points, the torques from linearized eqautions could be also obtained. The absolute errors of the torques for the two equations were calculated and plotted in Figure 13.

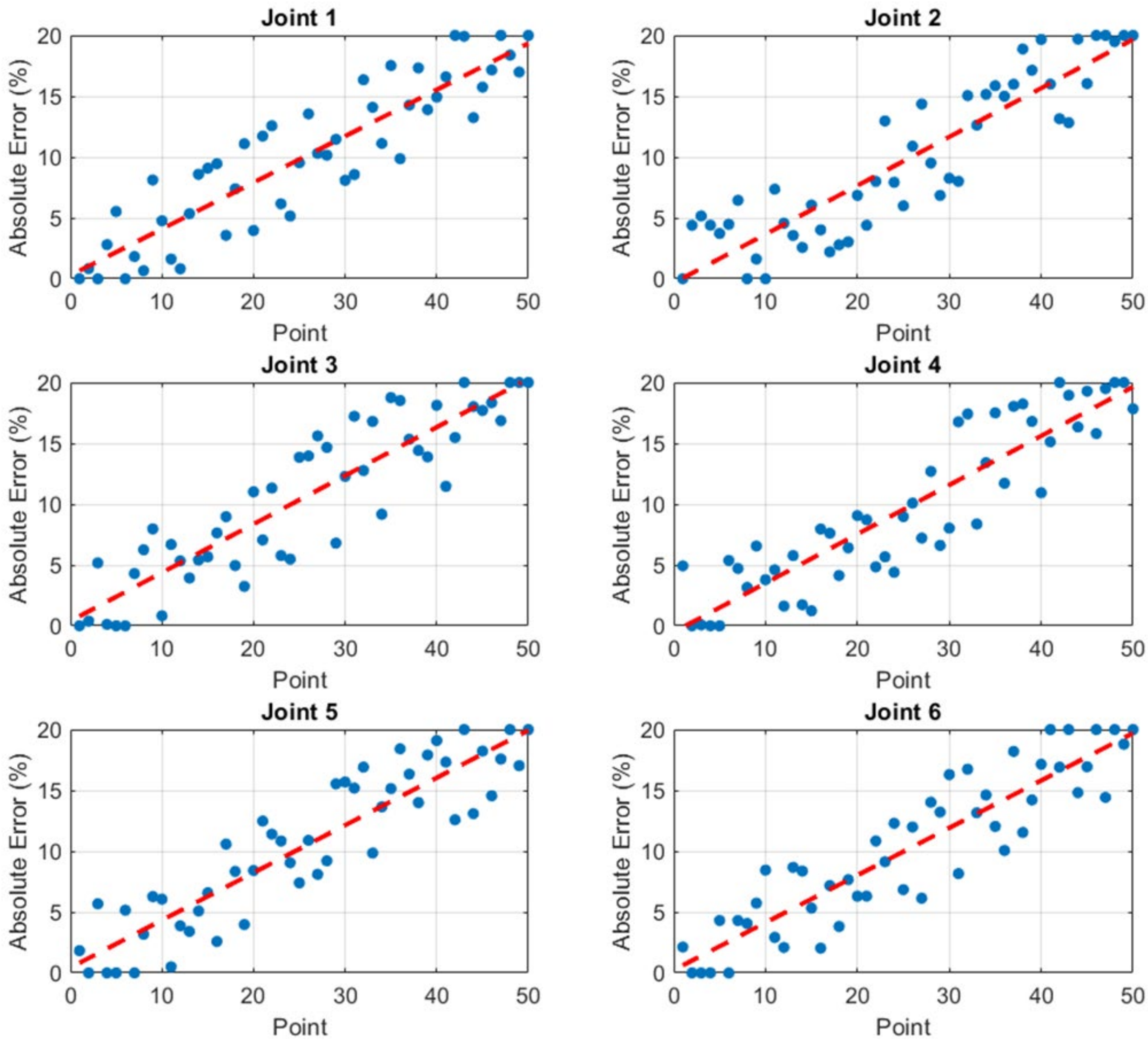

Figure 13. Absolute error between linear and nonlinear equations at 50 operating points.

