

URDF로부터 DH 파라미터를 구성하는 일반적인 방법

A Universal Method for Constructing DH parameters from Unified Robot Description Format

유병기¹·이준영¹·박상현¹·김무림[†]

Byeonggi Yu¹, Junyoung Lee¹, Sang hyun Park¹, Maolin Jin[†]

Abstract: This paper introduced how to construct Denavit-Hartenberg (DH) parameters from the Unified Robot Description Format (URDF). URDF is convenient for describing a robot even though the robot is very complex. On the other hand, DH convention is not an easy notation for many novices who want to describe a robot. Therefore, most vendors provide URDF and users prefer to use URDF to describe a robot. However, some controllers or algorithms are based on DH parameters to perform kinematics, dynamics, control, etc. To connect URDF and DH parameters, we present a three-step approach to construct DH parameters from URDF. The first step is to define the joint axis for constructing DH parameters. The second step is constructing DH parameters to define joint character. The final step is constructing DH parameters to define the coordinate frame of the child link. This approach is based on intuitive vector calculation and guarantees the uniqueness of DH parameters. To verify our approach, we applied our approach to a simple one-link robot, a manipulator with 6 DOF, and a quadruped robot with 3 DOF per leg. We verified that our approach worked well based on forward kinematic results.

Keywords: Denavit-Hartenberg (DH) Parameters, Unified Robot Description Format (URDF), Robot Manipulator

1. 서 론

Unified Robot Description Format (URDF)는 eXtensible Markup Language (XML)를 이용하여 로봇의 링크와 관절 정보 등을 기술하는 형식으로 최근에 많이 사용되고 있는 형식이다^[1]. URDF는 2개의 좌표계의 관계를 6개의 파라미터를 사용하여 정의하고 있으며 3차원 벡터로 정의된 임의 방향의 관절축을 정의할 수 있다.

한편 로봇공학 분야에서는 지난 수십 년간 기구학 등을 계산하기 위해 Denavit-Hartenberg (DH) 파라미터 표시방법을

사용해왔다^[2]. DH 표시방법은 4개의 파라미터(관절축의 회전 각도(θ_i), 관절축 방향 원점의 오프셋(d_i), 링크의 길이(a_i), 그리고 관절축의 비틀림각(α_i))와 z축 관절을 이용하여 두 좌표계(coordinate frame) 사이의 관계를 정의한다^[3]. 삼차원 공간에서 위치와 방향을 정의하기 위한 6개의 자유도를 단 4개의 파라미터로 정의하기 때문에 로봇을 간결하게 기술할 수 있는 장점이 있다.

URDF와 DH 파라미터 두 방식 모두 로봇 기구학을 기술하는데 사용될 수 있으나 편의성 측면에서 차이점이 존재한다. 먼저 DH 파라미터 표시방법은 2개의 구속조건(i.e. DH 조건)이 필요하여^[4] 주어진 로봇에 제한적으로 좌표계를 할당해야 한다. 특정 경우에는 구속조건을 만족시키기 위해 좌표계를 로봇 바디 외부에 배치하게 되므로 직관적이지 않은 문제가 있으며, 때로는 기구학적 정보가 필요한 위치에 좌표계를 지정하지 못하여 따로 좌표계를 지정하여 계산해야 하는 문제가 있다. 그에 반해 URDF는 DH 표시 방법과 달리 링크의 좌표계를 지정하는데 아무런 제한 조건이 없으며 관절의 축 방향 또한 제한이 없다. 결과적으로 로봇 공학에 익숙하지 않은 사용

Received : Nov. 16. 2022; Revised : Dec. 15. 2022; Accepted : Jan. 9. 2023

※ This work (research) was supported by ITECH R&D program of MOTIE/KEIT. (Project No. 20014398 Development of a kinematically versatile, easy-to-use, safety-supported, AI-integrable, and high speed (≥ 5 kHz) robot controller)

1. Senior Researcher, KIRO, Pohang, Korea (byeonggi, junyoung, shpark@kiro.re.kr)

† Chief Researcher, Corresponding author: Director in Human-Centered Robotics Research Division, KIRO, Pohang, Korea (mulimkim@kiro.re.kr)

자라도 기구학적 특성을 알고 싶은 위치에 좌표계를 지정하고 관절 종류와 축 방향을 지정하여 용이하게 로봇을 URDF로 기술할 수 있다. 뿐만 아니라 사용자가 3D CAD 프로그램에서 설계한 로봇에 링크 좌표계와 관절축을 지정하여 자동으로 URDF를 생성할 수도 있다⁵⁾.

사용자 입장에서는 URDF는 DH 파라미터 표시방법에 비해 직관적이고 편리하게 로봇 기구학을 기술 할 수 있는 방법이다. 그러나 알고리즘 호환성, 소프트웨어 지원, 팀 협동, 단순화 및 계산 효율성 등의 이유 때문에 DH 파라미터로 로봇을 기술할 필요가 있다. DH 파라미터 표시방법은 로봇공학에서 오랫동안 활용되어 로봇 공학자들에 익숙하고, 많은 연구자들이 DH 파라미터에 기반하여 기구학, 동역학 라이브러리를 만들고 역기구학의 폐쇄형 해 존재 여부 판단⁶⁾, 캘리브레이션 알고리즘⁷⁾ 등 방대한 자료를 이미 구축하였다. 이런 자료를 활용하려면 사용자 편의성이 높은 URDF로 로봇을 기술하고, 이 기술된 정보를 DH 파라미터 기반으로 재구성 해야한다.

URDF를 DH 파라미터로 재구성하는 작업은 시간이 많이 드는 작업이며, 만약 이 두가지 방법에 모두 익숙하지 않다면 변경 작업을 할 때 치명적 오류를 범할 수 있다. 따라서 DH 파라미터를 활용하고 있는 라이브러리나 제어시스템에 3D CAD 등에서 용이하게 생성 가능한 URDF를 적용 할 수 있도록 URDF 파일에서 DH 파라미터를 자동으로 얻는 것이 필요하다.

URDF에서 DH 파라미터를 구성하는 일은 쉽지 않다. URDF의 좌표계를 DH 조건에 맞게 재작성해야 하며, 로봇의 자유도가 클수록 URDF에서 DH 파라미터를 구성하는 일은 복잡해지고 많은 시간이 소요된다. 또한 임의 방향의 관절축이 이미 선언되었다면 간단한 구성을 가진 로봇이라도 그에 대응하는 DH 파라미터를 구성하는 일은 간단하지 않다.

URDF에서 DH 파라미터를 구성하는 기존 연구를 살펴보면 URDF에서 할당된 좌표계 대신 DH 조건을 만족하는 다른 좌표계를 도입하여 4개의 DH 파라미터를 구한다⁸⁾. 즉 원본 좌표계의 정보를 구하지 못한다. 또한 공개된 변환 프로그램은 DH 조건이 만족된 URDF를 요구하기 때문에⁹⁾ DH 조건을 만족하지 않는 대부분의 URDF에서는 DH 파라미터를 얻을 수 없다¹⁰⁾. 따라서, 일반적인 URDF 파일로부터 DH 조건을 만족하지 않더라도 자동으로 DH 파라미터를 얻을 수 있어야 한다.

본 논문에서는 URDF 파일에서 DH 조건이 만족되지 않아도 DH 파라미터를 구성할 수 있는 방법을 제안한다. 2장에서는 DH 파라미터 구성 방법을 3단계로 나누어 설명한다. 3장에서는 실제 DH 조건을 만족하지 않는 URDF에서도 DH 파라미터를 구성할 수 있음을 보이고 정기구학 계산을 수행하여 제안한 방법의 타당성을 확인한다. 본 논문의 결론은 4장에서 서술한다.

