Understanding URDF: A Dataset and Analysis

Tola, Daniella; Corke, Peter

Published in:
IEEE Robotics and Automation Letters

DOI (link to publication from Publisher):
10.1109/LRA.2024.3381482

Publication date:
2024

Document Version
Accepted author manuscript, peer reviewed version

Citation for published version (APA):
Tola, D., & Corke, P. (2024). Understanding URDF: A Dataset and Analysis. *IEEE Robotics and Automation Letters*, *9*(5), 4479-4486. Article 10478618. https://doi.org/10.1109/LRA.2024.3381482

Understanding URDF: A Dataset and Analysis

Daniella Tola[1] and Peter Corke[2] *Fellow, IEEE*

***Abstract*—The complexity of robot systems is rising which makes it increasingly effective to simulate before deployment. To do this, a model of the robot's kinematics or dynamics is required, and the most commonly used format is the Unified Robot Description Format (URDF). This article presents the first dataset of URDF files with metadata and analysis. The dataset contains 322 URDF files from various industrial and research organizations, and the metadata describes each robot, its type, manufacturer, and the source of the model. The files correspond to 195 unique robot models – the excess URDFs correspond to either multiple definitions across sources, or URDF variants of the same robot. We analyze the files in the dataset and provide information on how they were generated, which mesh file types are most commonly used, and compare models of multiply-defined robots. The purpose of this article is to create foundational knowledge about URDF files, how they are created and used, and generate insight into the current state of the URDF format. Publishing the dataset, analysis, and the scripts and tools used enables others using, researching or developing URDFs to easily access this data and use it in their own work.**

***Index Terms*—Simulation and Animation, Kinematics, Software Tools for Benchmarking and Reproducibility**

I. INTRODUCTION

MODELING and simulation are key parts of the process of developing robotic systems [1], [2]. Their use has been increasing over the past years, as has their complexity [3]. Modeling and simulation reduces engineering cost and risk by allowing experimentation with parameters, algorithms, and work cell layout before committing to expensive physical hardware. This need has driven the development of numerous simulation tools, such as Gazebo, Webots, Unity, and RoboDK. Each of these tools has its own native model type, and exchanging models between these tools can be cumbersome if a common format is not used.

The Unified Robot Description Format (URDF) was introduced in 2009 by the developers of the Robot Operating System (ROS) as a format to describe the kinematics, dynamics, and geometries of robots, independently of software programs [4]. A URDF file is an XML-based file with an extension of *.urdf*. URDF files allow robotics developers to describe a robot using a universal format, which can be imported and exported by different tools to visualize or simulate the robot. The number of tools supporting URDF files is growing, for instance, Unity recently (2019) added support for URDF, and MATLAB supported the import of URDF files in their 2018 release, and the exportation of URDF files in their 2023 release. This illustrates a great interest in continuously supporting and using the format. In addition to URDF, three other open-source object description formats are common [5], [6], these are the Simulation Description Format (SDF), Universal Scene Description (USD), and MuJoCo File Format (MJCF). Table I shows an overview of commonly used robot simulation tools [7], and which of the four mentioned formats each tool supports. Of the 12 simulation tools, 11 support URDF, indicating its wide adoption.

Simulation tool	URDF	SDF	USD	MJCF
CoppeliaSim	✔	✔	✗	✗
Drake	✔	✔	✗	✔
Gazebo	✔	✔	(✔)	(✔)
MATLAB	✔	✔	✗	✗
MuJoCo	✔	✗	✗	✔
NVidia Isaac	✔	✗	✔	✔
PyBullet	✔	✔	✗	✔
RoboDK	✗	✗	✗	✗
Robotics Toolbox for Python	✔	✗	✗	✗
RViz	✔	✗	✗	✗
Unity	✔	✗	✗	✗
Webots	(✔)	✗	✗	✗
Total 12	11	5	2	5

TABLE I: Simulators and the formats they support (as of 12/2023). The parentheses indicate that the format is supported by a supplied translation tool.

Despite URDF being an important and ubiquitous file format, it lacks the rigour of a formal standard. There are no official guidelines on working with, or developing, URDFs [6], for example naming conventions, coding patterns, and the folder layout of multi-file URDF models.

The contributions of this article are a novel curated dataset of URDF models, *harvested from the wild*, various analyses that show, for the first time, how URDF is used in practice, and the problems related to its use. The dataset can be used to find common usage and coding patterns across the URDF files – providing defacto guidance in the absence of a formal standard – as well as enabling benchmarking and evaluation of URDF-related tools. Features of the dataset include:

- Diverse collection of URDF files that represent various types of robotic devices, such as manipulator arms, end effectors, quadrupeds, and wheeled mobile bases.
- Diverse sources of URDF files, including professional research and industrial organizations.
- Annotated with human and machine-readable metadata describing the type of robot, manufacturer, and a URL pointing to the original URDF location.
- Accompanied by a Python tool and scripts for analyzing

Manuscript received: December, 23, 2023; Accepted February, 19, 2024.
This paper was recommended for publication by Lucia Pallottino upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Innovation Foundation Denmark through the MADE FAST project.