As can be seen from Figure 13, the absolute torque error of the six joints gradually increased with the increase of operating points and was approximately linear with the number of operating points For the initial 20 random operating points, the error remained at a 10% level; however, upon reaching 40 points, the error escalated to 20%. One potential explanation for this phenomenon may be attributed to the accumulation of previous errors within the moving linearization algorithm. Therefore, it is reasonable to anticipate a continuous growth in absolute error should additional operating points be introducead.

- **Bode plot analysis**

The above simulations evaluate the overall performance of the linearized dynamics system. To further evaluate the feasibility of moving linearization, a comprehensive examination of the entire system's characteristics is necessary. A prevalent method for this is the Bode plot [14, 21, 31], which enables visualization of transfer function stability, resonant frequency, and disturbance robustness.

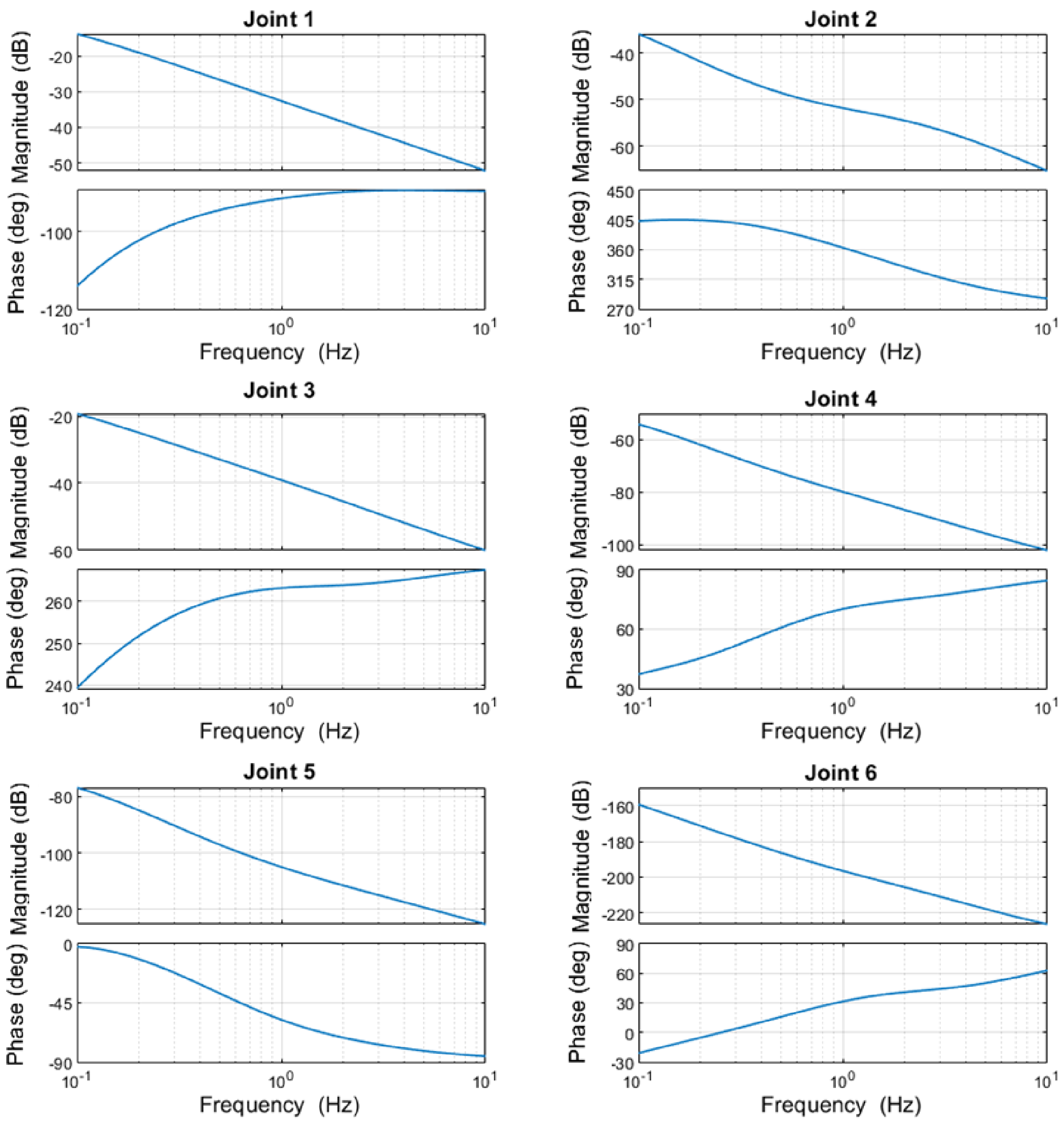

Figure 14. Bode plot of a specific operating point.

