

Article

Development of Manipulator Digital Twin Experimental Platform Based on RCP

Zhe Dong *, Xiaoyao Han, Yuntao Shi, Weifeng Zhai and Song Luo

School of Electrical and Control Engineering, North China University of Technology, Beijing 100041, China

* Correspondence: dongzhe@ncut.edu.cn

Abstract: From the perspective of teaching and researching, we developed a manipulator digital twin experiment platform (named the remote experience platform, REP) based on a rapid control prototype (RCP). The platform consisted of a controlled target, a real-time controller, rapid prototype configuration software, and supervisory control software. The controlled target was a 6-DOF manipulator, divided into a physical entity and its digital twin. The 3D model and mathematical model of the manipulator were constructed as an experimental entity in a digital space. The whole system provided flexible and intuitive experimental scenes without the restraints of time and place. Based on RCP technology, students can design various complex control strategies using simulation tools such as Matlab/Simulink, then convert the graphical model into executable code to be performed in target hardware. The framework and development methods of the proposed system are elaborated in this paper. An example is demonstrated, including invocation of algorithms, one-click code generation and compilation, real-time verification and online parameter adjustment, and more. The feasibility and practicability of the system are verified through the PID control experiment of the manipulator.



Citation: Dong, Z.; Han, X.; Shi, Y.; Zhai, W.; Luo, S. Development of Manipulator Digital Twin Experimental Platform Based on RCP. *Electronics* **2022**, *11*, 4196. <https://doi.org/10.3390/electronics11244196>

Academic Editors: Juan M. Corchado, Byung-Gyu Kim, Carlos A. Iglesias, In Lee, Fuji Ren and Rashid Mehmood

Received: 9 November 2022

Accepted: 11 December 2022

Published: 15 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: RCP; digital twin; code generate; experimental teaching

1. Introduction

In recent years, with the development of the IOT (internet of things) technology such as real-time networks and digital twins, education informatization has become possible [1]. Particularly over the past few years, due to the impact of COVID-19, education at many universities has been limited; therefore, the role of online education is becoming increasingly prominent [2]. However, experimental teaching, which is equally important and indispensable compared with theoretical teaching, is difficult to carry out online because of the dependence on hardware [3]. As far as the current experimental mode is concerned, the remote experimental platform provides an effective solution [4,5].

In today's industrial manufacturing, manipulators are widely used because of their high production efficiency and low labor costs [6,7]. As the manipulator is a typical object of engineering majors, experiments concerning the development and implementation of manipulators are significant. A complete manipulator experiment course includes basic motor control, forward and inverse kinematics analysis, path planning, etc. As the manipulator development process is long and difficult, the design and implementation of a practical manipulator requires various technical skills such as theoretical analysis, controller design, software programming, hardware design, and system integration. At the same time, the development procedure is iterative, and the learning efficiency is reduced accordingly. The existing experimental platform often relies on the development environment provided by the robot manufacturer. Due to the closed development environment, it is difficult to design, develop, and research from the bottom layer [8–10].

A digital twin provides a link connecting the physical world and the digital world, and this has received extensive attention [11–13]. The proposal of this concept effectively combines virtuality and reality, which increases the intuition and visualization of the

experiment. Based on the digital twin, the experiment can be carried out without relying on physical objects. By establishing an accurate digital model of the physical entity and the communication channel between the physical entity and the model, the two-way mapping between the digital model and the physical entity is realized. Compared with traditional modeling and simulation, a digital twin has the advantages of a shorter design cycle, higher reliability, lower overhaul frequency, and lower maintenance cost. Therefore, it is an effective means for remote and online experiments [14–16].

RCP technology provides a rapid development method from model to code, and finally, to object. It can greatly shorten the development cycle, improve the iteration speed, and reduce the development cost [17,18]. It has been widely used in engineering research and development in the aerospace industry, automotive electronics, and other fields. At present, many companies provide their unique RCP software and hardware solutions for different research fields, such as dSPACE in Germany, RT-LAB in Canada, and others. However, these products are mainly oriented to enterprises and they are expensive. Therefore, they are not suitable for academic application [19–23].