[1]D. Tola is with the Department of Electrical and Computer Engineering, Aarhus University, Aarhus, Denmark dt@ece.au.dk
[2]P. Corke is with the Queensland University of Technology (QUT) Centre for Robotics, QUT, Brisbane, Australia peter.corke@qut.edu.au

Digital Object Identifier (DOI): see top of this page.

URDF files, allowing users to easily analyze new URDF files and reproduce the results in this article.

- Open sourced to allow others to contribute to it, or to perform their own analyses.

Other collections of URDF robot models [8], [9] have fewer models[1] and do not include a detailed analysis of their URDF files.

Section II provides a brief introduction to URDF files. The dataset is presented in Section III, and the results of various analyses of the dataset in Section IV. The construction of the dataset, how to reproduce its results, and an introduction to a work-in-progress tool for analyzing URDFs, are all presented in Section V. We conclude in Section VI with our main findings and recommendations.

II. WHAT IS URDF?

A URDF model is a human-readable XML file describing the kinematic structure, dynamic parameters, visual representation, and collision geometries of a robot. The file may include references to other files that contain 3D geometries of the robot's components. We refer to such a set of files as a URDF Bundle, introduced below in Section II-B. The concepts behind a URDF file, and its associated components, are briefly described in this section.

A. URDF File

URDF was developed as a specification that can include all relevant modeling details of a robot within a single file. It was initially introduced within the ROS ecosystem, but its standalone characteristic and XML basis has allowed it to be adopted by many different tools outside of ROS.

The minimal URDF file contains the name of the robot and a single link. Links are rigid bodies that can be connected using joints, and are described by inertial, visual, and collision properties [10]. The visual and collision properties describe the shapes of the links as either simple geometric primitives (such as cubes and cylinders) or more complex Computer-aided Design (CAD) meshes. A commonly used format for both visualization and collision meshes in URDF is STL (with file extension *.stl*), which represents simple 3D surface geometries without color or texture information. Another format, COLLADA (with file extension *.dae*), is typically used for visualization meshes as it supports both color and texture information. The OBJ format (with file extension *.obj*), supports color, texture, and free-form curves, allowing for higher levels of detail for visualization, where the color and texture data are stored in a separate *(.mtl)* file. The only required specification of a link is its name, which *must* be unique within the model.

Joints connect two links, a parent and a child link. The parent link is closer to the base (or root) of the robot, and the child link is closer to the tool tip[2]. The main specifications of joints are the type (kinematics), dynamics, and safety limits [11], with the most common joint types being revolute and prismatic. The required specifications of a joint are its name, type, and the names of its parent and child links. For more information on URDF, we refer to the ROS wiki page [12].

[1]For example, [8] has less than 100 robot models.

[2]URDF cannot model a closed-kinematic chain.

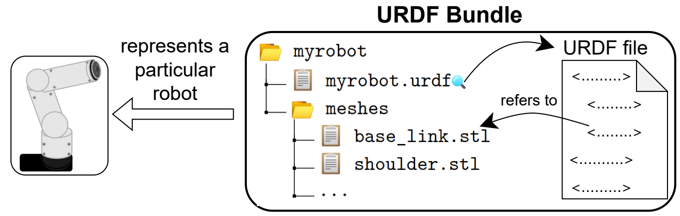

Fig. 1: Example of a URDF Bundle. It is common for mesh files to be located in a separate folder.

B. URDF Bundle

A robot modeled with URDF typically comprises the URDF file and separate mesh files describing the physical appearance of each of the robot's links. We distinguish between the URDF file itself (with the *.urdf* file extension) and the set of files comprising the URDF file and meshes, by referring to the latter as a *URDF Bundle*, see Fig. 1. This URDF Bundle contains the URDF file called *myrobot.urdf* together with the meshes of the robot located within the *meshes* folder. The URDF file refers to the mesh files of the links using relative paths.

C. Xacro

Xacro [13] is both a macro language and preprocessor for XML, that is commonly used in ROS. Xacro can perform variable substitution, math calculations, file inclusion, and conditional blocks. Xacro allows URDF files to be configured or customized, based on the application, to reduce redundancy and simplify the maintainability of the models.

The URDF file with xacro tags has an extension of *(.xacro)* and is processed by the xacro preprocessor (a command line tool). The preprocessor combines multiple files and resolves macro and property definitions and outputs a plain URDF file with no xacro tags. This process is especially beneficial when dealing with complex robotic structures [14].

III. DATASET

The URDF Bundles in this dataset consist solely of robots with meshes representing either the visual properties, or both the visual and collision properties of the robots. The dataset is publicly available in a GitHub repository [15].

A. Definitions

URDF Bundle vs. robot: A physical robot can be described by a URDF Bundle.

