URDF Generator for Manipulator Robot

Yeon Kang, Donghan Kim
Department of Electronics
Kyung Hee University
Yong-in, Republic of Korea
lovely_kite@khu.ac.kr, donghani@khu.ac.kr

Kwangjin Kim
Department of Electronics
Cemware M04 133dong,
1, Gwanak-ro, Gwanak-gu, Seoul, Republic of Korea
ceo@cemware.com

Abstract—**As the robots market grows, the demand for manipulators and the market size are also increasing. Various methods are used for designing and applying the manipulator. Among them, the most efficient in terms of cost and time is a simulation. Therefore, the importance of simulation is increasing, and various studies are being conducted on this. The proposed system in this paper automates the generation of URDF files for manipulator simulation. The developed system receives DH parameters and joint types. To verify this system, URDF is generated using Universal Robot's UR5 6-axis manipulator. Then, import the generated URDF file into V-REP, one of the commercial simulator, and confirm that the proper manipulator is outputted. The result shows the validity of the developed system.**

Keywords-manipulator; urdf; simulation; dh parameter

I. INTRODUCTION

Currently, manipulators are widely applied to industrial robots. In addition, the scope of application is expanding to include professional service robots, home robots, and medical robots. According to Siemens.Corp forecast [1], the global robot market will be 42.9 Billions US dollars in 2020 and 66.9 Billions US dollars in 2025; according to Martin Haegele [2], 1.4 million new industrial robots will enter the factory in 2019. As the robot market grows larger, the proportion of manipulators will increase. In order to design and adapt these manipulators in various environments, a variety of methods have been used, including mounting manipulators on the field and testing them. Among them, simulation is widely used because of it's cost and time efficient. Therefore, the importance of these manipulator simulation lines is increasing.

Recently, there have been many studies to utilize and analyze these manipulator simulations. In [3], the authors compared V_REP (Coppelia Robotics.Corp) and Gazebo (Open Source Robotics Foundation) among the commercial simulators in terms of ROS(Robot Operating System) integration, world modeling, robot model modifications, programmatic control and CPU(Center Processing Unit) usage. [4] simulated a 5-DOF(Degree of the Freedom) manipulator based on MATLAB (Mathworks.Corp) Simulink, and proposed a solution for path following in 3D space. Also, [5] proposed a method for faster and more efficient computation of forward kinematics and inverse kinematics of 3-DOF manipulators using DH (Denavit-Hartenberg) parameters through simulation. In the literature [6] the authors proposed a method to obtain the trajectory of a manipulator to effectively overcome a flexible load and a constraint force that are difficult to control, dynamic modeling and simulation. In addition, there are several works that try to analyze various types of manipulators through simulation [7-9]. While in [10] the authors have developed a simulator to learn about robotic arms, this simulator can test robot kinematics, dynamics, and differential movements. Proposal such as [11] presented a method for finding a joint and link tree to simulate Baxter (rethink robotics.Corp) robots on the ROS. The author used forward kinematics to calculate the robot's working radius. On the other hand, there are other proposed using the same principle of simulating robot manipulators using DH parameters [12-14]. In [15], the authors proposed the simulation environment for user-designed manipulator with generate the code in XML format.

In this paper, we have developed a system that automatically generates a URDF (Universal Robotic Description Format) file using only the DH parameter [16] that contains the basic information of the manipulator and the type of each joint. The method to verify the system is to input UR5 6-axis manipulator of Universial Robot's DH parameter into the developed system, and confirm the generated URDF. In order to verify the generated URDF it's importing it into commercial simulator V-REP.

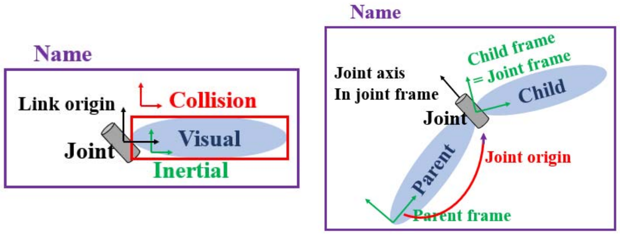
The rest of the paper is organized as follow: In section □, URDF and DH parameters are reviewed. In section □, describe the system and algorithm of the URDF generate system. In Section □, URDF generating using the DH parameters of UR5 is verified. Finally, in section □, conclusions and directions for future research are presented.