For instance, Figure 14 presents the Bode plot of the linearized system at an operating point. Utilizing this plot, researchers can identify suitable control strategies, tuning parameters, and design enhancements. By analyzing Bode plots for all points within the workspace, it becomes possible to determine the success of the entire linearization process. However, as this analysis is beyond the scope of the current project, it will not be addressed in this paper.

5. Experiments

In the previous work, a series of simulations were conducted utilizing the proposed mathematical models for the UR5 manipulator. Nonetheless, under real-world operating conditions, the impact of various factors, including manufacturing tolerances, friction, and environmental elements on the UR5's motion, cannot be disregarded. Therefore, it is essential to carry out an experiment to assess the accuracy and applicability of the developed model in the actual motion of the manipulator.

5.1. Experiment Setup and Procedure

In the experiment, the UR5 was mounted on a bench and programmed to follow a circular trajectory at varying constant speeds. Throughout the experiment, data on joint rotation angles (θ), velocities ($\dot{\theta}$), and motor currents (I) were gathered using sensors integrated into the robot's joints. The experimental setup is shown in Figure 15. The experiment encompassed three distinct speed groups, ranging from 50mm/s to 150mm/s, with increments of 50mm/s per group. The end-effector of the UR5 consistently traced circles with a 100mm radius (identical to the circular trajectory in the kinematics simulation) for each speed group. To ensure consistent data, each speed group is tested three times, meaning that the robot arm will track a circle at the same speed three times.

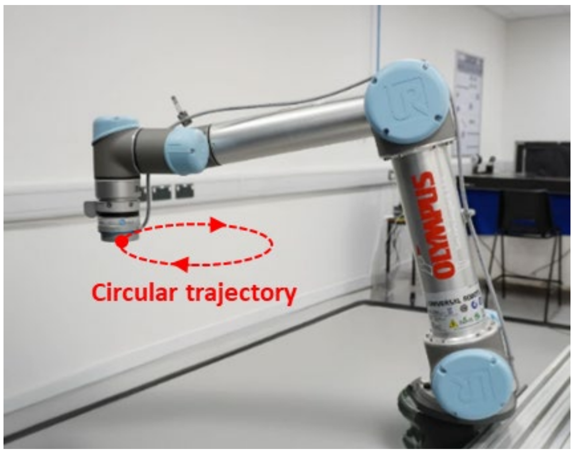

Figure 15. Experimental Setup of the UR5

5.2. Data Processing

According to the manufacturer [28], the sampling rate of the sensors at joints is 8 ms, which means that $\theta, \dot{\theta}$ and I were recorded in every 8ms. This relatively high sampling rate, compared to other robots, ensures accurate and reliable data acquisition of robot status [5, 6]. Figure 16 shows the rotational angles and velocities of the six joints recorded during the experiment as a function of time.

