Journal of Electronic Research and Application, 2025, Volume 9, Issue 1
http://ojs.bbwpublisher.com/index.php/JERA
ISSN Online: 2208-3510
ISSN Print: 2208-3502

Development of Virtual Simulation System Based on Motion Control Card and Unity Platform

Shuangquan Wen[1]*, Pengxiong Wang[2]

[1]School of Mechanical and Electrical Engineering, Shenzhen Polytechnic University, Shenzhen 518055, China
[2]Dongguan Googol Automation Technology Co., LTD., Dongguan 523459, China

***Corresponding author:** Shuangquan Wen, sqwen@szpu.edu.cn

Abstract: Digital twin can simulate and monitor the state and behavior of physical entities in the real world, helping enterprises to better understand and manage real-world physical systems, improve production efficiency, reduce costs, and improve safety and reliability. In this paper, we use GTS motion control card and Unity engine to build a digital twin system, and control a virtual industrial automation handling platform including two screw servo axes and multiple sensors through the physical GTS motion control card. The control card program controls the motion of the virtual model through transmission control protocol (TCP) communication, and the virtual model system feeds back the signal to the control card program to achieve the virtual and real synchronous digital twin effect. The digital twin system uses Unity engine to create a highly realistic virtual environment, and can run on multi-platform terminals.

Keywords: Digital twin; GTS motion control card; Unity engine; TCP communication

Online publication: February 12, 2025

1. Introduction

In the field of industrial automation, digital twin technology can help enterprises better understand and manage physical systems in the real world, improve production efficiency, reduce costs, improve safety and reliability, and predict the occurrence of system failures before they occur, and perform predictive maintenance. In this paper, the computer installed with the GTS motion control card of GoogolTech is used to control the virtual model built in the Unity engine environment, and the virtual model can also feedback the signal to the motion control card program to realize the digital twin effect. This paper will discuss the design process of the virtual simulation system in detail.

2. Overall system design

The virtual simulation system designed in this paper includes two computers, one is an industrial computer installed with GTS motion control card, which runs GTS upper computer software. The actual physical equipment can be controlled by the control card. The other computer is the virtual platform computer, which runs the virtual model program built by Unity environment. The virtual model is the digital twin of the physical device.

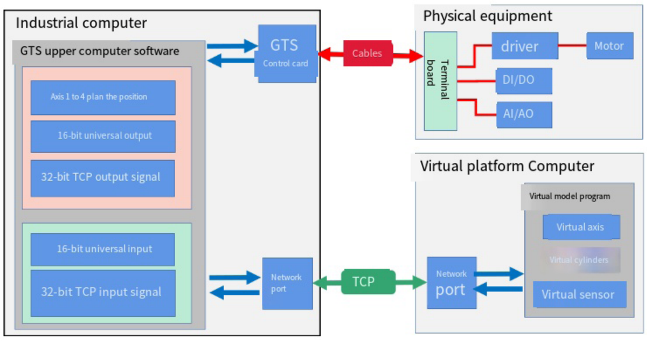

Figure 1. Schematic diagram of the communication mode of the virtual simulation system

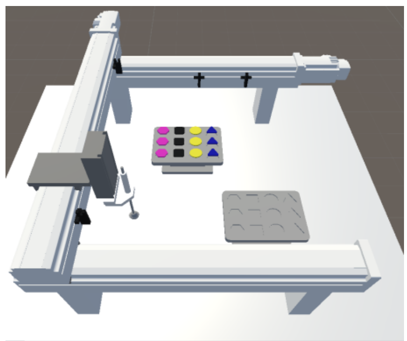

Figure 2. Virtual model of the handling platform

Figure 1 shows the data interaction and communication mode among GTS host computer software, virtual model program, and physical equipment. The motion control card of the industrial computer is connected to the terminal board of the physical equipment through the control cable, and the driver, digital input/digital output (DI/DO) and other components are connected under the terminal board. The industrial computer communicates with the virtual platform computer through TCP protocol, and sends the data of the motion control card to the virtual model program, and the virtual model program also feeds back the signal to the control card program.

