

URDF Based Generalized 3D Printer Simulation in MaxWhere Environment

Botond Horváth^{1*}, András József Czmerk¹

¹ Department of Mechatronics, Optics and Mechanical Engineering Informatics, Faculty of Mechanical Engineering, Budapest University of Technology and Economics, Műgyetem rkp. 3., H-1111 Budapest, Hungary

* Corresponding author, e-mail: horvath.botond@mogi.bme.hu

Received: 04 June 2026, Accepted: 14 August 2026, Published online: 07 October 2026

Abstract

With the advancement of technology, cyber-physical systems are increasingly gaining ground. Today, virtual reality is appearing in numerous areas - not only in industrial applications or popular video games but also in education, facilitating a deeper understanding of various processes. The topic of this paper is the implementation of a generalized 3D printer simulation using the 3D virtual software MaxWhere. This simulation can demonstrate the operation of any Cartesian printer by providing its 3D models and URDF description, thereby offering an easily accessible and manageable demonstration tool to support education. For the sake of user-friendliness, an AI-based URDF generation application is also presented.

Keywords

3D printer, cyber-physical systems, generative AI, LLM, URDF, industry 4.0, VR

1 Introduction

Visualizing a general 3D printer in space based on a URDF (Unified Robot Description Format) model has proven to be a feasible and effective way to make additive-manufacturing concepts tangible for students.

In the case of Cartesian printers, printing occurs via linear movements along the three axes (X, Y, Z); thus the individual elements - which practically represent each axis - are connected to one another by translational joints, see Fig. 1. [1]

The objective of this work was to create a visually engaging and realistic simulation that can be customized to any Cartesian 3D printer model and can be installed in

an arbitrary virtual environment, providing a wide range of applications and enhancing the user experience, thus making it easier to understand.

This paper presents first a URDF-driven, generalized Cartesian 3D printer component for the MaxWhere virtual space [2] that can represent an arbitrary printer's geometry and kinematics instead of a single hard-coded model. Secondly, an AI-assisted web application is described that generates a compliant URDF description for a given printer from a natural-language prompt, removing the need for manual URDF authoring. Third, the reliability of the AI-generated URDF files were evaluated across three common Cartesian kinematic layouts and were reported where the generation succeeds.

2 Scientific background

2.1 Simulations in manufacturing

Virtual simulations play a particularly important role in engineering and industrial applications, especially in mechanical engineering. Modeling the behavior of machines, equipment, components, and final products makes it possible to detect design errors early, thereby saving considerable time, energy, and money [3, 4]. Mourtzis et al. [5] review these developments across the manufacturing life cycle and

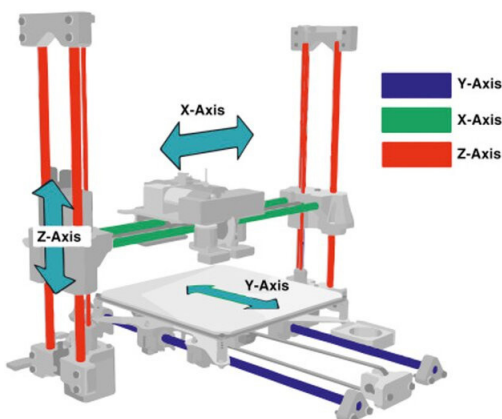


Fig. 1 The structure of a Cartesian 3D printer

highlight persistent gaps in simulation interoperability and industrial adoption, while Zhang et al. [6] focus specifically on how simulation supports intelligent, data-driven manufacturing, and Garwood et al. [7] and Afazov [8] examine simulation tools for energy efficiency and process-chain modeling, respectively.

FEM (Finite Element Method) software, which is also popular in university education, can be used to perform structural simulation. Simulations dealing with the design, testing, analysis, and optimization of industrial processes and the creation of industrial environments also serve a valuable purpose, as they allow a new system to be tested before live deployment, such as Plant Simulation [9] or Process Simulate [10].

2.2 3D printer simulations

It is also advisable to first test manufacturing processes using simulation software, thus avoiding the production of possible rejects or damage to the equipment. However, for additive manufacturing, no such deep demonstration option is available.

CNC Simulator, which according to its website was the first 3D printer simulation in the world [11], offers a quite basic solution to the problem. In this software, the user interacts with a realistic printer, can input custom G-code for printing, start and stop the simulation via a control panel, and adjust the printing speed. While the simulation is running, the interface displays the currently executed line of G-code. However, the visualization of the print process is quite basic, and the printer model components are schematic and simplified, and only the default machine can be used for printing. Nevertheless, it is effective for demonstrating the printing process and executing G-code, although a more visually engaging and realistic model could potentially enhance its utility.

2.3 Cartesian 3D printers

The primary advantage of the Cartesian printers lies in the simplicity of their mechanism, which accounts for their outstanding popularity and low cost; indeed, the majority of 3D printers operate within a Cartesian coordinate system, where every point of the model is associated with three values that determine its specific position in space [12, 13]. Although the presence of multiple axes and motors increases the potential for inaccuracies, these systems are easily calibrated, allowing significant errors to be eliminated [14]. Since the Cartesian mechanism is the most widespread and mechanically transparent, and

therefore the most representative case, we focus on it in greater detail throughout this paper.