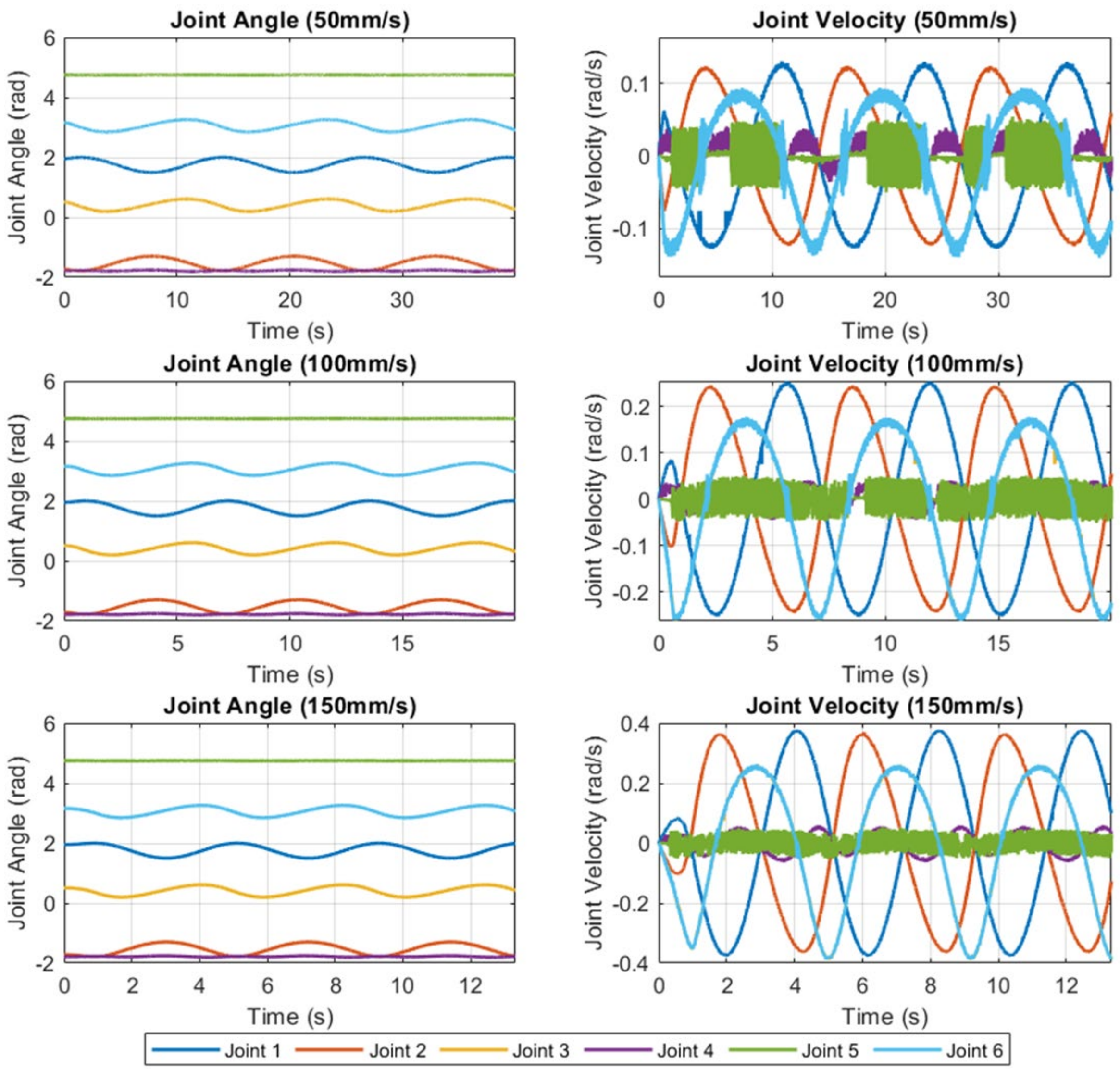

Figure 16. Joint angles and joint velocities over time for three different tracking speeds.

The sensor noise issue has a relatively large impact on joint 4 and joint 5 due to the minimal rotation they exhibit in the experiment. A distinct periodicity can be observed in the position and velocity data for the remaining joints, whose joint angle changes are almost the same for all three velocity groups. As the tracking speed increases, the peak of the joint velocity also increases gradually. In addition to this, when the UR5 tracked the circle at a faster speed, it can be seen that the data obtained for the joint velocities are relatively clean and free of noise. Therefore, in the subsequent analysis, I would use the test group with the most pronounced variation in values and less noise, i.e., 150mm/s.

5.3. Dynamics analysis

Since the UR5 does not have acceleration sensors to directly obtain joint acceleration, Equation (60) was used to calculate acceleration from $t = 0$ by considering rotation angles and rotation speeds. The equation is as follows:

$$\theta(t+\Delta t) = \theta(t) + \dot{\theta}(t)\Delta t + \frac{1}{2}\ddot{\theta}(t)\Delta t^2 \tag{60}$$

Where $\theta(t)$, $\dot{\theta}(t)$ and $\ddot{\theta}(t)$ are the position, velocity, and acceleration at the initial time, $\theta(t+\Delta t)$ is the position at the subsequent time. Δt equals to sampling rate, which is 8ms. The joint torques were then computed using the obtained data with the proposed dynamics algorithm to compare with the measured current. At the same time, a time-shift denoising technique was applied to partially remove the signal noise and make the torque curves smoother.

