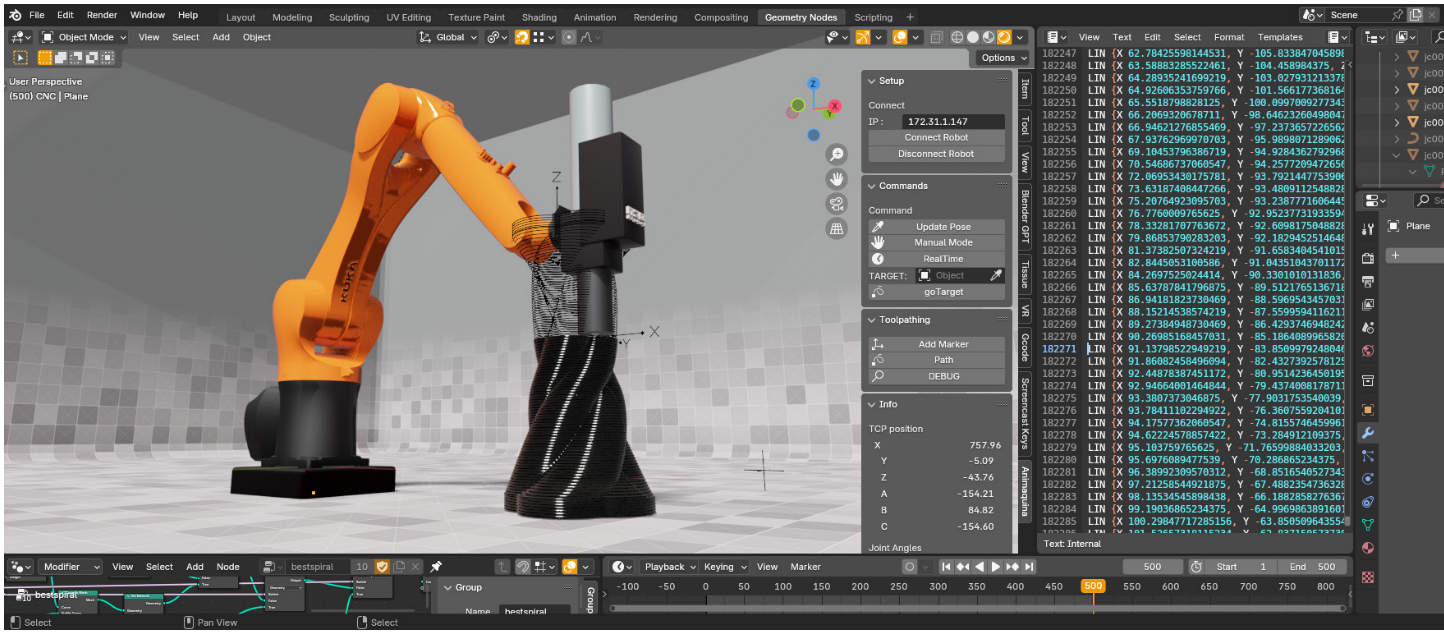
Animaquina

Luis Pacheco
Florida International University

A Framework for Democratizing Architectural Robotics



1

1 The Animaquina Interface.

ABSTRACT

The complexities in robot integration, programming, and control pose significant challenges for designers in the field of architectural robotics. This paper identifies key opportunities to democratize the field by enhancing usability and simplifying integration processes. It introduces *Animaquina*, a toolkit designed to address these challenges through two main innovations: an intuitive interactive toolpathing approach and a modular, wireless hardware framework for simplified integration of sensors, actuators, and end-effectors. The paper begins by analyzing current architectural robotic tools to identify areas of opportunity. It then describes a prototype that implements the proposed strategies. This prototype is used for testing an iterative workflow, serving as the means to explore and validate the proposed functionality. The paper showcases an example workflow in robotic 3D printing and concludes by outlining the future roadmap for Animaquina's development as an open-source platform to democratize architectural robotics.

INTRODUCTION

In his influential talk "Inventing on Principle," Bret Victor argues that creators need an immediate connection to what they create (Victor, 2012). He demonstrates how providing direct, interactive control and continuous visual feedback transforms creative processes. In the field of architectural robotics, the disconnect between design intent and robotic execution is particularly acute. Current robotic systems often rely on offline programming workflows, where designers specify robotic toolpaths in CAD/CAM software and then upload them to the robot for execution (Braumann and Brell-Cokcan 2011). This decoupled approach makes it difficult for designers to understand how design decisions impact the robot's behavior, leading to a trial-and-error process to achieve desired results.

Inspired by Victor's principle, this paper explores how providing a more immediate, interactive connection between design intent and robotic execution can streamline access to robot arm toolpathing, making it more user-friendly and approachable. The paper also proposes making architectural robotics more accessible by developing a plug-and-play library of hardware components, heavily inspired by the work described in the Modular-Things project (Read et al. 2023) and the Urumbu minimal machine-building framework (Gershenfeld, Bolsée, and Hart 2022). These projects demonstrate how open-source, modular toolkits can enable novice users to develop complex physical computing systems, potentially impacting the adoption of these technologies in non-expert settings like academic labs, smaller companies, and personal fabrication.