The physical equipment is an industrial automation handling platform including two screw servo axes and multiple sensors. The appearance of the virtual model in Unity environment is shown in **Figure 2**.

3. GTS motion control card host computer program design

The host computer program is divided into seven subsystems according to the function of the user management system, log system, input/output (IO) control system, axis system, manual part, automatic program, TCP communication and so on. Each subsystem function is independent, and can call each other, combine with each other to achieve more functions. The upper computer program is mainly implemented by two classes AxisCtrl and IOCtrl.

Among them, the AxisCtrl class realizes the most basic functions of the axis system. This class opens a series of functions for other subsystems to control the axis and obtain axis information, etc. The axis system can call axis resources uniformly, which is convenient for the realization of various functions, axis resource scheduling, axis security protection, etc. The functions provided by this class mainly include GetAxisFree(), AxisMovePos1(), AxisGoHome(), AxisEmgStop(), GetAxisPrfPos(), GetAxisPrfVel(), GetAxisEncPos(), and GetAxisEncVel(), etc.

The IOCtrl class realizes the most basic functions of IO control system, this class opens a series of functions for other subsystems to read and write IO values, IO control system achieves unified management of IO information. The functions provided by this class mainly include GetDICurVal(), GetDOCurVal(), SetDOVal(), GetAIVal(), GetDacVal(), SetDacVal(), GetDICurValFromTcp() and so on.

4. Unity virtual simulation system design

The virtual model program is developed using Unity engine, its role is to receive the motion data of GTS host computer program, combine Unity engine and script, simulate the behavior and state of real equipment. The virtual model program is divided into three parts: user interface (UI), model and script.

UI interface can set TCP server internet protocol (IP) parameters, reset the model, exit the program and other operations, can display the system status, display log information, and so on.

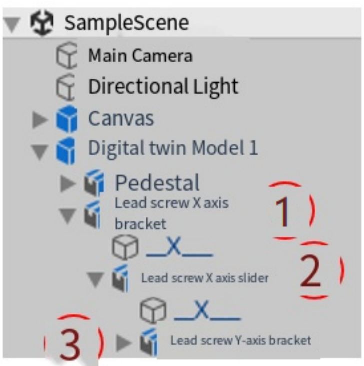

Figure 3. Hierarchy of physical union

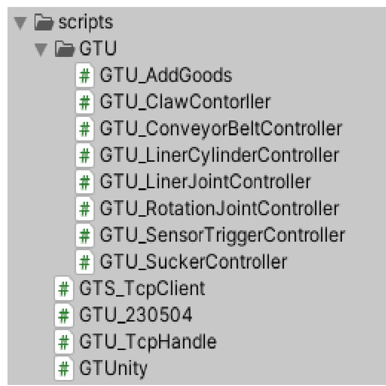

Figure 4. Overview of script of handling platform

The 3D model of the handling platform is static after importing into Unity. In order to obtain realistic physical behaviors in the simulation environment, it is necessary to use hierarchically organized objects to build physical combinations. A physical bond is a set of bonds organized in the form of a logical tree, in which each parent-child relationship reflects the relative motion of mutual constraints. As shown in **Figure 3**, this paper simulates the linear motion of the X, Y axis, with (1) “Lead screw X axis bracket” as the root, followed by (2) “Lead screw X axis slider,” and then (3) “Lead screw X axis bracket,” combination in turn. In addition, the collision body component is used to define the shape of the physical collision of the game object. The collision body that needs to be set up for the handling platform has various shapes of materials, suckers, nozzles, sensors, shaft sliders, light blockers, and so on. Rigid body components also need to be set for each material to obtain material effects such as gravity.

Scripts are used to create graphical effects that control the physical behavior of the object. The script used by the virtual handling platform is shown in **Figure 4**, which realizes the functions of the axis movement of the handling platform, the action of the cylinder, the suction of the material by the suction cup, the sensor feedback, and the TCP communication. Among them, GTU_LinerJointController is a script to realize the linear motion of the axis. The script is mounted on the game object corresponding to the axis.