URDF variant: URDF files can represent different features of the same robot, depending on the application of the URDF. For example, the Kuka LBR iiwa 14 robot has the URDF variants *spheres collision*, *no collision*, *spheres dense elbow collision*, etc., and the Atlas robot has URDF variants of *convex hull* and *minimal contact*. These URDF

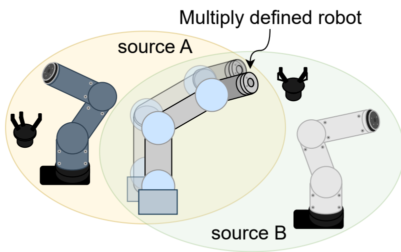

Fig. 2: Illustration of a multiply defined robot (middle) with two different URDF models from two different sources.

variants seem to be created for different applications or simulation purposes, and as not all of them include explanations, it may be difficult to choose the most appropriate URDF file for a given application.

Multiply defined robot: URDF Bundles representing the same robot can be found in multiple sources, see Fig. 2. For instance, the UR5e robot from Universal Robots can be found in both the `matlab` and `ros-industrial` sources which are introduced below. We characterize such a robot as being multiply defined. The files of a multiply defined robot are not necessarily identical, and it is not guaranteed that the behavior of the different models are equivalent, i.e., that the forward kinematics are the same.

B. Dataset Overview

The URDF Bundles in our dataset have been gathered from six different sources described below. Table II shows the total number of URDF Bundles and their URDF variants from each source, and Fig. 3 shows the number of URDF Bundles by robot type. The sources are:

ros-industrial: is an open-source project with the goal of supporting ROS for manufacturing and automation [16]. The project builds on ROS, and includes URDF Bundles and drivers for specific robots. The developers and contributors of the project are mostly research organizations, but are supported by some of the manufacturers.

matlab: is a commercial platform for programming and numeric computing which contains a number of toolboxes [17]. The Robotics System Toolbox is shipped with URDF Bundles of commonly-used robots. These Bundles (from MATLAB R2022b) are also included in our dataset and have their own individual licensing.

robotics-toolbox: is an open-source Python toolbox providing various tools for working with kinematics and dynamics, visualizations, and path planning [18]. The toolbox is developed and maintained by researchers. The URDF Bundles provided by the `robotics-toolbox` are modified from other sources.

drake: is an open-source Python and C++ toolbox providing tools for modeling dynamical systems, working with kinematics and dynamics, and solving mathematical programs [19]. The toolbox is developed and maintained by researchers and developers. The URDF Bundles provided by `drake` are modified from other sources, where they have appended a readme or license file to describe the modifications and origins of the files.

oems: is an assortment of URDF Bundles provided directly by the Original Equipment Manufacturers of the robots.

random: is an assortment of URDF Bundles from various GitHub repositories. These URDF Bundles may have been developed by researchers, developers, or others.

TABLE II: An overview of the URDF Bundles (incl. variants) and URDF variants in the dataset.

Source	#URDF Bundles	#URDF Variants
`ros-industrial`	108	1
`matlab`	52	2
`robotics-toolbox`	44	15
`drake`	16	12
`oems`	35	6
`random`	67	39
Total	**322**	**75**

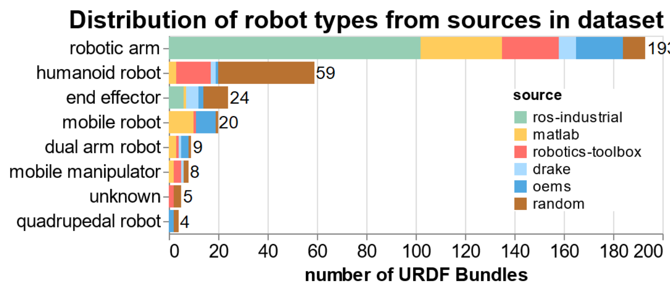

Fig. 3: Number of URDF Bundles of each robot type.

The dataset contains two directories, one with the URDF Bundles, the other with scripts for reproducing the results in this article. The *urdf_files* directory contains all of the URDF Bundles in the dataset categorized by the source. Information about the source and robots (URDF Bundles) is included using human and machine-readable JSON files. Each URDF Bundle is described using meta information that includes the name, type, manufacturer, relative URDF file location, unique dataset id, source, URL, and whether the URDF Bundle has manually been generated using xacro while creating the dataset.

IV. ANALYSIS

This section provides an analysis of our curated dataset, where we look into a selected number of properties of the URDFs in the dataset. We have chosen to analyze these properties based on what we believe is interesting to understand about how URDF files are constructed and used. By making the dataset publicly available, it allows others to analyze additional properties. We outline the manufacturers and their provision of URDFs; common folder structures in the dataset; detection of identical files; comparisons of multiply defined robots; most common errors when parsing URDFs; commonly used CAD formats for visualization and collision; and other quantitative information such as author contact information and the licensing of the URDF Bundles.