II. PARAMETERS FOR MANIPULATOR SIMULATION

A. URDF

URDF is an XML(extensible Markup Language) format for representing a robot model. It represents the appearance of the robot and its intended action. It is used to generate robotic models in various commercial simulators such as V-rep or Gazebo. As shown in Fig. 1(a), the URDF link is divided into name, which defines the link itself, and inertial, visual, and collision. Definitions for these variables are given in Table □. As shown in Fig. 1(b), the URDF joint is divided into a name defining the joint itself, a type defining the joint type, and an origin, parent, child, and axis. Definitions for these variables are given in Table □.

978-1-5386-9245-5/19/$31.00 ©2019 IEEE
DOI 10.1109/IRC.2019.00101



(a)

(b)

Figure 1. URDF Variables (a) Link (b) Joint

TABLE I. VARIABLES OF URDF LINK

Link	Definition
collision	Set information for link collision counting
visual	Set visualization information for links
origin	Set visualization information for links
mass	Weight of link [kg] set
inertial	Set inertia information for the link
geometry	Enter the shape of the model (Box, cylinder, sphere)

TABLE II. VARIABLES OF URDF JOINT

Joint	Definition
parent	Joint Parent Link
child	Joint Child Link
origin	Convert parent link coordinate system to child link coordinate system
axis	Rotation axis setting
limit	Set joint speed, force, and radius

B. DH Parameter

DH parameter represents kinematic analysis of the kinematic chain or robot manipulator. It consists of the following four parameters [17][18]:

- d = offset along previous z to the common normal
- θ = angle about previous z, from old x to new x
- a = length of the common normal, assuming a revolute joint, this is the radius about previous z
- α = angle about common normal, from old z axis to new z axis

Also, there are some constraints on the relationships between the axis:

- the x_n-axis is perpendicular to both the z_{n-1} and z_n axis
- the x_n-axis intersects both z_{n-1} and z_n axis
- the origin of joint n is at the intersection of x_n and z_n
- y_n completes a right-handed reference frame based on x_n and z_n

As shown in Fig. 2, these four parameters can be used to describe the relationship between the two links, Link $\mathrm{i}-1$ and Link i.

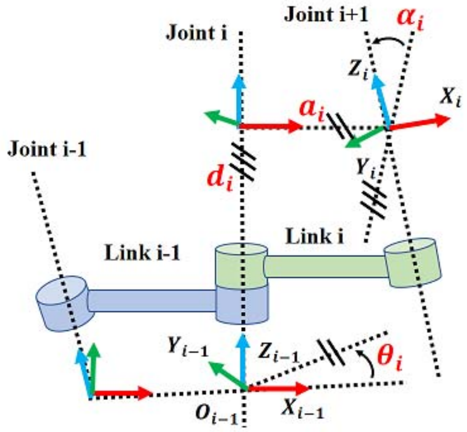


Figure 2. Four basic DH parameters

III. SYSTEM CONFIGUTION

In this section, we provide a URDF generation system. The system configuration is shown in Fig. 3. URDF generate system is based on following steps:

Step 1: The URDF generation system receives the values of d, a and α in the DH parameter through the GUI built on PyQt5.

Step 2: The parameters from the Step 1, PyQt5 and input it into the URDF generation algorithm.

Step 3: Import the generated URDF file into the simulator.

Step 4: Perform the simulation using the manipulator model generated by imported URDF file.

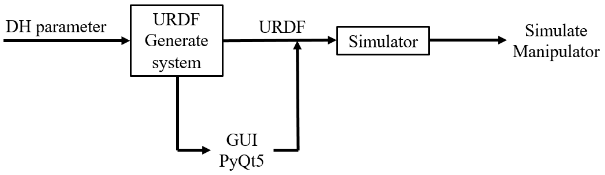
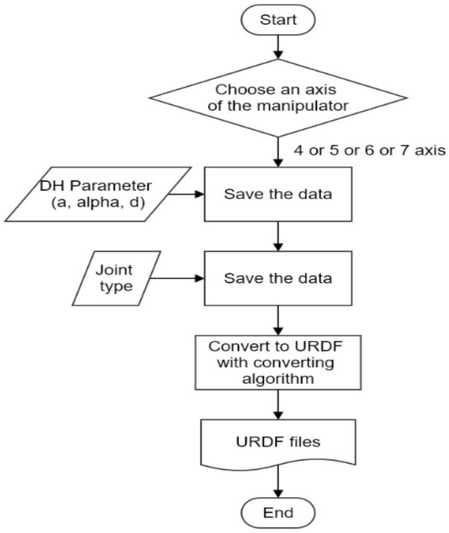