As shown in **Figure 5**, the Axis Dir property can be set to constrain the direction of movement of the axis, Speed controls the speed of movement of the axis, and Goal Pos sets the position of the axis. This script also provides the SetGoalPos() method to set the axis Goal Pos value. The script is associated with the Articulation Body assembly component of the game object. Through the TCP protocol, it can receive the axis position data from the GTS host computer, call the SetGoalPos() method, and input the axis position into the script, thus realizing the movement of the shaft model.

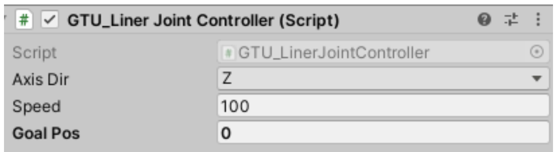

Figure 5. GTU_LinerJointController property panel

GTS_TcpClient is the function of TCP communications scripts. As TCP client, it can set the server's IP, connect the server, get GTS personal computer (PC) data, send data to the GTS PC. The GTU_TcpHandle class is a script that manages TCP function and UI interface in the virtual model program. GTU_230504 is a specific script for this project. In this script, the association between the model action and the GTS host computer data is realized. The attribute panel of this script is shown in **Figure 6**. The binding of Tcp_ is GTU_TcpHandle component. By calling the method provided in GTU_TcpHandle component, it can read or write TCP data, and then control the action of the model or feedback the signal of the virtual sensor. The DataList contains a set of data items, which are the configuration of the virtual model. If the AxisIndex type is selected as axis, then this is the axis number, read the position of the axis with AxisIndex from the TCP data, and assign the position data to the GTU_LinerJointController component in the GameObject. If the DIIndex/DIOffs type is a suction cup or a linear cylinder, then these two values are used as the DI address, and the value of this address is read from the TCP data and assigned to the corresponding script component in the GameObject. If the DOIndex/DOOffs type is selected as the sensor trigger, then the sensor trigger value of the corresponding GTU_SensorTriggerController component in the GameObject will be read, and the sensor trigger value will be written in the TCP data with DOIndex/DOOffs as the DO address.

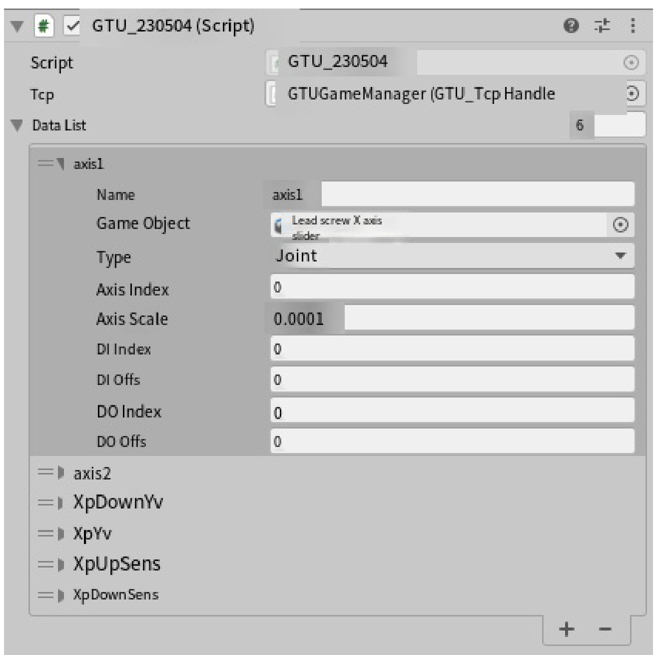

Figure 6. GTU_230504 property panel

The rest of the script includes linear controller GTU_LinerCylinderController cylinder, sensor trigger controller GTU_SensorTriggerController and sucker controller GTU_SuckerController, etc.