Animaquina, is an open-source toolkit that combines intuitive software for interactive toolpathing with a modular, plug-and-play hardware for simplified integration of sensors, actuators, and end-effectors on industrial robots. This paper provides an analysis of current trends and tools in architectural robotics to identify opportunities for making more accessible, user-friendly, and democratized tools. It then presents the core components of the Animaquina framework, including the design principles, system architecture, user interface, and hybrid workflows. Next, it highlights a workflow produced with the framework, and discusses the implications and future potential of open-source hardware and software platforms for increasing access and usability in architectural robotics.

HISTORICAL BACKGROUND

In the early 2000s, architects began integrating and researching Computer-Aided Manufacturing (CAM) tools into their practice, enhancing fabrication capabilities and precision for design realization. Architectural technologists of this era began exploring the emergence of digital fabrication in architecture, bringing the application of manufacturing processes and techniques into architecture (Kolarevic 2001). Many also documented the impact of CAM on design and architecture (Schodek et al. 2004, Kolarevic 2004). These works highlight how digital fabrication tools revolutionized design materialization, focusing on the new morphogenetic possibilities and the use of industrial CAM tools to produce more complex geometries.

Between 2009 to 2012 the limitations of traditional CAM tools became evident, prompting architects to look for alternatives like robotic arms to facilitate mass customization. In this period, early parametric design tools for robot milling marked a pivotal transition towards robotic control in architecture. For example, the Rhinoceros 3D / Grasshopper plugin KUKA|PRC explored robotics in architecture and construction, leading to a parametric approach for generating KUKA robot language (KRL) code through Visual Programming Language (VPL) (Brell-Cokcan and Braumann 2010; Braumann and Brell-Cokcan 2011). This featured a Rhino Grasshopper component, demonstrating its application in CAM workflows. Further developments integrated user interfaces with microcontroller boards for industrial robot interaction. Further research provided insights into accessible robot tooling for robotic fabrication workflows (Braumann and Brell-Cokcan 2012).

This era saw a surge in tools using serialized instructions for one-way robot control, with some experimenting with direct or real-time control methodologies for the integration of fabrication constraints into architectural design (Pigram and McGee 2011). Other software initiatives like HAL, incorporated components like an OpenSoundControl (OSC) message translator, kinematic solvers, and a bridge connecting virtual and physical robots, contributing to real-time control and multidimensional device communication (Schwartz 2013). These works explored robotic parametric robotic arm programming and the possibility of simulating the robot movements in real-time, bringing the use of robotic arms as feasible tools for architectural design and manufacturing.

The period from 2012 to 2015, marks a shift from the traditional CAM approach towards integrating craftsmanship and creativity into robotic systems. Firefly, a Grasshopper plugin bridging digital and physical spaces, simplified architectural prototyping by creating instant feedback loops between hardware and the Rhino/Grasshopper environment, enabling real-time control and bidirectional communication between sensors and actuators (Payne and Johnson 2013). Important research on simulation and fabrication highlighted the benefits of real-time simulation in bridging design and robotic fabrication, emphasizing iterative problem-solving. This research demonstrated how real-time simulation feedback could enhance design refinement and robot programming optimization (Braumann and Brell-Cokcan 2014). Several studies collectively demonstrated the evolution of architectural robotics beyond mere CAM tools, towards more creative and integrated applications that blend digital design with physical craftsmanship.

Between 2015 to 2020 there is a transformation towards collaborative robotics, enabling real-time feedback loops in the design process. Research in these years focused on developing intuitive methods for interacting with and controlling robotic arms in design and fabrication. Collaborative and interactive tools emerged, like using skin as a canvas for digital design and fabrication. This allowed it to generate responsive toolpaths that adapt to material deformations and incorporate human input (Gannon, Grossman, and Fitzmaurice 2015). Other studies highlighted safety through sensor and position feedback, enabling enhanced haptic interactions and creating a more interactive environment. (Braumann, Stumm, and Brell-Cokcan 2016). Similarly, studies offering insights into KUKA's MXautomation remote control interface introduced dataflow modes for real-time control, focusing on adaptive buffer size modification for responsive robotic control (Munz, Braumann, and Brell-Cokcan 2016).