Based on the above reasons, this paper proposes a remote experimental platform (REP) for the manipulator experiment. The platform consists of a controlled target, real-time controller, rapid prototype configuration software, and supervisory control software. The controlled target was a 6-DOF manipulator, which is divided into a physical entity and its digital twin. The real-time controller was developed based on RCP. The 3D model and mathematical model of the manipulator were constructed as an experimental entity in digital space. The platform provides an integrated solution for students to learn the manipulator from the bottom layer and understand how it works. Using this platform, we can not only learn the traditional control algorithm but also verify our own algorithm in real time and improve students' basic skills and comprehensive quality.

The structure of this paper is as follows: The second part describes the overall system design. In the third part, the design of the REP is introduced. In the fourth part, the teaching method adopted in this paper is compared with the traditional teaching method, and the teaching experiment is carried out with examples. Finally, the fifth section discusses future challenges and concludes the paper.

2. Overall System Design

The goal of the REP is to provide a manipulator control algorithm verification platform for students and teachers. This platform provides a variety of experimental methods which can not only carry out simulation experiments and physical experiments alone but can also realize functions such as virtual–real interactions of the manipulator. First of all, the platform required a section that could automatically generate code from the graphical model and then compile it to an executable file automatically. In addition, the controlled object section needed to have both the real manipulator and its mathematical model. It was used to complete the virtual–real interaction and other functions of the manipulator. Finally, in the section on data monitoring, not only was the data display interface required, but also the display of the 3D interface of the manipulator was required. The visual display of the data can make the movement of the manipulator more intuitive. The system diagram of the teaching experiment platform is shown in Figure 1.

The proposed REP system consists of six main components, namely the REP-Link, REP-Toolbox, REP-DT (digital twin), REP-Server, REP-View and REP-3D, respectively. The REP-Link was mainly used to transfer the graphical simulation models into executable files running in real-time controller with one click. The REP-Toolbox is a hardware interface toolbox customized for manipulator, which was embedded in Matlab/Simulink. The REP-DT provided the mathematical models for the digital twin of the manipulator. The REP-Server managed various services of this system, providing a data service, web service and platform service. The REP-View was the data visualization interface for supervisory control of the experiment. The REP-3D is a 3D engine where the digital twin of the

manipulator ran, which was embedded in the REP-View. Each component is described in detail in the next chapter.