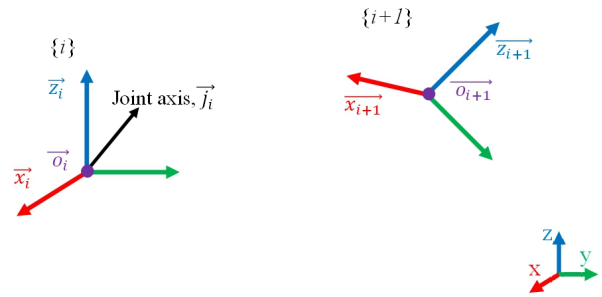
2. URDF→DH 파라미터 구성 방법

2.1 문제 정의

URDF에서 DH 파라미터를 구성하기 위해서는 임의 방향의 관절 축에 의한 두 좌표계 사이의 관계를 찾아야 한다. 이를 위해 [Fig. 1]을 바탕으로 임의 방향의 관절 axis \mathbf{j}_i 에 대하여 좌표계 $\{i\}$ 와 좌표계 $\{i+1\}$ 의 관계를 3 단계를 거쳐 확인하며, 좌표계 $\{i\}$ 는 $\{i\}$ 로 표기하였다. 본 논문에서 사용된 표기법은 [Table 1]에 정리하였다.

2.2 Step 1: 관절 축 정의를 위한 DH 파라미터 구성

첫번째 단계로 URDF에서 관절 축 \mathbf{j}_i 는 임의의 방향을 가질 수 있기 때문에 \mathbf{j}_i 를 z축으로 가지는 새로운 좌표계가 필요한데 [Fig. 2]에 이 과정을 나타내었다. 먼저 \mathbf{x}_{i+1} 을 도입하였는데 \mathbf{x}_{i+1} 는 관절 축 \mathbf{j}_i 와 $\{i\}$ 의 z축의 벡터곱(cross product)로 정의



[Fig. 1] Two coordinate frames and two arbitrary joint axes. To construct DH parameters from URDF, \mathbf{x}_i , \mathbf{z}_i , \mathbf{o}_i , \mathbf{x}_{i+1} , \mathbf{z}_{i+1} , \mathbf{o}_{i+1} , and \mathbf{j}_i are required. \mathbf{x}_i , \mathbf{z}_i , \mathbf{x}_{i+1} , \mathbf{z}_{i+1} , and \mathbf{j}_i are herein unit vectors

[Table 1] List of Notations used in this paper

$\{i\}$	Coordinate frame i
\mathbf{j}_i	Joint axis at $\{i\}$
\mathbf{x}_i	x axis at $\{i\}$
\mathbf{z}_i	z axis at $\{i\}$
\mathbf{o}_i	origin at $\{i\}$
\mathbf{x}_{it1}	x axis at $\{it1\}$
\mathbf{x}_{it2}	x axis at $\{it2\}$
\mathbf{o}_{it1}	origin at $\{it1\}$
\mathbf{o}_{it2}	origin at $\{it2\}$
α_{i1}, θ_{i1}	DH parameters from $\{i\}$ to $\{it1\}$
$\theta_{i2}, d_{i2}, a_{i2}, \alpha_{i2}$	DH parameters from $\{it1\}$ to $\{it2\}$
θ_{i3}, d_{i3}	DH parameters from $\{it2\}$ to $\{i+1\}$

하였다. 만약 \mathbf{j}_i 가 고정 관절로 0-벡터라면 $\{i\}$ 의 z 축을 이용하여 \mathbf{j}_i 를 정의할 수 있다. \mathbf{x}_{it1} 가 구해지면 DH 파라미터 중 α 와 θ 를 구할 수 있게 된다. 이때 조인트 오프셋과 링크 길이가 존재하지 않기 때문에 d 와 a 는 0이 된다. \mathbf{x}_{it1} 의 수식과, 부호를 고려한 α_{i1} 와 θ_{i1} 의 수식은 다음과 같다.

$$\mathbf{x}_{it1} = \mathbf{j}_i \times \mathbf{z}_i \quad (1)$$

$$\cos \alpha_{i1} = \mathbf{j}_i \cdot \mathbf{z}_i \quad (2)$$

$$\sin \alpha_{i1} = (\mathbf{j}_i \times \mathbf{z}_i) \cdot \mathbf{x}_{it1} \quad (3)$$

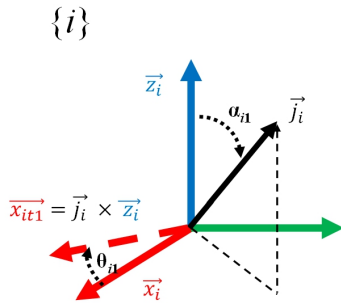
$$\cos \theta_{i1} = \mathbf{x}_i \cdot \mathbf{x}_{it1} \quad (4)$$

$$\sin \theta_{i1} = (\mathbf{x}_i \times \mathbf{x}_{it1}) \cdot \mathbf{z}_i \quad (5)$$

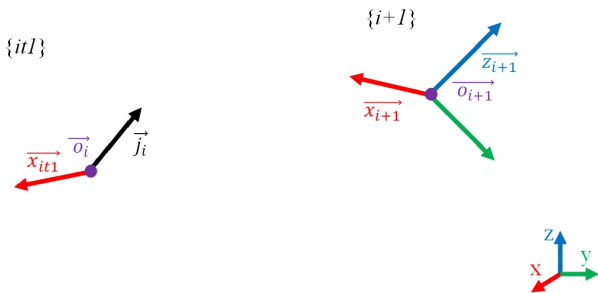
결과적으로 임의의 방향을 가지는 관절 축을 표현하기 위해서 2개의 파라미터가 필요하며 제시된 수식을 통해 찾을 수 있다. 이러한 표현 방법은 구면좌표계(spherical coordinate system)의 표현과 동일하다.

2.3 Step 2: 관절 특성 DH 파라미터 구성

관절 축을 z축으로 가지는 새로운 좌표계 $\{it1\}$ 을 지정하게 되



[Fig. 2] Step 1: DH parameters to define the joint axis. \mathbf{x}_{it1} is introduced to find α_{i1} and θ_{i1}



[Fig. 3] Step 2: DH parameters for joint characteristics. Using constructed $\{it1\}$, DH parameters for joint characteristics can be constructed

면 이제는 관절 축의 특성(e.g. 회전(rotation), 병진(translation) etc.)을 표현하는 관계를 찾을 수 있게 된다[Fig. 3]. 그러나 $\{i+1\}$ 의 x축, \mathbf{x}_{i+1} 이 \mathbf{j}_i 과 수직이고 교차해야만(i.e. DH 조건) 직접적으로 DH 파라미터를 찾을 수 있다. 그렇지 않을 경우 $\{it1\}$ 를 생성한 경우와 동일하게 또다른 $\{it2\}$ 를 중간에 지정해야 하며, 이때 3가지 경우의 수가 존재한다.