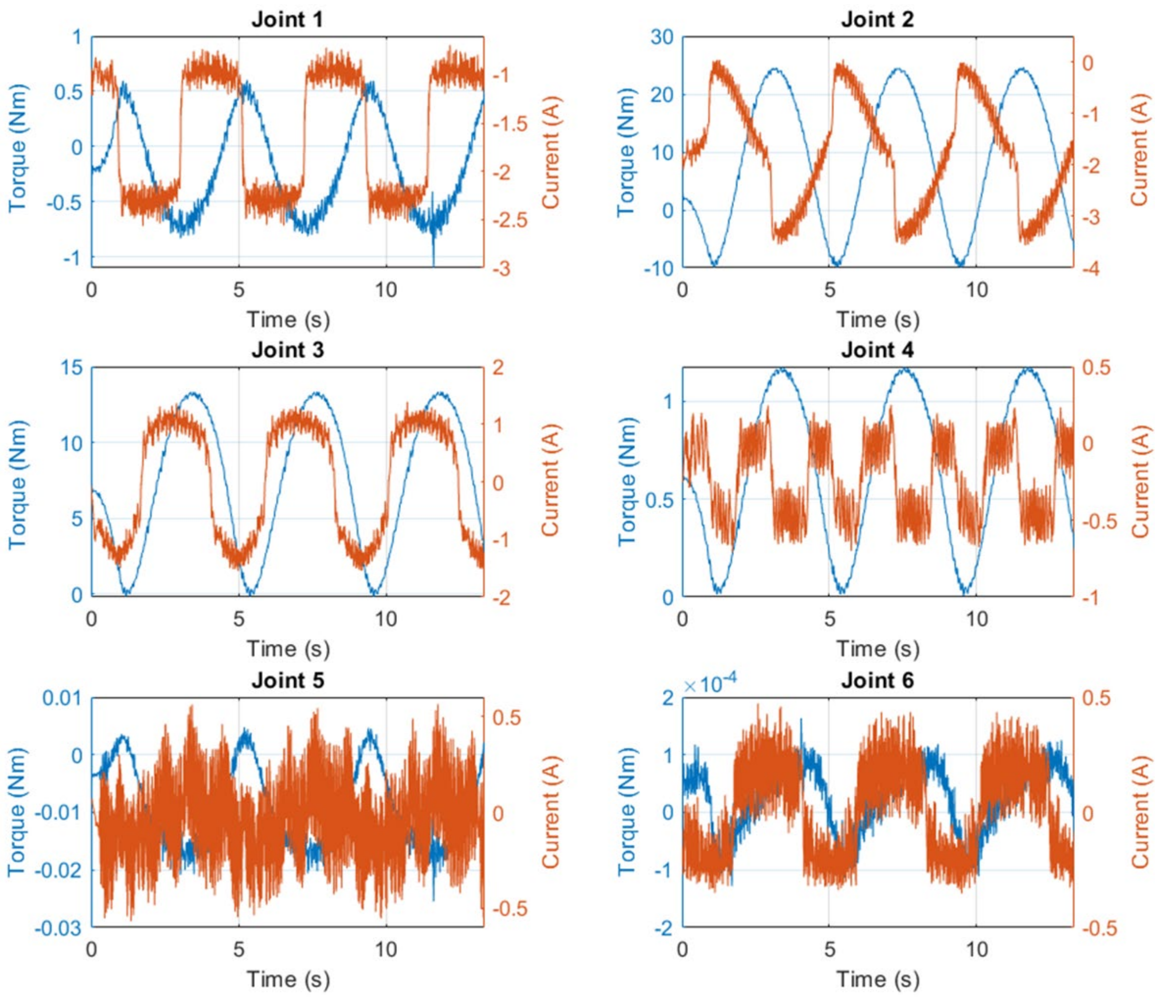

Figure 17. Calculated joint torques and measured motor currents for the 150mm/s group.