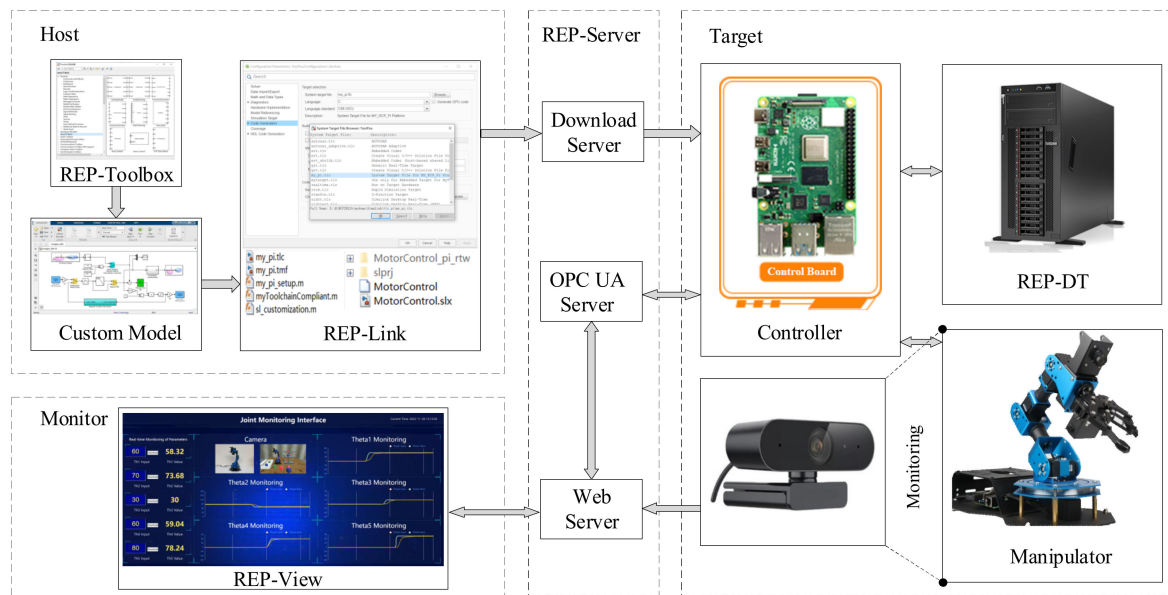


Figure 1. System diagram of the experiment platform.

3. Remote Experiment Platform Design

3.1. Design and Implementation of the Rapid Control Prototype System

This section contains the design and implementation of the REP-Link and REP-Toolbox.

The function of the REP-Link is to generate source code from the Matlab/Simulink model and then compile it to produce executable files. In the REP-Link, all steps are automated, and thus users can complete the process from model to algorithm implementation with just one click. At the same time, REP-Link should have the function of remotely starting and stopping the real-time model and adjusting the model parameters online.

The main process of developing the REP-Link includes the development of a TLC (target language compiler) file and TMF (template makefile) file, the customization of the grt_main.c program, the establishment of a cross-compilation environment, etc. TLC was used to configure system customization options and code generation options. The corresponding hook file was developed at this stage. After the user selects the system TLC file, all options can be automatically configured. The grt_main.c is the main function of the entire project. It is a custom template file for program, and multiple interfaces are reserved for code insertion. The TMF is the template of the makefile. After the model was built, clicking the build button first generated the code for the model, and then the TMF was called upon to compile and generate an executable file.

First of all, the Simulink model was compiled into a textual .rtw model description file. In this file, the configuration and attribute information of all modules such as input, output, parameters, and status were described. Second, the embedded coder used TLC tools to execute .tlc files and turn .rtw into code. Finally, TMF generated model.mk and compiled the generated code and template code together to generate an executable file. The automatic code generation process is shown in Figure 2.

According to the special needs of the manipulator, the corresponding development was carried out in the TMF file, which mainly included the following steps:

- (1) Compiler, header files, and related library file paths.

A cross compiler is built according to the chip model of the manipulator controller. The compiler path is added to the TMF file. In order to realize other functions, the header file and library file path also need to be added.

(2) Compiler optimization options and libraries required for linking.

According to different requirements and custom optimization options, appropriate optimization parameters make the code more efficient. At the same time, in order to use the static library or dynamic library provided by the third party, the library path needs to be specified when compiling.

(3) Compilation rules

The required C source code is first compiled into .o files according to dependencies and then linked into executable files according to intermediate files and libraries.

(4) External mode

The REP-Link integrates the external mode of Matlab/Simulink. It needs to include the source files related to the external mode and add it to the compilation process.