2.4 3D printer simulation in MaxWhere

At the beginning of this development, we created a virtual simulation of the Craftbot Plus Pro 3D printer (Fig. 2) that can help students to understand the basic principles of the most widespread additive technique, Material Extrusion, focused on the Fused Filament Fabrication (FFF) process because of its accessibility and relevance in educational contexts [15]. A printed product can be built up step by step based on the arbitrarily chosen G-code while the machine moves accordingly.

The printer simulation proved to be sufficiently realistic, allowing users to closely observe the processes through free movement within the 3D space, thus enhancing the software's educational value as a visualization tool. By increasing the simulation speed, the printing process is completed more quickly, thereby minimizing the wait time for the final result. A comparison between the objects produced by the physical and virtual printers reveals that they are nearly identical, demonstrating that the model effectively replicates real-world printing quality in the virtual space.

This project can be the initial state of a generalized virtual Cartesian 3D printer which can simulate the behavior of an arbitrary machine according to the users' needs.

3 Materials and methods

3.1 MaxWhere environment

MaxWhere is a software [2] platform that provides interactive 3D virtual environments to enhance how people work,



Fig. 2 The simulation of Craftbot Plus Pro in MaxWhere

learn, and collaborate. It is designed to simulate real-world spaces in a digital format, making information processing more intuitive and engaging. MaxWhere bridges the generational gap between educators and students by providing a familiar and proven environment for teachers while immersing students in a stimulating 3D space. Csapó et al. [16] relate this multidimensional layout to classical memory-palace techniques, showing that a persistent 3D spatial organization helps users form and recall complex conceptual relationships, while Ujbányi et al. [17] report similar cognitive-load benefits in transparent MaxWhere working environments allowing faster information processing and a deeper understanding of complex connections. Berki [18] further quantifies this benefit, measuring a significantly higher subjective sense of presence. Specifically optimized for education and industrial demonstration applications, MaxWhere allows multiple users to simultaneously enter the same 3D space and work in real time [19, 20], which sets it apart from competitors like Unity Engine [21] or Unreal Engine [22].

MaxWhere spaces are powered by the MaxWhere framework, which runs on top of the MaxWhere Engine, which combines 3D graphics with web-based content. High-level programming in MaxWhere spaces is done by a JavaScript API (Node.js) [23].

The Where Object Model (WOM) is a tree-structured representation of a three-dimensional space and all its elements. The WOM can contain 3D objects, 2D surfaces, light sources, WebView objects, and custom components. These components are encapsulated objects that combine visual appearance with functionality. A given MaxWhere space can contain many components. This paper uncovers a special 3D printer component that can be installed in any MaxWhere space with MaxWhere version 7.

3.2 Craftbot Plus Pro printer

The Craftbot Plus Pro (Fig. 3 [24]) is a high-performance plug-and-play desktop FFF 3D printer developed by CraftUnique, designed for both professional and educational use.

3.3 Creality Ender 3 printer

Creality Ender 3 (Fig. 4 [25]) is one of the most popular and affordable entry-level Fused Deposition Modeling (FDM) 3D printers on the market. Known for its reliability, open-source design, and strong community support, the Ender 3 offers excellent value for beginners and hobbyists alike.

Despite its low cost, the Ender 3 is capable of producing high-quality prints after basic calibration and maintenance. Its open-source firmware allows for extensive modifications



Fig. 3 Craftbot Plus Pro



Fig. 4 Creality Ender 3

and upgrades, such as automatic bed leveling, silent stepper drivers, or improved extruders thus making it so popular [26].

3.4 URDF

URDF (Unified Robot Description Format) is the most popular robot modeling format, originally developed for ROS [27], but now used in many other areas. It is an XML file used to describe the physical and kinematic structure of a robot. It defines a robot's components - such as links, joints, sensors, and actuators - as well as their physical properties like mass, inertia, and visual geometry [28].

3.5 FastAPI

FastAPI is a cutting-edge, high-performance web framework designed for building APIs with Python, leveraging standard Python type hints to deliver a seamless developer experience. Built upon the robust foundations of Starlette for web routing [29] and Pydantic for data validation [30], it stands out as one of the fastest Python frameworks

currently available, reaching performance levels comparable to Node.js and Go [31]. Its core strength lies in its native support for asynchronous programming, which allows it to handle concurrent workloads with exceptional efficiency. Beyond raw speed, FastAPI drastically accelerates the development lifecycle by automatically generating interactive API documentation via Swagger UI [32] and ReDoc [33], ensuring that endpoints are always synchronized with the codebase. Ultimately, FastAPI represents a modern shift in backend architecture, providing a scalable, production-ready solution that is equally well-suited for microservices, machine learning deployments, and high-traffic cloud environments [34, 35].

3.6 Vercel

Vercel is a leading cloud platform designed to optimize the development, hosting, and deployment of modern web applications, serving as the primary infrastructure for the Next.js framework. It revolutionizes the developer experience by offering a seamless Git-integrated workflow, where every push to a repository triggers an automatic build and deploys a unique preview URL, enabling real-time collaboration and testing. At its core, Vercel focuses on delivering exceptional end-user performance through its global edge network, which serves content from locations closest to the user to minimize latency. The platform excels at handling hybrid rendering strategies, effortlessly managing Static Site Generation (SSG), Server-Side Rendering (SSR), and Serverless Functions within a single environment. By abstracting away complex infrastructure management and providing built-in features like automatic SSL, custom domain handling, and advanced analytics, Vercel allows engineering teams to focus entirely on building high-quality frontend experiences while maintaining the scalability and reliability required for production-grade applications [36, 37].