A. Manufacturers' Provision of URDFs

In total there are 32 different manufacturers of the robots in the dataset, see Fig. 4 for the distribution. Of these:

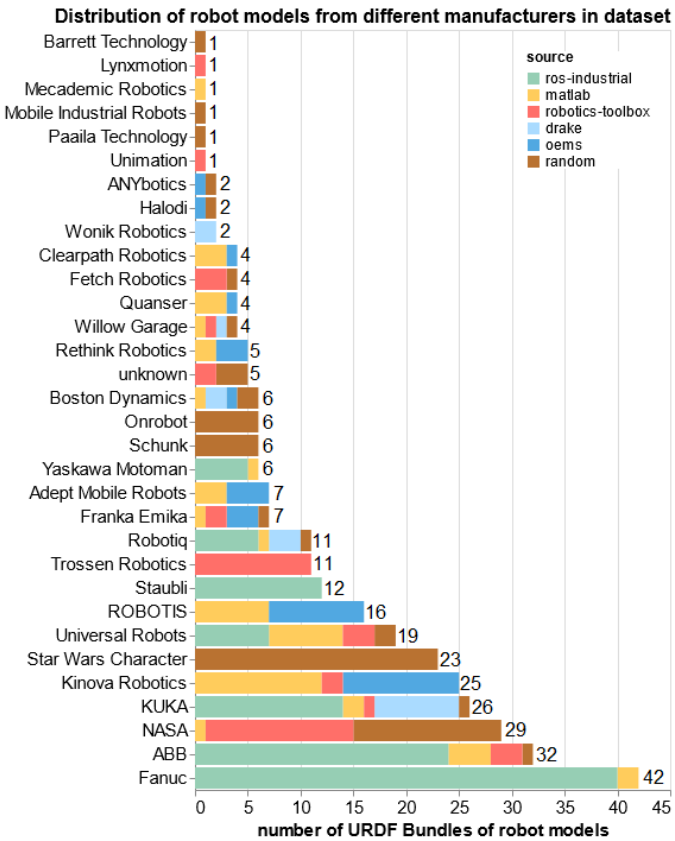

Fig. 4: Bar plot of manufacturers and the number and sources of URDF Bundles of their robot models in the dataset.

- One is unknown, as the robot name is not provided, and four are test URDF files that do not represent a robot. These are all marked as *Unknown* in the dataset.
- One is fictional, the robot is R2-D2 from the movie Star Wars, marked as *Star Wars Character* in the dataset.
- Three are no longer operational (at the time of writing); these are *Unimation*, *Willow Garage*, and *Adept Mobile Robots*.

We have divided the manufacturers into four application categories based on what their robots are mainly used for:

industry: the robots are mainly used in applications for making profits. Examples are robots in manufacturing and service robots.

non-profit: the robots are mainly used for research, education, or for hobbyists.

industry & non-profit: the robots are widely used for both industrial and non-profit purposes.

other: represents manufacturers that do not fit into any of the previous categories. This category is not taken into account when calculating general numbers on manufacturers. The two manufacturers in this category are *Unknown* and *Star Wars Character*.

This categorization is based on our subjective opinion, and considers the following information:

- our own knowledge about the manufacturers,
- checking each manufacturer's website to see the type of customers they are targeting. For example, some of the manufacturers explicitly use the keywords *industrial robots* or *research* or *education*,
- looking at the types of robots they develop. For example, if they develop small robot kits, they can be associated with research or hobbyists.

TABLE III: Number of manufacturers that supply a URDF Bundle for at least one of their robots, based on the target application of the manufacturer's robots. The percentage of robots describes how many robots in the dataset represent models from manufacturers of the specific application.

Application categories	**URDF supplied by manufacturer**	**% of robots from category in dataset**
industry	8/20 (40%)	77
non-profit	6/7 (86%)	12
industry & non-profit	2/3 (67%)	10
other	—	1
Total	16/30 (53%)	100

TABLE IV: Number of manufacturers that provide URDF Bundles versus how many manufacturers link to these URDF Bundles directly from their website. As the table shows 4/16 manufacturers provide information about URDF on their websites. The column 'no website' shows the manufacturers that are no longer operational (at the time of writing).

Has URDF	*total*	**'URDF' in search**			
		not found	**found**	**no search bar**	**no website**
yes	16	7	4	3	2
no	14	9	0	4	1

Table III shows the number and percentage of manufacturers that provide a URDF Bundle for at least one of their robots. 40% of the industry-targeted manufacturers supply at least one URDF Bundle (77% of the dataset). 86% of manufacturers targeting non-profit applications supply a URDF Bundle of at least one of their robots (12% of the dataset). The list of manufacturers, their categorization, and the procedure to determine if a manufacturer supplies URDF Bundles, can be found in the GitHub repository [20]. It is important to note that these numbers may differ in reality, as some manufacturers provide URDF files through collaborations with third party organizations.

Given the value of simulation we would expect manufacturers to help users by making URDF Bundles readily available, and also provide references or relevant information about URDF on their website. To test this hypothesis we visited the website of the manufacturers in the dataset and checked if the term 'URDF' could be found when searching their websites. The results are shown in Table IV.