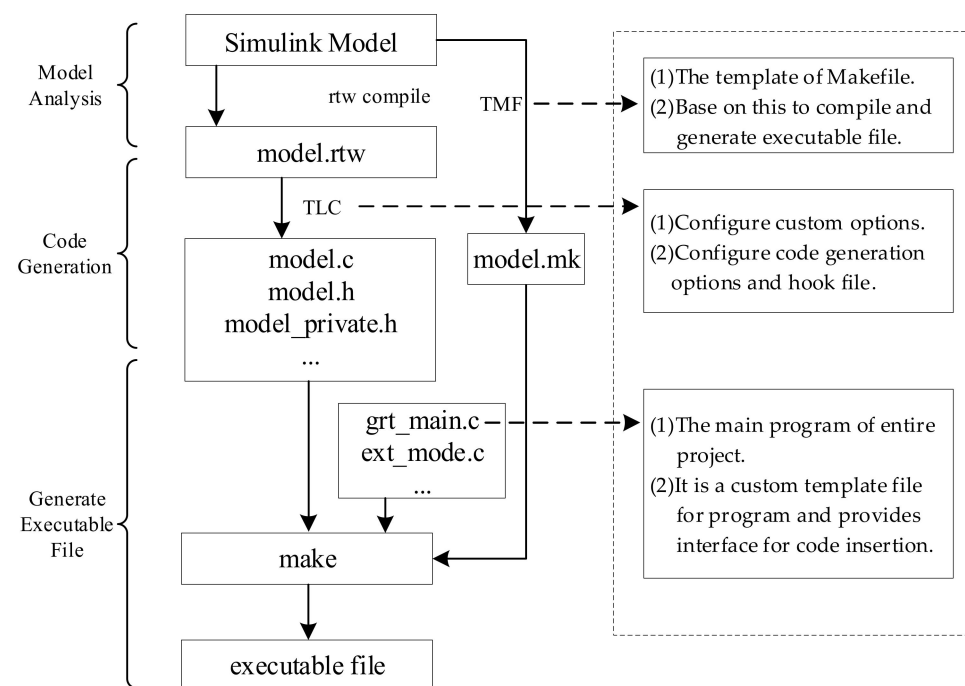


Figure 2. Automatic code generation flow.

The REP-Toolbox provides a Simulink toolbox related to the manipulator. It is mainly used to drive the movement of the manipulator. The development of this part is based on the C Mex S function. According to the communication protocol, each function of the manipulator is encapsulated into the form of a module library. It can be directly dragged by the user when needed. The design process of the manipulator drive module is shown in Figure 3.

Through the above process, the functions of the manipulator are encapsulated. This includes the control of joint rotation, joint position feedback, proportional conversion of control parameters and path planning. The custom module library is shown in Figure 4.

3.2. Design and Implementation of Digital Twinning

This section contains the design and implementation of the REP-DT and REP-3D.

The REP-DT is the mathematical model of the digital twin of the manipulator, including kinematics and dynamics models. Firstly, the manipulator coordinate system was established as shown in Figure 5. The DH parameter list of the manipulator was established as shown in Table 1. Then, derivation of the kinematics equation of the manipulator was undertaken and the parameters were added to achieve the kinematic model. The forward kinematics equation is as per Equation (1), and the inverse kinematics equation is as per

Equation (2). In addition, the mass, center of mass position, and inertia tensor of each link were measured. The manipulator dynamics model was built with Simscape using these parameters, as shown in Figure 6. Finally, the equations of the model were implemented by programming, packaged into modules and put into the REP-Toolbox for users to use.

