

Article

Algorithmic Modified Denavit–Hartenberg Modeling for Robotic Manipulators Using Line Geometry

Minchang Sung ¹ and Youngjin Choi ^{2,*}

¹ Department of Electrical and Electronic Engineering, Hanyang University, Ansan 15588, Republic of Korea; wdrac331@hanyang.ac.kr

² Department of Robotics, Hanyang University, Ansan 15588, Republic of Korea

* Correspondence: cyj@hanyang.ac.kr

Abstract: This paper proposes a kinematic modeling method for robotic manipulators by extracting the modified Denavit–Hartenberg (MDH) parameters using line geometry. For single-branched manipulators, various joint axes can be represented as lines using Plücker coordinates. The forward kinematics is derived by performing the product of matrices which are the exponential maps lifted from two kinds of exponential coordinates using the MDH parameters. For extracting MDH parameters, line geometry systematically analyzes the following: (1) the closest point between a point and line, (2) the closest distance and twist angle between two lines, (3) the common perpendicular line and its intersection points, and (4) classifies line relationships into collinear, distant parallel, intersected, and skewed cases. For each case, five parameters including twist angle, closest distance, common perpendicular direction vector, and both feet on a common perpendicular line are sequentially computed as results of the line geometry block. Finally, the aforementioned line geometry blocks are utilized to extract the four MDH parameters according to their definitions. The effectiveness of the proposed algorithm is verified by four examples including a typical Selective Compliance Assembly Robot Arm (SCARA) robot and three different commercial manipulators.

Keywords: line geometry; modified Denavit–Hartenberg (MDH) parameters; screw axis; kinematic modeling



Academic Editor: Douglas O’Shaughnessy

Received: 27 March 2025

Revised: 25 April 2025

Accepted: 28 April 2025

Published: 30 April 2025

Citation: Sung, M.; Choi, Y. Algorithmic Modified Denavit–Hartenberg Modeling for Robotic Manipulators Using Line Geometry. *Appl. Sci.* **2025**, *15*, 4999. <https://doi.org/10.3390/app15094999>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Kinematic modeling of articulated robots is commonly categorized into two approaches according to the number of parameters required to describe the relative motion between adjacent joint axes: screw-based and line-based formulations. The screw-based formulation employs six parameters, whereas the line-based counterpart needs four. Owing to its theoretical completeness, the screw framework has been widely adopted in robotics research [1–4]. In a screw-based model, the transformation of one link with respect to its predecessor is represented by an exponential map constructed from six exponential coordinates—three rotational and three linear components. Multiplying these exponential maps sequentially produces the forward kinematics of a single-branched manipulator and extends naturally to branched or closed-loop mechanisms.

When each joint axis is regarded as an infinite line, only four independent quantities—Plücker coordinates—are sufficient to describe its pose, because a line is invariant under self-rotation and self-translation. Consequently, line-based formulations require four parameters per axis. Although screw coordinates offer superior numerical robustness for calibration, the modified Denavit–Hartenberg (MDH) notation remains the interface in

industrial robot controllers. Direct identification in the MDH, however, often suffers from convergence issues caused by the coupling inherent in its parameter structure [5]. To alleviate this limitation, the present study proposes a post-processing algorithm that converts screw-identified parameters into the MDH format. A concise comparison among the standard Product-of-Exponential (PoE) form, MDH notation, and the classical DH convention—including differences in parameter count, transformation dependencies, frame consistency, and automation suitability—is provided in Table A1 in Appendix A.

Among traditional kinematic conventions, the Denavit–Hartenberg (DH) notation remains the most widely adopted. Originally introduced by Denavit and Hartenberg, it expresses the relative transformation between successive frames with four scalar parameters [6,7]. Two variants have since emerged: the distal form [8] and the proximal form—also called the Modified DH (MDH) notation—[1,9,10]. This work adopts the proximal form because every parameter is referenced to the same coordinate frame, which simplifies implementation and numerical treatment [11–14]. Deriving MDH parameters, however, requires locating the common perpendicular between consecutive joint axes; line geometry therefore becomes an essential computational tool in our framework.

The studies most closely related to this work are those of Faria et al. [15] and Liu et al. [16]. Faria et al. extract line parameters by commanding axis-specific rotations and subsequently convert them to DH parameters, but their procedure requires user-defined joint-excitation trajectories for each axis during data acquisition and manual assignment of each frame's x -axis. Liu et al. automate the extraction of conventional DH parameters, yet their method handles only revolute joints and is incompatible with the MDH formulation. The algorithm proposed in the present study overcomes all of these limitations: it requires no constrained excitation, supports both revolute and prismatic joints, and outputs parameters directly in the MDH notation.

Recent studies [17–20] have focused on kinematic parameter identification based on screw theory, typically performed with a robot equipped with sensors attached to a TCF for calibration. Although direct identification is not performed in this study, the proposed method can be subsequently applied to convert the identified screw-based parameters into an MDH representation. This allows for seamless integration with MDH-based kinematic modeling frameworks commonly used in industrial and simulation environments.

The paper is organized as follows: Section 2 introduces the exponential map using MDH parameters, the exponential coordinates using MDH, the joint twist, and the joint matrix. Section 3 analyzes the relationships between line and point and between two lines, finds the common perpendicular line and its feet, and finally lists up the important contents into the line geometry block. Section 4 proposes how to extract the MDH parameters using two consecutive line geometry blocks. Several examples are employed to verify the proposed method in Section 5, and finally we draw conclusions.

2. Exponential Map Using MDH

Since a joint connects the adjacent links, a single branched articulated robotic manipulator having N joints has $N + 1$ links and an adjacent transformation is defined between a pair of adjacent links or joints. This adjacent transformation, a homogeneous transformation matrix, is described in terms of four MDH parameters. The last frame of the robot is called the tool coordinate frame (TCF) where robotic tasks are assigned. The forward kinematics of the robotic manipulator is expressed as:

$${}^0T_{tcf} = {}^0T_1(\theta_1, d_1){}^1T_2(\theta_2, d_2) \cdots {}^{N-1}T_N(\theta_N, d_N){}^NT_{tcf},$$

where θ_i and d_i denote i th generalized joint variables which can be one or two degrees-of-freedom (dof) according to the joint type, the prefixed superscript and postfix subscript

of the homogeneous transformation stand for the reference frame and target frame, respectively, e.g., ${}^N T_{tcf}$ represents the constant adjacent transformation from frame $\{N\}$ to $\{tcf\}$.

2.1. MDH Parameters

The MDH notation describes the relationship between adjacent joint axes at zero-configuration which represents when every joint variable is zero. To define the four MDH parameters, the coordinate frames are determined according to the following procedures:

1. Set the \hat{z} -axis in the direction of the joint axis.
2. Define the \hat{x} -axis as the common perpendicular direction to the adjacent joint axes. In cases where adjacent axes are collinear, the previous \hat{x} -axis is retained as the current \hat{x} -axis.
3. Determine the \hat{y} -axis using the right-hand rule.

Figure 1 illustrates how to set up two coordinate frames $\{i - 1\}$ and $\{i\}$ from three consecutive joint axes $i - 1, i$ and $i + 1$. After setting up both coordinate frames, four MDH parameters are defined as follows:

1. Link twist (α_i): this parameter represents the angle from \hat{z}_{i-1} to \hat{z}_i about the \hat{x}_{i-1} -axis
2. Link length (a_i): this denotes a distance from \hat{z}_{i-1} to \hat{z}_i along the \hat{x}_{i-1} -axis
3. Joint offset distance (d_{i0}): this is the distance from \hat{x}_{i-1} to \hat{x}_i along the \hat{z}_i -axis
4. Joint offset angle (θ_{i0}): this expresses the angle from \hat{x}_{i-1} to \hat{x}_i about the \hat{z}_i -axis.

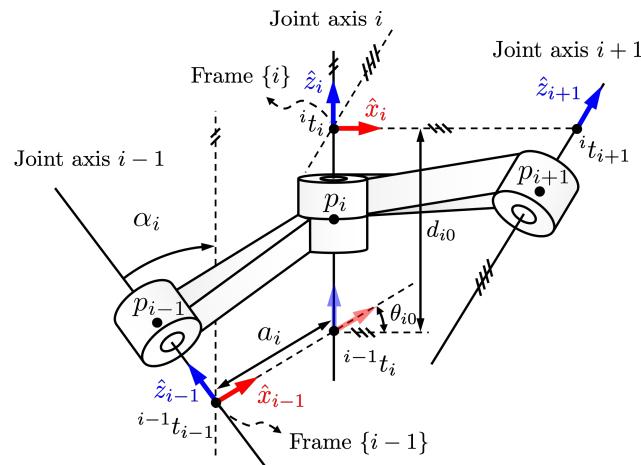


Figure 1. Modified Denavit–Hartenberg (MDH) parameters, where p_{i-1} denotes the point on the $(i - 1)$ th joint axis, the \hat{z}_{i-1} -axis is set along the $(i - 1)$ th joint axis, the \hat{x}_{i-1} -axis is determined to be a common perpendicular direction from the \hat{z}_{i-1} -axis to the \hat{z}_i -axis, and the origin of the frame $\{i - 1\}$ denoted by $i^{-1}t_{i-1}$ is an intersection point of the \hat{x}_{i-1} -axis and the \hat{z}_{i-1} -axis. Also, four MDH parameters are denoted as α_i , a_i , θ_{i0} , and d_{i0} .

It is noted that three consecutive joint axes are required to define four MDH parameters, which are used to obtain the adjacent transformation between the frame $\{i - 1\}$ and the frame $\{i\}$. Figure 1 shows a typical example of MDH parameters. Note that the terms “joint offset distance” and “joint offset angle” correspond to the standard MDH parameters d_i and θ_i , respectively. The subscript “0” is introduced to distinguish between constant geometric parameters and variable joint inputs (e.g., $q_{i,\theta}$, $q_{i,d}$) and to clarify their role in generating the transformation matrix. Using the four constant MDH parameters, the adjacent transformation can be represented by adding joint variables $q_{i,\theta}$ for the revolute joint type and $q_{i,d}$ for the prismatic joint type. For this purpose, i th generalized joint variables θ_i and d_i are defined differently according to the joint types as follows: where h implies a pitch to denote a moving distance along the joint axis per one revolution for the

helical joint. It is noted that three consecutive joint axes are required to define four MDH parameters, which are used to obtain the adjacent transformation between the frame $i - 1$ and the frame i , as summarized in Table 1.

Table 1. The generalized joint variables with MDH parameters.

Joint Type	Generalized Joint Variables (θ_i, d_i)
Revolute (R) one-dof	$\theta_i \leftarrow \theta_{i0} + q_{i,\theta}$ $d_i \leftarrow d_{i0}$
Prismatic (P) one-dof	$\theta_i \leftarrow \theta_{i0}$ $d_i \leftarrow d_{i0} + q_{i,d}$
Helical (H) one-dof	$\theta_i \leftarrow \theta_{i0} + q_{i,\theta}$ $d_i \leftarrow d_{i0} + hq_{i,\theta}$
Cylindrical (RP) two-dof	$\theta_i \leftarrow \theta_{i0} + q_{i,\theta}$ $d_i \leftarrow d_{i0} + q_{i,d}$

2.2. Exponential Map Using MDH

In this work, Lie group methods are employed to establish a connection between the Modified Denavit–Hartenberg (MDH) parameters and Lie algebra in the context of robot forward kinematics. For readers unfamiliar with the fundamentals of Lie groups and Lie algebras in robotic modeling, we recommend [3,4] as accessible introductions. For a more advanced treatment of Lie group integration and numerical methods, refer to [21]. In addition, screw theory, which underpins many geometric interpretations used here, is thoroughly discussed in [2]. The adjacent transformation from the coordinate frame $\{i - 1\}$ to the frame $\{i\}$ in Figure 1 can be decomposed into two exponential maps because the link twist α_i and the link length a_i are represented with respect to the \hat{x}_{i-1} -axis and the generalized joint variables d_i and θ_i with respect to the \hat{z}_i axis as follows:

$${}^{i-1}T_i(\theta_i, d_i) = \text{expm}([\lambda_{x_{i-1}}])\text{expm}([\lambda_{z_i}]), \quad (1)$$

where the matrix exponential, belonging to the Lie group $SE(3)$, is defined by using the matrix $[\lambda]$, belonging to Lie algebra $\mathfrak{se}(3)$, as follows:

$$\text{expm}([\lambda]) = \sum_{k=0}^{\infty} \frac{[\lambda]^k}{k!} \quad (2a)$$

$$[\lambda_{x_{i-1}}] = \left[\begin{array}{ccc|c} 0 & 0 & 0 & a_i \\ 0 & 0 & -\alpha_i & 0 \\ 0 & \alpha_i & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \end{array} \right] \quad (2b)$$

$$[\lambda_{z_i}] = \left[\begin{array}{ccc|c} 0 & -\theta_i & 0 & 0 \\ \theta_i & 0 & 0 & 0 \\ 0 & 0 & 0 & d_i \\ \hline 0 & 0 & 0 & 0 \end{array} \right]. \quad (2c)$$