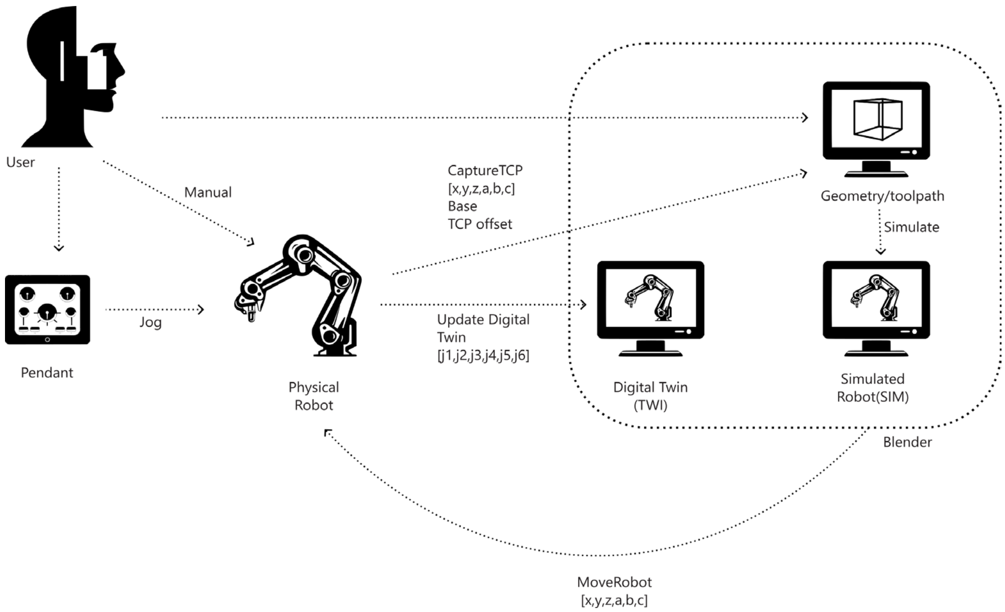
Animation-based tools for robot programming simplified robot movement interface through a keyframing method, departing from traditional code-based approaches (Poustinchi 2019). Other open-source tools emerged, further democratizing access to robotic control in creative fields. MIMIC for Maya, developed by Autodesk Research, offered a plugin for intuitive robot programming within the Maya 3D animation software. It allowed users to leverage familiar animation tools for robot path planning and simulation, bridging the gap between creative intent and robotic execution (Autodesk Robotics Lab 2019). The Robots Add-on by Vicente Soler (Visose) for Grasshopper provided a multi-robot open-source alternative within the Rhino and Grasshopper environment. This tool enabled users to program and simulate various robot brands within a unified feature parity (Soler 2020).

These developments collectively advanced the integration of robotic technology in architecture and design, addressing various challenges and opening new possibilities for human-robot collaboration. Importantly, this era saw the emergence of new paradigms in robotic control, moving beyond traditional programming methods towards more intuitive, visual, and interactive approaches. The introduction of animation-based tools, real-time feedback systems, and open-source alternatives marked a significant shift in how designers and architects could interact with and control robotic systems, paving the way for more creative and accessible applications in the field.

In the past few years the field of architectural robotics has had important advancements, with notable developments in the realms of real-time interaction, open-source methodologies, and the advent of incorporating Artificial Intelligence (AI) in architectural robotics workflows. Studies exploring the potential of real-time interactions with industrial robots in creative domains highlighted the challenges inherent in existing methods, which predominantly depended on computational geometry. These studies have advocated for a flow-based approach, emphasizing the importance of integrating cyclical data flows to surmount these challenges. Through analyzing specific projects that benefited from real-time control, the studies demonstrate the incorporation of diverse sensors capable of instantaneously altering robot movements (Braumann and Singline 2021).

Robot Ex Machina, presents a framework for real-time robot programming and control within the fields of architecture, design, and art. This framework, operates based on a concurrency control paradigm, employs an action-state model (or state machine) for synchronized control. This approach facilitates the efficient management of "actions" triggered by specific conditions or events, facilitating real-time adjustments during the fabrication process, and creating an environment that promotes creativity and experimentation (García del Castillo 2019). The exploration of different environments more akin to defining flows in robotic fabrication through visual programming research has showcased an approach to interactive robotic fabrication, enhancing the flexibility of design and production processes in architectural robotics using Unity Visual Scripting (UVS) (Braumann, Gollob, and Singline 2022).

There is also research on the exploration of open-source approaches to robotic fabrication in architecture. Outlining both the benefits and challenges associated with employing open-source methodologies in robotic prototyping for architectural design and construction (Rossi et al. 2022). Finally, an important development is in the framework COMPAS, which integrates robotics as part of a larger ecosystem for collaboration within architecture, engineering, and construction, offering libraries for robotic control (Van Mele et al. 2017-2021). This framework exemplifies the



2 Diagram illustrating the proposed architecture for the robot driver to enable an interactive toolpath development workflow.

trend towards more comprehensive, open-source solutions that facilitate interdisciplinary collaboration in architectural robotics.

These recent developments represent a shift towards more mature interactive interfaces, open and accessible tools, and the potential for AI-driven approaches in architectural robotics. The integration of robotics into larger collaborative frameworks highlights the field's move towards more holistic, interdisciplinary approaches to architectural robotics and fabrication.

