

Design and Implementation of an Intelligent Control System for 3D Printers Based on Unity3D

Yichen Zhou, Yang Xiang, Can Jian, Jiawei Jiang, Mingchang Zhang, Kunpeng Li, Tianhao Zhao, Yanze Jin

Shanghai University of Engineering Science, Shanghai 201620, China

Abstract: With the rapid development of additive manufacturing technology, the demand for high-precision control and real-time monitoring of 3D printing systems has become increasingly urgent. To address issues such as step loss errors, insufficient real-time performance, and weak remote monitoring capabilities in traditional 3D printers, this paper proposes an intelligent 3D printing system that integrates digital twin technology with incremental PID closed-loop control. By constructing a virtual-physical synchronized digital twin model and combining it with Unity3D for high-precision virtual simulation, the system achieves real-time mapping of the motion trajectory and state parameters of the physical printer. A Raspberry Pi is employed as the core controller, integrating the MKS SERVO42C closed-loop driver board and an incremental PID algorithm to resolve step loss issues during high-speed operation of stepper motors, thereby improving printing accuracy. A network communication architecture based on the HTTP protocol is designed to enable real-time data interaction among the Raspberry Pi, the main control board, and the virtual twin. Additionally, a MySQL database is utilized for storing and analyzing printing process data. The system significantly optimizes printing efficiency and quality.

Keywords: Unity3D; Digital Twin; PID Closed-loop Control; 3D Printing System; Real-time Monitoring.

1. Introduction

With continuous advancements in additive manufacturing, 3D printing has been increasingly adopted across multiple industries. Compared with conventional machining methods, this technology offers significant advantages, including mold-free fabrication, design flexibility, and the capability to manufacture complex structures. In sectors such as manufacturing and healthcare, 3D printing can substantially reduce production cycles, lower costs, and enable product innovation, thereby broadening its application in design and development.[1][2]

Although mature control system solutions for 3D printing have been developed abroad, domestic technological progress remains relatively lagging, with weaker adaptability and a lack of reliable control platforms. Consequently, enterprises often encounter challenges such as low efficiency and unstable print quality in practical operations. Enhancing the performance and reliability of domestic 3D printing control systems has thus become an urgent task.

To this end, this paper proposes a Raspberry Pi-based digital factory control system, which leverages digital twin technology to enable real-time monitoring and simulation of the 3D printing process. The system aims to overcome spatial limitations, while simultaneously improving both printing efficiency and output quality.[4][7]

2. Overall Design of Digital Twin 3DPrinter

2.1. System Architecture Design

This 3D printer is developed based on the open-source Klipper system, using the MKS Monster8 V2.0 as the main control board, in combination with the MKS SERVO42C closed-loop stepper motor driver board to achieve closed-loop control and prevent print failures caused by step loss. A

Raspberry Pi 4B running the Fluidd system integrates both the operating system and the web-based Klipper interface, enabling remote task management and real-time monitoring of printer status.[9]

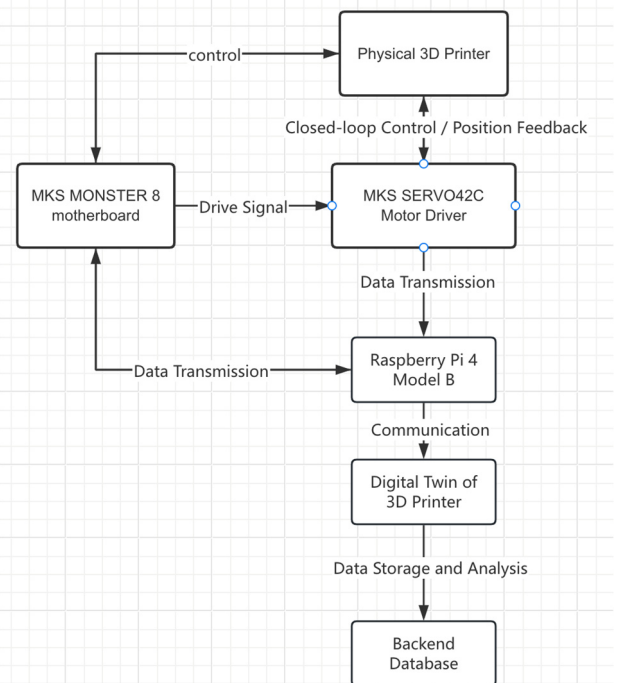


Figure 1. Architecture Diagram of the Digital Twin Control System

To accurately reproduce the behavior of the physical printer, a digital twin was developed in Unity3D. The twin simulates key parameters such as nozzle position, movement speed, extrusion volume, nozzle temperature, bed temperature, and current print file status. Communication between the physical printer and the digital twin is achieved via the HTTP protocol, ensuring real-time synchronization. The Unity application is

external disturbances.

Suitable for digital control systems (e.g., single-chip microcomputers or PLC control) to avoid excessive control output caused by calculation overflow.

3. Digital Twin Construction in Unity3D

3.1. Virtual Simulation of the Digital Printer

The printer model was first created in SolidWorks and imported into Unity for simulation. The designed printer adopts a UM2 Core X-Y motion structure, whose behavior is reproduced in Unity using interpolation functions. Interpolation methods, including linear interpolation and cubic Bezier curves, are used to generate smooth motion paths between keyframes. Unity’s built-in Lerp and SmoothStep functions allow precise computation of nozzle trajectories.

Printing paths originate from G-code generated by slicing software. G-code specifies motion trajectories, extrusion volumes, and thermal parameters. By parsing G-code, motion control points are extracted and interpolated to reconstruct nozzle trajectories. Additional simulation features include mesh filling and layer merging to reproduce deposition thickness and layer transitions, enabling a highly realistic virtual print process.