4. Virtual simulation program and GTS host computer program data interaction

TCP protocol is used to communicate between the virtual model program and the GTS host computer program, and the communication connection is shown in **Figure 1**. The data transmitted by the GTS host computer to the virtual model program includes the planned position of axis 1-4 and the general output. Its data structure is defined as a string of characters consisting of "axis 1 planning position # axis 2 planning position # axis 3 planning position # axis 4 planning position # general output," which is sent to the virtual model program through TCP protocol. The data sent by the virtual model program to the GTS host computer is the virtual sensor signal, which is also defined as a string of characters and sent to the GTS host computer via the TCP protocol.

In the TCP communication form of the GTS host computer, there are mainly three independent threads, namely ReadDataFast(), DealMessage(), and SendDataFast(), for data processing. Among them, ReadDataFast() is responsible for reading TCP messages at an interval of 5 ms and saving the read information to the data buffer area. DealMessage() is responsible for decoding the data in the data buffer area, extracting the useful data, and assigning it to the data variables. SendDataFast() is responsible for sending data at an interval of 10 ms. It edits the data of the GTS host computer into a string and sends it to the virtual model program via the TCP protocol. The GTS host computer does not need to consider the update of the sent data because in the SendDataFast thread, it directly reads the latest data of the control card, edits it into a string, and sends it out via TCP. If the GTS host computer needs to use the data input via TCP, it can call the method GetDICurValFromTcp() of the IOCtrl class to obtain the current TCP data.

In the GTU_TcpHandle script of the virtual model program, there are also three independent threads with the same functions for data processing. Similarly, the virtual model program does not need to consider the update of the sent data. After calling the SetDOVal method of the GTU_TcpHandle script, the data is saved to the sending buffer area. The virtual model program directly reads the data in the buffer area, edits it into a string, and sends it out. If the virtual model program needs to use the data input via TCP, it can call the methods GetDIVal() or GetCurAxisPos() of the GTU_TcpHandle script to obtain the latest TCP data.

5. Conclusion

During the current stage of the transformation of the manufacturing industry towards intelligent manufacturing, digital twins have achieved a deep integration of the physical world and the virtual world. In this paper, with the GoogolTech motion control card as the host computer and the Unity engine as the simulation platform, the digital twin effect of the linkage between the virtual and the real has been realized. The constructed virtual simulation system is also applicable to experimental and practical teaching.

Funding

Research Startup Project of Shenzhen Polytechnic University "Research and Development of High-Speed and High-Resolution 2D/3D Combined Vision Sensor" (Project No. 6022312003K).

Disclosure statement

The authors declare no conflict of interest.

References

[1] Cao YF, Wang HS, Zheng CK, 2021, An Experimental Study Based on Digital Twin. Machine Building & Automation, 50(4): 32–34. https://doi.org/10.19344/j.cnki.issn1671-5276.2021.04.008

[2] Liu L, 2021, Construction of Virtual Simulation System of Gantry Machine Tool Based on Digital Twin and the Research on Collision Detection Technology, thesis, Chongqing University of Technology.

[3] Fang F, Yuan YP, Wang B, et al., 2023, Digital Twin System in Textile Machinery Manufacturing Workshop Based on Unity3D. Industrial Control Computer, 36(8): 143–144. https://doi.org/10.3969/j.issn.1001-182X.2023.08.057

[4] Lin CZ, Huang H, Zhang SX, 2023, Design of Digital Twin Platform for Automated Production Line Based on Unity. Journal of Tangshan College, 36(3): 38 + 43, 84. https://doi.org/10.16160/j.cnki.tsxyxb.2023.03.007

[5] Qi J, 2020, Cross-Border Empowerment, Unity Real-Time 3D Technology Helps the Implementation of Digital Twins. Intelligent Manufacturing, 2020(12): 25–27.

[6] Chen YJ, Zeng SY, He HJ, et al., 2023, Research and Implementation of Digital Twin System for Industrial Robot Palletizing. Computer Integrated Manufacturing Systems, 29(6): 1930–1940. https://doi.org/10.13196/j.cims.2023.06.012

[7] Wang HJ, Wang, ZF, 2020, Research on PLC Simulation Teaching Platform Based on Unity. International Conference on Intelligent Design (ICID), 2020: 15–18. https://doi.org/10.1109/ICID52250.2020.00011