THE ANIMAQUINA FRAMEWORK

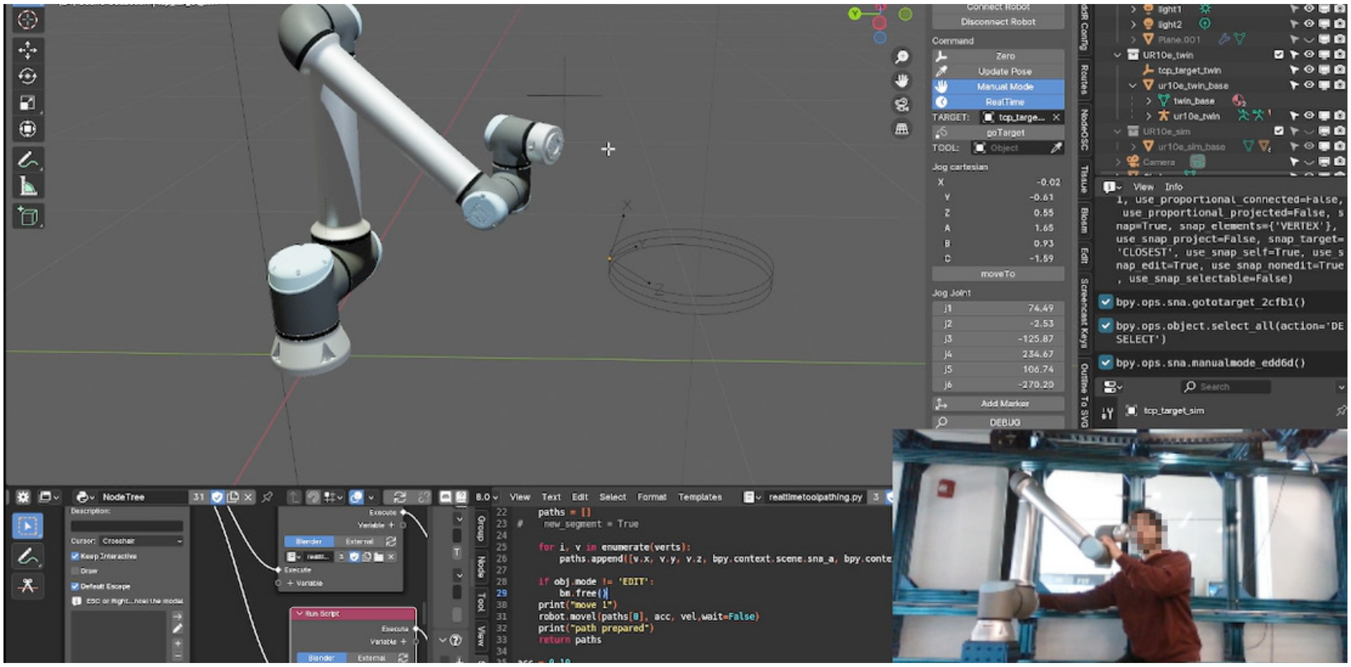
This section introduces Animaquina, a software tool under development that is being built to address three areas of improvement to democratize architectural robotics: 1) Facilitating interactive toolpathing, a paradigm shift from conventional, static programming methods to a more dynamic, user-friendly interface that takes advantage of real-time feedback while designing toolpaths and trajectories. 2) Simplifying the integration of hardware within robotic systems, especially the versatility and adaptability of end-effectors to embrace interactive workflows; 3) Advocating for an open development approach in architectural robotics. The current landscape is dominated by proprietary systems that can stifle innovation and accessibility. By embracing open-source principles, like providing accessible templates, code, and documentation, the field can foster a collaborative environment that encourages innovation, knowledge sharing, and community-driven development.

Animaquina's development adopts an experimental and design-based research and development methodology, an approach well-suited for developing and refining products. The process is iterative and involves the creation of prototypes, testing in real-world scenarios, and subsequent refinement based on feedback and analysis. This methodology enables a practical evaluation of the Animaquina toolkit while ensuring continuous improvement through iterative design cycles. This methodology is described in detail by Henrik Vestad and Martin Steinert in the research paper entitled: "Creating Your Own Tools: Prototyping Environments for Prototype Testing" (Vestad and Steinert 2019). The following describes the various stages of the project design.

Interactive Toolpathing: A digital twin approach

The concept of interactive toolpathing holds significant promise for the realm of architectural robotics. Traditional robotic programming methods, often rigid and serialized, limit the creative and experimental capabilities of designers and architects. By adopting the principles of interactive toolpathing, as seen in the fields of digital fabrication and 3D printing, a more intuitive and adaptable approach to robotic programming can be achieved. This approach resonates with Bret Victor's vision of inventing on principle, emphasizing real-time interaction and user-friendly interfaces (Victor 2012).

Central to solving this paradigm shift is the use of digital twins, which are virtual representations of physical robots and their



3 Comparing the digital twin and the physical robot, the UR10e digital twin is instantly updated when manually moving the physical robot.

3

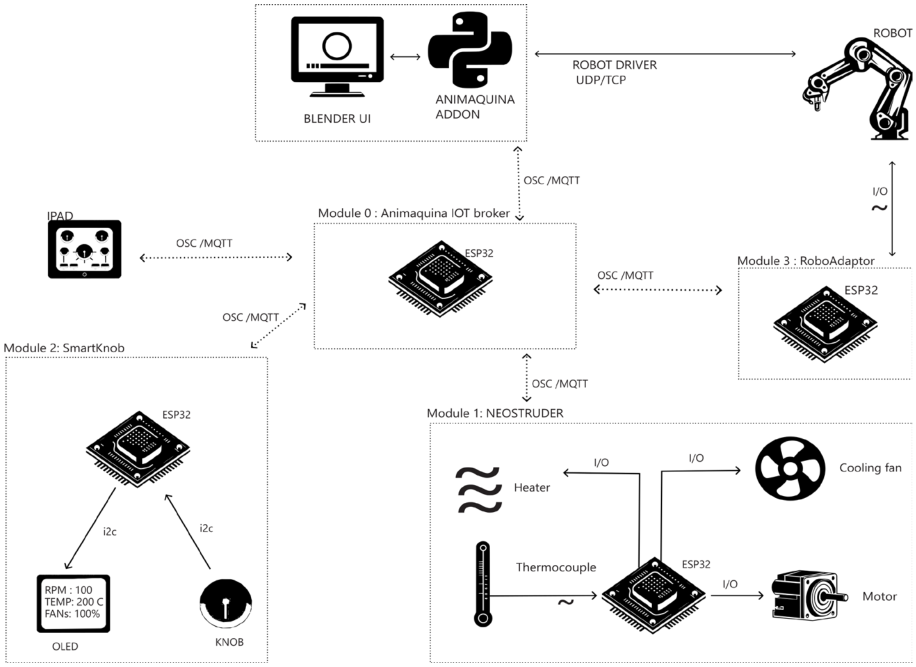
operational contexts. These digital counterparts allow for the real-time visualization of the physical robot, simulation of possible trajectories, and modification of toolpaths based on external parameters, offering immediate visual feedback of the current state of the physical robot. The application of interactive toolpathing in architectural robotics mirrors the successful use in 3D printing, as outlined in studies like "Interactive Digital Fabrication Machine Control Directly within a CAD Environment" (Fossdal, Heldal, and Peek 2021) and "p5.fab: Direct Control of Digital Fabrication Machines from a Creative Coding Environment" (Subbaraman and Peek 2022). By adopting similar strategies, architects and designers can engage in a more interactive and responsive process of robotic programming, akin to the flexibility experienced in digital fabrication.