2.3.1 Case 1: Skew lines

첫번째 경우는 가장 일반적인 경우로 관절 축 \mathbf{j}_i 와 \mathbf{z}_{i+1} 이 비스듬한(skew)인 경우이다[Fig. 4]. 두 직선이 서로 평행(parallel)하지도 교차(intersecting)하지도 않은 경우, 두 직선과 동시에 수직인 유일한(unique) 직선이 존재한다^[11]. 이 직선은 유일하기 때문에 유일한 DH 파라미터를 제공할 수 있다. 이 직선을 찾기 위해서는 관절 축 \mathbf{j}_i 와 \mathbf{z}_{i+1} 이 만들어낸 두 직선과 동시에 수직으로 만나는 직선의 위치 벡터(\mathbf{o}_{it1} , \mathbf{o}_{it2}), 2개를 찾아내야 한다. 관절 축 \mathbf{j}_i 와 \mathbf{z}_{i+1} 가 만들어내는 두 직선의 방정식은 다음과 같다.

$$\mathbf{r}_1 = \lambda_a \mathbf{j}_i + \mathbf{o}_i \quad (6)$$

$$\mathbf{r}_2 = \lambda_b \mathbf{z}_{i+1} + \mathbf{o}_{i+1} \quad (7)$$

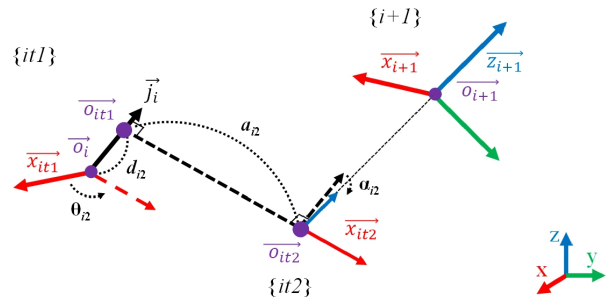
\mathbf{o}_{it1} , \mathbf{o}_{it2} 가 각각 \mathbf{r}_1 , \mathbf{r}_2 위에 있는 위치 벡터들이므로 식 (6,7)과 동시에 수직으로 만나는 조건을 만족시킬 수 있는 λ_1 , λ_2 를 찾으면 \mathbf{o}_{it1} , \mathbf{o}_{it2} 는 식 (8,9)에 의해 결정된다.

$$\mathbf{o}_{it1} = \lambda_1 \mathbf{j}_i + \mathbf{o}_i \quad (8)$$

$$\mathbf{o}_{it2} = \lambda_2 \mathbf{z}_{i+1} + \mathbf{o}_{i+1} \quad (9)$$

두 직선과 수직으로 만나는 직선의 조각, \mathbf{r}_3 는 $\mathbf{o}_{it2} - \mathbf{o}_{it1}$ 을 이용하여 구할 수 있다(식 (10)).

$$\mathbf{r}_3 = \mathbf{o}_{it2} - \mathbf{o}_{it1} = \lambda_2 \mathbf{z}_{i+1} + \mathbf{o}_{i+1} - (\lambda_1 \mathbf{j}_i + \mathbf{o}_i) \quad (10)$$



[Fig. 4] Case 1: Skew lines

\mathbf{r}_3 를 구하게 되면 결과적으로 \mathbf{x}_{it2} 를 구할 수 있게 된다(식 (11)). \mathbf{r}_3 가 $\mathbf{r}_1, \mathbf{r}_2$ 와 각각 수직인 조건을 이용하여 두개의 구속 조건, 식(12,13)를 구할 수 있다. 식(12,13)을 λ_1, λ_2 로 묶어 정리하면 연립 선형 방정식(simultaneous linear equation), 식(14)을 구할 수 있다. 이를 이용하여 식(15)에서 λ_1, λ_2 를 찾아내어 $\mathbf{o}_{it1}, \mathbf{o}_{it2}$ 를 구할 수 있다.

$$\mathbf{x}_{it2} = \mathbf{r}_3 / \|\mathbf{r}_3\| \quad (11)$$

$$\mathbf{r}_3 \cdot \mathbf{j}_i = (\lambda_2 \mathbf{z}_{i+1} + \mathbf{o}_{i+1} - (\lambda_1 \mathbf{j}_i + \mathbf{o}_i)) \cdot \mathbf{j}_i = 0 \quad (12)$$

$$\mathbf{r}_3 \cdot \mathbf{z}_{i+1} = (\lambda_2 \mathbf{z}_{i+1} + \mathbf{o}_{i+1} - (\lambda_1 \mathbf{j}_i + \mathbf{o}_i)) \cdot \mathbf{z}_{i+1} = 0 \quad (13)$$

$$\begin{bmatrix} -\|\mathbf{j}_i\|^2 & \mathbf{j}_i \cdot \mathbf{z}_{i+1} \\ -\mathbf{j}_i \cdot \mathbf{z}_{i+1} & \|\mathbf{z}_{i+1}\|^2 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} -(\mathbf{o}_{i+1} - \mathbf{o}_i) \cdot \mathbf{j}_i \\ -(\mathbf{o}_{i+1} - \mathbf{o}_i) \cdot \mathbf{z}_{i+1} \end{bmatrix} \quad (14)$$

$$\begin{aligned} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} &= \begin{bmatrix} -\|\mathbf{j}_i\|^2 & \mathbf{j}_i \cdot \mathbf{z}_{i+1} \\ -\mathbf{j}_i \cdot \mathbf{z}_{i+1} & \|\mathbf{z}_{i+1}\|^2 \end{bmatrix}^{-1} \begin{bmatrix} -(\mathbf{o}_{i+1} - \mathbf{o}_i) \cdot \mathbf{j}_i \\ -(\mathbf{o}_{i+1} - \mathbf{o}_i) \cdot \mathbf{z}_{i+1} \end{bmatrix} \\ &= \begin{bmatrix} -1 & \mathbf{j}_i \cdot \mathbf{z}_{i+1} \\ -\mathbf{j}_i \cdot \mathbf{z}_{i+1} & 1 \end{bmatrix}^{-1} \begin{bmatrix} -(\mathbf{o}_{i+1} - \mathbf{o}_i) \cdot \mathbf{j}_i \\ -(\mathbf{o}_{i+1} - \mathbf{o}_i) \cdot \mathbf{z}_{i+1} \end{bmatrix} \quad (15) \end{aligned}$$

식(15)에서 역행렬(inverse matrix)이 존재하기 위해서는 행렬식(determinant)이 0이 아니어야 하며, 다시 말하면 $\mathbf{j}_i \cdot \mathbf{z}_{i+1} \neq 1$ 의 조건을 만족하면 되는데 이 조건은 \mathbf{j}_i 와 \mathbf{z}_{i+1} 가 평행하지 않다는 뜻이다. 위 식은 비스듬한 조건일 때 사용하므로 역행렬은 항상 존재한다.

[Fig. 4]에서 4개의 DH 파라미터를 시각적으로 표현하였으며 이들을 구하는 식은 다음과 같다.

$$\cos \alpha_{i2} = \mathbf{j}_i \cdot \mathbf{z}_{i+1} \quad (16)$$

$$\sin \alpha_{i2} = (\mathbf{j}_i \times \mathbf{z}_{i+1}) \cdot \mathbf{x}_{it2} \quad (17)$$

$$\cos \theta_{i2} = \mathbf{x}_{it1} \cdot \mathbf{x}_{it2} \quad (18)$$

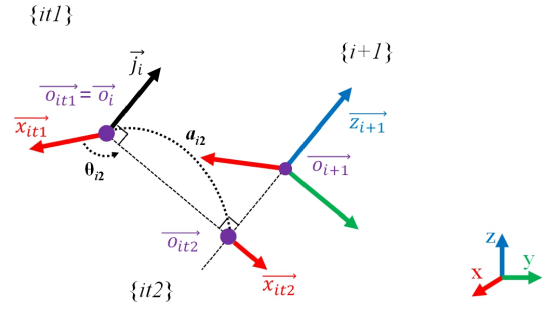
$$\sin \theta_{i2} = (\mathbf{x}_{it1} \times \mathbf{x}_{it2}) \cdot \mathbf{j}_i \quad (19)$$

$$d_{i2} = (\mathbf{o}_{it1} - \mathbf{o}_i) \cdot \mathbf{j}_i \quad (20)$$

$$a_{i2} = (\mathbf{o}_{it2} - \mathbf{o}_{it1}) \cdot \mathbf{x}_{it2} \quad (21)$$