In other words, $[\lambda]$ implies Lie algebra $\mathfrak{se}(3)$ and $\text{expm}([\lambda])$ is a Lie group $SE(3)$ lifted from the Lie algebra $[\lambda]$, where $\lambda \in \mathbb{R}^6$ means the exponential coordinate vector. The series expansion of $\text{expm}([\lambda])$ using Equation (2a) results in:

$$\expm([\lambda_{x_{i-1}}]) = \left[\begin{array}{ccc|c} 1 & 0 & 0 & a_i \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad (3a)$$

$$\expm([\lambda_{z_i}]) = \left[\begin{array}{ccc|c} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ \hline 0 & 0 & 0 & 1 \end{array} \right]. \quad (3b)$$

Since Lie algebra $\mathfrak{se}(3)$ is isomorphic to a six-dimensional real vector \mathbb{R}^6 , we can use it interchangeably

$$\lambda = \begin{bmatrix} \eta \\ \xi \end{bmatrix} \in \mathbb{R}^6 \leftrightarrow [\lambda] = \left[\begin{array}{ccc|c} 0 & -\xi_z & \xi_y & \eta_x \\ \xi_z & 0 & -\xi_x & \eta_y \\ -\xi_y & \xi_x & 0 & \eta_z \\ \hline 0 & 0 & 0 & 0 \end{array} \right] \in \mathfrak{se}(3),$$

where $\eta = [\eta_x, \eta_y, \eta_z]^T \in \mathbb{R}^3$ and $\xi = [\xi_x, \xi_y, \xi_z]^T \in \mathbb{R}^3$ imply the linear and angular components of the exponential coordinate vector $\lambda \in \mathbb{R}^6$, respectively. Thus, we can find the exponential coordinates of MDH from Equations (2b) and (2c), respectively, as follows:

$$\lambda_{x_{i-1}} = \begin{bmatrix} a_i \\ 0 \\ 0 \\ \frac{0}{\alpha_i} \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad \lambda_{z_i} = \begin{bmatrix} 0 \\ 0 \\ d_i \\ 0 \\ 0 \\ \theta_i \end{bmatrix}. \quad (4)$$

2.3. Joint Twist Using MDH

Since the exponential map (or adjacent transformation) of Equation (1) using MDH is obtained by the matrix product of Equations (3a) and (3b), the joint twist is derived from the left multiplication rule [21,22] as follows:

$$[\dot{i}^{-1}V_i] = i^{-1}T_i^{-1}(\theta_i, d_i)^{i-1}\dot{T}_i(\theta_i, d_i) \quad (5a)$$

$$= [\dot{\lambda}_{z_i}] \quad (5b)$$

$$\therefore i^{-1}V_i = \dot{\lambda}_{z_i}, \quad (5c)$$

where $i^{-1}V_i = \dot{\lambda}_{z_i} \in \mathbb{R}^6$ implies the i th joint twist relative to the frame $\{i-1\}$ represented in the frame $\{i\}$. It is noted that i th joint twist $i^{-1}V_i$ is represented in the frame $\{i\}$. Now, we can define i th joint matrix E_i by separating the time-derivative of joint variable \dot{q}_i from the joint twist as follows:

$$i^{-1}V_i = \dot{\lambda}_{z_i} := E_i\dot{q}_i, \quad (6)$$

where $\dot{\lambda}_{z_i} = [0, 0, \dot{d}_i, 0, 0, \dot{\theta}_i]^T = [0, 0, \dot{q}_{i,d}, 0, 0, \dot{q}_{i,\theta}]^T$ because the joint offset distance d_{i0} and the joint offset angle θ_{i0} as MDH parameters are constants. For given four joint types $\{R, P, H, RP\}$, each joint has its own joint matrix E_i and time-derivative of joint variable $\dot{q}_i = \{\dot{q}_{i,\theta}, \dot{q}_{i,d}\}$ as follows:

$$\begin{aligned}
R : E_i \dot{q}_i &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \dot{q}_{i,\theta} & P : E_i \dot{q}_i &= \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \dot{q}_{i,d} \\
H : E_i \dot{q}_i &= \begin{bmatrix} 0 \\ 0 \\ \frac{h}{0} \\ 0 \\ 0 \\ 1 \end{bmatrix} \dot{q}_{i,\theta} & RP : E_i \dot{q}_i &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_{i,\theta} \\ \dot{q}_{i,d} \end{bmatrix}.
\end{aligned} \tag{7}$$

The joint matrix E_i represented in the frame $\{i\}$ can be transformed into either the base frame $\{0\}$ or the tool coordinate frame $\{tcf\}$; for this purpose, the adjoint transformation is introduced as follows:

$${}^0E_i = {}^0\text{Ad}_i E_i \quad \text{and} \quad {}^{tcf}E_i = {}^{tcf}\text{Ad}_i E_i, \tag{8}$$

where 0E_i and ${}^{tcf}E_i$ are called spatial screw and body screw in [4], respectively, and the adjoint transformation is defined by

$${}^a\text{Ad}_b = \begin{bmatrix} {}^aR_b & [{}^a r_b] {}^aR_b \\ 0_3 & {}^aR_b \end{bmatrix} \in \mathbb{R}^{6 \times 6}, \tag{9}$$

using the homogeneous transformation ${}^aT_b({}^aR_b, {}^a r_b)$, in which ${}^aR_b \in SO(3)$ implies a three-dimensional rotation matrix, $[{}^a r_b]$ represents three-dimensional skew-symmetric matrix obtained from a three-dimensional displacement vector ${}^a r_b \in \mathbb{R}^3$, and 0_3 implies three-dimensional zero matrix. Till now, we have revealed how the MDH parameters influence the exponential map, the exponential coordinate vector, the joint twist, and the joint matrix. The next section will discuss the relationships between the joint axes using the line geometry, ultimately, to find the MDH parameters.

3. Line Geometry

A line $L(\hat{z}, v)$ can be represented by a set of three-dimensional direction vector \hat{z} and three-dimensional moment vector v , where \hat{z} and v are called the Plücker coordinate after Julius Plücker [23,24]. The Plücker coordinate to express the line has four independent elements because the line has two constraints that it remains the same line when rotated about itself and translated in its direction [25]. To improve readability and reduce the complexity of the main text, detailed derivations of the equations are provided in Appendix B.

Here, $\hat{z} \in \mathbb{R}^3$ is a unit vector to denote any direction in a three-dimensional space and $v \in \mathbb{R}^3$ is the moment of the line described from an arbitrary point p on the line represented in the frame $\{0\}$ as shown in Figure 2, which can be expressed as follows:

$$v = p \times \hat{z}. \tag{10}$$

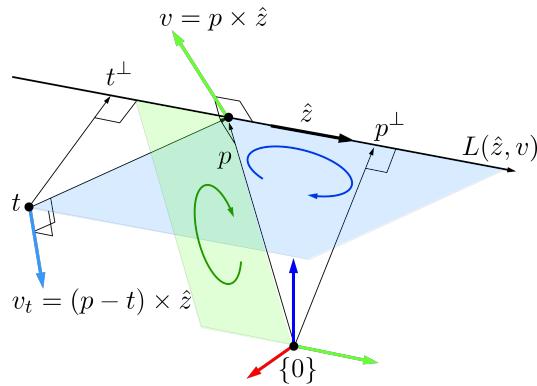


Figure 2. Relationships between a line $L(\hat{z}, v)$ and a point p on the line, and between the line and a point t out of the line in three-dimensional space.

3.1. Relationships Between Line and Points

Since the moment vector v is orthogonal to \hat{z} by Equation (10), it provides information about the closest point from the origin of frame $\{0\}$ as shown in Figure 2. Namely, the closest point p^\perp on the line L from the origin of frame $\{0\}$ is computed by

$$p^\perp = \hat{z} \times v, \quad (11)$$

where the closest distance from the origin of frame $\{0\}$ to the line L is $\|p^\perp\| = \|\hat{z} \times v\|$. Furthermore, the moment v_t of the line L about an arbitrary point t out of the line in Figure 2 is defined by

$$v_t = v - t \times \hat{z}, \quad (12)$$

and the closest point t^\perp on the line L from a point t out of the line in Figure 2 is computed by

$$t^\perp = t + \hat{z} \times v_t, \quad (13)$$

where it is represented in the frame $\{0\}$, and the closest distance from an arbitrary point t out of the line to the line L is $\|t - t^\perp\| = \|\hat{z} \times v_t\|$. It is interesting that the closest point t^\perp on the line from a point t out of the line is obtained by taking the cross-product of the direction vector \hat{z} and the moment v_t .

3.2. Relationships Between Two Lines

Another important property of the Plücker coordinate is the reciprocal product between two lines. When two lines are denoted as $L_1(\hat{z}_1, v_1 = p_1 \times \hat{z}_1)$ and $L_2(\hat{z}_2, v_2 = p_2 \times \hat{z}_2)$, as shown in Figure 3, it is defined as follows:

$$L_1 * L_2 := \hat{z}_1 \cdot v_2 + \hat{z}_2 \cdot v_1, \quad (14)$$

where the reciprocal product implies the moment of line L_1 about line L_2 or the moment of L_2 about L_1 . It is interesting that the reciprocal product is independent of the location of p_1 on L_1 or p_2 on L_2 , and it is only dependent on the direction vectors \hat{z}_1 and \hat{z}_2 . In addition, the reciprocal product is scalar, not a vector. Let us denote the moment v_{21} of the line $L_2(\hat{z}_2, v_2)$ about the point t_1 using Equation (12):

$$v_{21} = v_2 - t_1 \times \hat{z}_2 = (t_2 - t_1) \times \hat{z}_2, \quad (15)$$

where t_1 and t_2 are feet of the common perpendicular line to the two lines L_1 and L_2 as shown in Figure 3.

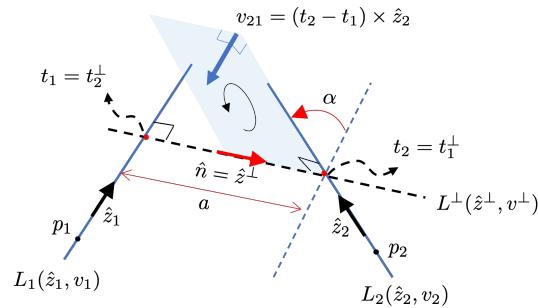


Figure 3. Relationships between two lines $L_1(\hat{z}_1, v_1 = p_1 \times \hat{z}_1)$ and $L_2(\hat{z}_2, v_2 = p_2 \times \hat{z}_2)$, where a is the distance between two lines, t_1 and t_2 are feet of the common perpendicular line $L^\perp(\hat{z}^\perp, v^\perp)$ to two lines, and $\hat{n} = \hat{z}^\perp$ is the direction vector from t_1 to t_2 .

Let us define the plane Π containing \hat{z}_2 and v_{21} as shown in Figure 4, which is orthogonal to the common perpendicular line and parallel to \hat{z}_1 . Consequently, the plane Π contains \hat{z}_1 as shown in Figure 4. Using the fact that the link twist α is the angle of rotation from \hat{z}_1 to \hat{z}_2 about the common perpendicular line, the reciprocal product can be recomputed as follows:

$$L_1 * L_2 = -\|v_{21}\| \sin \alpha, \quad (16)$$

where $\|v_{21}\|$ implies the distance between L_1 and L_2 . Thus, it provides the information on the link length a in Figure 1 as follows:

$$a = \|v_{21}\| = -\frac{L_1 * L_2}{\sin \alpha} = -\frac{\hat{z}_1 \cdot v_2 + \hat{z}_2 \cdot v_1}{\|\hat{z}_1 \times \hat{z}_2\|}. \quad (17)$$

As a matter of fact, the reciprocal product becomes zero either when the two lines are intersected ($a = 0$) or when the two lines are parallel ($\alpha = 0$), where the link twist α in Figure 1 is defined by the angle of rotation from \hat{z}_1 to \hat{z}_2 about the common perpendicular line, which is calculated as follows:

$$\alpha = \text{atan2}(\sin \alpha, \cos \alpha) = \text{atan2}(\|\hat{z}_1 \times \hat{z}_2\|, \hat{z}_1 \cdot \hat{z}_2), \quad (18)$$

where it is effective for a positive range of $0 \leq \alpha \leq \pi$. To cover its negative range of $-\pi \leq \alpha \leq 0$, the sign variable denoted by s_α will be considered later.