The initial development of the *Animaquina* prototype has enabled testing of various workflows with robotic arms. This version allows for bidirectional data streaming between the physical robotic arm and Blender's Animaquina interface. Specifically, it streams the Tool Center Point (TCP) location, TCP offset, Base offset, and Joint Angles, enabling the virtual robot to reflect the physical position of the robotic arm in near real-time. Users can also move a target marker in the 3D viewport to simulate trajectories from the current position and move the physical robot to the simulated target position when desired. This bidirectional workflow allows users to digitize and record positions with the robot, which can then be used as references to map physical features in the workspace.

The *Animaquina* Control Panel (Fig. 1) has several key functionalities. Users can connect to a robot via UDP network by specifying the robot's IP and model. Once connected, the panel displays Commands, Toolpathing, and Info subpanels. The Commands subpanel allows users to update the digital twin's pose with the physical robot's current joint angles, enable manual robot movement (if supported), and activate real-time updates. Users can also select a 3D object as a target and move the robot to that position. The Toolpathing panel enables users to add markers at the current TCP coordinates, generate and animate paths from selected curves or vertices, and export robot code. The play button sends the current animation to the robot in real-time, allowing for dynamic toolpath adjustments. A debug feature automatically positions the robot and tool in the 3D scene by retrieving TCP and Base offsets. The Info subpanel displays the robot's joint angles and TCP cartesian coordinates numerically.

Simplified Integration Enhancing End-Effector Compatibility in Architectural Robotics

Current robotic systems, particularly in architectural and creative applications, often face significant challenges in integrating various end-effectors with robotic arms. These challenges stem from the dependency on specific robot programming languages, compatibility issues with different voltages, and the necessity of using custom Programmable



4

4 Diagram illustrating the integration of a commercially available Massive Dimension Pellet Extruder (MDPE), a hardware interface for manual adjustment, an adapter to read and write to the robot I/O interface, and an iPad app showcasing the system's flexibility.

Logic Controllers (PLCs). Such complexities not only increase the technical barrier for users but also limit the versatility and adaptability of robotic systems in creative applications.

To address these challenges, this study proposes a framework that decouples the control of robot motion from end-effector operation. By adopting a modular approach, this framework allows for the independent operation and programming of end-effectors, irrespective of the robotic arm's make or model. This separation simplifies the integration process significantly, as it eliminates the need for end-effectors dependent on a specific robot's programming language or electrical specifications. This system is designed to wirelessly connect to the Blender Addon, where data is streamed to inform the Digital Twin interface and to update the end-effector parameters on the fly.

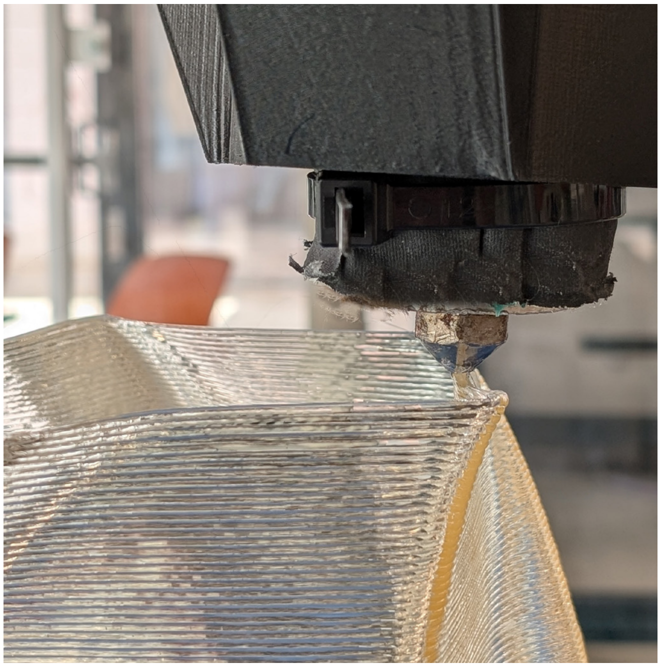
The current prototype comprises three independent modules. The first module streams digital inputs and outputs from the robot to the extruder during program execution (Fig. 6). The second is a hardware interface featuring a screen and knobs for displaying and adjusting the extruder's RPM, temperature, and fan speed (Fig. 7). The third module controls the hardware, generating signals to manage the motor, relay, and fan while streaming back the current temperature value (Fig. 5). This modular setup enables flexible integration and real-time control of the extruder in robotic fabrication processes. To demonstrate the system's adaptability, an iPad app was developed for remote parameter adjustments.