Figure 17 displays the calculated joint torques and motor currents over time for 150mm/s. The figure indicates a strong correlation between torque and current changes. Assuming a 1/4 period phase lag relative to the current can be attributed to time delay [4, 21], the joint torque obtained through the closed-form dynamic equation can be considered to vary with current magnitude. However, due to the nonlinear current-torque relationship for UR5's AC motors and varying current-torque equations across different AC motor types [6], it is challenging to establish a definitive function to verify the mathematical relationship between torque and current magnitudes. Nonetheless, this does not preclude the conclusion that the derived dynamic equation can represent the actual motion of the UR5.

5.4. Linearization analysis

To analyze the accuracy and applicability of the moving linearization, a constant current-torque relationship proposed by [6] was applied to obtain joint torques with measured motor currents.

$$\tau = K_p \cdot I \tag{61}$$

Where K_p was determined by averaging the ratio of the maximum torque to the maximum current for each joint as shown in Figure 17, equal to 1.67. After that, every operating point on the 150mm/s group was input into the linearization algorithm to calculate A matrices. The maximum real part eigenvalues of the A matrices were calculated and plotted in Figure 18.

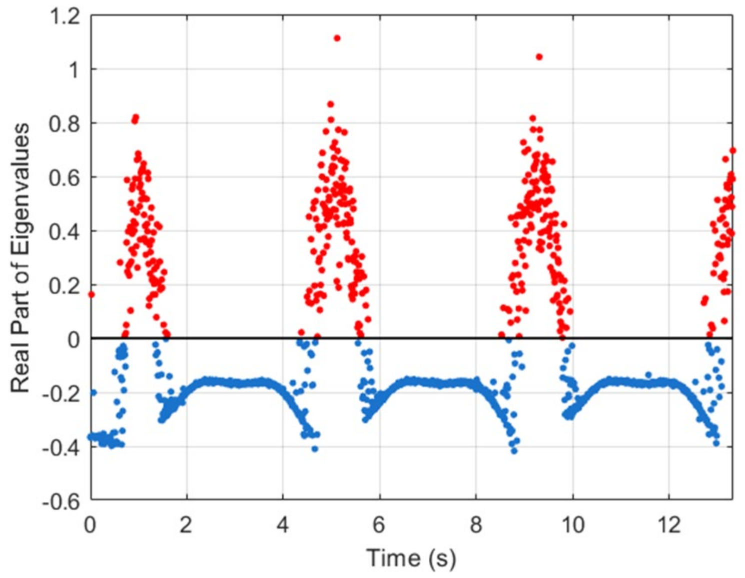

Figure 18. Stability analysis of the 150mm/s group.

The maximum real part eigenvalue of the A matrix was obtained for 67% of the operating points less than zero. Therefore, it could be assumed that the linearization is valid for 67% of the circular trajectory tracking experiment at 150mm/s. In addition to this, the linearized analysis exhibits a similar periodicity to the joint angle and velocity variation in Figure 16. The most favorable linearization outcomes are observed when the joint is in the initial state, starting from a static motion. The linearized system tends to become progressively unstable as the joint velocity increases.

6. Discussion

The analysis and results of the modeling, simulation, and experiment of the UR5 have been described in detail in their respective sections. This section will discuss the applicability and accuracy of the developed mathematical model of the UR5 based on the completed simulations and experiments.

- **Kinematics Discussion**

The proposed kinematic model can primarily be validated through simulation, as the external environment and real-world working conditions have minimal impact on the manipulators' kinematics. During the simulation, workspace analysis and circular trajectory tracking were carried out to validate the model. The results showed no variation between the simulation and proposed kinematic equations in the virtual environment. This demonstrates the effectiveness of the derived kinematic model in representing the UR5 manipulator's forward and inverse motion characteristics throughout the entire workspace. It is expected that the kinematic equations will maintain high performance and accuracy in the actual UR5. However, further investigation is needed to support this expectation.

- **Dynamics Discussion**

Regarding the dynamics model, both simulation and experiment were conducted to evaluate its accuracy, as there are many internal and external factors that may affect the dynamic system of UR5.