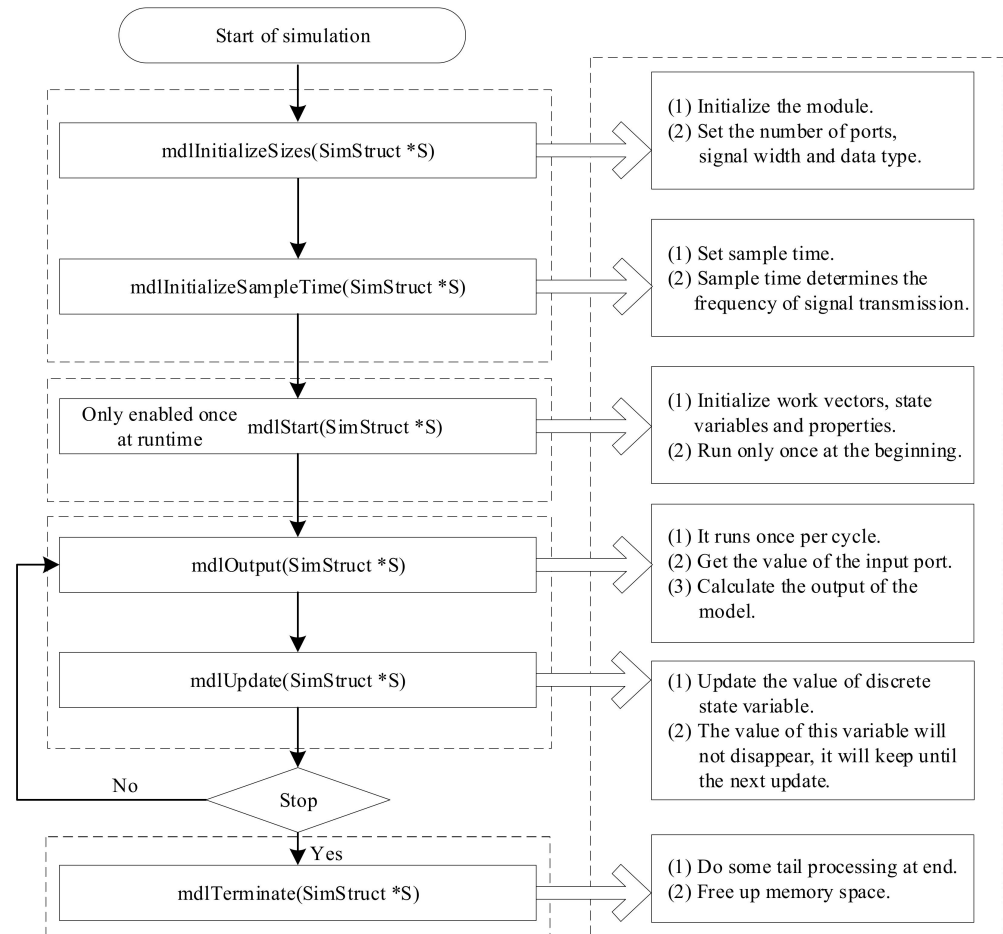


Figure 3. Design flow of device driver module based on Simulink.

The following are the meanings of the DH parameters:

di: Offset along previous z to the common normal;

θi: Angle about previous z, from old x to new x;

ai: Length of the common normal;

αi: Angle about common normal, from old z axis to new z axis.

$$\begin{cases}
 n_x = c_1 c_5 s_4 (s_2 s_3 - c_2 c_3) - c_1 c_4 c_5 (c_2 s_3 + s_2 c_3) + s_1 s_5 \\
 o_x = c_1 s_5 s_4 (c_2 c_3 - s_2 s_3) + c_1 c_4 s_5 (c_2 s_3 + s_2 c_3) + s_1 c_5 \\
 a_x = c_1 c_4 (c_2 c_3 - s_2 s_3) + c_1 s_4 (c_2 s_3 + s_2 c_3) \\
 p_x = d_7 [c_1 c_4 (c_2 c_3 - s_2 s_3) + c_1 s_4 (c_2 s_3 + s_2 c_3)] + c_1 (a_3 s_2 s_3 + a_3 s_2 c_3 + a_2 c_2) \\
 n_y = -s_1 s_4 c_5 (s_2 c_3 + c_2 s_3) + s_1 c_4 c_5 (c_2 c_3 - s_2 s_3) - c_1 s_5 \\
 o_y = s_1 s_4 s_5 (c_2 c_3 - s_2 s_3) + s_1 s_5 c_4 (s_2 c_3 + c_2 s_3) - c_1 c_5 \\
 a_y = s_1 c_4 (c_2 c_3 - s_2 s_3) + s_1 s_4 (s_2 c_3 + c_2 s_3) \\
 p_y = d_7 [s_1 c_4 (c_2 c_3 - s_2 s_3) + s_1 s_4 (s_2 c_3 + c_2 s_3)] + s_1 (a_3 s_2 c_3 + a_3 c_2 s_3 + a_2 s_2) \\
 n_z = -s_4 c_5 (s_2 c_3 + c_2 s_3) + c_4 c_5 (c_2 c_3 - s_2 s_3) \\
 o_z = s_4 s_5 (s_2 c_3 + c_2 s_3) - c_4 s_5 (c_2 c_3 - s_2 s_3) \\
 a_z = c_4 (s_2 c_3 + c_2 s_3) + s_4 (c_2 c_3 - s_2 s_3) \\
 p_z = d_7 [c_4 (s_2 c_3 + c_2 s_3) + s_4 (c_2 c_3 - s_2 s_3)] + a_3 s_2 c_3 + a_3 c_2 s_3 + a_2 s_2
 \end{cases} \quad (1)$$