2.3.2 Case 2: Parallel lines

식(15)은 \mathbf{j}_i 와 \mathbf{z}_{i+1} 가 평행하면(i.e. $\mathbf{j}_i \times \mathbf{z}_{i+1} = 0$) 사용이 불가능하므로 다른 접근 방법이 필요하다. [Fig. 5]에서 이러한 경우를 보여주고 있다. \mathbf{j}_i 와 \mathbf{z}_{i+1} 가 평행할 경우 두 직선을 유



[Fig. 5] Case 2: Parallel lines

일하게 연결해주는 공통 법선(i.e. common normal line)이 없으므로 임의로 특정한 직선을 지정해야 한다. 이를 위하여 \mathbf{o}_{it1} 을 \mathbf{o}_i 로 사용하여 유일한 직선을 설정하였다. 이러한 조건에서 $d = 0$ 이 된다. 또한 \mathbf{j}_i 와 \mathbf{z}_{i+1} 이 평행하므로 $\alpha = 0$ 이 된다. 나머지 a 와 θ 를 구하기 위해 필요하는 식들은 아래와 같다.

$$\mathbf{o}_{it1} = \mathbf{o}_i \quad (22)$$

$$\mathbf{o}_{it2} = \mathbf{o}_{i+1} - ((\mathbf{o}_{i+1} - \mathbf{o}_i) \cdot \mathbf{z}_{i+1}) \mathbf{z}_{i+1} \quad (23)$$

$$\mathbf{r}_3 = \mathbf{o}_{it2} - \mathbf{o}_{it1} \quad (24)$$

$$\mathbf{x}_{it2} = \mathbf{r}_3 / \|\mathbf{r}_3\| \quad (25)$$

$$\cos \theta_{i2} = \mathbf{x}_{it1} \cdot \mathbf{x}_{it2} \quad (26)$$

$$\sin \theta_{i2} = (\mathbf{x}_{it1} \times \mathbf{x}_{it2}) \cdot \mathbf{j}_i \quad (27)$$

$$a_{i2} = (\mathbf{o}_{it2} - \mathbf{o}_{it1}) \cdot \mathbf{x}_{it2} \quad (28)$$

2.3.2.1 Special case: Coincidence lines

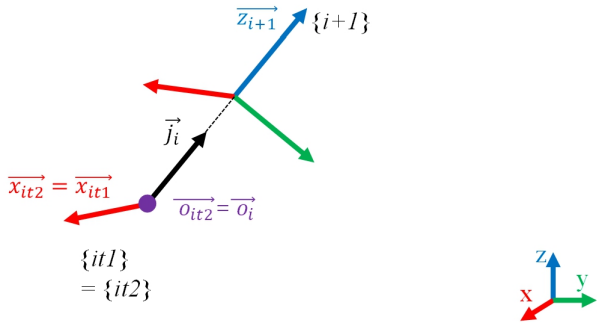
평행 보다 더 특수한 경우로, \mathbf{j}_i 와 \mathbf{z}_{i+1} 이 같은 직선 상에 놓여져 있을 수 있다(i.e. $\mathbf{j}_i \times \mathbf{z}_{i+1} = 0$) [Fig. 6]. 이 경우 DH 파라미터를 전부 0으로 두어 다음 단계를 위한 {it2}를 {it1}과 동일하게 만들었다.

$$\mathbf{o}_{it2} = \mathbf{o}_i \quad (29)$$

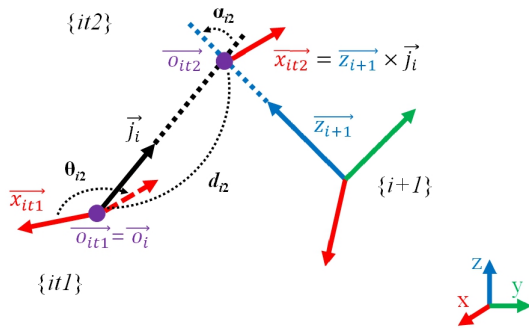
$$\mathbf{x}_{it2} = \mathbf{x}_{it1} \quad (30)$$

2.3.3 Case 3: Intersecting lines

\mathbf{j}_i 와 \mathbf{z}_{i+1} 가 교차하는 경우(i.e. $(\mathbf{j}_i \times \mathbf{z}_{i+1}) \cdot (\mathbf{o}_{i+1} - \mathbf{o}_i) = 0$) [Fig. 7]에도 유일하게 결정되는 하나의 점을 찾을 수 있으므로 이 점에서 새로운 {it2}를 결정하여 DH 파라미터를 결정할 수 있다. 두 라인 $\mathbf{r}_1, \mathbf{r}_2$ 이 교차할 경우 식(6)과 식(7)이 같



[Fig. 6] Special case: Coincidence lines



[Fig. 7] Case 3: Intersecting lines

은 값을 가지도록 만드는 λ_1, λ_2 를 찾아 교차점을 찾을 수 있다. 또한 교차하는 경우 링크 길이, a 는 항상 0이 된다. DH 파라미터를 구하기 위한 식들은 다음과 같다.

$$\mathbf{o}_{it1} = \mathbf{o}_i \quad (31)$$

$$\mathbf{x}_{it2} = \mathbf{z}_{i+1} \times \mathbf{j}_i \quad (32)$$

$$\lambda_1 \mathbf{z}_{i+1} + \mathbf{o}_{i+1} = \lambda_2 \mathbf{j}_i + \mathbf{o}_i \quad (33)$$

$$\begin{bmatrix} \mathbf{z}_{i+1} & -\mathbf{j}_i \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = [-(\mathbf{o}_{i+1} - \mathbf{o}_i)] \quad (34)$$

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = [\mathbf{z}_{i+1} \quad -\mathbf{j}_i]^{-1} [-(\mathbf{o}_{i+1} - \mathbf{o}_i)] \quad (35)$$

$$\mathbf{o}_{it2} = \lambda_2 \mathbf{z}_{i+1} + \mathbf{o}_{i+1} \quad (36)$$

식 (34)가 여분(redundant)하게 보일 수 있으나(x, y, z 축 3개의 식) 두 선이 교차하는 경우 자유도를 하나 잃기 때문에 풀랭크 행렬(full rank matrix)가 된다. 또한 행렬식이 0인 경우(i.e. $\mathbf{j}_i \times \mathbf{z}_{i+1} = 0$, 평행조건)는 교차하는 경우에는 발생하지 않으므로 항상 역행렬이 존재하며 유일해(unique solution)가 존재한다. 결과적으로 DH 파라미터 들을 구하기 위한 식들은 다음과 같다.

$$\cos \alpha_{i2} = \mathbf{j}_i \cdot \mathbf{z}_{i+1} \quad (37)$$

$$\sin \alpha_{i2} = (\mathbf{j}_i \times \mathbf{z}_{i+1}) \cdot \mathbf{x}_{it2} \quad (38)$$

$$\cos \theta_{i2} = \mathbf{x}_{it1} \cdot \mathbf{x}_{it2} \quad (39)$$

$$\sin \theta_{i2} = (\mathbf{x}_{it1} \times \mathbf{x}_{it2}) \cdot \mathbf{j}_i \quad (40)$$

$$d_{i2} = (\mathbf{o}_{it2} - \mathbf{o}_{it1}) \cdot \mathbf{j}_i \quad (41)$$

2.4 Step 3: 자식 링크 DH 파라미터 구성

Step 2를 통해 {it2}는 {i+1}과 동일 방향의 coaxial z축을 가지게 된다[Fig. 8]. 즉 링크 길이와 비틀림 각은 존재하지 않고 오로지 조인트 각과 링크 오프셋만을 가진다. DH 파라미터를 구하기 위한 식들은 다음과 같다.