B. Common URDF Folder Structures

Four of the most commonly used folder structures in the dataset were identified and quantified, see Table V. Folder structure A, shown in Fig. 5, characterizes the structure used for multiple URDF Bundles from a single manufacturer, where multiple URDF Bundles within a directory typically represent a particular robot and its different URDF variants. The same structure can be used for single URDF Bundles,

TABLE V: Four of the most common folder structures in sources and the number of URDF Bundles using them.

Source	Structure			
	A	B	C	D
`ros-industrial`	1	55	–	4
`matlab`	17	5	3	–
`robotics-toolbox`	8	–	1	–
`drake`	3	–	–	–
`oems`	9	–	1	–
`random`	4	–	1	4
Total	42	60	6	8

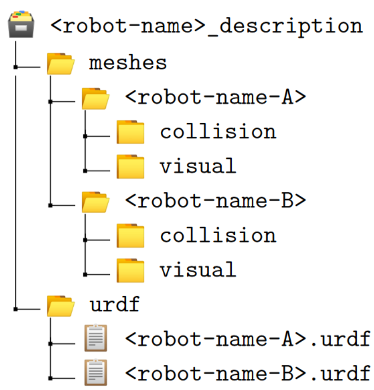

Fig. 5: Folder structure A for multiple URDF Bundles.

where there is only one *.urdf* file in the `urdf` folder, and the `collision` and `visual` folders are directly under the `meshes` folder. In some cases, the `collision` and `visual` folders do not exist, and the CAD files are placed directly in the `meshes` folder. Structure B is similar to structure A with the difference that the root folder is named `<manufacturer-name>_<robot-name>_support`. This structure is also used for both single or multiple URDF Bundles. Structure C is nearly identical to structure A, however, the `urdf` folder is instead named `robots`. Structure D is also similar to structure A, however, the root directory name ends with `visualization` instead of `description`. Structures C and D are only used for single URDF Bundles. Structure D was found to only be used for end effectors. It is important to note that not all the URDF folders follow the described structures, we have chosen to only present the most commonly used structures and count their occurrence in the dataset.

C. Xacro Generated URDF Bundles

The number of URDF Bundles in the dataset that were generated with and without xacro are shown in Table VI. As the results show most of the URDF Bundles (95%) have been generated using xacro.

D. Multiply Defined Robots

In total 60 robots are multiply defined in the dataset with 130 URDF Bundles from different sources, implying an average of 2.2 sources per multiply defined robot. Table VII shows a comparison of features of the URDF Bundles of the multiply defined robots. It shows that the feature with

TABLE VI: Number of URDF Bundles that have been generated in the dataset with and without xacro. The column 'By us' represents the URDF Bundles that we generated while creating the dataset. The column 'By others' represents URDF Bundles created by others.

Source	By us	By others	
	using xacro	using xacro	without xacro
`ros-industrial`	90	18	0
`matlab`	0	49	3
`robotics-toolbox`	17	24	3
`drake`	0	16	0
`oems`	26	6	3
`random`	12	49	6
Total	145	162	15

TABLE VII: Feature differences across the multiply defined robots. The *any* feature indicates at least one difference was found between the URDF Bundles, while the *any excl. lines* represents the same, however, excluding the number of lines.

Feature discrepancies	#Robots
number of joints	9
number of links	9
CAD file type	6
forward kinematics	11
number of lines	38
any	38
any excl. lines	12

the largest number of differences is the number of lines in the URDF files. The number of joints and links differed in 9 multiply defined robots, where we found some of the differences were additional joints and links representing the world and end effectors.

One of the surprising results is that 11 multiply defined robots had different forward kinematics. Different forward kinematics implies the behavior of the robot models vary depending on which source the URDF Bundle originates from. To better understand these discrepancies across URDFs, in the following sections, we analyze the parsing of URDF files, the CAD formats used for visualization and collision, and compare the link and joint names in different URDFs.

E. Parsing the URDF Files

Each URDF file was validated and the errors and warnings were extracted, using the official ROS URDF parser [21] from the `urdfdom` package [22] version 3.1.0. Information about validation using other URDF parsers is provided in Section V. The results showed that 11/322 URDF files failed the parser, with the errors summarized in Table VIII. The source with the highest number of 4 URDF files failing is `random`. Only one URDF file (from `drake`) resulted in a warning, which was that the link material was undefined. A more detailed description of the errors follows below:

- **(E1) Issue with joint limits:** revolute and prismatic joints require joint limit specification of effort and velocity. If none of these attributes is provided, then the URDF file results in an error.

TABLE VIII: The URDF files' parsing results using the official ROS parser. The errors are described in Section IV-E.