3.7 OpenAI

The OpenAI API is a cloud-based platform that provides access to OpenAI's advanced artificial intelligence models, such as GPT-4, DALL-E, and Whisper [38]. It allows developers to integrate natural language understanding, text generation, image creation, and speech recognition into their own applications.

With the OpenAI API, users can build chatbots, virtual assistants, content generators, code helpers, and many other AI-powered tools. It is designed to be flexible and easy to use, supporting both simple requests and complex workflows

through RESTful endpoints. In short, the OpenAI API gives developers powerful access to cutting-edge AI capabilities for a wide variety of real-world applications [39].

3.8 LangChain

LangChain is an open-source framework designed to simplify the development of applications powered by large language models (LLMs) [40]. It provides a modular architecture that enables developers to integrate language models with external data sources, APIs, and custom logic in a structured and scalable manner. In this work, the Python implementation of LangChain was employed to orchestrate interactions with LLMs, allowing for the construction of complex processing pipelines through components such as chains, agents, and memory modules. Chains enable the sequential execution of multiple processing steps, while agents introduce dynamic decision-making capabilities by selecting appropriate tools based on the input context [41]. Additionally, LangChain facilitates prompt management and template reuse, improving consistency and maintainability of LLM interactions. Its support for document loading, vector stores, and retrieval-augmented generation (RAG) - the technique introduced by Lewis et al. [42], who combine a pretrained generator with a retrieved document index to ground outputs in external knowledge - makes it particularly suitable for applications requiring the integration of domain-specific knowledge. Overall, LangChain significantly reduces development overhead and enhances flexibility when building robust, LLM-driven systems.

4 Build the printer

4.1 URDF links

In the URDF model, the four primary components must first be defined, consisting of the machine base and the parts representing the three axes. By default, all parameters for the individual links are considered zero, requiring only the link name and its corresponding mesh file to be specified. These files must be placed in a predefined folder of the component to ensure accessibility based solely on their filenames.

To ensure the realistic operation of the printer, additional links are necessary; while the core functions can be well-represented without them, certain elements common to all printers should be included. Specifically, the component acting as the print bed must be designated for proper functioning, and it is also recommended to visualize the filament spool model. Furthermore, as every printer includes a control panel, its position must be defined within the URDF

structure to designate the placement of the smart tablet serving as the control interface. If cooling is required for the print head, a fan and its associated mesh file must also be integrated into the model. Finally, auxiliary points must be created to visualize the filament hanging from the spool and the tube guiding it to the head, allowing for a Bézier curve to be fitted to the specified points.

4.2 URDF Joints

The relationships between the various components are defined by the joints connecting them, which are described as joints within the model. With the exception of the machine base, every individual link must be connected to the machine via some type of joint. These joints connect the axes either to the base or to one another, depending on the specific mechanical configuration. Since the system in question is a Cartesian printer, these are implemented as prismatic joints. It is crucial to define their orientation such that they designate the direction of positive displacement for the given axis. Furthermore, the relative positions of the parent and child elements must also be specified here. If a fan is included, it is connected to the print head via a revolute joint, where the direction of the positive rotation axis must be defined. All other links are attached to their respective parent links using fixed joints, ensuring they are coupled with the component they are intended to move with.

4.3 URDF processing

The structure of the printer is built within the component based on the provided URDF model.

First, the XML file is read to allow for the interpretation of its content, where the `printerSource` variable determines which URDF model is utilized for the given component.

By converting the read file into a JSON object, it becomes easily processable; for this purpose, we used the `Parser()` function from the `xml2js` library [43]. The file, converted to JSON format, was first read into an empty string, and then a JSON object was created using the `JSON.parse()` function.

Within the object formed from the processed URDF model, a link and a joint array were created, containing the respective links and joints. By observing the structure of the robot object and iterating through its elements, the necessary information can be filtered out.

Based on conventional naming standards, the sub-assemblies can be selected from the object during iteration. The process involves checking for the existence of such an element among the links; if found, its mesh URL is

saved, followed by an iteration through the joints to store the associated position and orientation. A similar process is applied to the other parts, with the exception of the axes and the auxiliary points, the latter of which do not have associated mesh files.

A different approach is required for the axes: it must be determined which part fulfills the role of which axis before the bodies in the program can be mapped to the axes. This is easily determined by extracting the elements of the `xyz` attribute of the axis tag stored in the URDF into an array, where movements along each axis correspond to individual elements of the array. Since this is a Cartesian printer, exactly one element will always be 1, while the others remain 0. As the fan is the only moving part besides the axes, a conditional branch must ensure it is handled separately so that only the axes are processed.

In the `MaxWhere` space, bodies can be created using the `wom.create()` function, utilizing the previously extracted data. The identifier of every object created includes the component's unique ID to avoid potential conflicts when multiple components are instantiated within the same space. The dimensions and resolution of the control panel must be specified when calling the component. Using these parameters, along with the data extracted from the URDF model, a billboard can be created to load arbitrary content - in this case, the web-based user interface.

At this stage, the empty nodes used in the original program (such as `printBaseCube`, `printBaseTube`, and `printCoordTracker`) are initialized to assist during motion. In this instance, providing an identifier is sufficient, as all other values default to zero. Furthermore, since it is not mandatory to assign a body to nodes - and in this case, no visual representation is desired - no file path needs to be specified. However, to fit a curve to the auxiliary points of the Bézier curves, mesh files had to be created, as this is not possible with empty nodes. Therefore, we created a unit cube for each point; these are negligibly small and practically unnoticeable, thus they do not interfere with the simulation while fulfilling their intended function.