Here, $c_i = \cos \theta_i$, $s_i = \sin \theta_i$.

$$\begin{cases} \theta_1 = \arctan(p_y - n_y a_7 - a_y d_7, p_x - n_x a_7 - a_x d_7) \\ \theta_2 = \arctan\left(\frac{(a_3 c_3 + a_2)(c_1 m + s_1 n) + a_3 s_3 t}{(a_3 c_3 + a_2)^2 + a_3^2 s_3^2}, \frac{(a_3 c_3 + a_2)t - a_3 s_3(c_1 m + s_1 n)}{(a_3 c_3 + a_2)^2 + a_3^2 s_3^2}\right) \\ \theta_3 = \arctan\left(\pm \sqrt{1 - \left(\frac{c_1^2 m^2 + s_1^2 n^2 + 2s_1 c_1 m n + t^2 - a_2^2 - a_3^2}{2a_2 a_3}\right)}, c_3\right) \\ \theta_4 = \arctan(a_z, c_1 a_x + s_1 a_y) - \theta_3 - \theta_2 \\ \theta_5 = \arctan(s_1 n_x - c_1 n_y, s_1 o_x - c_1 o_z) \end{cases} \quad (2)$$

In Equation (2), $m = p_x - n_x a_7 - a_x d_7$, $n = p_y - n_y a_7 - a_y d_7$, $t = p_z - n_z a_7 - a_z d_7$.