Error	#URDF files	Sources
E1	3	drake
E2	4	random (2), robotics-toolbox (2)
E3	1	random
E4	1	oems
E5	3	random
E6	11	random (4), oems (3), drake (2), robotics-toolbox (2)

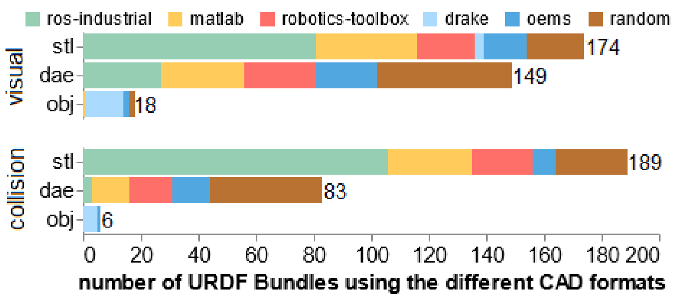

Fig. 6: Number of URDF Bundles referring to different CAD formats based on mesh use.

- **(E2) No link elements found in URDF file:** at least one link is required in a URDF file, otherwise there is no kinematic structure represented.
- **(E3) Non-unique link:** each link name must be unique in a URDF file to distinguish between them when assigning them as parent or child links in joints.
- **(E4) No name given for robot:** the name of the robot in the URDF file must be specified.
- **(E5) Parent link not found:** a joint requires both a parent and child link for the parser to be able to place the joint in the kinematic structure.
- **(E6) XML parsing failed:** parsing of the xml file failed.

F. CAD Formats

The number of URDF Bundles with meshes based on the CAD format is shown in Fig. 6. STL is the most commonly used, followed by COLLADA and OBJ. Not all URDF Bundles contain collision meshes, as there are 341 URDF Bundles with visual meshes, but only 278 with collision meshes. Collision meshes are most commonly STL format. Furthermore, some URDF Bundles combine different CAD formats for visual meshes, explaining the fact that there are 341 URDF Bundles with visual meshes compared to the total of 322 URDF Bundles in the dataset.

G. Identical Files

The Linux command `fdupes` [23] was used to find identical files across the different sources. The `fdupes` command finds identical files within a given set of directories by comparing file sizes, MD5 signatures, and comparing the files byte-by-byte. Before running the `fdupes` command on the files, we removed white spaces, tabs, and changed the carriage return of all files to be DOS (CRLF).

TABLE IX: Number of identical files across sets of sources for two robots. The {} contain the sources that share the specified number of identical files.

Robot	File type	Sources	#Identical files
PR2	stl	{matlab, drake}	8
		{matlab, robotics-toolbox, drake, random}	31
		{robotics-toolbox, random}	9
	dae	{matlab, random}	18
		{matlab, robotics-toolbox, random}	5
Franka Panda	stl	{matlab, oems, random}	9
		{matlab, random}	1
	dae	{robotics-toolbox, oems}	10

TABLE X: Number of URDF files where joints or links contain the words *world* or *flange* in their name.

Source	world	flange
ros-industrial	3	204
matlab	30	14
robotics-toolbox	12	0
drake	1	0
oems	24	0
random	112	0

Examples of identical files across sources for two robots are shown in Table IX. The sources with the *.stl* files differ from the sources with the *.dae* files, and the number of identical files across the sources varies. This makes it difficult to infer how these files have propagated and evolved across the sources. For example, the PR2 has 8 identical files common to `matlab` and `drake` which are not part of the 31 identical files common to `matlab`, `robotics-toolbox`, `drake`, and `random`.

H. Joint and Link Naming Conventions

Table X shows the sources and the number of robots that have links or joints with names containing *world* or *flange*. These specific words were found to be one of the main differences between the joints and links of the multiply defined robots in Section IV-D. As illustrated, `ros-industrial` URDF files have a significant number of occurrences of the word *flange*, implying they may use this convention when developing URDF files. The word *flange* only appeared in `ros-industrial` and `matlab`; which have identical mesh files (STL or COLLADA), across 11 URDF Bundles.

I. Author Contact Information

In many cases, it may be relevant to know who to contact when working with a URDF model of a robot. This could be indicated by a comment in the URDF file that included the name or email address of the author. We counted the number of URDF files containing words or symbols related to contact information of authors and found that the word '*author*' occurred in total in only 13 files, the symbol '@' that can be associated with emails occurred in 16 files, and '*.com*' occurred in 13 files. This shows that providing contact information in URDF files is not common practice.

TABLE XI: Different licenses used in sources of the dataset.

Source	Apache License v2.0	BSD 3-Clause	BSD 2-Clause	MIT License
`ros-industrial`	57	45	6	-
`matlab`	10	36	5	1
`robotics-toolbox`	-	-	-	-
`drake`	-	16	-	-
`oems`	13	17	-	-
`random`	5	3	-	6
Total	85	117	11	7

J. Licensing

The URDF Bundles in the dataset are protected by different licenses based on the Open Source Initiative [24]. The four main licenses found in the dataset are shown in Table XI. These licenses are very similar, as they are permissive, meaning developers can use and modify the files, and make their own new versions of them. The minor differences between these licenses are related to non-endorsement and patenting rights. When protecting open-source software through licenses, it is natural to choose the most commonly used license within the community, which in this case as shown in Table XI is the BSD 3-Clause license.