4.4 Parent-child relationships

The hierarchical relationships between elements can be determined based on the joint connections defined in the URDF model. To achieve this, we iterated through the joints, individually examining both the parent and child elements. The selected elements were then linked to one another using the appropriate function within `MaxWhere`, so the kinematic chain can be built.

Once these relationships are established - since every individual element can be traced back to the machine base as the ultimate ancestor - it is sufficient to render only the base; all descendant elements will be rendered accordingly as a result. Following this logic, the loading of the Creality Ender 3 and Craftbot Plus Pro printers into the Virtual Classroom based on their URDF models was successfully completed by using two printer components with different configurations in the same space (Fig. 5).

4.5 Kinematic chain

The user interface enables the visualization of the given printer's kinematic chain in a tree structure (Fig. 6). To display this hierarchy, we utilized the treeflex library [44], pre-defining four nodes since the frame of every Cartesian printer consists of its base and three axes. In this context, the primary objective was to define the relationships between these elements based on the URDF model. In every case, the machine base serves as the top-level root. Based on the defined joints, whichever axis does not have the base as its parent is placed at the lowest level, while its parent is derived from the joint



Fig. 5 Different printers loaded in the space via URDF model

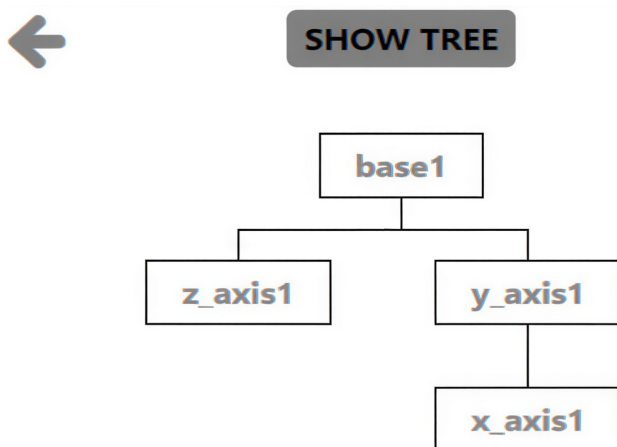


Fig. 6 Kinematic chain of the Craftbot Plus Pro on the control panel

definitions; the fourth element is the descendant of the base to which no further elements are connected. This graph statically represents the connections even if the individual nodes are empty; therefore, we kept the structure hidden until data becomes available, at which point a function toggles its visibility.

4.6 Move the axes

To operate the generalized printer, the movement logic had to be reconstructed, as the original implementation was designed for a single specific printer configuration [15]. We developed a unified function to handle the movement of the individual axes, which expects JSON objects containing the node names and target coordinates within an array. For the execution of the movement, we utilized the MaxWhere Node.animate() method, specifying 'position' as the attribute to be animated, while the target ('to') and 'duration' parameters are extracted from the array elements. Although the method allows various curve types such as circular paths or Bézier curves, we selected a 'linear' type for cubic interpolation, as the system employs translational joints.

The program parses the new coordinates from the G-code and determines whether printing is in progress based on the extrusion rate. An increase in extrusion indicates filament consumption, signifying an active printing process. To calculate the movements, the current positions of the mobile bodies must be retrieved. By utilizing these values along with the initial positions of the machine base and the axes, the relative position of the print head compared to the bed can be determined, as this reflects the displacement across all three axes.

A vector contains the new coordinates from the subsequent command. The relative movement to be executed in the current step is defined by the diff variable, while the distance variable represents the length of the displacement. The duration of the movement is calculated based on this distance: it is divided by the feed rate, then multiplied by 60 to convert the value from [mm/min] to seconds. Since the time is scalable, the resulting duration is divided by the time scale factor to determine the final length of the animation for the given step. Based on the incoming coordinates, the new target positions for each axis are determined by adding the new coordinates - with the correct algebraic sign - to the initial positions, aided by the movement direction vectors extracted from the URDF model. Once all target coordinates are defined, the movement of the individual axes is executed.

5 URDF generation with AI

In order to enable users with limited knowledge of URDF structures to customize the simulation, it was necessary to develop a web application capable of generating robot description files for any arbitrary Cartesian 3D printer.

5.1 Prompt engineering

To achieve this, OpenAI's Large Language Model (LLM) provided the necessary support. By utilizing this model with appropriate prompting, the URDF description for the desired printer is generated.

A critical aspect of this generative technique is the correct identification of the hierarchical relationships between the axes. This is essential for the proper operation of the given printer; however, as these relationships are not always self-evident, the corresponding prompt had to be refined numerous times before the program reached its sufficiently effective form.

Another critical point is determining the orientation of the axes. It depends on the specific construction of each printer whether a movement in a given direction described by the G-code requires a positive or negative translational displacement of the corresponding axis. If the program can correctly determine these parameters as well, the simulation will be able to move appropriately based on the generated URDF.

5.2 Web application

A webapp was created where the user can enter the name of the printer to be used in the simulation and after the generation the URDF model is displayed and can be downloaded as .xml file with a custom name (Fig. 7).