3.2. User Interaction with the Digital Printer

A comprehensive human-machine interaction framework was built using Unity’s C# scripts, event system, UI components, and I/O modules.

3.2.1. Interaction Function Modules

View Control:

```

MoveInput
Vector3 move = Vector3.zero;

        move.x = Input.GetAxis("Horizontal") *
moveSpeedWithKeyBoard * Time.deltaTime;
        move.y = Input.GetAxis("Vertical") *
moveSpeedWithKeyBoard * Time.deltaTime;
        move.z = Input.GetAxis("Sagittal") *
moveSpeedWithKeyBoard * Time.deltaTime;

        if (!MouseStatus.isPointerOnUI(canvas))
        {

                if (Math.Abs(Input.GetAxis("Mouse
ScrollWheel")) >= 0.1)
                        move.z = Input.GetAxis("Mouse
ScrollWheel") * moveSpeedWithScroll * Time.deltaTime;

                if (Input.GetMouseButton(2))
                {
                        move.x = -Input.GetAxis("Mouse X") *
moveSpeedWithMouse * Time.deltaTime;
                        move.y = -Input.GetAxis("Mouse Y") *
moveSpeedWithMouse * Time.deltaTime;
                }
        }
return move;

```

Keyboard inputs are read via Unity's Input.GetAxis()

method (e.g., WASD keys control forward, backward, left, and right movement; mouse scroll wheel adjusts view zoom). Mouse drag rotation uses Transform.Rotate(), enabling 360 degree observation of printer status.

Print Control:

Monitors mouse click events (Input.GetMouseButtonDown()) to trigger print job start, pause, or termination. Via UnityWebRequest, it sends HTTP commands to the Raspberry Pi to control the physical printer's behavior.

```

public Reset resetX;
public Reset resetY;
public Reset resetZ;
public GCodeIdentify gCodeIdentify;
public PrinterManager printerManager;

/// <summary>
/// </summary>
public void doReset()
{
        Console.createMessage(3, " Start resetting ");

        resetX.stop = resetY.stop = resetZ.stop = false;
        Reset.isReset = 0;

        StartCoroutine(resetX.axisReturn());
        StartCoroutine(resetY.axisReturn());
        StartCoroutine(resetZ.axisReturn());
}

```

Information Feedback:

Key parameters such as printing progress, nozzle temperature, and layer height are displayed in real-time on the UI panel. Status information is dynamically updated via the TextMeshPro component, with support for multi-language switching.

```

public TextMeshProUGUI statusText;
void UpdateStatus(string message)
{
        statusText.text = " Current status:" + message;
}

```

4. System Data Mgmt & Anal.

4.1. Design & Impl. of Communication Module

The system uses Fluidd and Moonraker as control and communication interfaces. Fluidd provides a lightweight web UI for managing print tasks, uploading G-code, and monitoring progress, while Moonraker is a Python-based web server that exposes APIs via HTTP, WebSocket, or MQTT.

The Raspberry Pi communicates with the MKS Monster8 V2.0 board via Unix domain sockets, ensuring efficient inter-process communication. Between Unity and the Raspberry Pi, HTTP requests are used for state queries and control commands.

4.2. Data Storage and Analysis

The backend is implemented using Go, while the frontend employs Vue, with MySQL as the database. Key parameters (time, height, speed, temperature, etc.) are stored in structured form. Analysis algorithms are applied to detect anomalies such as overheating, extrusion inconsistencies, or nozzle misalignments.

5. Conclusion

This study presents an intelligent 3D printing monitoring and control system built upon digital twin technology, achieving real-time synchronization and remote control of the virtual and physical printers. Techniques including incremental PID control, G-code parsing, and process data analysis enhanced printing precision, stability, and intelligence.

The contributions of this work lie in digital twin modeling, intelligent control algorithms, and data-driven analysis, offering new perspectives for smart manufacturing and additive manufacturing. Future improvements should focus on adaptive PID control, intelligent data analytics, and integration with cloud computing and AI algorithms to further enhance system intelligence.

Competing Interests

The authors declare that they have no competing of interest.

References

- [1] Zhang Jian. Research and Implementation of a Control System for Desktop FDM 3D Printers [D]. Jiangsu University of Science and Technology, 2019.
- [2] Liu Lugang, Wang Kun, Guo Haidong. Design and Fabrication of 3D Printers [M]. 2023.
- [3] Bai Yuhui. Mechanical Structure Design of 3D Printers. Electronics Technology, 2023.
- [4] Liu Yonggang, Yao Liquan. Application of Digital Twin-Based Virtual Simulation in Remote Training for Intelligent Manufacturing. Journal of Hubei Open Vocational College, 2023.
- [5] Liu Dawei, Zhang Guoquan, Li Zeshan, et al. Design and Application of a Unity3D-Based Virtual Simulation Platform for Engineering Graphics. Electromechanical Engineering Technology, 2022.
- [6] Tan Renming, Zhang Renjie, Jiang Tao. Design of a Stepper Motor Position Closed-Loop Control System Based on STM32. Journal of Heilongjiang Institute of Technology, 2023.
- [7] Lu Jianhua, Long Chaoping. Exploration of Raspberry Pi-Based Embedded Course Teaching. Education Forum, 2019.
- [8] Lin Rui. Research and Design of an IoT-Based Industrial Monitoring Platform [D]. Yangtze University, 2021.
- [9] Wang Yichen. Comparison and Study of Incremental and Positional PID Algorithms. Industrial Control Computer, 2018.
- [10] Jin Tao, Ma Jie, Yang Lichuang. Design of a WiFi-Based Remote Control System for 3D Printers. Sensors and Microsystems, 2018.