Figure 3. URDF generate system configuration using DH parameter

GUI built on PyQt5 is given by Fig.4, Fig. 5 and Fig. 6. As shown in Fig. 4, it describes the flow of the GUI system. Select the number of axis of the manipulator which want to simulate, as shown in Fig. 5. Then as shown as Fig. 6, enter the values of a_i, α_i, and d_i in the DH parameter of the manipulator. Finally, if you select joint type, URDF file is created by generating algorithm. The generating algorithm uses the correlation between α_i value and a_i, d_i, and the joint axis to determine the link length and origin x, y, z values to enter into URDF, and the joint axis x, y, z values as shown in Algorithm. 1. The key parameter among the input parameters is α_i. By definition, α_i is a parameter that indicates how much the two related joints and links rotate. Thus, as the value of α_i changes, the associated link and joint axis are affected. According to [19], the values α can take is set presented in eq.(1).

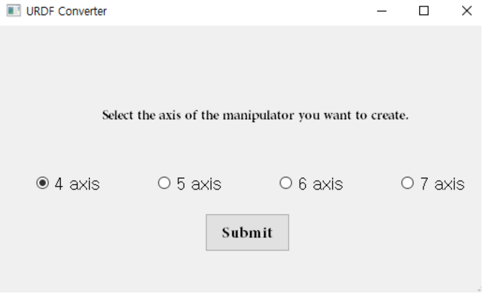
$$\alpha \in \{0, \frac{\pi}{2}, -\frac{\pi}{2}, \pi\} \quad (1)$$

Suppose convert the URDF of the N-axis manipulator. If the α of the i^{th} DH parameter is 0 or π, then the axis and joint axis of the link from i^{th} to N^{th}, and the origin, y, z values of the link and joint do not change. And the d value of the DH parameter is the length of the N^{th} link. If the i^{th} α is $^{\pi}/_{2}$, the y and z axis of each link are swapped from $(i+1)^{th}$ to N^{th} links, and the length of the link is also swapped from y to and swap z. However, the origin, x, y, and z values of all links and joints remain unchanged. Finally, if the i^{th} α is $-^{\pi}/_{2}$, the y and z axis of each link are swapped from $(i+1)^{th}$ to the N^{th} link similarly to when α is $^{\pi}/_{2}$. In this case, the values of y and z among the origin, y, and z values of link and joint are also swapped. This process is performed from the first axis to the N^{th} axis, then from the second axis to the N^{th} axis, and finally from the $(N-1)^{th}$ axis to the N^{th} axis. Through this process, the parameters are sorted and converted into URDF and output.



Start
Choose an axis of the manipulator
4 or 5 or 6 or 7 axis
DH Parameter (a, alpha, d)
Save the data
Joint type
Save the data
Convert to URDF with converting algorithm
URDF files
End

Figure 4. Flowchart of URDF generate GUI



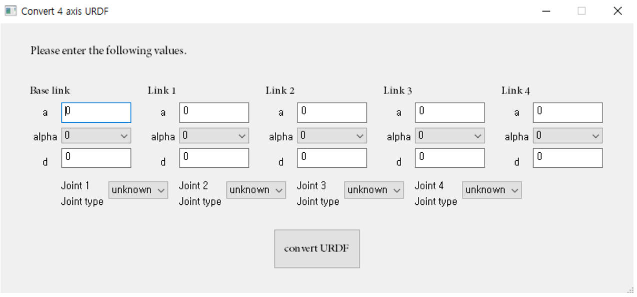
URDF Converter

Select the axis of the manipulator you want to create.

● 4 axis ○ 5 axis ○ 6 axis ○ 7 axis

Submit

Figure 5. URDF generate GUI with main window



Convert 4 axis URDF

Please enter the following values.