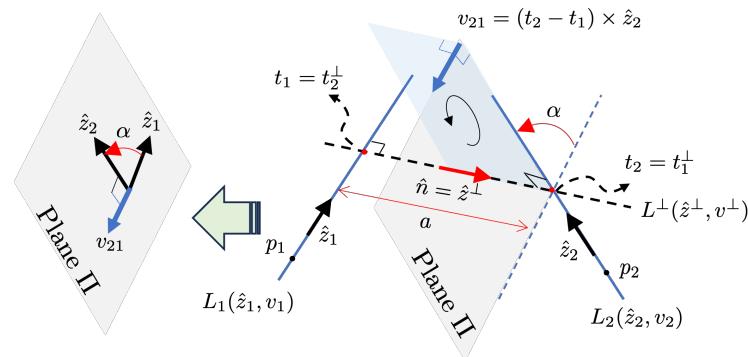


Figure 4. The plane Π contains \hat{z}_2 and v_{21} , which is orthogonal to the common perpendicular line and parallel to \hat{z}_1 ; consequently, it contains \hat{z}_1 as well, where α implies the twist angle of rotation from \hat{z}_1 to \hat{z}_2 about the common perpendicular line, and \hat{z}_2 is perpendicular to v_{21} .

3.3. Common Perpendicular Line

Let us find the common perpendicular line denoted by $L^\perp(\hat{z}^\perp, v^\perp)$ in Figure 4. Then, its direction vector \hat{z}^\perp emanating from t_1 is obtained as follows:

$$\hat{z}^\perp = \hat{n} = \frac{\hat{z}_1 \times \hat{z}_2}{\|\hat{z}_1 \times \hat{z}_2\|}, \quad (19)$$

in fact, it is corresponding to the \hat{x} -axis required to express the MDH parameters in Figure 1, and the moment vector v^\perp is computed by:

$$v^\perp = t_2 \times \hat{z}^\perp. \quad (20)$$

Since we have obtained the common perpendicular line, we are ready to derive two intersection points t_1 and t_2 in Figure 4, which are called feet of common perpendicular line, where the former is obtained as an intersection between L_1 and L^\perp and the latter between L_2 and L^\perp .

3.4. Feet of Common Perpendicular Line

The aforementioned intersection points (or feet) t_1 and t_2 in Figure 4 can be found through the following procedures. In fact, the related proof can be found in [25] as well. To begin with, let us set up a coordinate frame $\hat{z}_1\hat{u}\hat{n}$ at the origin $\{0\}$ as shown in Figure 5, which is composed of three perpendicular unit vectors \hat{z}_1 , $\hat{n} = \frac{\hat{z}_1 \times \hat{z}_2}{\|\hat{z}_1 \times \hat{z}_2\|}$, and $\hat{u} = \hat{n} \times \hat{z}_1$. The basic idea in [25] is to project the point t_2 onto the coordinate frames $\hat{z}_1\hat{u}\hat{n}$ in Figure 5 as follows:

$$t_2 = (t_2 \cdot \hat{z}_1)\hat{z}_1 + (t_2 \cdot \hat{u})\hat{u} + (t_2 \cdot \hat{n})\hat{n}. \quad (21)$$

For the third term $(t_2 \cdot \hat{n})$ of the right-hand side of Equation (21), we can observe the following:

$$t_2 \cdot \hat{n} = -\frac{v_2 \cdot \hat{z}_1}{\|\hat{z}_1 \times \hat{z}_2\|}. \quad (22)$$

For the second term $(t_2 \cdot \hat{u})$, the definitions of \hat{u} and \hat{n} are used as follows:

$$t_2 \cdot \hat{u} = -v_1 \cdot \hat{n}. \quad (23)$$

For the first term $(t_2 \cdot \hat{z}_1)$, the direction vector \hat{z}_2 lies in the plane spanned by \hat{z}_1 and \hat{u} as shown in Figure 5, such that

$$\begin{aligned} \hat{z}_2 &= (\cos \alpha)\hat{z}_1 + (\sin \alpha)\hat{u} \\ &= (\hat{z}_1 \cdot \hat{z}_2)\hat{z}_1 + \|\hat{z}_1 \times \hat{z}_2\|\hat{u}. \end{aligned}$$

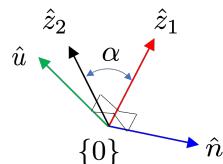


Figure 5. $\hat{z}_1\hat{u}\hat{n}$ coordinate frame, where \hat{z}_2 lies in the $\hat{z}_1\hat{u}$ plane, and thus $\sin \alpha = \|\hat{z}_1 \times \hat{z}_2\|$ and $\cos \alpha = \hat{z}_1 \cdot \hat{z}_2$ for $0 \leq \alpha \leq \pi$.

then, using Equation (23), we have:

$$t_2 \cdot \hat{z}_2 = (\hat{z}_1 \cdot \hat{z}_2)(t_2 \cdot \hat{z}_1) - v_1 \cdot (\hat{z}_1 \times \hat{z}_2). \quad (24)$$

Also, from the following:

$$\hat{z}_2 \times v_2 = t_2 - (\hat{z}_2 \cdot t_2)\hat{z}_2,$$

t_2 is obtained as:

$$t_2 = \hat{z}_2 \times v_2 + (\hat{z}_2 \cdot t_2)\hat{z}_2.$$

Then using Equation (24), $(t_2 \cdot \hat{z}_1)$ is derived using $(t_2 \cdot \hat{z}_2)$:

$$t_2 \cdot \hat{z}_1 = v_2 \cdot (\hat{z}_1 \times \hat{z}_2) + (\hat{z}_1 \cdot \hat{z}_2)^2(t_2 \cdot \hat{z}_1) - (\hat{z}_1 \cdot \hat{z}_2)(v_1 \cdot (\hat{z}_1 \times \hat{z}_2)),$$

since $(t_2 \cdot \hat{z}_1)$ appears in both sides of the above equation, rearranging it yields

$$t_2 \cdot \hat{z}_1 = \frac{(v_2 - (\hat{z}_1 \cdot \hat{z}_2)v_1) \cdot \hat{n}}{\|\hat{z}_1 \times \hat{z}_2\|}. \quad (25)$$

By inserting the three terms of Equations (22), (23), and (25) into Equation (21), finally, we have t_2 , using $a \times (b \times c) = (a \cdot c)b - (a \cdot b)c$ and $\hat{n} = \frac{\hat{z}_1 \times \hat{z}_2}{\|\hat{z}_1 \times \hat{z}_2\|}$, as following form:

$$t_2 = \frac{v_2 \times (\hat{z}_1 \times (\hat{z}_1 \times \hat{z}_2)) - (v_1 \cdot (\hat{z}_1 \times \hat{z}_2))\hat{z}_2}{\|\hat{z}_1 \times \hat{z}_2\|^2}. \quad (26)$$

On the other hand, t_1 is obtained from Equation (26) for t_2 by switching subscripts 1 and 2, as follows:

$$\therefore t_1 = \frac{v_1 \times (\hat{z}_2 \times (\hat{z}_2 \times \hat{z}_1)) - (v_2 \cdot (\hat{z}_2 \times \hat{z}_1))\hat{z}_1}{\|\hat{z}_1 \times \hat{z}_2\|^2}. \quad (27)$$

It is noted that the derived t_1 and t_2 are computed by using the information only on two lines $L_1(\hat{z}_1, v_1)$ and $L_2(\hat{z}_2, v_2)$. The next section summarizes the five results of line geometry such as link length a (17), link twist α (18), \hat{x} -axis \hat{n} (19), and feet of common perpendicular line t_2 (26) and t_1 (27).

3.5. Line Geometry Block, $\mathcal{LG}(L_1(\hat{z}_1, v_1), L_2(\hat{z}_2, v_2))$

The method to express how two adjacent lines (or joint axes) are located each other can be classified into four cases: collinear, distant parallel, intersected, and skewed. To begin with, criterion 1 is defined to check whether two lines are parallel or not, which is performed using the two direction vectors:

$$\|\hat{z}_1 \times \hat{z}_2\| = 0 \rightarrow \|\hat{z}_1 \times \hat{z}_2\| < \sin(\epsilon_\theta), \quad (28)$$

where $\epsilon_\theta \in \mathbb{R}^+$ denotes a small positive constant with units of radians, introduced to handle numerical issues when comparing angular values. If criterion 1 of Equation (28) is satisfied, the two lines are parallel to each other. The parallel lines can be further classified into two cases: collinear (parallel) and distant (parallel). For this purpose, criterion 2-1 is defined as the distance between the closest points from the origin of the coordinate frame to the lines using Equation (11) as follows:

$$\|p_1^\perp - p_2^\perp\| = 0 \rightarrow \|p_1^\perp - p_2^\perp\| < \epsilon_d, \quad (29)$$

where

$$\begin{aligned} p_1^\perp &= \hat{z}_1 \times v_1 = \hat{z}_1 \times (p_1 \times \hat{z}_1) \\ p_2^\perp &= \hat{z}_2 \times v_2 = \hat{z}_2 \times (p_2 \times \hat{z}_2). \end{aligned}$$

where $\epsilon_d \in \mathbb{R}^+$ is a small positive threshold with units of meters, introduced to ensure numerical robustness when comparing distances between points. If criterion 2-1 is satisfied, the two lines are collinear (parallel); otherwise, they are distant (parallel). These parallel cases are summarized in Figure 6.

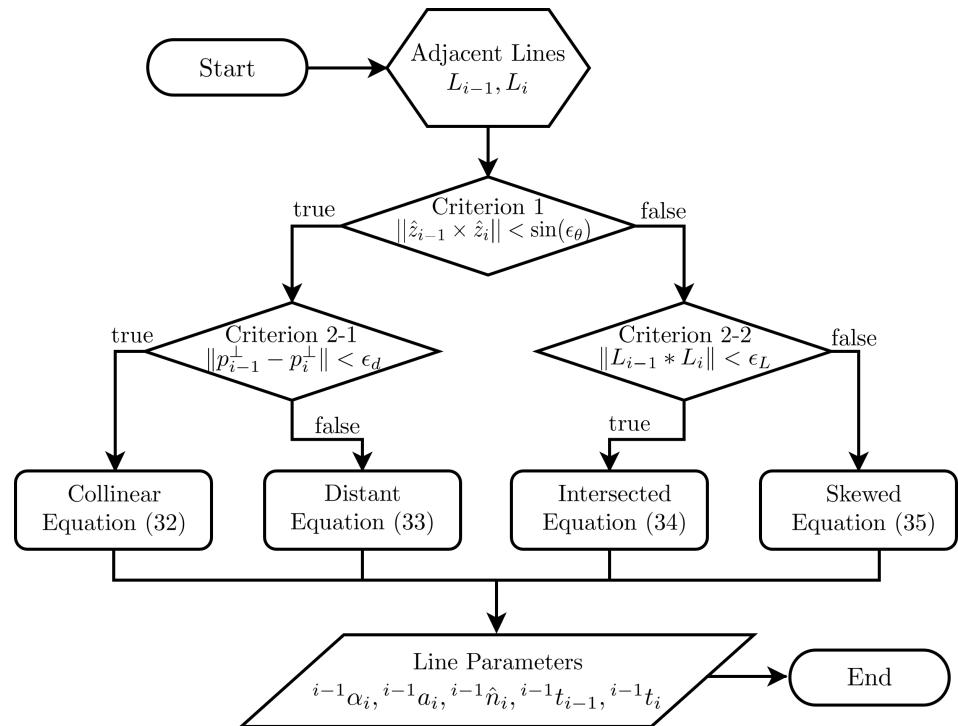


Figure 6. Flowchart of the line geometry block $\mathcal{L}\mathcal{G}(L_{i-1}, L_i)$. Given two input lines, $L_{i-1}(\hat{z}_{i-1}, v_{i-1} = p_{i-1} \times \hat{z}_{i-1})$ and $L_i(\hat{z}_i, v_i = p_i \times \hat{z}_i)$, the block uses two criteria to classify their relationship into one of the four cases: *Collinear*, *Distant*, *Intersected*, or *Skewed*. Based on the case, it returns five parameters: $i-1\alpha_i$, $i-1a_i$, $i-1\hat{n}_i$, $i-1t_{i-1}$, and $i-1t_i$.

On the other hand, if criterion 1 of Equation (28) is not satisfied, the two lines are not parallel to each other. The non-parallel lines can be further classified into two cases: intersected and skewed. For this purpose, criterion 2-2 is defined as the existence of the intersection point, which is conducted using the reciprocal product (16):

$$|L_1 * L_2| = 0 \rightarrow |L_1 * L_2| < \epsilon_L. \quad (30)$$

where $\epsilon_L \in \mathbb{R}^+$ is a small positive threshold used to evaluate the reciprocal product. Note that unlike ϵ_θ (in radians) and ϵ_d (in meters), the reciprocal product is inherently dimensionless. If criterion 2-2 of Equation (30) is satisfied, two lines are intersected at one point, otherwise, they are skewed.