V. Construction and Reproducibility

This section presents our rationale for constructing the dataset. Additional notes and information can be found in the GitHub repository [20].

A. Construction of the Dataset

We have constructed the dataset to be representative of general URDF Bundles that can be generated and found on the internet, and have therefore chosen different types of sources with the following classifications:

- ROS-related sources (`ros-industrial`)
- Commercialized tools (`matlab`)
- Original Equipment Manufacturers (`oems`)
- Common tools used by roboticists (`robotics-toolbox`, `drake`)
- Various repositories that users may find when searching for URDF Bundles (`random`)

Although it may be suspected that the quality of the URDF Bundles from the `random` source is lower than an industrial source (e.g. MATLAB), it is important to include all representations of URDF to understand what a general user may find in their search for URDF Bundles. The dataset may be biased, by the fact that the `ros-industrial` source contains the larger fraction (34%) of the URDF Bundles. This may affect the results when analyzing, for example, folder structures, URDF file generation using xacro, and the types of meshes.

Our dataset is publicly available, making it possible for others to contribute their URDF Bundles, and perform analyses on the newly-added data. Instructions on how to add new URDF Bundles or sources are described in the dataset repository.

TABLE XII: Overview of the URDF parsers currently supported by our URDF analyzer tool.

Parser	Version	Origin
yourdfpy	0.0.52	PyPi
urdfpy	0.0.22	PyPi
pybullet	3.2.5	PyPi
robotics-toolbox	1.0.3	PyPi
MATLAB (R2022b)	9.13	PyPi
ROS parser (urdfdom)	3.1.0	conda-forge

B. Reproducing Results

To produce the results in this article, the dataset contains accompanying Python scripts, located in the *scripts/paper_results* directory of the GitHub repository [15]. All the CSV files with the results presented in this article can be found in the GitHub repository with the dataset material [20].

C. URDF Analysis Tool

In addition to the scripts for analyzing the dataset, we are developing a Python-based tool for analyzing URDF Bundles. The tool is publicly available and can be found in the GitHub repository [25]. It has been created to operate as a standalone command line tool, but can also be used together with the dataset. The tool combines the capabilities of other URDF tools and can be used independently of ROS. The tool can be used to generate the following information:

Parser comparison results: the tool currently supports 6 different URDF parsers, presented in Table XII. Results of running all the parsers on all the URDF files in the dataset are shown in Table XIII.

URDF files parsing results: this feature can be used to analyze each URDF file specifically with regards to which URDF parsers failed or succeeded with it. A URDF file is defined as being successfully parsed, if the result of loading the URDF file with the parser, contains a Python object and is not ‘None’. There may be warnings while loading the URDF files, but as long as they can be loaded into an object, we count them as being successfully parsed.

Comparison of multiply defined robots: provides information on multiply defined robots, describing the name of the robot, manufacturer, type, sources, and if there are discrepancies in the number of joints or links, mesh types, forward kinematics, and the number of lines in the URDF files.

Model information: constructs a table with information on the number and names of the joints and links in each URDF file, and the types of CAD formats used.

Up to date information about the tool is provided in the repository.

The results in Table XIII imply that the so-called Unified Robot Description Format may not be as unified as one would expect. This may indicate a lack of documentation, forcing the creators of the different URDF parsers to develop parts of the parsing mechanisms with an incomplete understanding of the URDF rules/schema, resulting in inconsistent outcomes across different parsers.

TABLE XIII: Success rate for parsing URDF files from different sources with various parsers. The underlined values signify the source and parser are from the same organization.

Source	total	*yourdfpy*	*urdfpy*	*pybullet*	*robotics-toolbox*	*MATLAB*	*ROS parser*
`ros-industrial`	108	106	0	90	108	108	108
`matlab`	52	21	0	39	50	52	52
`robotics-toolbox`	44	25	0	36	42	42	42
`drake`	16	8	0	0	14	14	14
`oems`	35	17	1	30	31	31	32
`random`	67	43	6	57	61	62	63
Total	322	220	7	252	306	309	311

VI. CONCLUSION AND RECOMMENDATIONS

In this article we presented a novel dataset of URDF Bundles accompanied by analyses of the files. The main highlights of the analyses are:

- Around half (16/30) of the original robot manufacturers supply a URDF Bundle for at least one of their robots.
- Reliance on xacro is very common, 307/322 (95%) URDF Bundles were generated using xacro.
- Parsers vary in performance, 11/322 URDF files resulted in errors when parsing them with the official ROS parser – the most common error being "XML parsing failed". The rules/schemas they follow are inconsistent, and there is no gold-standard test suite.
- STL is the most used CAD format for visualizations.
- Metadata such as author, date, version and modification history is rare. This could be provided using XML comments but there is no convention for doing this.
- URDF Bundles are copied and evolved across the internet in an undisciplined way. Different models of the same robot may have different appearance or kinematics. Without good metadata it is impossible to know the provenance of files or which evolved from which.
- The most commonly used open source software license to protect URDF files is the BSD 3-Clause license.