	Base link	Link 1	Link 2	Link 3	Link 4
a	0	0	0	0	0
alpha	0	0	0	0	0
d	0	0	0	0	0

Joint 1 Joint type	Joint 2 Joint type	Joint 3 Joint type	Joint 4 Joint type
unknown	unknown	unknown	unknown

convert URDF

Figure 6. URDF generate GUI with 4 axis

Algorithm 1 URDF generating algorithm

```
Input: DH parameters: a, α, d
Input: N axis
Sub URDF generating algorithm()
  set all axis y[1: N] = 0 and axis z[1: N] = 1
  set all origin y[1: N] = 0 and origin z[1: N] = 1
  set i = 1
  while i = N − 1
    if α[i] = 0 or π
      a, d, link and joint axis are not change
    elseif α[i] = π/2
      switch all a[1: N]and d[1: N]
      switch all joint axis y[1: N]and joint z[1: N]
    else
      switch all a[1: N]and d[1: N]
      switch all axis y[1: N]and z[1: N]
      switch link origin y[i + 1: N] and
       link origin z[i + 1: N]
      switch joint origin y[i + 1: N] and
       joint origin z[i + 1: N]
    i = i + 1
end
```

IV. VERIFICATIONE

In order to verify the performance of the system, we generated URDF using the DH parameters of UR5 (Universal Robots.corp) which is a 6 axis manipulator widely used as industrial robot. At this time, the geometry of the link is set to box, and the thickness of the link is set to 0.01[m] * 0.01[m]. Table □ shows the DH parameters for UR5. For all joints, theta[rad] is 0. As a result of inputting the above DH parameter into the URDF generator UI, URDF file is generated. After importing it into V-REP, we compared it with UR5 modeling. The results are shown in Fig. 7. The result, confirmed that all 5 links length of the joints of the modeling made of the actual URDF and the generated URDF are all the same. Also, it shows the axial direction of link 1, 2, 3 and 5 is y axis, and the axial direction of link 4 is z axis, made with generated URDF. It is same with the axial direction of links, made with actual URDF. As a results, it confirmed that the axial direction and the link lengths of the joints of the modeling made of the actual URDF and the generated URDF are all the same.

TABLE III. DENAVIT-HARTENBERG PARAMETERS OF UR5[20]

Kinematics	a [m]	d [m]	alpha [rad]
Joint 1	0	0.089159	$\pi/2$
Joint 2	-0.425	0	0
Joint 3	-0.39225	0	0
Joint 4	0	0.10915	$\pi/2$
Joint 5	0	0.09465	$-\pi/2$
Joint 6	0	0.0823	0

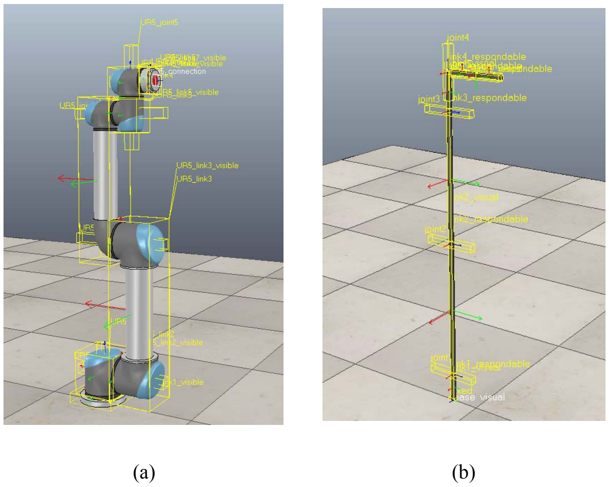


Figure 7. Compare (a) UR5 and (b) URDF modeling in V-REP

V. CONCLUSION

In this work, we have accomplished what we have aimed. Only DH parameter and joint type are necessary in order to generated URDF automatically. We imported the generated URDF into one of the commercial simulator V-REP, and confirmed that the generate URDF process was successful. As shown in Fig. 7, the manipulator's link and joint axis are changed. The shape of the link and the thickness of the link are arbitrarily set to box and 0.01, which makes the problem simpler when constructing the transformation algorithm. We hope this system will be of great help for URDF generation to use manipulators. Additional study is necessary because created manipulators using the proposed URDF transformation system show that all links and joints are connected in a straight line, and it means the transformation system can not generate URDF that are exactly` the same shape as the actual manipulator. Analyzing this problem, we found that DH parameters do not tell how far the child link of the manipulator is in the x and y directions of the parent link.

Future research regarding the generate the URDF files for multiple manipulators. It will be a great help to people who design and adapt manipulators.

ACKNOWLEDGMENT

This research was supported by the Ministry of Trade, Industry & Energy(MOTIE, Korea) under Industrial Technology Innovation Program(No. 10067414, 10070258, 10076477, 10080348) and the Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(No. 2016R1D1A1A02936946).