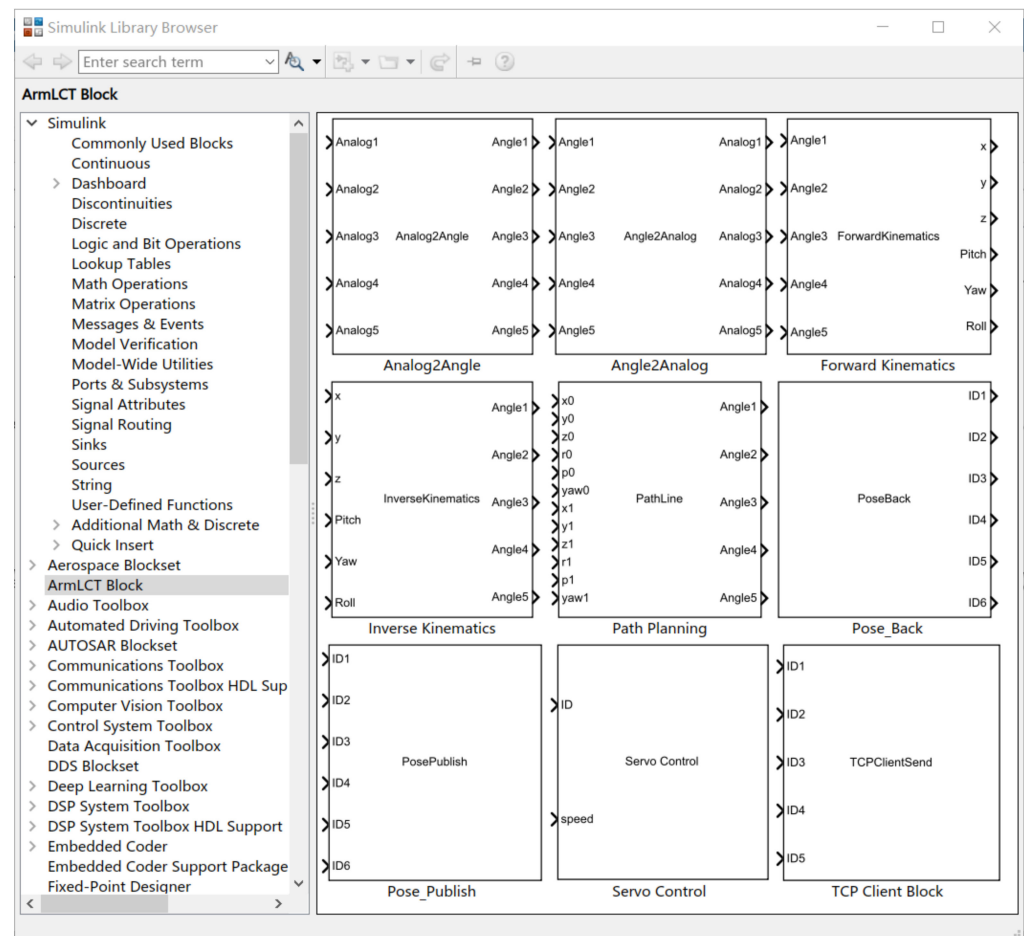


Figure 4. Custom module libraries.

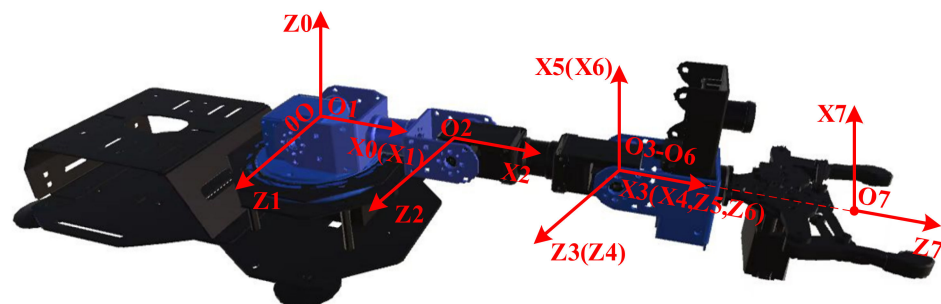
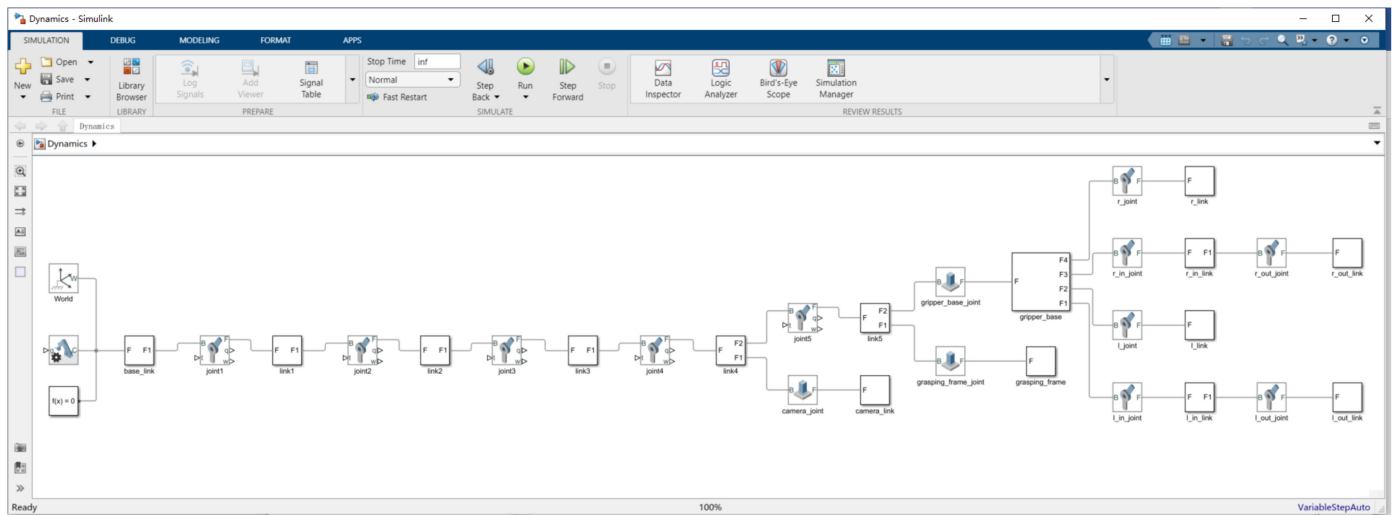