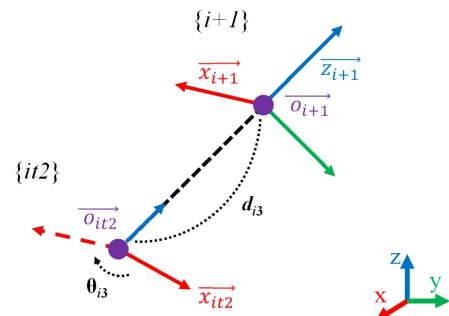
$$\cos \theta_{i3} = \mathbf{x}_{i+1} \cdot \mathbf{x}_{it2} \quad (42)$$

$$\sin \theta_{i3} = (\mathbf{x}_{i+1} \times \mathbf{x}_{it2}) \cdot \mathbf{z}_{i+1} \quad (43)$$

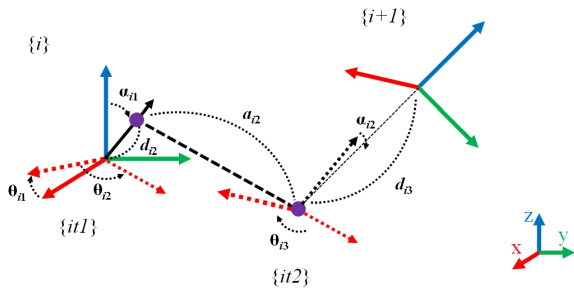
$$d_{i3} = (\mathbf{o}_{i+1} - \mathbf{o}_{it2}) \cdot \mathbf{z}_{i+1} \quad (44)$$

2.5 요약

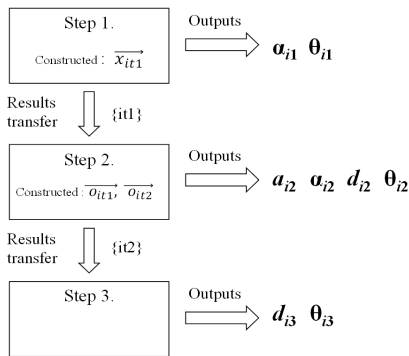
[Fig. 9]에서 모든 단계를 합친 그림을 볼 수 있다. 각 단계에서 어떤 벡터가 DH 파라미터를 구하기 위해 구성되고 또한 결과로 어떤 DH 파라미터가 계산되는지 [Fig. 10]에서 확인할 수 있다. 결론적으로 {i}와 {i+1}의 관계는 총 8개의 DH 파라미터로 결정이 될 수 있다. 또한 중간 더미 좌표계, {it1}과 {it2} 사이에서만 관절 변수가 도입이 되며 나머지 관계들, {i}와 {it1}, {it2}와 {i+1}은 고정 관절로 변수가 없는 동차행렬변환(homogeneous transformation matrix)를 가지게 된다[Fig. 11]. 이러한 접근 방법을 이용하게 되면 URDF에서 정의된 부모 링크와 자식 링크 사이의 관계를 DH 파라미터로 구성할 수 있다.



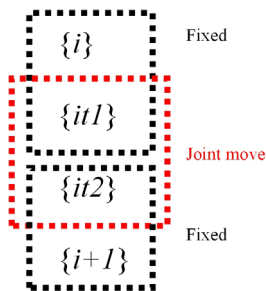
[Fig. 8] Step 3: DH parameters for child link



[Fig. 9] Summary of constructing DH parameters



[Fig. 10] Flow chart to construct DH parameters from URDF



[Fig. 11] Relationship between coordinate frames

3. DH 파라미터 생성 결과

3.1 시험 환경

Chapter 2에서 설명한 DH 파라미터 구성 방법을 사용하여 URDF에서 DH 파라미터를 구성해 보았다. 이를 위해 DH 파라미터로 직접적으로 표현할 수 없는 3가지 로봇의 URDF를 사용하였다(Case 1: 1축 링크 로봇, Case 2 : 6축 로봇 Neumeeka사의 Indy7, Case 3 : Unitree의 사족보행로봇 GO1). 또한 구성된 DH 파라미터의 타당성을 확인하기 위하여 말단장치(end effector)의 정기구학 결과를 비교하였다. 참고로 각 단계에서 구한 DH 파라미터가 전부 0일 경우, 동차행렬변환이 단위 행렬(identity matrix)이 되므로 DH 파라미터 결과에 표시

하지 않았다. 모든 결과는 MATLAB Robotics System Toolbox을 사용하여 시각화 하였다.

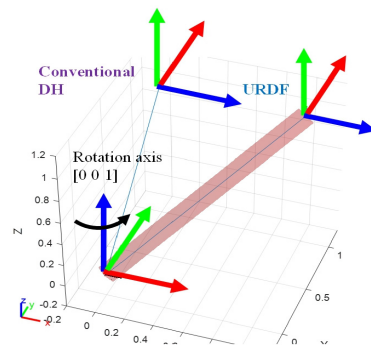
3.2 시험 결과

3.2.1 Case 1: 1축 링크 로봇

[Table 2]와 [Fig. 12]에 보여준 URDF로 기술된 1축 로봇은 기존의 DH 파라미터 표시방법을 사용할 경우 로봇 바디 외부에 좌표계가 설정된다[Fig. 12]. 결과적으로 기존 DH 파라미터를 사용할 경우 직관적이지 못하며 로봇 말단의 위치를 확인할 수 없는 문제가 존재한다. 본 논문에서 제안한 방법을 사용하면 이러한 문제를 해결할 수 있다. DH 파라미터를 구성한 결과는 [Table 3]과 [Fig. 13]에서 확인할 수 있다. 주어진 관절 궤도(trajecory)([Fig. 14(a)])에 대해서 DH 파라미터 표시 방법과 URDF 방식으로 계산한 말단 장치의 궤적이 동일함을 [Fig. 14]과 [Fig. 15]에서 확인할 수 있었다.