Open-Source Development and Dissemination Approach

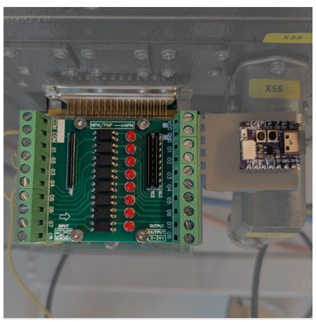
The proposed open-source approach fosters inclusivity and collaboration by providing a shared platform for developing, refining, and disseminating ideas, tools, and techniques across a broad community of users and developers.

Blender, an open-source 3D creation suite, emerges as an exemplary platform in this context. Its open-source nature ensures free accessibility, fostering a diverse community of users and developers. This contrasts with the historical reliance on proprietary software like Rhino and Grasshopper in architectural robotics, which limits accessibility and constrains collaborative development. Blender's fully functional Python interpreter allows for easy integration of various Python libraries, including those for physical computing, computer vision and artificial intelligence, making it an ideal environment for developing and testing innovative robotic applications. Establishing an open protocol for communication and code development is crucial for this approach. Standardization allows users to easily adapt and integrate different end-effectors, external axes, and other components into their robotic systems, creating a common framework that ensures compatibility and ease of use across various hardware and software elements.

Creating a community library is an important part of this open development model. In this repository, users can share custom toolpath algorithms, end-effector designs, or entire robotic control systems. This communal resource accelerates learning, fosters innovation, and allows users to build upon each other's work, reducing development time and encouraging experimentation.



5



6

7

5 Massive Dimension Pellet Extruder in operation. An Esp32 generates signals for motor movement, heater relay, and fan control based on the incoming messages.

6 Module for streaming Digital I/O into data from the robot to the extruder during offline program execution.

7 Human hardware input interface displaying motor RPM, fan speed, and temperature, with manual adjustment knobs for each parameter connection.

PROOF OF CONCEPT: MINIMAL VIABLE PRODUCT (MVP)

As a proof of concept, an experiment was conducted to demonstrate the system's capabilities. The experiment showed that after digitizing a printing surface, it is possible to parametrically develop a toolpath that conforms to the surface's normal vector. This surface can be updated in the 3D environment based on external data, allowing real-time adjustment of the trajectory parameters as the robot moves. Data can be streamed to the robot to update its next position. Simultaneously, Animaquina streams data to the end-effector, controlling extrusion flow, fan speed, and temperature. This bidirectional data flow enables adaptive and responsive robotic fabrication, showcasing the system's potential for complex, dynamic printing tasks

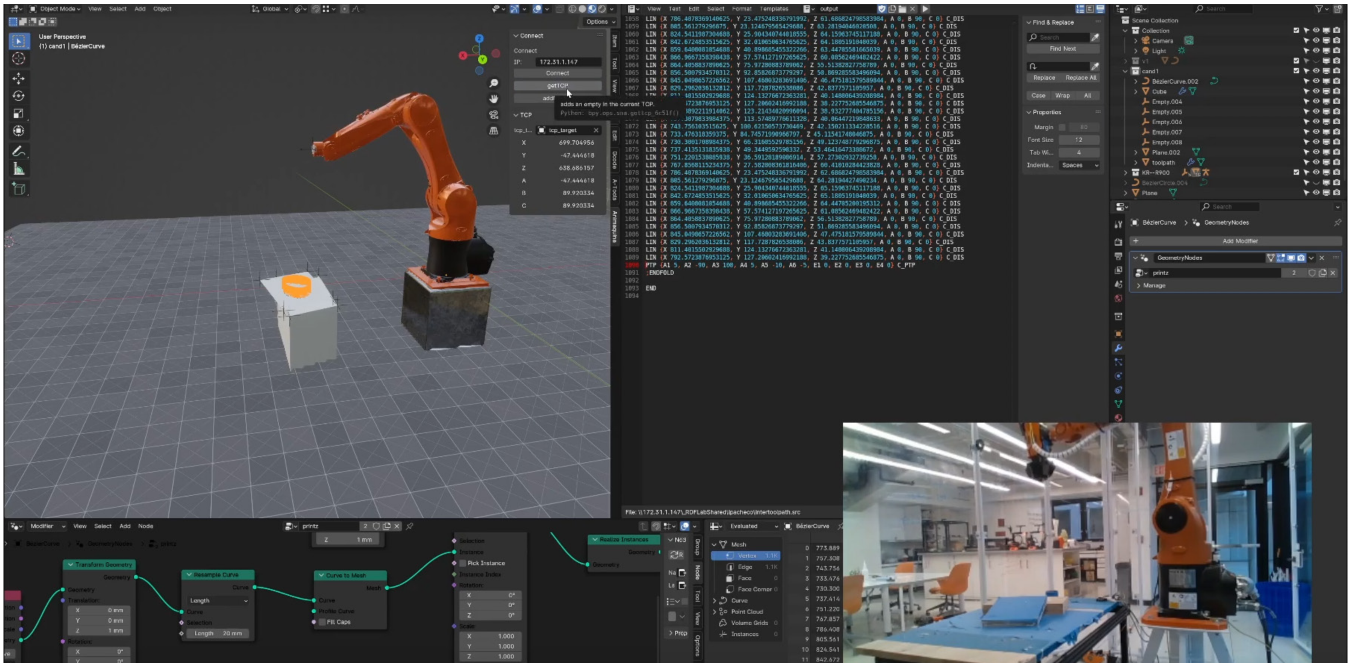
A second experiment was conducted during the fabrication of 3D printed columns for an artistic installation (Fig. 9). In this case, the extruder speed (RPM) and fan speeds were dynamically adjusted to account for under-extrusion on overhang areas. This real-time adjustment capability proved to be a valuable tool for reducing waste and recovering from potential print failures. The ability to modify printing parameters on-the-fly demonstrates the system's adaptability to changing conditions during the fabrication process. This approach aligns with recent trends in adaptive fabrication techniques, The successful implementation of dynamic parameter adjustment not only improved print quality but also showcased the potential for more efficient and resilient 3D printing processes in architectural-scale applications.