In the simulation, 50 arbitrary operating points were input into the derived dynamics equation and the widely accepted dynamics equation to compare their joint torques. The results showed that their joint torques were nearly identical for the first five joints, though some discrepancies were observed in the sixth joint. However, the maximum absolute average error is still below 10%, a typical figure for assessing the accuracy of robot models. Therefore, it should be expected that if friction and other nonlinear effects are ignored, the proposed dynamics model could represent the system of the UR5 to a large extent. Another result of the simulation is that a gradual rise in torque deviation exists from the initial joint to the last joint. The torque error at a joint is approximately equal to the sum of the errors at its preceding joints. The potential reason for this may be that the nonlinear characteristics of the joint become more pronounced as it gets closer to the end, exhibiting error accumulation. Therefore, this effect should occur in all dynamic models and not only in the model proposed in this paper.

As for the circular trajectory tracking experiment, the torques obtained from the dynamic equation were compared with measured motor currents. It should be noted that due to joints 4 and 5 having small rotations during the experiment, the data from these two joints are subject to signal error issues. Therefore, operating points from the high-speed group should be selected to perform validation calculations. Since the current-torque relationship is undetermined, it is not possible to validate the proposed equation through establishing a mathematical relationship between calculated torques and measured current. However, their consistent trend indicates that the dynamics model can at least represent the system of the UR5 analytically.

- **Linearization Discussion**

The moving linearization technique presented in the report was assessed through simulation and experiment. In both simulation and experiment, stability analysis was conducted. The results show that the linear dynamics equations obtained by this linearization method have approximately a 70% probability of behaving as a stable system, both in the whole working space and in a specific range. This figure is relatively high compared to other methods. However, this linearization method starts to become less effective when the operating point has a faster joint velocity. In addition, the simulation also conducted an absolute analysis to compare the linear and nonlinear equations. According to the generated plot, it can be claimed that the variation between the two equations increases when more operating points are involved in the linearization. The possible reason for this could be that only the first-order Taylor series was used for linearization, rather than a higher-order Taylor series. Overall, the moving linearization achieved better stability but has some drawbacks when dealing with a large number of operating points or operating points with high joint velocity.

7. Conclusions

This report presented a set of comprehensive mathematical models for the UR5 manipulator, including kinematics, dynamics, and dynamics linearization. The models were implemented in Matlab and evaluated for accuracy and utility through simulations and experiments. The research outcomes provide a valuable resource for researchers aiming to improve the UR5's performance and develop novel controllers. Our findings contribute to the existing knowledge by offering the most complete mathematical model of the UR5 to date, facilitating further research in advanced control strategies. In conclusion, this research lays a solid foundation for enhancing the UR5 manipulator's performance and broadening its applications across various industries. The mathematical models and Matlab implementation presented in this paper offer valuable tools to advance the research and application of the UR5 robot in various industries.

8. References

[1] J. J. Craig, *Introduction to robotics: mechanics and control*. Pearson Educacion, 2005.

[2] J. Iqbal, R. U. Islam, and H. Khan, "Modeling and analysis of a 6 DOF robotic arm manipulator," *Canadian Journal on Electrical and Electronics Engineering,* vol. 3, no. 6, pp. 300-306, 2012.

[3] S. Wahyuningtri, D. Adzkiya, and H. Nurhadi, "Motion Control Design and Analysis of UR5 Collaborative Robots Using Proportional Integral Derivative (PID) Method," in *2021 International Conference on Advanced Mechatronics, Intelligent Manufacture and Industrial Automation (ICAMIMIA)*, 8-9 Dec. 2021 2021, pp. 157-161, doi: 10.1109/ICAMIMIA54022.2021.9807805.

[4] N. Guoxian and L. Liang, "Workspace Analysis and Dynamics Simulation of Manipulator based on MATLAB," *IOP Conference Series: Materials Science and Engineering,* vol. 825, p. 012001, 05/29 2020, doi: 10.1088/1757-899X/825/1/012001.

[5] S. Robla-Gómez, V. M. Becerra, J. R. Llata, E. González-Sarabia, C. Torre-Ferrero, and J. Pérez-Oria, "Working Together: A Review on Safe Human-Robot Collaboration in Industrial Environments," *IEEE Access,* vol. 5, pp. 26754-26773, 2017, doi: 10.1109/ACCESS.2017.2773127.

[6] K. Kufieta, "Force estimation in robotic manipulators: Modeling, simulation and experiments," *Department of Engineering Cybernetics NTNU Norwegian University of Science and Technology,* 2014.