[Table 2] URDF information of 1-link robot

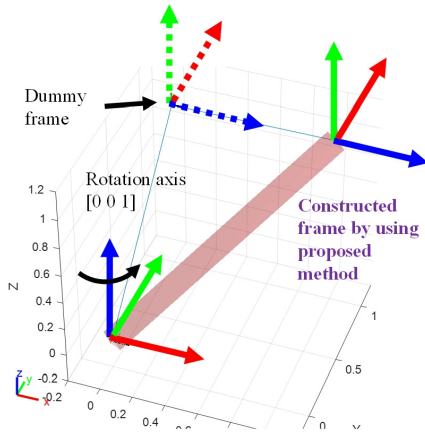
	Parent	Child	Origin xyz	RPY	Axis	Type
Joint 1	Base	Link1	0 0 0	0 0 0	0 0 1	Revolute q_1
Joint 2	Link1	end	1 1 1	$\pi/2$ 0 $\pi/2$	0 0 0	Fixed



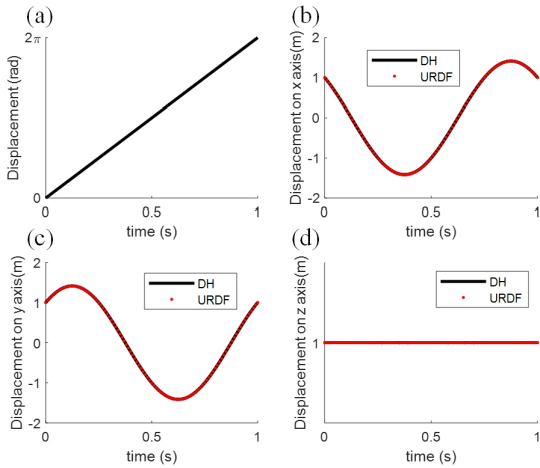
[Fig. 12] Robots constructed by URDF from [Table 2] and conventional DH parameter($\theta = q_1 + \pi/2, d = 1, a = 1, \alpha = \pi/2$). In this figure, we notice that the coordinate frame constructed by conventional DH parameter is out of the robot body. This situation makes users hard to use conventional DH parameter and know the position of end effector

[Table 3] DH parameters from URDF of 1-DOF robot

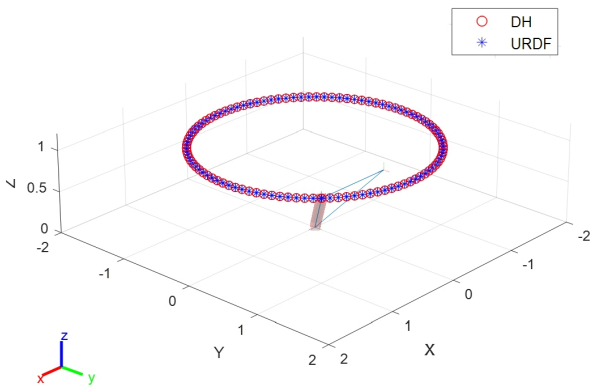
θ_i	d_i	a_i	α_i
$q_1 + \pi/2$	1	1	$\pi/2$
0	0	1	0



[Fig. 13] The robot constructed by DH parameters from [Table 3]. In this figure, the end effector is identical with the end effector in the robot constructed from URDF



[Fig. 14] Forward kinematic results of end effector of 1-DOF robot from DH parameters and URDF. (a) Joint trajectory for q_1 . (b) x axis displacement of end effector. (c) y axis displacement of end effector. (d) z axis displacement of end effector



[Fig. 15] The trajectories of end effectors of 1-DOF robots from URDF and DH parameters. For the given trajectory of the joint axis, the results of the forward kinematics of the end effectors are the same between the two robots

3.2.2 Case 2: 6축 manipulator 로봇

Indy7의 URDF 파일은 (주)뉴로메카에서 공개한 파일을 사용하였다^[12]. 또한 URDF에서 베이스 링크의 병진(translation)을 수행하여 베이스의 위치를 공간상 임의의 위치에 놓았을 때 DH 파라미터를 구할 수 있음을 보이고자 하였다. [Table 4]가 Indy7의 URDF 정보이며, 관절 축은 전부 z축으로 설정되어 있으나 DH 조건이 지켜져 있지 않아 직접적으로 DH 파라미터를 찾아낼 수 없다. URDF가 시각화된 [Fig. 16]에서 x axis가 전 단계 z axis와 만나지 않아 DH 조건이 지켜지지 않음을 확인할 수 있다.

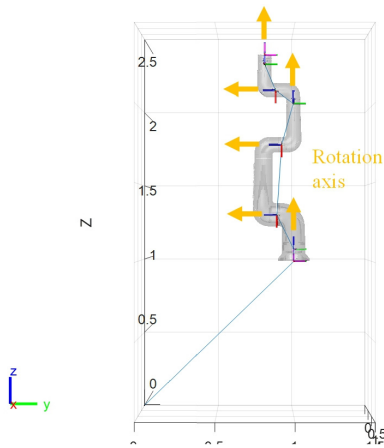
Indy7의 URDF에서 DH 파라미터를 구성한 결과는 [Table 5]에서 확인 가능하다. 또한 DH 파라미터를 이용하여 표현된 로봇은 [Fig. 17], [Fig. 18]에서 확인 가능하다.

DH 파라미터로 표현된 로봇을 확인하면 URDF에서 수행된 베이스 링크, 링크 0의 병진이 제대로 수행 되었음을 확인할 수 있었고 URDF와 동일한 위치에 좌표계가 생성된 것을 [Fig. 19]을 통해 확인할 수 있다.

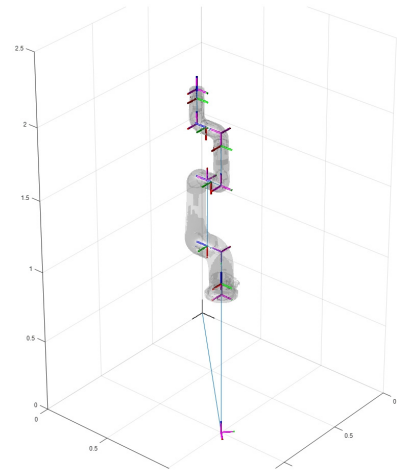
또한 [Fig. 20]에서 DH 파라미터와 URDF로 만들어진 로봇들이 주어진 관절 궤도([Fig. 20(a)])에 대해서 동일한 말단장치 궤적을 가진다는 것을 확인 했으며 [Fig. 21]에 나타난 3차원 공간에서도 동일함을 확인할 수 있었다.

[Table 4] URDF information of indy7

	Parent	Child	Origin xyz	RPY	Axis	Type
Joint 1	World	Link0	1, 1, 1	0 0 0	0 0 0	Fixed
Joint 2	Link0	Link1	0, 0, 0.0775	0 0 0	0 0 1	Revolute q_1
Joint 3	Link1	Link2	0, -0.109, 0.222	$\pi/2$, $\pi/2$, 0	0 0 1	Revolute q_2
Joint 4	Link2	Link3	-0.45, 0, -0.0305	0 0 0	0 0 1	Revolute q_3
Joint 5	Link3	Link4	-0.267, 0, -0.075	$-\pi/2$, 0, $\pi/2$	0 0 1	Revolute q_4
Joint 6	Link4	Link5	0, -0.114, 0.083	$\pi/2$, $\pi/2$, 0	0 0 1	Revolute q_5
Joint 7	Link5	Link6	-0.168, 0, 0.069	$-\pi/2$, 0, $\pi/2$	0 0 1	Revolute q_6
Joint 8	Link6	TCP	0, 0, 0.06	0 0 0	0 0 0	Fixed



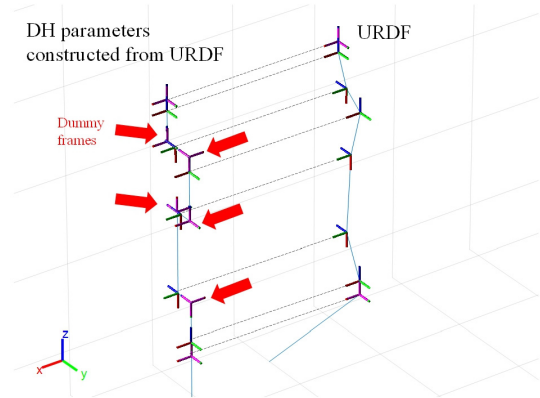
[Fig. 16] Indy7 coordinate frames from URDF