CONCLUSION: CHARTING THE PATH FORWARD FOR THE ANIMAQUINA TOOLKIT

This initial testing of the MVP for Animaquina has provided some key improvements and areas of research that need to be addressed, I propose the further development of the Animaquina toolkit as an Integrated Development Environment designed to lower the barriers to entry in the field of architectural robotics and to foster an environment of creativity. Comprising three key components:

A Blender add-on, the main element of Animaquina, offers an intuitive and powerful Digital Twin toolpathing interface. Leveraging Blender's robust animation, keyframing, and parametric capabilities, this add-on simplifies the process of programming and simulating robotic movements while allowing for real-time feedback and control for the physical robot, end-effectors, and sensors. This integration not only makes toolpathing more accessible but also opens new possibilities for creative exploration, allowing users to visualize and adjust robotic actions "on the fly."

A modular hardware toolkit specifically designed to ease the integration of sensors and actuators for the development of end-effectors, sensors, and external axes. This addresses one of the fundamental challenges in architectural robotics – the complexity of integrating diverse hardware components. By providing a standardized yet flexible platform, it enables users to experiment with and create a wide range of end-effectors and sensors, enhancing the versatility of robotic systems as a plug-and-play ecosystem.



8

8 Non-parallel printing surface mapped using Tool Center Point (TCP) data directly in Blender.

Establish a repository that functions as a comprehensive library of tools and resources. This repository will not only serve as a platform for sharing and accessing tools but also as a collaborative space for continuous development and innovation. By adopting a GPL license, the toolkit ensures that all contributions remain open and accessible, encouraging a culture of sharing and collective growth among users. The development of use case workflows that benefit from the proposed features can be initially focused on hybrid digital fabrication processes to focus on a growing area in architectural robotics research that can benefit from this material and end-effector informed workflow.

In conclusion, Animaquina's goal is to bring together innovations that have been previously explored independently in the realm of architectural robotics. It aims to make the field more accessible, versatile, and collaborative by providing tools that make it easier to program, integrate, and promote open development in architectural robotics.