The line geometry block $\mathcal{L}\mathcal{G}(L_1, L_2)$ to be designed in this section receives two lines (L_1, L_2) as inputs, selects one from the four cases using the criteria, and yields five parameters ($\alpha, a, \hat{n}, t_1, t_2$) as outputs, where it is noted that two lines as the inputs consist of the direction vectors \hat{z}_1, \hat{z}_2 of joint axes and the points p_1, p_2 on joint axes as follows:

$$L_1(\hat{z}_1, v_1 = p_1 \times \hat{z}_1) \quad \text{and} \quad L_1(\hat{z}_2, v_2 = p_2 \times \hat{z}_2). \quad (31)$$

3.5.1. Collinear

If $\|\hat{z}_1 \times \hat{z}_2\| < \sin(\epsilon_\theta)$ and $\|p_1^\perp - p_2^\perp\| < \epsilon_d$ are satisfied, the two lines are collinear with the following five parameter descriptions:

$$\alpha = \text{atan2}(\|\hat{z}_1 \times \hat{z}_2\|, \hat{z}_1 \cdot \hat{z}_2), \quad (32a)$$

$$a = 0, \quad (32b)$$

$$\hat{n} = 0_{3 \times 1}, \quad (32c)$$

$$t_1 = t_2 = p_1, \quad (32d)$$

where link twist α can be either 0° or 180° . In fact, the two lines are either in the same direction ($\alpha = 0^\circ$) or in the opposite direction ($\alpha = 180^\circ$) according to the link twist value. In the collinear case, the common perpendicular direction \hat{x} -axis is not defined, and thus the \hat{x} -axis of the previous frame is identically assigned to the new frame.

3.5.2. Distant (Parallel)

If $\|\hat{z}_1 \times \hat{z}_2\| < \sin(\epsilon_\theta)$ and $\|p_1^\perp - p_2^\perp\| \geq \epsilon_d$ are satisfied, the two lines are distant parallel to each other with the following five parameter descriptions:

$$\alpha = \text{atan2}(\|\hat{z}_1 \times \hat{z}_2\|, \hat{z}_1 \cdot \hat{z}_2), \quad (33a)$$

$$a = \|\hat{z}_1 \times (v_2 \cos \alpha - v_1)\|, \quad (33b)$$

$$\hat{n} = \frac{\hat{z}_1 \times (v_2 \cos \alpha - v_1)}{\|\hat{z}_1 \times (v_2 \cos \alpha - v_1)\|}, \quad (33c)$$

$$t_1 = p_1, \quad (33d)$$

$$t_2 = p_1 + \hat{z}_1 \times (v_2 \cos \alpha - v_1), \quad (33e)$$

where it is noted that $\cos \alpha = 1$ when the two parallel lines are in the same direction and $\cos \alpha = -1$ in the opposite direction. Additionally, the closest points p_1^\perp on line L_1 and p_2^\perp on line L_2 yield the link length a and the common perpendicular direction \hat{n} as follows:

$$\begin{aligned} a\hat{n} &= p_2^\perp - p_1^\perp = (\hat{z}_2 \times v_2) - (\hat{z}_1 \times v_1) \\ &= (\hat{z}_1 \cos \alpha \times v_2) - (\hat{z}_1 \times v_1) = \hat{z}_1 \times (v_2 \cos \alpha - v_1). \end{aligned}$$

In fact, t_1 is an arbitrary point on line $L_1(\hat{z}_1, v_1)$, but we choose t_1 equal to p_1 for practical use. Then, $t_2 = p_1 + \hat{z}_1 \times (v_2 \cos \alpha - v_1)$ since $a\hat{n} = t_2 - t_1$.

3.5.3. Intersected

If $\|\hat{z}_1 \times \hat{z}_2\| \geq \sin(\epsilon_\theta)$ and $|L_1 * L_2| < \epsilon_d$ are satisfied, the two lines are intersected at one point $t_1 = t_2$ with the following five parameter descriptions:

$$\alpha = \text{atan2}(\|\hat{z}_1 \times \hat{z}_2\|, \hat{z}_1 \cdot \hat{z}_2), \quad (34a)$$

$$a = 0, \quad (34b)$$

$$\hat{n} = \frac{\hat{z}_1 \times \hat{z}_2}{\|\hat{z}_1 \times \hat{z}_2\|}, \quad (34c)$$

$$t_1 = ((v_1 \cdot \hat{z}_2)I_3 + \hat{z}_1 v_2^T - \hat{z}_2 v_1^T) \frac{\hat{z}_1 \times \hat{z}_2}{\|\hat{z}_1 \times \hat{z}_2\|^2}, \quad (34d)$$

$$t_2 = t_1, \quad (34e)$$

where I_3 is a three-dimensional identity matrix. To find the intersection point, let us rearrange the second term of t_1 (27) using $(a \cdot c)b = a \times (b \times c) + (a \cdot b)c$ as follows:

$$(v_2 \cdot (\hat{z}_2 \times \hat{z}_1))\hat{z}_1 = v_2 \times (\hat{z}_1 \times c) + (v_2 \cdot \hat{z}_1)c,$$

where $c := \hat{z}_2 \times \hat{z}_1$, then t_2 of Equation (27) is rewritten as follows:

$$t_2 = \frac{(v_1 \times (\hat{z}_2 \times c)) - (v_2 \times (\hat{z}_1 \times c)) - (v_2 \cdot \hat{z}_1)c}{\|\hat{z}_1 \times \hat{z}_2\|^2}.$$

Also, by using the property of cross-product operator, e.g., $a \times (b \times c) = (ba^T - (a \cdot b)I_3)c$, we have

$$\begin{aligned} t_2 &= (\hat{z}_2 v_1^T - (v_1 \cdot \hat{z}_2)I_3 - \hat{z}_1 v_2^T) \frac{\hat{z}_2 \times \hat{z}_1}{\|\hat{z}_1 \times \hat{z}_2\|^2} \\ &\therefore = ((v_1 \cdot \hat{z}_2)I_3 + \hat{z}_1 v_2^T - \hat{z}_2 v_1^T) \frac{\hat{z}_1 \times \hat{z}_2}{\|\hat{z}_1 \times \hat{z}_2\|^2}, \end{aligned}$$

where the moments of the intersected lines are the same with each other, e.g., $v_1 = v_2 = v^\perp$ because $t_1 = t_2$.

3.5.4. Skewed

If $\|\hat{z}_1 \times \hat{z}_2\| \geq \sin(\epsilon_\theta)$ and $|L_1 * L_2| \geq \epsilon_d$ are satisfied, the two lines are skewed (not co-planar) with following the five parameters descriptions:

$$\alpha = \text{atan2}(\|\hat{z}_1 \times \hat{z}_2\|, \hat{z}_1 \cdot \hat{z}_2), \quad (35a)$$

$$a = -\frac{L_1 * L_2}{\|\hat{z}_1 \times \hat{z}_2\|}, \quad (35b)$$

$$\hat{n} = \frac{\hat{z}_1 \times \hat{z}_2}{\|\hat{z}_1 \times \hat{z}_2\|}, \quad (35c)$$

$$t_1 = \frac{v_1 \times (\hat{z}_2 \times (\hat{z}_2 \times \hat{z}_1)) + (v_2 \cdot (\hat{z}_1 \times \hat{z}_2))\hat{z}_1}{\|\hat{z}_1 \times \hat{z}_2\|^2}, \quad (35d)$$

$$t_2 = \frac{v_2 \times (\hat{z}_1 \times (\hat{z}_1 \times \hat{z}_2)) + (v_1 \cdot (\hat{z}_2 \times \hat{z}_1))\hat{z}_2}{\|\hat{z}_2 \times \hat{z}_1\|^2}, \quad (35e)$$

where the five results of line geometry such as link length a (17), link twist α (18), \hat{x} -axis \hat{n} (19), and feet of common perpendicular line t_2 (26) and t_1 (27) are fully exploited for the skewed case.

The line geometry block denoted by \mathcal{LG} summarizes the five parameters chosen according to criteria as shown in Figure 6, where the line geometry block is described by

$$\{^{i-1}\alpha_i, ^{i-1}a_i, ^{i-1}\hat{n}_i, ^{i-1}t_{i-1}, ^{i-1}t_i\} = \mathcal{LG}(L_{i-1}, L_i), \quad (36)$$

where the inputs are two lines and the outputs are five parameters yielded from one of the four cases, which is selected using two criteria. As suggested in Figure 1, three joint axes (a set of three consecutive lines L_{i-1} , L_i , and L_{i+1}) are required to extract four MDH parameters and thus two line geometry blocks are needed to describe four MDH parameters for i th joint.

4. Algorithmic MDH Parameter Extraction

This section describes how to algorithmically extract MDH parameters using the line geometry blocks suggested in the previous Section 3.5. This method can be directly applied to the single-branched articulated robotic manipulators. To extract a set of four MDH

parameters for each joint, three consecutive lines (joint axes) are needed. Thus, $N + 2$ line information is required with respect to the base frame $\{0\}$ including tcf denoted by $\{N + 1\}$ for N -joint manipulators. Three coordinate axes of the base frame are defined by $\hat{x}_0 = [1, 0, 0]^T$, $\hat{y}_0 = [0, 1, 0]^T$ and $\hat{z}_0 = [0, 0, 1]^T$, respectively. Algorithm 1 summarizes the inputs and the outputs through the computations of intermediate variables. The inputs are listed as follows:

$$\begin{aligned} \text{base frame (line)} &: \{\hat{z}_0, p_0\}, \hat{x}_0, \\ \text{1st joint (line)} &: \{\hat{z}_1, p_1\}, \\ &\vdots \\ \text{Nth joint (line)} &: \{\hat{z}_N, p_N\}, \\ \text{tcf (line)} &: \{\hat{z}_{N+1}, p_{N+1}\}, \end{aligned}$$

where $p_0 = [0, 0, 0]^T$. The adjacent two lines are applied consecutively to the line geometry blocks for i th joint as follows:

$$\begin{aligned} \{{}^{i-1}\alpha_i, {}^{i-1}a_i, {}^{i-1}\hat{n}_i, {}^{i-1}t_{i-1}, {}^{i-1}t_i\} &\leftarrow \mathcal{LG}(L_{i-1}, L_i), \\ \{{}^i\alpha_{i+1}, {}^i a_{i+1}, {}^i \hat{n}_{i+1}, {}^i t_i, {}^i t_{i+1}\} &\leftarrow \mathcal{LG}(L_i, L_{i+1}), \end{aligned}$$

for $i = 1, 2, \dots, N$.

To extract four MDH parameters, we need to define \hat{x}_i -axis to be the common perpendicular direction denoted by ${}^i\hat{n}_{i+1}$ as shown in Figure 1. As a special case, the common perpendicular direction is not defined in the collinear case as suggested in Equations (32a)–(32d), and thus we assign the previous \hat{x}_{i-1} -axis to \hat{x}_i as follows:

$$\hat{x}_i = \begin{cases} \hat{x}_{i-1} & \text{if } L_i \text{ and } L_{i+1} \text{ is collinear} \\ {}^i\hat{n}_{i+1} & \text{otherwise,} \end{cases} \quad (37)$$

where it is noted that the origin of frame $\{i\}$ is specified as ${}^i t_i$. Once defining \hat{x}_i -axis through Equation (37), the sign of link twist α_i and link length a_i can be determined by using the dot-product between \hat{x}_{i-1} and ${}^{i-1}\hat{n}_i$. To begin with, the sign is assigned into +1 when two lines are parallel. Subsequently, if $\hat{x}_{i-1} \cdot {}^{i-1}\hat{n}_i > 0$, then the sign will be +1, and if $\hat{x}_{i-1} \cdot {}^{i-1}\hat{n}_i < 0$, then the sign will be -1. As a summary, the sign is determined as follows:

$$s_{ix} = \begin{cases} 1 & \text{if } \|\hat{z}_{i-1} \times \hat{z}_i\| < \sin(\epsilon_\theta) \\ \frac{\hat{x}_{i-1} \cdot {}^{i-1}\hat{n}_i}{|\hat{x}_{i-1} \cdot {}^{i-1}\hat{n}_i|} & \text{otherwise.} \end{cases} \quad (38)$$

The sign of the joint offset angle θ_{i0} can be also determined by using the dot-product between \hat{z}_i and $(\hat{x}_{i-1} \times \hat{x}_i)$ as follows:

$$s_{iz} = \begin{cases} 1 & \text{if } \|\hat{x}_{i-1} \times \hat{x}_i\| < \sin(\epsilon_\theta) \\ \frac{\hat{z}_i \cdot (\hat{x}_{i-1} \times \hat{x}_i)}{|\hat{z}_i \cdot (\hat{x}_{i-1} \times \hat{x}_i)|} & \text{otherwise,} \end{cases} \quad (39)$$

where it is noted that the sign of joint offset distance d_{i0} does not have to be considered because it is obtained already by dot-product implying the projection onto the \hat{z}_i -axis. Using the signs of s_{ix} (38) and s_{iz} (39), four MDH parameters are determined in the following sections.

Algorithm 1 Algorithmic MDH parameter extraction.