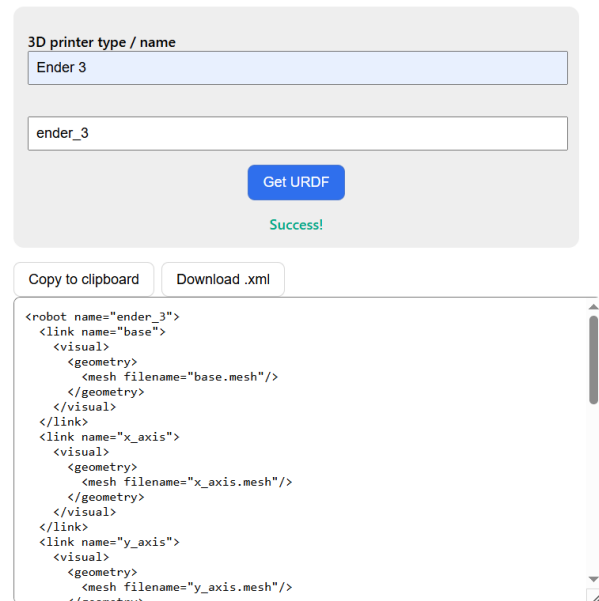
Proper prompting is very important. In the python code the prompt describes the content structure of the URDF so the LLM should only customize it for the actual printer model. The downloaded file should be placed into the proper folder of the MaxWhere space and when running the simulation, the printer is built up by the used mesh files and is moved according to it.

The web application has been hosted to become reachable to everyone who would like to customize the 3D printing space.

5.3 Implementation

The created system is an asynchronous web service that generates URDF representations from natural language input using a large language model (LLM). The backend is implemented using FastAPI, which

URDF Generator



```
<robot name="ender_3">
  <link name="base">
    <visual>
      <geometry>
        <mesh filename="base.mesh"/>
      </geometry>
    </visual>
  </link>
  <link name="x_axis">
    <visual>
      <geometry>
        <mesh filename="x_axis.mesh"/>
      </geometry>
    </visual>
  </link>
  <link name="y_axis">
    <visual>
      <geometry>
        <mesh filename="y_axis.mesh"/>
      </geometry>
    </visual>
  </link>
</robot>
```

Fig. 7 URDF generator app

leverages the ASGI standard to enable non-blocking, concurrent request handling.

The system exposes a RESTful API, where the primary endpoint accepts structured input describing an existing 3D printer. Input validation is handled by Pydantic, ensuring type safety and robustness in request processing. The service can also serve a static frontend for user interaction.

The generative component integrates an LLM via LangChain, interfacing with the OpenAI API (e.g., GPT-4.1). The model is configured with low temperature to ensure deterministic and reproducible outputs.

Output quality is enforced through structured prompt engineering. A detailed system prompt explicitly defines the expected URDF structure, including kinematic chains, parent–child relationships between axes, and coordinate conventions. This is complemented by additional constraint rules provided as separate messages, reinforcing hierarchical dependencies among axes. Together, these form a quasi-declarative rule system embedded within the model context.

The raw model output undergoes post-processing to remove formatting artifacts (e.g., Markdown code blocks) and extract clean XML content. Optionally, the generated URDF can be downloaded to disk.

Overall, the application can be interpreted as a neuro-symbolic system, where the neural component (LLM) performs semantic interpretation and code generation, while the symbolic component is realized through explicitly

defined structural and kinematic constraints in the prompt. This hybrid approach enables the generation of structured robot models from minimal natural language descriptions.

6 URDF generation

The software was tested on Cartesian 3D printers with different kinematic structures to assess the accuracy of the generation process. We evaluated the generated files against four criteria: whether the number of links is correct, whether the joint parenting in the kinematic structure yields the correct kinematic chain, whether the axis directions in the joints are correct, and whether the printing-bed component is properly identified.

6.1 Bed Slinger

The Bed Slinger is the most iconic and affordable Cartesian design. In this setup, the motion is split between the print head and the build plate: the bed moves back and forth along the Y-axis, while the print head moves side-to-side along the X-axis and up/down along the Z-axis on a vertical gantry (e.g., Prusa i3 MK4, Creality Ender 3, Bambu Lab A1, LulzBot TAZ Workhorse) [45].

The results (Fig. 8) show that the generation for Bed Slinger printers is quite accurate, and the printing bed (Y-axis) can be determined correctly. The kinematic chain was also correct for every printer, but in some cases the axis direction for the Y-axis was faulty.

6.2 Box-Frame

In the Box-Frame design, the print head handles all horizontal movement (X- and Y-axes) on a gantry at the top of the machine's frame, while the build plate moves only vertically (Z-axis). This keeps the printed part much more stable, since it is not "slung" back and forth during printing (e.g., CraftBot Plus Pro, Ultimaker S5, Creality Ender 6) [46].

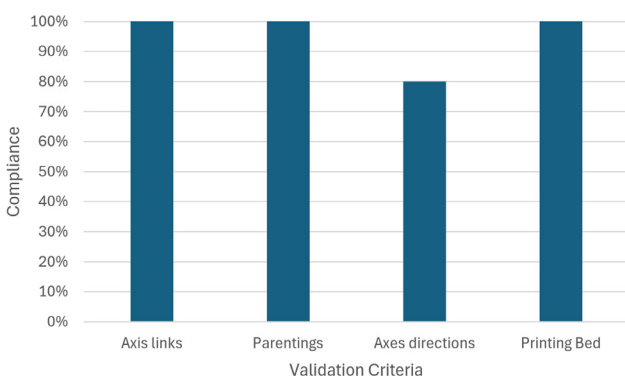


Fig. 8 Generation rate for Bed Slinger printers

As it can be seen in Fig. 9, the generation is not reliable for Box-Frame printers. Although the axis links can be described perfectly, the joint parentings gone wrong in most cases, the printing bed cannot be reliably identified in most cases, which causes the wrong axis directions.