REFERENCES

[1] "Robots: Building New Business Models." Siemens. last modified n.d., accessed Dec 20, 2018, https://www.siemens.com/innovation/en/home /pictures-of-the-future/digitalization-and-software/autonomous-systems-facts-and-forecasts.html

[2] Martin Hagele, "Double-Digit Growth Highlights a Boom in Robotics", IEEE Robotics & Automation Magazine, 2017, Volume: 24, Issue: 1.

[3] Lucas Nogueira, "Comparative Analysis Between Gazebo and V-REP Robotic Simulators", Seminario Interno de Cognicao Artificial(SICA), 2014.

[4] Velarde-Sanchez J.A., Rodriguez-Gutierrez S.A., Garcia-Valdovinos L.G., Pedraza-Ortega J.C, "5-DOF Manipulator Simulation based on MATLAB-Simulink methodology", International Conference on Electronics Communications and Computers (CONIELECOMP), 2010.

[5] Yun-Long Wan, Hong-Xia Xu, "On dynamics simulation of 3 DOF manipulator", 35th Chinese Control Conference (CCC), 2016.

[6] Huidong Ma, JinJin Guo, Weimin Ge, Xiaofeng Wang, "Dynamic Modeling and Simulation of Flexible Load System Operated by Reconfigurable Manipulators Under Different Constraints", IEEE International Conference on Mechatronics and Automation (ICMA), 2018.

[7] Jihui Zhou, Lin Li, Lan Ye and Qingsong Cao, "Research on hardware-in-the-loop real-time simulation system for vibration control of flexible manipulator," 2009 IEEE 10th International Conference on Computer-Aided Industrial Design & Conceptual Design, Wenzhou, 2009, pp. 2297-2300.

[8] T. Aofei, L. Yan, X. Junbao and Y. Zhihua, "Kinematics Simulation of Equivalent Models for Cable Manipulators," 2010 International Conference on Digital Manufacturing & Automation, Changsha, 2010, pp. 881-885.

[9] J. Luo, L. Cong and B. Li, "The Kinematics Simulation of a New Type of 3R Parallel Robot Manipulator," 2010 Second International Workshop on Education Technology and Computer Science, Wuhan, 2010, pp. 320-323.

[10] Fernando Gonzalez and Janusz Zalewski, "A Robotic Arm Simulator Software Tool for use in Introductory Robotics Courses", IEEE Global Engineering Education Conference(EDUCON), 2014, p.861-866.

[11] Michael Mortimer, Ben Horan, Matthew Joordens and Alex Stojcevski, "Seraching Baxter's URDF robot joint and link tree for active serial chains", System of Systems Engineering Conference(SoSE), 2015.

[12] J. Xiao, W. Han and A. Wang, "Simulation research of a six degrees of freedom manipulator kinematics based On MATLAB toolbox," 2017 International Conference on Advanced Mechatronic Systems (ICAMechS), Xiamen, 2017, pp. 376-380.

[13] J. Liu, R. Liu, X. Shen and L. Meng, "Research on Obstacle Avoidance of Space Manipulators Based on Cylindrical Bounding Box Model," 2018 IEEE International Conference on Mechatronics and Automation (ICMA), Changchun, 2018, pp. 1926-1931.

[14] L. Shuiping, L. Xueqin and W. Liang, "Design and Simulation of Immersing and Palletizing Robotic Manipulator for the De-NOx Catalyst Production Line," 2016 3rd International Conference on Information Science and Control Engineering (ICISCE), Beijing, 2016, pp. 914-917.

[15] H. M. Do, G. H. Kim, T. Choi, D. H. Kim and Y. Son, "Development of simulation model for modular and reconfigurable robots," 2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Xi'an, 2016, pp. 138-139.

[16] Craig, John. J, "Introduction to Robotics: Mechanics and Control", 2nd Edition, Addison Wesley Publishing. Reading, Massachusetts, 1989.

[17] Paul, Richard P. Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators. Cambridge, MA: MIT Press, 1981.

[18] Spong. Mark W, Vidyasagar. M. Robot Dynamics and Control. New York: John Wiley & Sons, 1989.

[19] A. Csiszar, "A combinatorial optimization approach to the Denavit-Hartenberg parameter assignment," 2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), Busan, 2015, pp. 1451-1456.

[20] "DH-Parameters." Universal Robots. last modified n.d., accessed Dec 23, 2018, https://www.universal-robots.com/how-tos-and-faqs/faq/ur-faq/parameters-for-calculations-of-kinematics-and-dynamics-45257/.