Require: $\{\hat{z}_0, p_0\}, \{\hat{z}_1, p_1\}, \dots, \{\hat{z}_N, p_N\}, \{\hat{z}_{N+1}, p_{N+1}\}, \hat{x}_0 = [1, 0, 0]^T$.
Ensure: $\{\hat{x}_1, \alpha_1, a_1, d_{10}, \theta_{10}\}, \dots, \{\hat{x}_N, \alpha_N, a_N, d_{N0}, \theta_{N0}\}$

```

1: for  $i = 1$  to  $N$  do
2:   for  $j = i - 1$  to  $i + 1$  do
3:      $v_j \leftarrow p_j \times \hat{z}_j$ 
4:      $L_j \leftarrow (\hat{z}_j, v_j)$ 
5:   end for
6:    $\{{}^{i-1}\alpha_i, {}^{i-1}a_i, {}^{i-1}\hat{n}_i, {}^{i-1}t_{i-1}, {}^{i-1}t_i\} \leftarrow \mathcal{LG}(L_{i-1}, L_i)$ 
7:    $\{{}^i\alpha_{i+1}, {}^i a_{i+1}, {}^i \hat{n}_{i+1}, {}^i t_i, {}^i t_{i+1}\} \leftarrow \mathcal{LG}(L_i, L_{i+1})$ 
8:   if  $L_i$  and  $L_{i+1}$  are collinear then
9:      $\hat{x}_i \leftarrow \hat{x}_{i-1}$ 
10:   else
11:      $\hat{x}_i \leftarrow {}^i \hat{n}_{i+1}$ 
12:   end if
13:   if  $\|\hat{z}_{i-1} \times \hat{z}_i\| < \sin(\epsilon_\theta)$  then
14:      $s_{ix} \leftarrow 1$ 
15:   else
16:      $s_{ix} \leftarrow (\hat{x}_{i-1} \cdot {}^{i-1}\hat{n}_i) / |(\hat{x}_{i-1} \cdot {}^{i-1}\hat{n}_i)|$ 
17:   end if
18:   if  $\|\hat{x}_{i-1} \times \hat{x}_i\| < \sin(\epsilon_\theta)$  then
19:      $s_{iz} \leftarrow 1$ 
20:   else
21:      $s_{iz} \leftarrow \hat{z}_i \cdot (\hat{x}_{i-1} \times \hat{x}_i) / |\hat{z}_i \cdot (\hat{x}_{i-1} \times \hat{x}_i)|$ 
22:   end if
23:    $\alpha_i \leftarrow s_{ix} {}^{i-1}\alpha_i$ 
24:    $a_i \leftarrow s_{ix} {}^{i-1}a_i$ 
25:    $d_{i0} \leftarrow \hat{z}_i \cdot ({}^i t_i - {}^{i-1}t_i)$ 
26:    $\theta_{i0} \leftarrow s_{iz} \text{atan2}(\|\hat{x}_{i-1} \times \hat{x}_i\|, \hat{x}_{i-1} \cdot \hat{x}_i)$ 
27: end for

```

4.1. Twist Angle α_i

It is defined as the angle from \hat{z}_{i-1} to \hat{z}_i about \hat{x}_{i-1} -axis as shown in Figure 1. Since the twist angle ${}^{i-1}\alpha_i$ provided from $\mathcal{LG}(L_{i-1}, L_i)$ is in the range of $0 \leq {}^{i-1}\alpha_i \leq \pi$, the sign s_{ix} should be reflected in the following form:

$$\alpha_i = s_{ix} {}^{i-1}\alpha_i. \quad (40)$$

4.2. Link Length a_i

It is defined as the distance from \hat{z}_{i-1} to \hat{z}_i along \hat{x}_{i-1} -axis as shown in Figure 1. Since the distance ${}^{i-1}a_i$ is positive, the sign s_{ix} is also reflected in the following form:

$$a_i = s_{ix} {}^{i-1}a_i. \quad (41)$$

4.3. Joint Offset Distance d_{i0}

It is defined as the distance from \hat{x}_{i-1} to \hat{x}_i along the \hat{z}_i -axis as shown in Figure 1. The distance projected into \hat{z}_i -axis is defined as follows:

$$d_{i0} = \hat{z}_i \cdot ({}^i t_i - {}^{i-1}t_i). \quad (42)$$

4.4. Joint Offset Angle θ_{i0}

It is defined as the angle from \hat{x}_{i-1} to \hat{x}_i about the \hat{z}_i -axis as shown in Figure 1. Since the \hat{x}_i -axis is determined by Equation (37), the sign s_{iz} should be reflected as follows:

$$\theta_{i0} = s_{iz}\text{atan2}(\|\hat{x}_{i-1} \times \hat{x}_i\|, \hat{x}_{i-1} \cdot \hat{x}_i). \quad (43)$$

The outputs of Algorithm 1 are consecutively arranged into:

$$\text{1st joint : } \{\hat{x}_1, \alpha_1, a_1, d_{10}, \theta_{10}\} \text{ from } L_0, L_1, L_2, \hat{x}_0,$$

$$\text{2nd joint : } \{\hat{x}_2, \alpha_2, a_2, d_{20}, \theta_{20}\} \text{ from } L_1, L_2, L_3,$$

$$\vdots$$

$$\text{Nth joint : } \{\hat{x}_N, \alpha_N, a_N, d_{N0}, \theta_{N0}\} \text{ from } L_{N-1}, L_N, L_{N+1},$$

where it is noted that the \hat{x}_0 -axis is required in advance and it is commonly chosen as $[1, 0, 0]^T$. To verify the completeness of Algorithm 1, several examples are suggested in the following section.

The algorithm does not include any iterative procedures and is composed of fixed geometric operations such as dot products and cross products. The line geometry block contains no loops, and Algorithm 1 uses a single for-loop over N joints. The overall computational complexity is $\mathcal{O}(N)$.

5. Verification

Algorithm 1 was implemented in both Python and MATLAB (The implementation for algorithmic DH parameter extraction is available at <https://github.com/MinchangSung0223/AlgorithmicMDH> (accessed on 27 April 2025)) and is also referenced in [26]. The algorithm was tested using PyBullet [27] and MATLAB simulations. For each joint i , the joint axis \hat{z}_i and a point p_i on the axis were extracted from the URDF (Unified Robot Description Format) model [28]. The verification was first conducted on a simple SCARA robot, followed by testing on three commercial robotic manipulators, as described in the following sections. All verifications were conducted entirely in simulation without using the experimental hardware.

5.1. SCARA

A typical SCARA robot consists of three revolute joints and one prismatic joint, where the left side of Figure 7 shows the line information as inputs and the right side shows the extraction results of MDH parameters. For the first joint, the following three consecutive lines are applied to the Algorithm 1:

$$\text{0th line : } \hat{z}_0 = [0, 0, 1]^T, p_0 = [0, 0, 0]^T, \hat{x}_0 = [1, 0, 0]^T,$$

$$\text{1st line : } \hat{z}_1 = [0, 0, 1]^T, p_1 = [0, 0, 0.375]^T,$$

$$\text{2nd line : } \hat{z}_2 = [0, 0, 1]^T, p_2 = [0, 0.5, 0.4]^T,$$

where $v_i = p_i \times \hat{z}_i$ for $i = 0, 1, 2$. The zeroth line and first line have a collinear relation, and the first line and second line have a distant parallel relation by the proposed criteria

in Section 3.5 regarding the line geometry block $\mathcal{LG}(L_{i-1}, L_i)$. And finally, the outputs of Algorithm 1 regarding the first joint are listed as follows:

$$\begin{array}{lll} \hat{x}_1 = [0, 1, 0]^T & & \text{for } \{1\} \text{ frame,} \\ {}^0t_1 = [0, 0, 0]^T & , {}^1t_1 = [0, 0, 0.375]^T & \text{on } \hat{z}_1\text{-axis,} \\ \alpha_1 = 0 & & \text{about } \hat{x}_0\text{-axis,} \\ a_1 = 0 & & \text{along } \hat{x}_0\text{-axis,} \\ d_{10} = 0.375 & & \text{along } \hat{z}_1\text{-axis,} \\ \theta_{10} = 90^\circ & & \text{about } \hat{z}_1\text{-axis,} \end{array}$$

where these results make us to denote the frame $\{1\}$ in the right side of Figure 7. The aforementioned processes are repeated until the fourth joint. As a result, a pair of input and output data for all joints are listed in Table 2 and denoted on the right side of Figure 7. During the processes, the relations between the lines are found in Table 3.

Table 2. MDH parameters of the Selective Compliance Assembly Robot Arm (SCARA) robot extracted when inputs are \hat{z}_i -axis, p_i on the i th joint axis, and its moments denoted by $v_i = p_i \times \hat{z}_i$, where outputs of Algorithm 1 are obtained as a set of \hat{x}_i -axis, link twist α_i , link length a_i , joint offset distance d_{i0} , and joint offset angle θ_{i0} for the i th joint. Additionally, ${}^{i-1}\vec{t}_i$ and ${}^i\vec{t}_i$ are both feet of the common perpendicular line.

	<i>i</i>	0	1	2	3	4	5
input	\hat{z}_i	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$
	\vec{p}_i	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0.375 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0.5 \\ 0.4 \end{bmatrix}$	$\begin{bmatrix} 0.5 \\ 0.5 \\ 0.4 \end{bmatrix}$	$\begin{bmatrix} 0.5 \\ 0.5 \\ 0.15 \end{bmatrix}$	$\begin{bmatrix} 0.5 \\ 0.5 \\ 0.15 \end{bmatrix}$
	\vec{v}_i	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0.5 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0.5 \\ -0.5 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -0.5 \\ 0.5 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -0.5 \\ 0.5 \\ 0 \end{bmatrix}$
output	\hat{x}_i		$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	
	${}^{i-1}\vec{t}_i$		$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0.5 \\ 0.375 \end{bmatrix}$	$\begin{bmatrix} 0.5 \\ 0.5 \\ 0.4 \end{bmatrix}$	$\begin{bmatrix} 0.5 \\ 0.5 \\ 0.4 \end{bmatrix}$	
	${}^i\vec{t}_i$	\times	$\begin{bmatrix} 0 \\ 0 \\ 0.375 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0.5 \\ 0.4 \end{bmatrix}$	$\begin{bmatrix} 0.5 \\ 0.5 \\ 0.4 \end{bmatrix}$	$\begin{bmatrix} 0.5 \\ 0.5 \\ 0.15 \end{bmatrix}$	\times
	α_i	0	0	0	0	π	
	a_i	0	0.5	0.5	0.5	0	
	d_{i0}	0.375	0.025	0	0	0.25	
	θ_{i0}	$\frac{\pi}{2}$	$-\frac{\pi}{2}$	0	0	0	

Table 3. Relations between adjacent joint axes computed from the SCARA model.

	b/w 0 and 1	1 and 2	2 and 3	3 and 4	4 and 5
relation	collinear	distant	distant	collinear	collinear

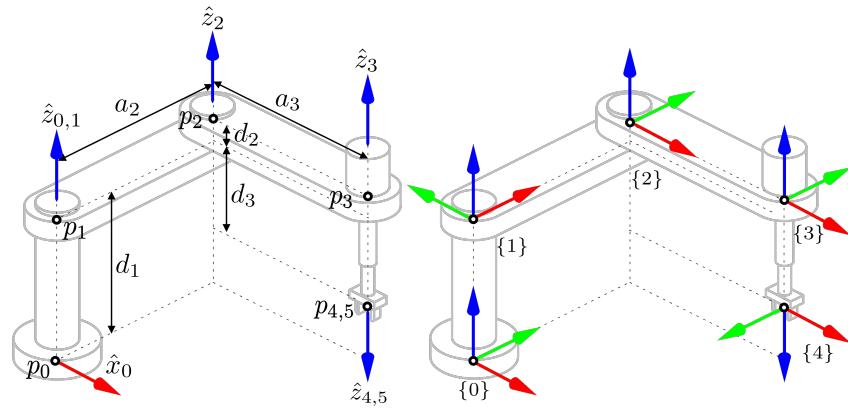


Figure 7. The SCARA robot example, where the left side shows six line information including the base and TCF and the right side illustrates the result of MDH parameters extracted using Algorithm 1, in which the red arrow denotes the \hat{x}_i -axis, the green arrow expresses the \hat{y}_i -axis, and the blue arrow represents the \hat{z}_i -axis, respectively. In addition, the circle \circ on the left side denotes the point p_i , and \circ on the right side expresses the origin of the i th frame.

Using the constant MDH parameters suggested in Table 2, the adjacent transformation can be represented by adding joint variables $q_{1,\theta}, q_{2,\theta}, q_{3,\theta}$ for the first, second, and third revolute joints and $q_{4,d}$ for the fourth prismatic joint. The generalized joint variables (θ_i, d_i) for each joint are summarized in Table 4.

Table 4. The generalized joint variables with MDH parameters for the SCARA robot.