9

9 Comparison of a 3D printed "Dancing Column" and its twin. Project by Joseph Choma and Luis Pacheco.

REFERENCES

1. Autodesk Robotics Lab. "Mimic." GitHub, 2023. https://github.com/AutodeskRoboticsLab/Mimic.
2. Braumann, Johannes, and Sigrid Brell-Cokcan. "Parametric Robot Control: Integrated CAD/CAM for Architectural Design." In Integration through Computation: Proceedings of the 31st Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA), 242-251. Banff, Alberta: ACADIA, 2011.
3. Braumann, Johannes, and Sigrid Brell-Cokcan. "Visual Robot Programming: Linking Design, Simulation, and Fabrication." In Proceedings of the 34th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA). Los Angeles: ACADIA, 2014.
4. Braumann, Johannes, E. Gollob, and Karl Singline. "Visual Programming for Interactive Robotic Fabrication Processes—Process Flow Definition in Robotic Fabrication." In Proceedings of the 40th Conference on Education and Research in Computer Aided Architectural Design in Europe (eCAADe). Ghent, Belgium: eCAADe, 2022. https://doi.org/10.52842/conf.ecaade.2022.2.427.
5. Braumann, Johannes, and Karl Singline. "Towards Real-Time Interaction with Industrial Robots in the Creative Industries." In IEEE International Conference on Robotics and Automation. Xi'an, China: IEEE, 2021. https://doi.org/10.1109/icra48506.2021.9561024.
6. Braumann, Johannes, Sven Stumm, and Sigrid Brell-Cokcan. "Towards New Robotic Design Tools: Using Collaborative Robots within the Creative Industry." In Proceedings of the 36th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA). Ann Arbor: ACADIA, 2016. https://doi.org/10.52842/conf.acadia.2016.164.
7. Brell-Cokcan, Sigrid, and Johannes Braumann. "A New Parametric Design Tool for Robot Milling." In Proceedings of the 30th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA). New York: ACADIA, 2010.
8. Fossdal, Fredrik Hilmar, Rogardt Heldal, and Nadya Peek. "Interactive Digital Fabrication Machine Control Directly Within a CAD Environment." In Symposium on Computational Fabrication. Virtual Event: ACM, 2021. https://doi.org/10.1145/3485114.3485120.
9. Gannon, Madeline, Tovi Grossman, and George Fitzmaurice. "Tactum: A Skin-Centric Approach to Digital Design and Fabrication." In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, 1779–1788. New York: ACM, 2015. https://doi.org/10.1145/2702123.2702581.
10. García del Castillo, Jose Luis. "Robot Ex Machina." In Proceedings of the 39th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA). Austin: ACADIA, 2019. https://doi.org/10.52842/conf.acadia.2019.040.
11. Gershenfeld, Neil, Quentin Bolsée, and Ryan Hart. "Urumbu: Minimal Machine Building." In Proceedings of the Symposium on Computational Fabrication. Seattle, WA, USA: ACM, 2022.
12. Kolarevic, Branko. Architecture in the Digital Age: Design and Manufacturing. New York: Spon Press, 2004.
13. Kolarevic, Branko. "Digital Fabrication: Manufacturing Architecture in the Information Age." In Proceedings of the 21st Annual Conference of the Association for Computer-Aided Design in Architecture (ACADIA). Buffalo: ACADIA, 2001. https://doi.org/10.52842/conf.acadia.2001.010.
14. Munz, Helmut, Johannes Braumann, and Sigrid Brell-Cokcan. "Direct Robot Control with mxAutomation: A New Approach to Simple Software Integration of Robots in Production Machinery, Automation Systems and New Parametric Environments." In Advances in Robot Design and Intelligent Control, 440–447. Cham: Springer International Publishing, 2016. https://doi.org/10.1007/978-3-319-26378-6_35.
15. Payne, Andrew O., and Jason K. Johnson. "Firefly: Interactive Prototypes for Architectural Design." Architectural Design 83, no. 2 (2013): 144–147. https://doi.org/10.1002/ad.1573.
16. Pigram, Dave, and Wes McGee. "Formation Embedded Design: A Methodology for the Integration of Fabrication Constraints into Architectural Design." In Proceedings of the 31st Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA). Calgary: ACADIA, 2011. https://doi.org/10.52842/conf.acadia.2011.122.
17. Poustinchi, Ebrahim. "Oriole Beta—A Parametric Solution for Robotic Motion Design Using Animation." In Proceedings of the 37th International Conference on Education and Research in Computer Aided Architectural Design in Europe (eCAADe) & 23rd Conference of the Iberoamerican Society Digital Graphics, Volume 2. Porto: eCAADe, 2019. https://doi.org/10.52842/conf.ecaade.2019.2.227.
18. Read, John R., Liam McElroy, Quentin Bolsée, Benjamin Smith, and Neil Gershenfeld. "Modular-Things: Plug-and-Play with Virtualized Hardware." In Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems. Hamburg, Germany: ACM, 2023.
19. Rossi, Andrea, Arjen Deetman, Alexander Stefas, Andreas Göbert, Carl Eppinger, Julian Ochs, Oliver Tessmann, and Philipp Eversmann. "An Open Approach to Robotic Prototyping for Architectural Design and Construction." In Advances in Architectural Geometry 2022, 96–107. Cham: Springer International Publishing, 2022. https://doi.org/10.1007/978-3-031-13249-0_9.
20. Schodek, Daniel L., Michael Bechthold, Kimo Griggs, Kenneth M. Kao, and Marco Steinberg. Digital Design and Manufacturing: CAD/CAM Applications in Architecture and Design. Hoboken, NJ: John Wiley & Sons, 2005.
21. Schwartz, Thibault. "HAL." In Rob | Arch 2012, edited by Sigrid Brell-Çokcan and Johannes Braumann, 92–101. Vienna: Springer, 2013.
22. Soler, Vicente. "Robots: Create and Simulate ABB, KUKA, UR, Staubli, Doosan and Franka Emika Robot Programs." GitHub,

2023. https://github.com/visose/Robots.

23. Subbaraman, Blair, and Nadya Peek. "p5.fab: Direct Control of Digital Fabrication Machines from a Creative Coding Environment." In Proceedings of the 2022 ACM Designing Interactive Systems Conference. Virtual Event: ACM, 2022. https://doi.org/10.1145/3532106.3533496.
24. Van Mele, Tom, and many others. "COMPAS: A Framework for Computational Research in Architecture and Structures." 2017-2021. https://doi.org/10.5281/zenodo.2594510.
25. Vestad, Henrik, and Martin Steinert. "Creating Your Own Tools: Prototyping Environments for Prototype Testing." Procedia CIRP 84 (2019): 707–712. https://doi.org/10.1016/j.procir.2019.04.225.

All drawings and images by the author.

Luis Arturo Pacheco is a Robotics and Digital Fabrication Research Specialist at Florida International University's School of Architecture and RDF lab. His research focuses on innovative workflows in additive manufacturing for construction and robot control interfaces. Pacheco holds a Master's in Robotics and Advanced Construction from IAAC, Barcelona, and a Bachelor's in Architecture from Universidad de La Salle Bajio, Mexico. As co-founder and former CEO of MakerMex, Mexico's first 3D printer manufacturing company, he led development of 3D printers, laser cutters, and customized robotic systems. Pacheco also co-founded MakerLab Leon and MakerScad. In 2016, he was recognized as one of Mexico's MIT Innovators under 35.