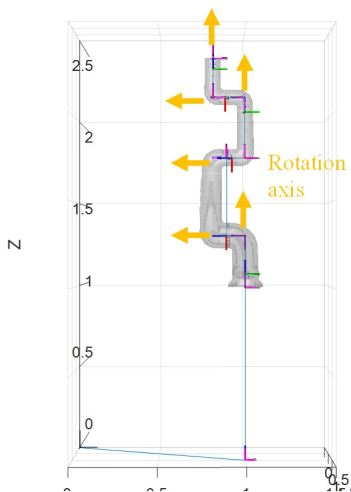
[Fig. 18] Indy7 coordinate frames from DH parameters with another viewpoint

[Table 5] DH parameters from URDF of indy7

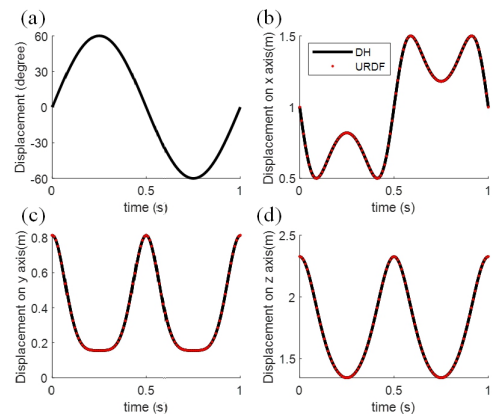
θ_i	d_i	a_i	α_i
$\pi/4$	0	$\sqrt{2}$	0
$-\pi/4$	1	0	0
q_1	0.0775	0	0
π	0.222	0	$-\pi/2$
$q_2+\pi/2$	0.109	0	0
π	0	0.45	0
$q_3+\pi$	-0.0305	0	0
$\pi/2$	-0.075	0	$-\pi/2$
q_4	0.267	0	0
π	0.083	0	$-\pi/2$
$q_5+\pi/2$	0.114	0	0
$\pi/2$	0.069	0	$-\pi/2$
q_6	0.168	0	0
0	0.06	0	0



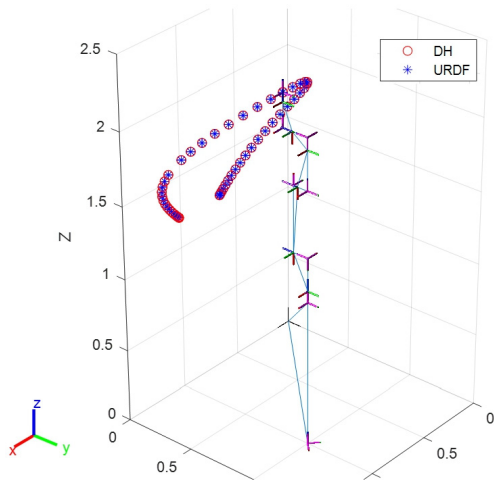
[Fig. 19] Indy7 coordinate frames from URDF and DH parameters. The coordinate frames of the URDF matches the coordinate frames of the DH parameter. Also, the dummy coordinate frames introduced in the DH parameter are observed for making DH parameters from URDF



[Fig. 17] Indy7 coordinate frames from DH parameters



[Fig. 20] Forward kinematic results of end effector of Indy7 from DH parameters and URDF. (a) Joint trajectories for $q_1 \sim q_6$. (b) x axis displacement of end effector. (c) y axis displacement of end effector. (d) z axis displacement of end effector



[Fig. 21] The trajectories of end effectors of Indy7 from DH parameters and URDF

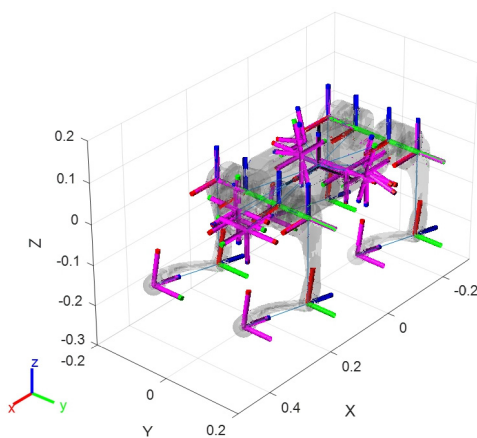
3.2.3 Case 3: 사족보행로봇

GO1의 URDF 파일은 Unitree에서 공개한 파일을 사용하였다^[13]. 사족보행로봇은 몸통을 기준으로 4개의 다리와 센서들이 부착되어 있는 형식으로 구성되어 있다[Fig. 22]. 4개 다리 모두 동일한 구조이므로 앞쪽 왼쪽 다리만 확인한다.

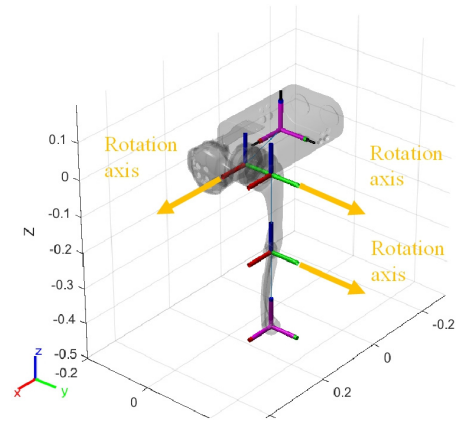
GO1의 URDF는 병진만 이용해 좌표계를 배치하여 직관적으로 이해할 수 있으나, 관절 축이 z축이 아니라 x축이나 y축으로 설정해 둔 것을 확인할 수 있다[Table 6][Fig. 23]. 즉 DH 파라미터를 이용하여 직접적으로 표현이 불가능하다.

GO1의 URDF에서 DH 파라미터를 구성한 결과는 [Table 7]에서 확인 가능하다. 또한 DH 파라미터를 이용하여 표현된 로봇은 [Fig. 24]에서 확인 가능하다.

또한 [Fig. 25]에서 DH 파라미터와 URDF로 만들어진 로봇들이 주어진 관절 궤도([Fig. 25(a)])에 대해서 동일한 말단장치 궤적을 가진다는 것을 확인하였다[Fig. 26].



[Fig. 22] GO1 whole body coordinate frames from URDF



[Fig. 23] GO1 front left leg coordinate frames from URDF

[Table 6] URDF information of GO1

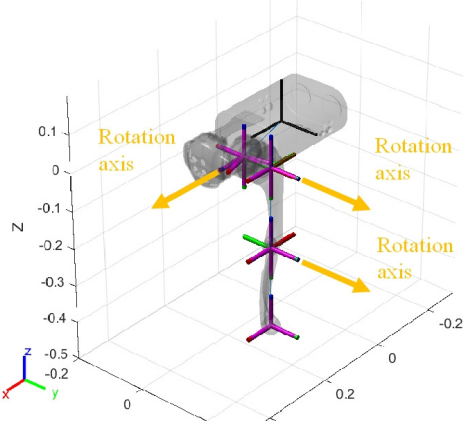
	Parent	Child	Origin xyz	RPY	Axis	Type
Joint 1	Base	Trunk	0, 0, 0	0 0 0	0 0 0	Fixed
Joint 2	Trunk	FL_hip	0.1881, 0.04675, 0	0 0 0	1 0 0	Revolute q_1
Joint 3	FL_hip	FL_thigh	0, 0.08, 0	0 0 0	0 1 0	Revolute q_2
Joint 4	FL_thigh	FL_calf	0, 0, -0.213	0 0 0	0 1 0	Revolute q_3
Joint 5	FL_calf	FL_foot	0, 0, -0.213	0 0 0	0 0 0	Fixed