Joints	Generalized Joint Variables (θ_i, d_i)
1st revolute	$\theta_1 \leftarrow \theta_{10} + q_{1,\theta} = \frac{\pi}{2} + q_{1,\theta}$ $d_1 \leftarrow d_{10} = 0.375$
2nd revolute	$\theta_2 \leftarrow \theta_{20} + q_{2,\theta} = -\frac{\pi}{2} + q_{2,\theta}$ $d_2 \leftarrow d_{20} = 0.025$
3rd revolute	$\theta_3 \leftarrow \theta_{30} + q_{3,\theta} = q_{3,\theta}$ $d_3 \leftarrow d_{30} = 0$
4th prismatic	$\theta_4 \leftarrow \theta_{40} = 0$ $d_4 \leftarrow d_{40} + q_{4,d} = 0.25 + q_{4,d}$

The forward kinematics can be obtained using the MDH parameters (α_i, a_i) about/along the \hat{x}_{i-1} -axis and the generalized joint variables (θ_i, d_i) about/along the \hat{z}_i -axis as follows:

$${}^0T_{tcf} = {}^0T_1(\theta_1, d_1){}^1T_2(\theta_2, d_2){}^2T_3(\theta_3, d_3){}^3T_4(\theta_4, d_4){}^4T_{tcf},$$

where tcf implies $\{5\}$ and i th adjacent transformation belonging to $SE(3)$ is obtained as, for $i = 1, 2, 3, 4$,

$${}^{i-1}T_i(\theta_i, d_i) = \text{expm}([\lambda_{x_{i-1}}])\text{expm}([\lambda_{z_i}]),$$

in which the exponential coordinates $\lambda_{x_{i-1}} \in \mathbb{R}^6$ and $\lambda_{z_i} \in \mathbb{R}^6$ isomorphic to the Lie algebras $[\lambda_{x_{i-1}}] \in \mathfrak{se}(3)$ and $[\lambda_{z_i}] \in \mathfrak{se}(3)$ are described using Equation (4) and summarized in Table 5.

Table 5. Exponential coordinates $\lambda_{x_{i-1}}$, λ_{z_i} , and joint matrices E_i for SCARA robot joints.

<i>i</i>	Joint Type	$\lambda_{x_{i-1}}$	λ_{z_i}	E_i
1	R	$[0, 0, 0, 0, 0, 0]^T$	$[0, 0, 0.375, 0, 0, \frac{\pi}{2} + q_{1,\theta}]^T$	$[0, 0, 0, 0, 0, 1]^T$
2	R	$[0.5, 0, 0, 0, 0, 0]^T$	$[0, 0, 0.025, 0, 0, -\frac{\pi}{2} + q_{2,\theta}]^T$	$[0, 0, 0, 0, 0, 1]^T$
3	R	$[0.5, 0, 0, 0, 0, 0]^T$	$[0, 0, 0, 0, 0, q_{3,\theta}]^T$	$[0, 0, 0, 0, 0, 1]^T$
4	P	$[0, 0, 0, \pi, 0, 0]^T$	$[0, 0, 0.25 + q_{4,d}, 0, 0, 0]^T$	$[0, 0, 1, 0, 0, 1]^T$

Additionally, we can derive the joint twist represented by ${}^{i-1}V_i = \dot{\lambda}_{z_i} = E_i \dot{q}_i$ and the joint matrix E_i as suggested in Section 2.3. Furthermore, the joint matrix represented in the base frame $\{0\}$ (${}^0E_i = {}^0\text{Ad}_i E_i$) and the joint matrix represented in the tool coordinate frame $\{tcf\}$ (${}^{tcf}E_i = {}^{tcf}\text{Ad}_i E_i$) can be obtained using the corresponding adjoint transformations ${}^0\text{Ad}_i$ and ${}^{tcf}\text{Ad}_i$, respectively. Two types of joint matrices denoted by 0E_i and ${}^{tcf}E_i$ are called spatial screw axes and body screw axes in [4], respectively.

5.2. SCARA with Modeling Uncertainties

In practical robotic systems, modeling uncertainties often lead to deviations in the direction and displacement vectors from their ideal values. Despite such deviations, the proposed algorithm is capable of robustly computing the MDH parameters from non-ideal inputs, provided that the joint axes are correctly defined as lines, as described in Section 3. In conventional approaches to kinematic modeling using MDH or DH parameterizations, the transformations from the base frame to the first joint frame and from the last joint frame to the tool coordinate frame are typically treated as known rigid transformations. If needed, they may be modeled with additional parameters and identified separately. In our method, it is also possible to estimate the MDH parameters associated with such rigid transformations by allocating additional virtual lines to represent them. However, because these transformations are typically fixed and known with high accuracy (e.g., derived from CAD models or mechanical assembly), our algorithm assumes them to be given and instead focuses on calibrating the internal joint transformations between successive links using the line geometry-based approach (see Algorithm 1). To evaluate the robustness of the proposed method, we introduced synthetic modeling uncertainties into a simple SCARA robot. Table 6 compares the nominal and perturbed (“actual”) joint parameters encoded in the URDF. Perturbations were applied to each joint with approximately 1° deviation in orientation and 0.01 m deviation in displacement, mimicking realistic sensor or modeling noise. Based on the perturbed URDF data in Table 6, the MDH parameters extracted using the proposed algorithm are listed in Table 7. Compared to the ideal case, the modeling uncertainties caused several joint axis pairs, previously identified as “distant” in line geometry, to be reclassified as “skewed”. As a result, the joint offset distances d_{i0} increased noticeably. The updated relationships between adjacent joint axes under perturbation are summarized in Table 8.

Table 6. Comparison of nominal and perturbed URDF parameters for the SCARA robot.

Joint Pair	Orientation (Roll, Pitch, Yaw) [deg]		Position (x, y, z) [m]	
	Nominal	Perturbed	Nominal	Perturbed
0–1	(0, 0, 0)	(0, 0, 0)	(0, 0, 0.375)	(0, 0, 0)
1–2	(0, 0, 0)	(1, 1, -1)	(0, 0.5, 0.025)	(0.01, 0.51, -0.024)
2–3	(0, 0, 0)	(1, -1, 1)	(0.5, 0, 0)	(0.51, -0.01, 0.01)
3–4	(0, 0, 0)	(-1, -1, -1)	(0, 0, -0.25)	(-0.01, -0.01, -0.29)
4–5	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)

Table 7. Extracted MDH parameters from the perturbed SCARA robot, using small positive threshold $\sin(\epsilon_\theta) = \epsilon_d = \epsilon_L = 10^{-16}$.

Parameter	Joint 1	Joint 2	Joint 3	Joint 4
α_i [rad]	0.0000	0.0247	0.0247	0.0247
a_i [m]	0.0000	0.3614	0.3739	0.0141
d_{i0} [m]	14.8930	-0.5166	-14.3041	-0.0100
θ_{i0} [rad]	0.7678	-1.5533	-1.5883	0.0000

Table 8. Relations between adjacent joint axes computed from the perturbed URDF model.

	b/w 0 and 1	1 and 2	2 and 3	3 and 4	4 and 5
Relation	collinear	skewed	skewed	skewed	collinear

5.3. Indy7

Let us consider the MDH parameters of Indy7 manufactured by Neuromeka, which has six revolute joints based on electrical current-based control. The left side of Figure 8 shows line information denoted by (\hat{z}_i, p_i) including the base frame and TCF, where the line information was extracted from the \hat{z} -axis and the origin of the joint frame written in the URDF in [29]. The right side of Figure 8 illustrates the coordinate frames defined by MDH parameters, which were extracted from Algorithm 1. Indeed, the MDH parameters of Indy7 have been discussed in [30], but the results are different from ours. The differences between the proposed method and the reference [30] happen because the proposed method makes use of the line geometry and [30] uses the user-defined \hat{x} -axis, respectively. Specifically, $d_{20} = 0$ in [30] while $d_{20} = 0.109$ in Table 9, and \hat{x}_5 -axis of [30], have opposite directions with the \hat{x}_5 -axis shown in Figure 8.

For Indy7, when the input data suggested in Table 9 are applied to Algorithm 1, the line geometry blocks are called by the algorithm, and the spatial relationships between the joint axes are determined as summarized in Table 10.

Table 9. Indy7, where upper part and lower part imply the input data to the Algorithm 1 and the output yielded from the algorithm, respectively.

i	0	1	2	3	4	5	6	7
\hat{z}_i	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$
input								
p_i	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0.0775 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -0.1090 \\ 0.2995 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -0.0785 \\ 0.7495 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -0.0035 \\ 1.0165 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -0.1175 \\ 1.0995 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -0.1865 \\ 1.2675 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -0.1865 \\ 1.3275 \end{bmatrix}$
v_i	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0.2995 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0.7495 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -0.0035 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1.0995 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -0.1865 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -0.1865 \\ 0 \\ 0 \end{bmatrix}$
i	0	1	2	3	4	5	6	7
\hat{x}_i		$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$	
output								
$i^{-1}t_i$	\times	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0.2995 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -0.1090 \\ 0.7495 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -0.0035 \\ 0.7495 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -0.0035 \\ 1.0995 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -0.1865 \\ 1.0995 \end{bmatrix}$	\times
i_t_i			$\begin{bmatrix} 0 \\ 0 \\ 0.2995 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -0.1090 \\ 0.2995 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -0.0035 \\ 0.7495 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -0.0035 \\ 1.0995 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -0.1865 \\ 1.0995 \end{bmatrix}$	

Table 9. Cont.

	i	0	1	2	3	4	5	6	7
output	α_i		0	$\frac{\pi}{2}$	0	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	
	a_i		0	0	0.4500	0	0	0	
	d_{i0}		0.2995	0.1090	-0.1055	0.3500	0.1830	0.1680	
	θ_{i0}		0	$\frac{\pi}{2}$	$\frac{\pi}{2}$	π	π	0	

Table 10. Relations between adjacent joint axes computed from the Indy7 model.

	b/w 0 and 1	1 and 2	2 and 3	3 and 4
relation	collinear	intersected	distant	intersected
	4 and 5	5 and 6	6 and 7	
relation	intersected	intersected	collinear	

The aforementioned selections yield the MDH parameters for Indy7 in the Algorithm 1. As a result, a pair of input and output data for all joints are listed in Table 9 and denoted on the right side of Figure 8. Note that any point p_i can be arbitrarily chosen along the line $L_i(\hat{z}_i, v_i)$ (i.e., the joint axis) since the MDH parameterization only requires that p_i lies on the axis direction \hat{z}_i . For example, if $p_2 = [0, 0, 0.2995]^T$ is selected as the intersection point between L_1 and L_2 , then the resulting value of d_{20} becomes zero.

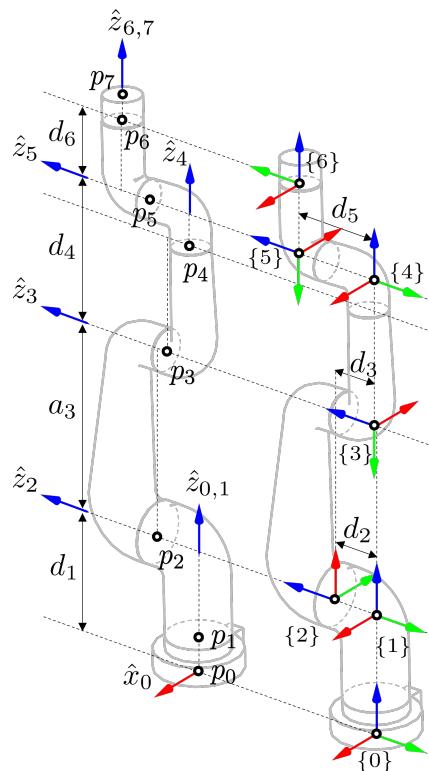


Figure 8. MDH parameters of Indy7, where the left side shows the input data regarding \hat{z}_i and p_i denoted by the circle \circ , and the right side illustrates the frames obtained from the MDH parameters. Numerical data of input/output for Algorithm 1 are given in Table 9.

5.4. KUKA LBR iiwa

The KUKA LBR iiwa is one of the widely used seven-dof robotic manipulators, which has seven revolute joints. The left side of Figure 9 shows line information denoted by (\hat{z}_i, p_i)

including the base frame and TCF, where the line information was extracted from the \hat{z} -axis and the origin of the joint frame written in the URDF in [31]. The right side of Figure 9 illustrates the coordinate frames defined by MDH parameters extracted from Algorithm 1. When the input data suggested in Table 11 are applied to Algorithm 1, the line geometry blocks are called by the algorithm, and the spatial relationships between the joint axes are determined as summarized in Table 12.

Table 11. KUKA LBR iiwa, where upper part and lower part imply the input data to the Algorithm 1 and the output yielded from the algorithm, respectively.

i	0	1	2	3	4	5	6	7	8
\hat{z}_i	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$
input	p_i	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0.1575 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0.3600 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0.5645 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0.7800 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0.9645 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1.1800 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1.2610 \end{bmatrix}$
v_i	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ -0.3600 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0.7800 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ -1.1800 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$
	\hat{x}_i	$\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$
	$i^{-1}t_i$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0.3600 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0.3600 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0.7800 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0.7800 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1.1800 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1.1800 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1.1800 \end{bmatrix}$
output	i_t_i	\times	$\begin{bmatrix} 0 \\ 0 \\ 0.3600 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0.3600 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0.7800 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0.7800 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1.1800 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1.1800 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1.2610 \end{bmatrix}$
	α_i	0	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$
	a_i	0	0	0	0	0	0	0	0
	d_{i0}	0.3600	0	0.4200	0	0.4000	0	0.0810	
	θ_{i0}	π	π	0	π	0	π	0	

Table 12. Relations between adjacent joint axes computed from the KUKA LBR iiwa model.

	b/w 0 and 1	1 and 2	2 and 3	3 and 4
relation	collinear	intersected	intersected	intersected
	4 and 5	5 and 6	6 and 7	7 and 8
relation	intersected	intersected	intersected	collinear

According to the above selection, the MDH parameters for KUKA LBR iiwa are obtained by the Algorithm 1. As a result, a pair of input and output data for all joints are listed in Table 11 and denoted on the right side of Figure 9.