6.3 Fixed Bed

In the Fixed Bed design, the build plate is completely stationary and often bolted to the base, while the entire gantry (the frame holding the print head) moves in all three directions (X, Y, and Z). This resembles how large-scale CNC milling machines or wood routers operate (e.g., Snapmaker Artisan, LulzBot TAZ Workhorse) [47].

For Fixed Bed structures the generated URDF files were perfect (Fig. 10). The reason can be that the printing bed is not one of the printer axes, but it is a part of the base link, it stays still, and the kinematic of the printer is serial.

7 Conclusion

The virtual 3D printer proved to be sufficiently realistic, allowing users to closely observe the processes through free movement within the 3D space with the help of a simulation framework in which an arbitrary Cartesian 3D printer - even with different structures, such as Bed Slinger,

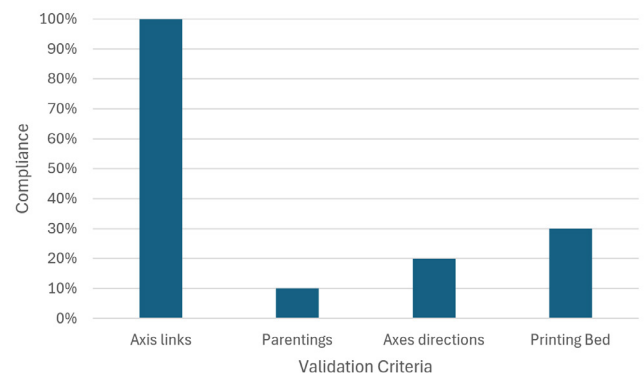


Fig. 9 Generation rate for Box-Frame printers

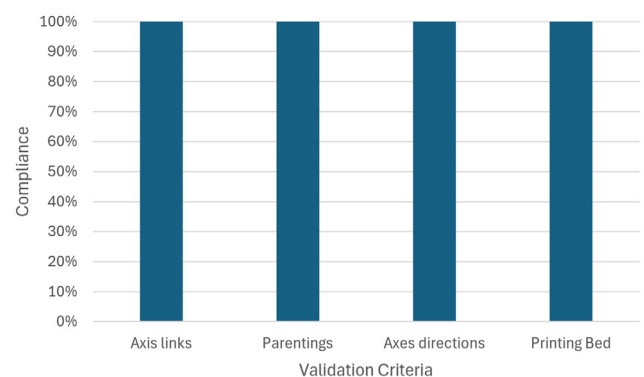


Fig. 10 Generation rate for Fixed Bed printers

Box-Frame, or Fixed Bed - can be automatically built in the virtual space based on its URDF model.

To take the user experience to the next level, an AI-based web application was developed that generates the URDF model of any Cartesian 3D printer; it produces usable robot descriptions in most cases, thereby making the digital printing lab substantially easier to customize. The website for URDF generation has been hosted on Vercel's cloud platform and is freely available [48].

The simulation can be extended into an immersive VR experience with the help of MaxWhere XR [49]. By establishing a connection to a real printer, a digital twin could

be created from this model in the virtual space. The simulation under development is available as a 3D printing workshop on the MaxWhere Portal [50].

Acknowledgement

The project supported by the Doctoral Excellence Fellowship Programme (DCEP) is funded by the National Research Development and Innovation Fund of the Ministry of Culture and Innovation and the Budapest University of Technology and Economics. This simulation is a product of the STEAM Upgrade Erasmus+ project.