[7] M. Pollák, M. Kočiško, D. Paulišin, and P. Baron, "Measurement of unidirectional pose accuracy and repeatability of the collaborative robot UR5," *Advances in Mechanical Engineering,* vol. 12, no. 12, p. 1687814020972893, 2020/12/01 2014, doi: 10.1177/1687814020972893.

[8] O. Egeland and J. T. Gravdahl, *Modeling and simulation for automatic control*. Marine Cybernetics Trondheim, Norway, 2002.

[9] P. M. Kebria, S. Al-wais, H. Abdi, and S. Nahavandi, "Kinematic and dynamic modelling of UR5 manipulator," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 9-12 Oct. 2016 2016, pp. 004229-004234, doi: 10.1109/SMC.2016.7844896.

[10] A. Chico *et al.*, "Hand Gesture Recognition and Tracking Control for a Virtual UR5 Robot Manipulator," in *2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM)*, 12-15 Oct. 2021 2021, pp. 1-6, doi: 10.1109/ETCM53643.2021.9590677.

[11] R. R. Serrezuela, A. F. C. Chavarro, M. A. T. Cardozo, A. L. Toquica, and L. F. O. Martinez, "Kinematic modelling of a robotic arm manipulator using Matlab," *ARPN Journal of Engineering and Applied Sciences,* vol. 12, no. 7, pp. 2037-2045, 2017.

[12] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. Wiley New York, 2006.

[13] P. I. Corke and O. Khatib, *Robotics, vision and control: fundamental algorithms in MATLAB*. Springer, 2011.

[14] B. Siciliano, O. Khatib, and T. Kröger, *Springer handbook of robotics*. Springer, 2008.

[15] R. S. Andersen, "Kinematics of a UR5," *Aalborg University,* 2018.

[16] T. Sugihara, "Solvability-unconcerned inverse kinematics by the Levenberg–Marquardt method," *IEEE transactions on robotics,* vol. 27, no. 5, pp. 984-991, 2011.

[17] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.

[18] J. Wittenburg, *Dynamics of systems of rigid bodies*. Springer-Verlag, 2013.

[19] H. Ayonga, "Fast Robot Optimization and Simulation Toolkit (FROST," ed.

[20] R. Tedrake, "Drake: Model-based design and verification for robotics," *URL https://drake. mit. edu,* 2019.
[21] H. Høifødt, "Dynamic modeling and simulation of robot manipulators," *Master of Science in Engineering Cybernetics, Norwegian University of Science and Technology,* 2011.
[22] W. Khalil and E. Dombre, *Modeling identification and control of robots*. CRC Press, 2002.
[23] A. De Luca and G. Oriolo, "Local incremental planning for nonholonomic mobile robots," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, 1994: IEEE, pp. 104-110.
[24] J.-J. E. Slotine and W. Li, *Applied nonlinear control* (no. 1). Prentice hall Englewood Cliffs, NJ, 1991.
[25] M. W. Spong and M. Vidyasagar, *Robot dynamics and control*. John Wiley & Sons, 2008.
[26] A. Marcos and G. J. Balas, "Development of linear-parameter-varying models for aircraft," *Journal of Guidance, Control, and Dynamics,* vol. 27, no. 2, pp. 218-228, 2004.
[27] P. Apkarian and R. J. Adams, "Advanced gain-scheduling techniques for uncertain systems," in *Advances in linear matrix inequality methods in control*: SIAM, 2000, pp. 209-228.
[28] U. Robots, "Ur5-user manual," ed: English (cit. on p. 49), 2013.
[29] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer.
[30] M. R. P. Ragazzon, "Robot manipulator collision handling in unknown environment without using external sensors," *Department of Engineering Cybernetics Norwegian University of Science and Technology,* 2012.
[31] Z. Bubnicki, *Modern control theory*. Springer, 2005.
[32] M. Giftthaler, M. Neunert, M. Stäuble, M. Frigerio, C. Semini, and J. Buchli, "Automatic differentiation of rigid body dynamics for optimal control and estimation," *Advanced Robotics,* vol. 31, no. 22, pp. 1225-1237, 2017.
[33] A. De Luca, "Decoupling and feedback linearization of robots with mixed rigid/elastic joints," *International Journal of Robust and Nonlinear Control: IFAC - Affiliated Journal,* vol. 8, no. 11, pp. 965-977, 1998.