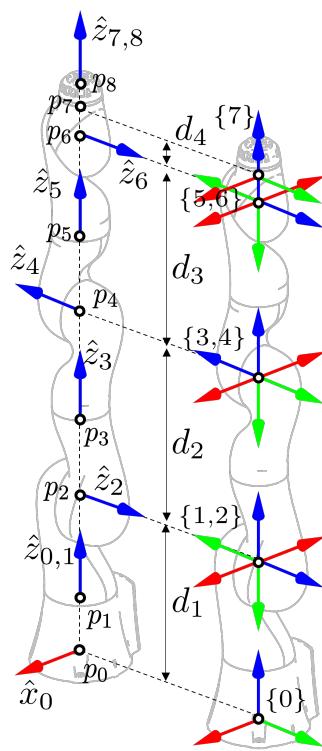


Figure 9. MDH parameters of KUKA LBR iiwa, where the left side shows the input data regarding \hat{z}_i and p_i denoted by the circle \circ , and the right side illustrates the frames obtained from the MDH parameters. Numerical data of input/output related to Algorithm 1 are given in Table 11.

5.5. UR5

The UR-5 is one of the widely used six-dof robotic manipulators, which has six revolute joints. The left side of Figure 10 shows line information denoted by (\hat{z}_i, p_i) including the base frame and TCF, where the line information was extracted from the \hat{z} -axis and the origin of the joint frame written in the URDF [32]. The right side of Figure 10 illustrates the coordinate frames defined by MDH parameters extracted from Algorithm 1. When the input data suggested in Table 13 are applied to Algorithm 1, the line geometry blocks are called by the algorithm, and the spatial relationships between the joint axes are determined as shown in Table 14.

According to the above selection, the MDH can be obtained. As a result, a pair of input and output data for all joints are listed in Table 13 and denoted on the right side of Figure 10.

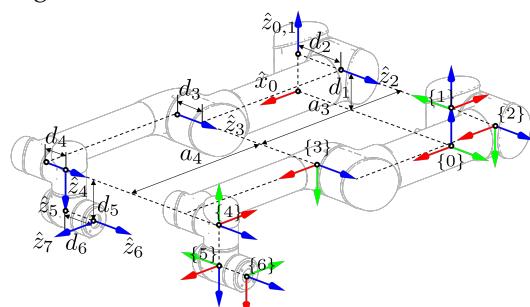


Figure 10. MDH parameters of UR5, where the left side shows the input data regarding \hat{z}_i and p_i denoted by the circle \circ , and the right side illustrates the frames obtained from the MDH parameters. Numerical data of input/output related to Algorithm 1 are given in Table 13.

Table 13. UR5, where the upper part and lower part imply the input data to the Algorithm 1 and the output yielded from the algorithm, respectively.

	i	0	1	2	3	4	5	6	7
input	\hat{z}_i	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	
	p_i	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0.0892 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0.1358 \\ 0.0892 \end{bmatrix}$	$\begin{bmatrix} 0.4250 \\ 0.0161 \\ 0.0892 \end{bmatrix}$	$\begin{bmatrix} 0.8173 \\ 0.0161 \\ 0.0892 \end{bmatrix}$	$\begin{bmatrix} 0.8173 \\ 0.1091 \\ 0.0892 \end{bmatrix}$	$\begin{bmatrix} 0.8173 \\ 0.1091 \\ -0.0055 \end{bmatrix}$	$\begin{bmatrix} 0.8173 \\ 0.1915 \\ -0.0055 \end{bmatrix}$
	v_i	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -0.0892 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -0.0892 \\ 0 \\ 0.4250 \end{bmatrix}$	$\begin{bmatrix} -0.0892 \\ 0 \\ 0.8173 \end{bmatrix}$	$\begin{bmatrix} -0.1091 \\ 0.8173 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0.0055 \\ 0 \\ 0.8173 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -0.0055 \\ -0.1915 \end{bmatrix}$
output	\hat{x}_i		$\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$	
	$i^{-1}t_i$		$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0.0892 \end{bmatrix}$	$\begin{bmatrix} 0.4250 \\ 0.1358 \\ 0.0892 \end{bmatrix}$	$\begin{bmatrix} 0.8173 \\ 0.0161 \\ 0.0892 \end{bmatrix}$	$\begin{bmatrix} 0.8173 \\ 0.1091 \\ 0.0892 \end{bmatrix}$	$\begin{bmatrix} 0.8173 \\ 0.1091 \\ -0.0055 \end{bmatrix}$	
	i_t_i	\times	$\begin{bmatrix} 0 \\ 0 \\ 0.0892 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0.1358 \\ 0.0892 \end{bmatrix}$	$\begin{bmatrix} 0.4250 \\ 0.0161 \\ 0.0892 \end{bmatrix}$	$\begin{bmatrix} 0.8173 \\ 0.1091 \\ 0.0892 \end{bmatrix}$	$\begin{bmatrix} 0.8173 \\ 0.1091 \\ -0.0055 \end{bmatrix}$	$\begin{bmatrix} 0.8173 \\ 0.1915 \\ -0.0055 \end{bmatrix}$	\times
	α_i		0	$\frac{\pi}{2}$	0	0	$\frac{\pi}{2}$	$\frac{\pi}{2}$	
	a_i		0	0	0.4250	0.3923	0	0	
	d_{i0}		0.0892	0.1358	-0.1197	0.0930	0.0946	0.0823	
	θ_{i0}		π	π	0	π	π	$\frac{\pi}{2}$	

Table 14. Relations between adjacent joint axes computed from the UR5 model.

	b/w 0 and 1	1 and 2	2 and 3	3 and 4
relation	collinear	intersected	distant	distant
	4 and 5	5 and 6	6 and 7	
relation	intersected	intersected	intersected	

6. Conclusions

This paper has presented how to extract the MDH parameters of single-branched robotic manipulators using the line geometry, ultimately, for kinematic modeling. The extracted MDH parameters could be used to define the adjacent transformation between the coordinate frames while finding out the exponential coordinates, joint twists, and joint matrices derived from the MDH parameters. The line geometry played core roles in extracting the MDH parameters based on the relation between line and points, the relation between two lines, a common perpendicular line, and the feet of a common perpendicular line, which are concatenated into the line geometry block. Also, it was important to classify how two adjacent lines are located relative to each other such as collinear, distant parallel, intersected, and skewed. Finally, the MDH extraction method was summarized in the algorithm to yield four parameters such as link twist, link length, joint offset distance, and joint offset angle. By combining parts of MDH parameters into the generalized joint variables, forward kinematics could be completed. The completeness

of the proposed algorithm was verified through the conventional SCARA robot and three different commercial Indy7, KUKA LBR iiwa, and UR5 robots. Although the kinematic modeling using the MDH parameters was not new, the proposed algorithm was able to overcome the disadvantage that the MDH parameters are differently defined by user-defined coordinate frames, because the line geometry provides a unique and systematic method.

Author Contributions: Conceptualization, M.S.; Methodology, M.S.; Software, M.S.; Validation, M.S.; Formal analysis, M.S.; Investigation, M.S.; Data curation, M.S.; Writing—original draft, M.S.; Writing—review & editing, M.S. and Y.C.; Supervision, Y.C.; Project administration, Y.C.; Funding acquisition, Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Institute of Civil-Military Technology Cooperation funded by the Defense Acquisition Program Administration and the Ministry of Trade, Industry, and Energy of the Korean government under grant No. 22-CM-EC-36, and in part by the Industrial Strategic Technology Development Program funded by the Ministry of Trade, Industry, and Energy, under grant No. 20023257, South Korea.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The source code and sample data used for the algorithmic MDH parameter extraction are openly available at <https://github.com/MinchangSung0223/AlgorithmicMDH> (accessed on 27 April 2025). Additional simulation results are available from the corresponding author upon reasonable request.

Acknowledgments: The authors thank the organizations that provided data and simulation models used in this study.

Conflicts of Interest: The authors declare no conflicts of interest. The sponsors had no role in the design, execution, interpretation, or writing of the study.

Abbreviations

- DH—Denavit–Hartenberg
- MDH—Modified Denavit–Hartenberg
- PoE—Product of Exponentials
- SCARA—Selective Compliance Assembly Robot Arm
- URDF—Unified Robot Description Format
- TCF—Tool Coordinate Frame

Appendix A. Comparison of Kinematic Parameterizations

Table A1. Comparison of PoE, MDH, and DH formulations for n -DOF serial manipulators.

Category	PoE	MDH	DH
# of parameters	$6 \times n$	$4 \times n$	$4 \times n$
${}^0T_{\text{tcf}}$ required	✓	✗	✗
${}^nT_{\text{tcf}}$ required	✗	✓	✓
Frame consistency	global	local	local
x -axis assignment	unnecessary	systematic	manual and ambiguous
Multi-DOF joint support	✓	✗	✗

Note: ✓ = supported; ✗ = not supported.

Appendix B. Derivation of Line Geometry Formulas

In this appendix, detailed derivations of the equations presented in the main text are provided.

Appendix B.1. Deviation of Equation (11)

This derivation shows the step-by-step procedure to obtain the closest point p^\perp :

$$p^\perp = p - (p \cdot \hat{z})\hat{z} \quad (\text{A1a})$$

$$= (\hat{z} \cdot \hat{z})p - (\hat{z} \cdot p)\hat{z} \text{ since } \hat{z} \text{ is a unit vector} \quad (\text{A1b})$$

$$= \hat{z} \times (p \times \hat{z}) \text{ using } a \times (b \times c) = (a \cdot c)b - (a \cdot b)c \quad (\text{A1c})$$

$$= \hat{z} \times v,$$

Appendix B.2. Deviation of Equation (13)

This derivation shows the step-by-step procedure to obtain the closest point t^\perp from an arbitrary point t :

$$t^\perp = p - ((p - t) \cdot \hat{z})\hat{z} \quad (\text{A2a})$$

$$= t + (\hat{z} \cdot \hat{z})(p - t) - ((p - t) \cdot \hat{z})\hat{z} \quad (\text{A2b})$$

$$= t + \hat{z} \times ((p - t) \times \hat{z}) \quad (\text{A2c})$$

$$= t + \hat{z} \times v_t, \quad (\text{A2d})$$

Appendix B.3. Deviation of Equation (16)

This derivation shows the procedure to calculate the moment vector v^\perp of the common perpendicular line connecting two lines L_1 and L_2 :

$$L_1 * L_2 = \hat{z}_1 \cdot v_2 + \hat{z}_2 \cdot v_1 \quad (\text{A3a})$$

$$= \hat{z}_1 \cdot (v_{21} + t_1 \times \hat{z}_2) + \hat{z}_2 \cdot v_1 \text{ using } a \cdot (b \times c) = (a \times b) \cdot c$$

$$= \hat{z}_1 \cdot v_{21} + (\hat{z}_1 \times (t_1 - p_1)) \cdot \hat{z}_2 \quad (\text{A3b})$$

$$= \hat{z}_1 \cdot v_{21} \quad (\text{A3c})$$

$$= \|\hat{z}_1\| \|v_{21}\| \cos\left(\alpha + \frac{\pi}{2}\right) \quad (\text{A3d})$$

$$= -\|v_{21}\| \sin \alpha, \quad (\text{A3e})$$

Appendix B.4. Deviation of Equation (20)