References

- [1] Rodrigues Carneiro, L. R., de Souza Tavares, J. J. P. Z. "Design and implementation of 3D printer for Mechanical Engineering Courses", *International Journal for Innovation Education and Research*, 9(3), pp. 293–312, 2021.
<https://doi.org/10.31686/ijer.vol9.iss3.3001>
- [2] MaxWhere "MaxWhere (version 7.1.1)", [computer program] Available at: <https://portal.maxwhere.com> [Accessed: 02 May 2026]
- [3] Jahangirian, M., Eldabi, T., Naseer, A., Stergioulas, L. K., Young, T. "Simulation in manufacturing and business: A review", *European Journal of Operational Research*, 203(1), pp. 1–13, 2010.
<https://doi.org/10.1016/j.ejor.2009.06.004>
- [4] Hosseinpour, F., Hajihosseini, H. "Importance of simulation in manufacturing", *World Academy of Science, Engineering and Technology*, 51(3), pp. 292–295, 2009.
- [5] Mourtzis, D., Doukas, M., Bernidaki, D. "Simulation in Manufacturing: Review and Challenges", *Procedia CIRP*, 25, pp. 213–229, 2014.
<https://doi.org/10.1016/j.procir.2014.10.032>
- [6] Zhang, L., Zhou, L., Ren, L., Laili, Y. "Modeling and simulation in intelligent manufacturing", *Computers in Industry*, 112, 103123, 2019.
<https://doi.org/10.1016/j.compind.2019.08.004>
- [7] Garwood, T. L., Hughes, B. R., Oates, M. R., O'Connor, D., Hughes, R. "A review of energy simulation tools for the manufacturing sector", *Renewable and Sustainable Energy Reviews*, 81(1), pp. 895–911, 2018.
<https://doi.org/10.1016/j.rser.2017.08.063>
- [8] Afazov, S. M. "Modelling and simulation of manufacturing process chains", *CIRP Journal of Manufacturing Science and Technology*, 6(1), pp. 70–77, 2013.
<https://doi.org/10.1016/j.cirpj.2012.10.005>
- [9] Siemens Digital Industries Software "Plant Simulation software (2412)", [computer program] Available at: <https://plm.sw.siemens.com/en-US/tecnomatix/products/plant-simulation-software/> [Accessed: 30 Oct 2023]
- [10] Siemens Digital Industries Software "Process Simulate (2412)", [computer program] Available at: <https://webinars.sw.siemens.com/de-DE/process-simulate/> [Accessed: 30 Oct 2023]
- [11] CNC Simulator Blog "The world's first 3D Printer Simulator", [online] Available at: <https://cncsimulator.com/blog/?p=224> [Accessed: 30 Oct 2023]
- [12] Tyagi, B., Raj, A., Gupta, H., Malik, G., Bhardwaj, L., Sharma, G. S., Uppukoden, J., Sahai, A., Sharma, R. S. "Modelling Structural Behaviour of the Cartesian, Polar, and Delta Material Extrusion Printers by Finite Element Analysis", *Journal of The Institution of Engineers*, 105(3), pp. 617–634, 2024.
<https://doi.org/10.1007/s40032-024-01066-8>
- [13] Tarimer, I., Yilmaz, E. "Designing and Manufacturing of a Cartesian Type 3 Dimensional Printer", *SETSCI Conference Proceedings*, 8, pp. 80–86, 2019.
<https://doi.org/10.36287/setsoci.4.5.018>
- [14] Omodolor Omodia, I. D., Livingstone, B., Foissard, E. "3D printer design", [online] Bachelor thesis, Universitat Politècnica de Catalunya, 2024. Available at: <https://hdl.handle.net/2117/416574> [Accessed: 02 May 2026]
- [15] Horváth, B. "Virtual 3D Printer Simulation for Educational Purposes in MaxWhere", In: 2025 IEEE 25th International Symposium on Computational Intelligence and Informatics (CINTI), Budapest, Hungary, 2025, pp. 779–784. ISBN 979-8-3315-5291-6
<https://doi.org/10.1109/CINTI67731.2025.11311822>
- [16] Csapó, Á. B., Horváth, I., Galambos, P., Baranyi, P. "VR as a Medium of Communication: from Memory Palaces to Comprehensive Memory Management", In: 2018 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Budapest, Hungary, 2018, pp. 389–394. ISBN 978-1-5386-7095-8
<https://doi.org/10.1109/CogInfoCom.2018.8639896>
- [17] Ujbányi, T., Stankov, G., Nagy, B. "A transparent working environment in MaxWhere virtual space", In: 2019 10th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Naples, Italy, 2019, pp. 475–478. ISBN 978-1-7281-4794-9
<https://doi.org/10.1109/CogInfoCom47531.2019.9089967>
- [18] Berki, B. "Sense of Presence in MaxWhere Virtual Reality", In: 2019 10th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Naples, Italy, 2019, pp. 91–94. ISBN 978-1-7281-4794-9
<https://doi.org/10.1109/CogInfoCom47531.2019.9089976>
- [19] Berki, B. "Navigation Power of MaxWhere: a Unique Solution", In: 2020 11th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Mariehamn, Finland, 2020, pp. 509–514. ISBN 978-1-7281-8214-8
<https://doi.org/10.1109/CogInfoCom50765.2020.9237904>