[Table 7] DH parameters from URDF of GO1

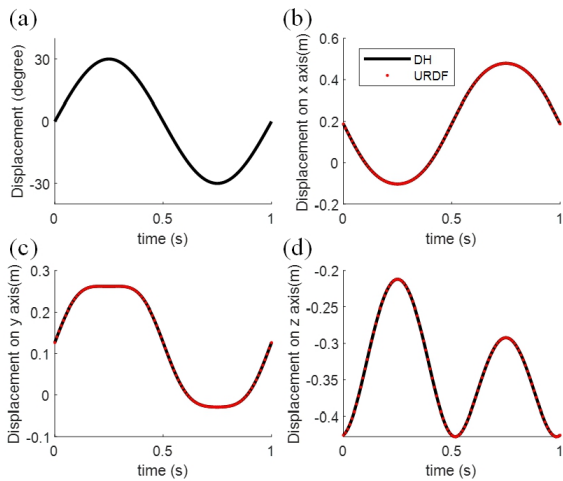
θ_i	d_i	a_i	α_i
0.243602	0	0.193823	0
-0.243602	0	0	0
$-\pi/2$	0	0	$-\pi/2$
$q_1+\pi$	0	0.08	$-\pi/2$
$-\pi/2$	0	0	0
0	0	0	$-\pi/2$
$q_2+\pi$	0	0	$-\pi/2$
π	-0.213	0	0
0	0	0	$-\pi/2$
$q_3+\pi$	0	0	$-\pi/2$
π	-0.213	0	0

4. 결론

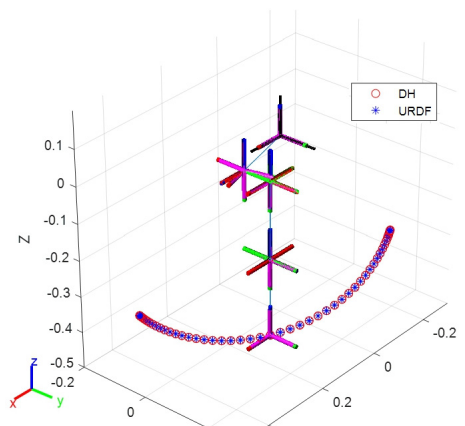
본 논문에서는 URDF에서 DH 파라미터를 구성하는 체계적이고 일반적인 방법을 제안하였다. 다양한 형상의 로봇에



[Fig. 24] GO1 front left leg coordinate frames from DH parameters



[Fig. 25] Forward kinematic results of end effector of GO1 from DH parameters and URDF. (a) Joint trajectories for $q_1 \sim q_3$. (b) x axis displacement of end effector. (c) y axis displacement of end effector. (d) z axis displacement of end effector



[Fig. 26] The trajectories of end effectors of GO1 from DH parameters and URDF

대해 URDF를 이용하여 얻은 정기구학 결과와, 제안하는 방법을 통하여 자동으로 생성된 DH 파라미터를 통하여 구한 정기구학 결과가 동일함을 확인하였다. 본 논문에서 제안하는 방법을 이용하면 DH 조건을 만족하지 않는 URDF 파일로부터도 DH 파라미터를 구할 수 있다는 점에서 의미를 가진다.

References

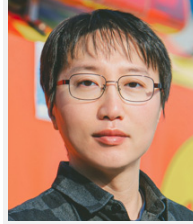
- [1] I. Sucas, J. Kay, urdf, *ROS Wiki*, [Online], <https://wiki.ros.org/urdf>, Accessed: Oct. 07, 2022.
- [2] J. J. Craig, *Introduction to Robotics : Mechanics and Control*, 3rd, Pearson/Prentice Hall, 2005, [Online], <https://books.google.co.kr/books?id=MqMeAQAAIAAJ>.
- [3] J. Denavit and R. S. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *Journal of Applied Mechanics*, vol. 22, no. 2, pp. 215-221, Jun., 2021, DOI: 10.1115/1.4011045.
- [4] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, 1st, Wiley, 2005, [Online], https://www.amazon.com/exec/obidos/tg/detail/-/0471649902/qid=1127861838/sr=8-1/ref=sr_8_xs_ap_i1_xgl14/104-3105527-1036767?v=glance&s=books&n=507846.
- [5] ros/solidworks_urdf_exporter, [Online], https://github.com/ros/solidworks_urdf_exporter, Accessed: Nov. 09, 2022.
- [6] W. Shanda, L. Xiao, L. Qingsheng, and H. Baoling, "Existence Conditions and General Solutions of Closed-form Inverse Kinematics for Revolute Serial Robots," *applied sciences*, vol. 9, no. 20, Oct, 2019, DOI: 10.3390/app9204365.
- [7] T. Zhang, L. Du, and X. Dai, "Test of robot distance error and compensation of kinematic full parameters," *Advances in Mechanical Engineering*, Jan., 2014, DOI: 10.1155/2014/810684.
- [8] J. He, M. Zhang, F. Xu, Q. Liu, and H. Li, "A Model Conversion Algorithm from URDF to DH for Camera Robot," *Culture-Oriented Science and Technology (CoST)*, Lanzhou, China, 2022, DOI: 10.1109/CoST57098.2022.00037.
- [9] mcevoyandy/urdf_to_dh, [Online], https://github.com/mcevoyandy/urdf_to_dh, Accessed: Oct. 07, 2022.
- [10] P. Corke, *Understanding URDF with MATLAB*, [Online], https://petercorke.com/wp-admin/admin-ajax.php?juwpfisAdmin=false&action=wpfd&task=file.download&wpfd_category_id=20&wpfd_file_id=1059&token=&preview=1, Accessed: Oct. 05, 2022.
- [11] G. A. Korn and T. M. Korn, *Mathematical Handbook for Scientists and Engineers: Definitions, Theorems, and Formulas for Reference and Review*, Courier Corporation, 2000, [Online], <https://books.google.co.kr/books?id=xUQe0RZhQnAC>.
- [12] neuromeka-robotics/indy-ros, [Online], <https://github.com/neuromeka-robotics/indy-ros>, Accessed: Oct. 05, 2022.
- [13] Unitree Robotics, [Online], <https://github.com/unitreerobotics>, Accessed: Oct. 13, 2022.



유 병 기

2013 한국항공대학교 항공우주기계공학과 (학사)
 2015 대구경북과학기술원 로봇공학과(석사)
 2022 대구경북과학기술원 로봇및기계전자 공학과(박사)
 2022~현재 한국로봇융합연구원 선임연구원

관심분야: Robotic Manipulation



박 상 현

2002년 한국과학기술원 기계공학과(학사)
 2004년 한국과학기술원 기계공학과(석사)
 2011년~2018년 로보스타 부장
 2018년~현재 한국로봇융합연구원 선임연구원

관심분야: 여유자유도 로봇, 모션 제어, 비선형 시스템 강인제어, 기구학 캘리브레이션, 유압 매니플레이터 제어



이 준 영

2013 대구경북과학기술원 로봇공학과(석사)
 2019 대구경북과학기술원 로봇공학과(공학 박사)
 2019~현재 한국로봇융합연구원 선임연구원

관심분야: 로봇틱스 및 어플리케이션



김 무 림

2008 한국과학기술원 기계공학과(공학박사)
 2008~2016 포항산업과학연구원 책임연구원
 2016~현재 한국로봇융합연구원 수석연구원 /본부장

관심분야: 재난대응로봇, 로봇 제어, 공장자동화, 협동로봇