Based on these analyses we recommend that:

- Manufacturers publish their own URDF Bundles with one static URL per robot or host them on GitHub to reveal the full revision history.
- Detailed guidelines and conventions are created for:
  - folder structures of URDF Bundles and for different types of robots,
  - CAD formats and variants of URDFs depending on the model's usage,
  - the naming of links and joints of different types of robots,
  - embedded metadata.

URDF is an important robotic technology and widely used, but it is unusual in not being a formal standard. There is no active "owner" committed to documentation, establishing conventions or extending it. Perhaps there is a role for the IEEE Robotics and Automation Society or others to remedy this situation.

REFERENCES

[1] L. Sanneman, C. Fourie, and J. A. Shah, "The state of industrial robotics: Emerging technologies, challenges, and key research directions," *CoRR*, vol. abs/2010.14537, 2020. [Online]. Available: https://arxiv.org/abs/2010.14537

[2] A. Afzal, D. S. Katz, C. Le Goues, and C. S. Timperley, "Simulation for robotics test automation: Developer perspectives," in *2021 14th IEEE Conference on Software Testing, Verification and Validation (ICST)*, 2021, pp. 263–274.

[3] H. Choi, C. Crump, C. Duriez, A. Elmquist, G. Hager, D. Han, F. Hearl, J. Hodgins, A. Jain, F. Leve, C. Li, F. Meier, D. Negrut, L. Righetti, A. Rodriguez, J. Tan, and J. Trinkle, "On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward," *Proceedings of the National Academy of Sciences*, vol. 118, no. 1, p. e1907856118, 2021. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.1907856118

[4] M. Quigley, B. Gerkey, and W. D. Smart, *Programming Robots with ROS: A Practical Introduction to the Robot Operating System*, 1st ed. O'Reilly Media, Inc., 2015.

[5] M. Ivanou, S. Mikhel, and S. Savin, "Robot description formats and approaches: Review," in *2021 International Conference "Nonlinearity, Information and Robotics" (NIR)*, 2021, pp. 1–5.

[6] D. Tola and P. Corke, "Understanding URDF: A Survey Based on User Experience," in *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, 2023, pp. 1–7.

[7] C. Symeonidis and N. Nikolaidis, "Chapter 18 - simulation environments," in *Deep Learning for Robot Perception and Cognition*, A. Iosifidis and A. Tefas, Eds. Academic Press, 2022, pp. 461–490. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780323857871000233

[8] S. Carón. (2024) Awesome Robot Descriptions. [Online]. Available: https://github.com/robot-descriptions/awesome-robot-descriptions

[9] Gepetto. (2024) Example Robot URDFs. [Online]. Available: https://github.com/Gepetto/example-robot-data

[10] ROS. (2022) URDF XML Link. [Online]. Available: https://wiki.ros.org/urdf/XML/link

[11] ——. (2022) URDF XML Joint. [Online]. Available: https://wiki.ros.org/urdf/XML/joint

[12] ——. (2023) XML Robot Description Format (URDF). [Online]. Available: https://wiki.ros.org/urdf/XML/model

[13] ——. (2022) Xacro. [Online]. Available: https://wiki.ros.org/xacro

[14] N. Albergo, V. Rathi, and J.-P. Ore, "Understanding xacro misunderstandings," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 6247–6252.

[15] D. Tola. (2023) A URDF Dataset. [Online]. Available: https://github.com/Daniella1/urdf_files_dataset

[16] ROS-Industrial. (2024) ROS-Industrial Github Repositories. [Online]. Available: https://github.com/ros-industrial

[17] MATLAB. (2023) importrobot. [Online]. Available: https://se.mathworks.com/help/robotics/ref/importrobot.html

[18] P. Corke and J. Haviland, "Not your grandmother's toolbox–the robotics toolbox reinvented for python," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 357–11 363.

[19] RobotLocomotion. (2024) drake. [Online]. Available: https://github.com/RobotLocomotion/drake

[20] D. Tola. (2023) Material for reproducing the results of the URDF dataset analysis. [Online]. Available: https://github.com/Daniella1/urdf_dataset_results_material

[21] ROS. (2023) urdfdom. [Online]. Available: https://github.com/ros/urdfdom

[22] Anaconda.org. (2024) urdfdom. [Online]. Available: https://anaconda.org/conda-forge/urdfdom

[23] L. man page. (2024) fdupes. [Online]. Available: https://linux.die.net/man/1/fdupes

[24] "Open Source Initiative," https://opensource.org/, [Online; accessed 25-06-2023].

[25] D. Tola. (2023) URDF Analyzer. [Online]. Available: https://github.com/Daniella1/urdf_analyzer