- [20] Berki, B. "Better Memory Performance for Images in MaxWhere 3D VR Space than in Website", In: 2018 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Budapest, Hungary, 2018, pp. 281–284. ISBN 978-1-5386-7095-8 <https://doi.org/10.1109/CogInfoCom.2018.8639956>
- [21] Unity Technologies "Unity Engine (6.0 LTS)", [computer program] Available at: <https://unity.com/products/unity-engine> [Accessed: 26 Jul 2026]
- [22] Epic Games "Unreal Engine (5.5)", [computer program] Available at: <https://www.unrealengine.com/> [Accessed: 26 Jul 2026]
- [23] OpenJS Foundation "Node.js (18.16.0)", [computer program] Available at: <https://nodejs.org/en> [Accessed: 26 Jul 2026]
- [24] Craftbot.com "Craftbot Plus Pro Plug & Play 3D Printer", [online] Available at: <https://craftbot.com/products/craftbot-plus-pro> [Accessed: 02 May 2026]
- [25] MatterHackers "Creality3D Ender 3 3D Printer Kit", [online] Available at: <https://www.matterhackers.com/store/l/creality3d-ender-3-3d-printer-kit/sk/M34AK4ZG> [Accessed: 14 May 2026]
- [26] Creality "Ender 3 3D Printer", [online] Available at: <https://www.creality.com/products/ender-3-3d-printer> [Accessed: 02 May 2026]
- [27] Open Robotics "ROS - Robot Operating System (Jazzy Jalisco)", [computer program] Available at: <https://www.ros.org/> [Accessed: 26 Jul 2026]
- [28] Tola, D., Corke, P. "Understanding URDF: A Survey Based on User Experience", In: 2023 IEEE 19th International Conference on Automation Science and Engineering (CASE), Auckland, New Zealand, 2023, pp. 1–7. ISBN 979-8-3503-2070-1 <https://doi.org/10.1109/CASE56687.2023.10260660>
- [29] Encode "Starlette (0.46.2)", [computer program] Available at: <https://starlette.dev/> [Accessed: 26 Jul 2026]
- [30] Pydantic "Pydantic (2.10.0)", [computer program] Available at: <https://pydantic.dev/docs/validation/latest/get-started/> [Accessed: 26 Jul 2026]
- [31] Google "Go Programming Language (1.24)", [computer program] Available at: <https://go.dev/> [Accessed: 26 Jul 2026]
- [32] SmartBear Software "Swagger UI (5.20.0)", [computer program] Available at: <https://swagger.io/open-source/swagger-ui/> [Accessed: 26 Jul 2026]
- [33] Redocly "Redoc (2.3.0)", [computer program] Available at: <https://redocly.com/redoc> [Accessed: 26 Jul 2026]
- [34] Lathkar, M. "Introduction to FastAPI", In: High-Performance Web Apps with FastAPI, Apress, 2023, pp. 1–28. ISBN 978-1-4842-9177-1 https://doi.org/10.1007/978-1-4842-9178-8_1
- [35] Lathkar, M. "Getting Started with FastAPI", In: High-Performance Web Apps with FastAPI, Apress, 2023, pp. 29–64. ISBN 978-1-4842-9177-1 https://doi.org/10.1007/978-1-4842-9178-8_2
- [36] Nguyen, K. N. N. "Development & deployment of a web server as an executable with Node.js, Express.js and Vercel/pkg" [online], Bachelor's Thesis, Metropolia University of Applied Sciences, 2026. Available at: <http://www.theseus.fi/handle/10024/744037> [Accessed: 01 May 2026]
- [37] Menghnani, M. "Designing Resilient Full-Stack Web Applications with Multi-Regional Cloud Deployment Strategies", In: 2025 5th Asian Conference on Innovation in Technology (ASIANCON), PIMPRI, India, 2025, pp. 1–6. ISBN 979-8-3503-5697-7 <https://doi.org/10.1109/ASIANCON66527.2025.11280706>
- [38] OpenAI Developers "OpenAI API (GPT-4o)", [computer program] Available at: <https://developers.openai.com/api/reference/overview/> [Accessed: 30 Jul 2026]
- [39] Roumeliotis, K. I., Tselikas, N. D. "ChatGPT and Open-AI Models: A Preliminary Review", Future Internet, 15(6), 192, 2023. <https://doi.org/10.3390/fi15060192>
- [40] LangChain "LangChain (0.3)", [computer program] Available at: <https://www.langchain.com/> [Accessed: 02 May 2026]
- [41] Schick, T., Dwivedi-Yu, J., Dessi, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., Scialom, T. "Toolformer: Language Models Can Teach Themselves to Use Tools", [preprint] arXiv, 09 Feb 2023. <https://doi.org/10.48550/arXiv.2302.04761>
- [42] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., ..., Kiela, D. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks", [preprint] arXiv, 12 Apr 2021. <https://doi.org/10.48550/arXiv.2005.11401>
- [43] xml2js "xml2js (0.6.2)", [computer program] Available at: <https://www.npmjs.com/package/xml2js> [Accessed: 31 Oct 2023]
- [44] Treeflex "Treeflex (2.0.1)", [computer program] Available at: <https://www.cssscript.com/semantic-hierarchy-tree-treeflex/> [Accessed: 05 Nov 2023]
- [45] van Driel, A., Govender, R. A. "Development of a Prototype Embedded 3D Printer with Dual Ink-Gel Extrusion Capability", South African Journal of Industrial Engineering, 36(4), pp. 1–12, 2025. <https://doi.org/10.7166/36-4-3172>
- [46] Raj, A., Tyagi, B., Kapoor, A., Parashar, A., Satsangi, A., Sahai, A., Sharma, R. S. "Optimization of Cartesian and polar 3D printer structures using finite element analysis: a comparative study on material selection and design enhancement", Proceedings of the Indian National Science Academy, 91(3), pp. 804–820, 2025. <https://doi.org/10.1007/s43538-024-00363-1>
- [47] El Messaoudi, D., Elyessefi, A., Buj-Corral, I., ElMesbahi, J., El Mrabet, M., Boulaala, M., Elmesbahi, A. "Design of a full-colors FDM 3D printer: key technical possibilities", International Journal on Interactive Design and Manufacturing, 18(1), pp. 569–589, 2023. <https://doi.org/10.1007/s12008-023-01639-4>
- [48] URDF Generator "URDF Generator (1.0)", [computer program] Available at: <https://urdf-generator-vercel.vercel.app/> [Accessed: 02 May 2026]
- [49] Horvath, I. "Innovative engineering education in the cooperative VR environment", In: 2016 7th IEEE International Conference on Cognitive Infocommunications, Wroclaw, Poland, 2016, pp. 359–364. ISBN 978-1-5090-2646-3 <https://doi.org/10.1109/CogInfoCom.2016.7804576>
- [50] MaxWhere "MaxWhere (2.0)", [computer program] Available at: <https://portal.maxwhere.com/mwweb/url.html?space=3d-printing-bett> [Accessed: 02 May 2026]