This derivation shows the procedure to calculate the moment vector v^\perp of the common perpendicular line connecting two lines L_1 and L_2 :

$$\begin{aligned}
v^\perp &= t_2 \times \hat{z}^\perp \\
&= \frac{t_2 \times (\hat{z}_1 \times \hat{z}_2)}{\|\hat{z}_1 \times \hat{z}_2\|} \text{ using } a \times (b \times c) = (a \cdot c)b - (a \cdot b)c \\
&= \frac{(t_2 \cdot \hat{z}_2)\hat{z}_1 - (t_2 \cdot \hat{z}_1)\hat{z}_2}{\|\hat{z}_1 \times \hat{z}_2\|} \\
&= \frac{((t_1 + t_2 - t_1) \cdot \hat{z}_2)\hat{z}_1 - (t_2 \cdot \hat{z}_1)\hat{z}_2}{\|\hat{z}_1 \times \hat{z}_2\|} \because (t_2 - t_1) \perp \hat{z}_2 \\
&= \frac{(\hat{z}_2 \cdot t_1)\hat{z}_1 - (\hat{z}_1 \cdot t_2)\hat{z}_2}{\|\hat{z}_1 \times \hat{z}_2\|} \text{ using } (a \cdot b)c = (a \cdot c)b - a \times (b \times c) \\
&= \frac{\hat{z}_1 \times v_2 - \hat{z}_2 \times v_1}{\|\hat{z}_1 \times \hat{z}_2\|} + \frac{\hat{z}_1 \cdot \hat{z}_2}{\|\hat{z}_1 \times \hat{z}_2\|}(t_1 - t_2) \text{ using } t_2 - t_1 = a\hat{n} = -\frac{\hat{z}_1 \cdot v_2 + \hat{z}_2 \cdot v_1}{\|\hat{z}_1 \times \hat{z}_2\|}\hat{n} \\
&= \frac{\hat{z}_1 \times v_2 - \hat{z}_2 \times v_1}{\|\hat{z}_1 \times \hat{z}_2\|} + \frac{(\hat{z}_1 \cdot \hat{z}_2)(\hat{z}_1 \cdot v_2 + \hat{z}_2 \cdot v_1)}{\|\hat{z}_1 \times \hat{z}_2\|^2}\hat{n}.
\end{aligned} \tag{A4}$$

Appendix B.5. Deviation of Equation (22)

This derivation shows how the scalar projection $t_2 \cdot \hat{n}$ of the intersection point t_2 onto the vector \hat{n} is computed:

$$t_2 = (t_2 \cdot \hat{z}_1)\hat{z}_1 + (t_2 \cdot \hat{u})\hat{u} + (t_2 \cdot \hat{n})\hat{n}. \tag{A5}$$

For the third term $(t_2 \cdot \hat{n})$ of the right-hand side of Equation (21), since we can observe the following

$$\begin{aligned}
v_2 \cdot \hat{z}_1 &= (t_2 \times \hat{z}_2) \cdot \hat{z}_1 \text{ using } (a \times b) \cdot c = a \cdot (b \times c) \\
&= -t_2 \cdot (\hat{z}_1 \times \hat{z}_2) \text{ using } \hat{n} = \frac{\hat{z}_1 \times \hat{z}_2}{\|\hat{z}_1 \times \hat{z}_2\|} \\
&= -(t_2 \cdot \hat{n})\|\hat{z}_1 \times \hat{z}_2\|,
\end{aligned}$$

then we have

$$\therefore t_2 \cdot \hat{n} = -\frac{v_2 \cdot \hat{z}_1}{\|\hat{z}_1 \times \hat{z}_2\|}. \tag{A6}$$

Appendix B.6. Deviation of Equation (23)

This derivation shows the procedure to calculate the projection of the intersection point t_2 onto the vector \hat{u} :

$$\begin{aligned}
t_2 \cdot \hat{u} &= t_2 \cdot (\hat{n} \times \hat{z}_1) \text{ using } a \cdot (b \times c) = (a \times b) \cdot c \\
&= (t_2 \times \hat{n}) \cdot \hat{z}_1 \text{ using } \hat{n} = \hat{z}^\perp = \frac{\hat{z}_1 \times \hat{z}_2}{\|\hat{z}_1 \times \hat{z}_2\|} \\
&= (t_2 \times \hat{z}^\perp) \cdot \hat{z}_1 \text{ using } v^\perp = t_2 \times \hat{z}^\perp \\
&= v^\perp \cdot \hat{z}_1 \text{ using Equation (20)} \\
&= -\frac{(\hat{z}_2 \times v_1) \cdot \hat{z}_1}{\|\hat{z}_1 \times \hat{z}_2\|} \\
&= -\frac{v_1 \cdot (\hat{z}_1 \times \hat{z}_2)}{\|\hat{z}_1 \times \hat{z}_2\|} \\
\therefore &= -v_1 \cdot \hat{n}.
\end{aligned} \tag{A7}$$

Appendix B.7. Deviation of Equation (25)

This derivation shows the procedure to calculate the scalar projection $t_2 \cdot \hat{z}_1$:

$$\begin{aligned} t_2 \cdot \hat{z}_1 &= \frac{(v_2 \cdot (\hat{z}_1 \times \hat{z}_2)) - (\hat{z}_1 \cdot \hat{z}_2)(v_1 \cdot (\hat{z}_1 \times \hat{z}_2))}{1 - (\hat{z}_1 \cdot \hat{z}_2)^2} \\ &= \frac{(v_2 \cdot (\hat{z}_1 \times \hat{z}_2)) - (\hat{z}_1 \cdot \hat{z}_2)(v_1 \cdot (\hat{z}_1 \times \hat{z}_2))}{\|\hat{z}_1 \times \hat{z}_2\|^2} \\ \therefore &= \frac{(v_2 - (\hat{z}_1 \cdot \hat{z}_2)v_1) \cdot \hat{n}}{\|\hat{z}_1 \times \hat{z}_2\|}. \end{aligned} \quad (\text{A8})$$

Appendix B.8. Deviation of Equation (26)

This derivation shows the procedure to obtain the explicit expression for intersection point t_2 :

$$\begin{aligned} t_2 &= (t_2 \cdot \hat{z}_1)\hat{z}_1 + (t_2 \cdot \hat{u})\hat{u} + (t_2 \cdot \hat{n})\hat{n} \\ &= \frac{(v_2 - (\hat{z}_1 \cdot \hat{z}_2)v_1) \cdot \hat{n}}{\|\hat{z}_1 \times \hat{z}_2\|}\hat{z}_1 - (v_1 \cdot \hat{n})\hat{u} - \frac{v_2 \cdot \hat{z}_1}{\|\hat{z}_1 \times \hat{z}_2\|}\hat{n} \\ &= \frac{(v_2 \cdot \hat{n})\hat{z}_1 - (v_2 \cdot \hat{z}_1)\hat{n}}{\|\hat{z}_1 \times \hat{z}_2\|} - (v_1 \cdot \hat{n})\left(\frac{(\hat{z}_1 \cdot \hat{z}_2)\hat{z}_1}{\|\hat{z}_1 \times \hat{z}_2\|} + (\hat{n} \times \hat{z}_1)\right) \\ &= \frac{v_2 \times (\hat{z}_1 \times \hat{n})}{\|\hat{z}_1 \times \hat{z}_2\|} - (v_1 \cdot \hat{n})\left(\frac{(\hat{z}_1 \cdot \hat{z}_2)\hat{z}_1 + (\hat{z}_1 \times \hat{z}_2) \times \hat{z}_1}{\|\hat{z}_1 \times \hat{z}_2\|}\right) \\ \therefore &= \frac{v_2 \times (\hat{z}_1 \times (\hat{z}_1 \times \hat{z}_2)) - (v_1 \cdot (\hat{z}_1 \times \hat{z}_2))\hat{z}_2}{\|\hat{z}_1 \times \hat{z}_2\|^2}. \end{aligned} \quad (\text{A9})$$

References

- Featherstone, R. Robot Dynamics Algorithms. Ph.D. Thesis, University of Edinburgh, Edinburgh, UK, 1984.
- Davidson, J.K.; Hunt, K.H.; Pennock, G.R. Robots and screw theory: Applications of kinematics and statics to robotics. *J. Mech. Des.* **2004**, *126*, 763–764. [[CrossRef](#)]
- Murray, R.M.; Li, Z.; Sastry, S.S. *A Mathematical Introduction to Robotic Manipulation*; CRC Press: Boca Raton, FL, USA, 2017.
- Lynch, K.M.; Park, F.C. *Modern Robotics*; Cambridge University Press: Cambridge, UK, 2017.
- Hayati, S.; Mirmirani, M. Improving the absolute positioning accuracy of robot manipulators. *J. Robot. Syst.* **1985**, *2*, 397–413. [[CrossRef](#)]
- Denavit, J.; Hartenberg, R.S. A kinematic notation for lower-pair mechanisms based on matrices. *J. Appl. Mech.* **1955**, *22*, 215–221. [[CrossRef](#)]
- Hartenberg, R.; Danavit, J. *Kinematic Synthesis of Linkages*; McGraw-Hill: New York, NY, USA, 1964.
- Kahn, M.E.; Roth, B. The near-minimum-time control of open-loop articulated kinematic chains. *J. Dyn. Sys. Meas. Control* **1971**, *93*, 164–172. [[CrossRef](#)]
- Featherstone, R. *A Program for Simulating Robot Dynamics*; Working Paper 155; Department of Artificial Intelligence, University of Edinburgh: Edinburgh, UK, 1982.
- Craig, J.J. *Introduction to Robotics*; Pearson Education: London, UK, 2006.
- Lipkin, H. A note on Denavit-Hartenberg notation in robotics. In Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Long Beach, CA, USA, 24–28 September 2005.
- Waldron, K.J.; Schmiedeler, J. *Chapter 1 Kinematics, Handbook of Robotics*; Springer: Berlin/Heidelberg, Germany, 2008.
- Corke, P.I. A Simple and Systematic Approach to Assigning Denavit–Hartenberg Parameters. *IEEE T. Rob.* **2007**, *23*, 590–594. [[CrossRef](#)]
- Lin, Y.; Zhao, H.; Ye, C.; Ding, H. A computationally efficient and robust kinematic calibration model for industrial robots with kinematic parallelogram. In Proceedings of the 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), Macau, China, 5–8 December 2017; IEEE: Piscataway, NJ, USA, 2017.
- Faria, C.; Vilaça, J.L.; Monteiro, S.; Erlhagen, W.; Bicho, E. Automatic Denavit-Hartenberg Parameter Identification for Serial Manipulators. In Proceedings of the IECON 2019—45th Annual Conference of the IEEE Industrial Electronics Society, Lisbon, Portugal, 14–17 October 2019; Volume 1, pp. 610–617. [[CrossRef](#)]
- Rajeevlochana, C.G.; Saha, S.K.; Kumar, S. Automatic Extraction of DH Parameters of Serial Manipulators using Line Geometry. In Proceedings of the 2nd Joint International Conference on Multibody System Dynamics, Stuttgart, Germany, 29 May–1 June 2012.

17. Huang, Q.; Zhang, X.; Pan, H.; Lu, K.; Cheng, Y. POE-based kinematic calibration for serial robots using left-invariant error representation and decomposed iterative method. *Robot. Auton. Syst.* **2025**, *186*, 104896. [[CrossRef](#)]
18. Yuan, Y.; Sun, W. An integrated kinematic calibration and dynamic identification method with only static measurements for serial robot. *IEEE/ASME Trans. Mechatron.* **2023**, *28*, 2762–2773. [[CrossRef](#)]
19. He, R.; Zhao, Y.; Yang, S.; Yang, S. Kinematic-parameter identification for serial-robot calibration based on POE formula. *IEEE Trans. Robot.* **2010**, *26*, 411–423.
20. Li, C.; Wu, Y.; Löwe, H.; Li, Z. POE-based robot kinematic calibration using axis configuration space and the adjoint error model. *IEEE Trans. Robot.* **2016**, *32*, 1264–1279. [[CrossRef](#)]
21. Iserles, A.; Munthe-Kaas, H.Z.; Nørsett, S.P.; Zanna, A. Lie-group methods. *Acta Numer.* **2000**, *9*, 215–365. [[CrossRef](#)]
22. Bullo, F.; Murray, R.M. *Proportional Derivative (PD) Control on the Euclidean Group*; CDS Technical Report 95-010; California Institute of Technology: Pasadena, CA, USA, 1995.
23. Hodge, W.V.D.; Pedoe, D. *Methods of Algebraic Geometry*; Cambridge University Press: Cambridge, UK, 1994; Volume 2.
24. Mason, M.T. *Mechanics of Robotic Manipulation*; MIT Press: Cambridge, MA, USA, 2001.
25. Jia, Y.B. Plücker coordinates for lines in the space. In *Problem Solver Techniques for Applied Computer Science, Com-S-477/577 Course Handout*; Iowa State University: Ames, Iowa, 2020.
26. Sung, M. Algorithmic MDH. Available online: <https://github.com/MinchangSung0223/AlgorithmicMDH> (accessed on 27 April 2025).
27. Coumans, E.; Bai, Y. Pybullet, a Python Module for Physics Simulation for Games, Robotics and Machine Learning. 2016. Available online: <http://pybullet.org> (accessed on 27 April 2025).
28. ROS.org. URDF. Available online: <http://wiki.ros.org/urdf> (accessed on 27 November 2023).
29. Neuromeka. Indy7 URDF. Available online: <https://github.com/neuromeka-robotics/indy-ros> (accessed on 27 November 2023).
30. Tadese, M.; Pico, N.; Seo, S.; Moon, H. A Two-Step Method for Dynamic Parameter Identification of Indy7 Collaborative Robot Manipulator. *Sensors* **2022**, *22*, 9708. [[CrossRef](#)] [[PubMed](#)]
31. KUKA. KUKA lbr iiwa URDF. Available online: https://github.com/rkg/lbr_iowa (accessed on 27 November 2023).
32. Universal Robotics. UR5 URDF. Available online: https://github.com/ros-industrial/universal_robot (accessed on 27 November 2023).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.