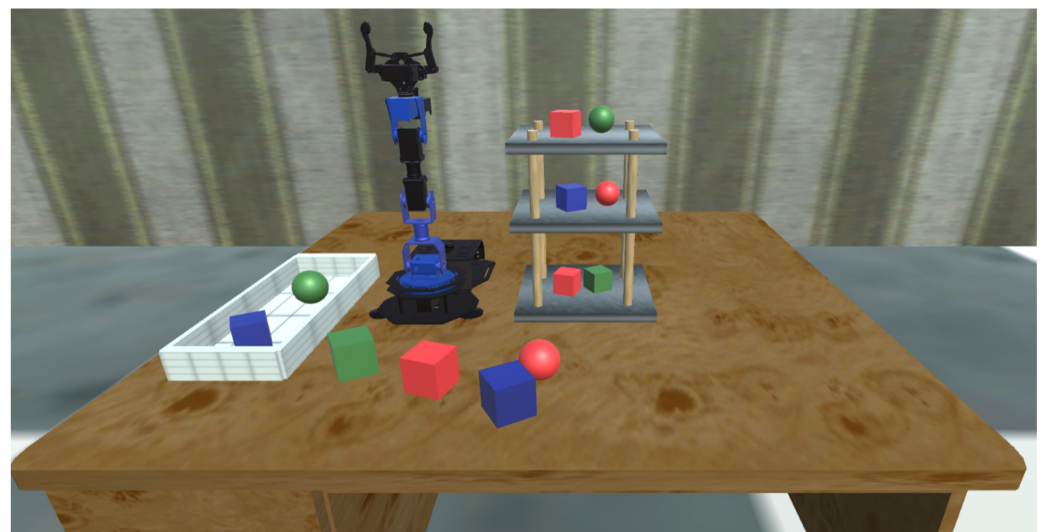
Figure 5. Manipulator coordinate system.

Table 1. DH parameter list.

i	θ_i	d _i	a _i	α_i
1	0	0	0	90
2	0	0	0.09736	0
3	0	0	0.09255	0
4	90	0	0	90
5	0	0.05556	0	0

**Figure 6.** Dynamic model in Simscape.

The REP-3D is the 3D model of the digital twin of the manipulator, which was implemented based on Unity 3D. The 3D model and virtual environment were created in SolidWorks. The model was imported into Unity 3D to assemble each module. Then, joint parent–child relationships, conditional constraints, the rotation range of each joint, and the collision properties of each module were added. The 3D model script of the manipulator was developed, including motion trajectory interpolation, UI control, communication scripts, etc. As the angle data of the manipulator of the 3D model came from the REP-Server, and the 3D model moved in real time. The 3D model of the manipulator is shown in Figure 7.

**Figure 7.** 3D scene of virtual manipulator.

3.3. Design and Implementation of Data Monitoring

This section contains the design and implementation of the REP-Server and REP-View.

The REP-Server was an important part of data transmission in this system, which included download server, OPC UA server and web server. After the REP-link was used to generate an executable file, the download server downloaded the file to the target. The communication with the target was developed based on the Simulink external mode. During the compilation of the model to generate executable files, the system automatically generated an rtw intermediate description file. This file contained the signal and parameters of the model, sampling period, external communication address and port, and other important information. This intermediate file was used to gather information about the model. The IP address and port in the file were used to communicate with the target. The OPC UA server ran model-related data structures that store real-time data about the manipulator. At the same time, a web server for data monitoring was also running in the REP-Server. Its data came from the OPC UA server. After parameters were modified in the REP-View, the web server synchronized the data to the OPC UA server to achieve the control effect. Figure 8 shows the data update process of the OPC UA server on the left and the main interface of the developed OPC UA server on the right.

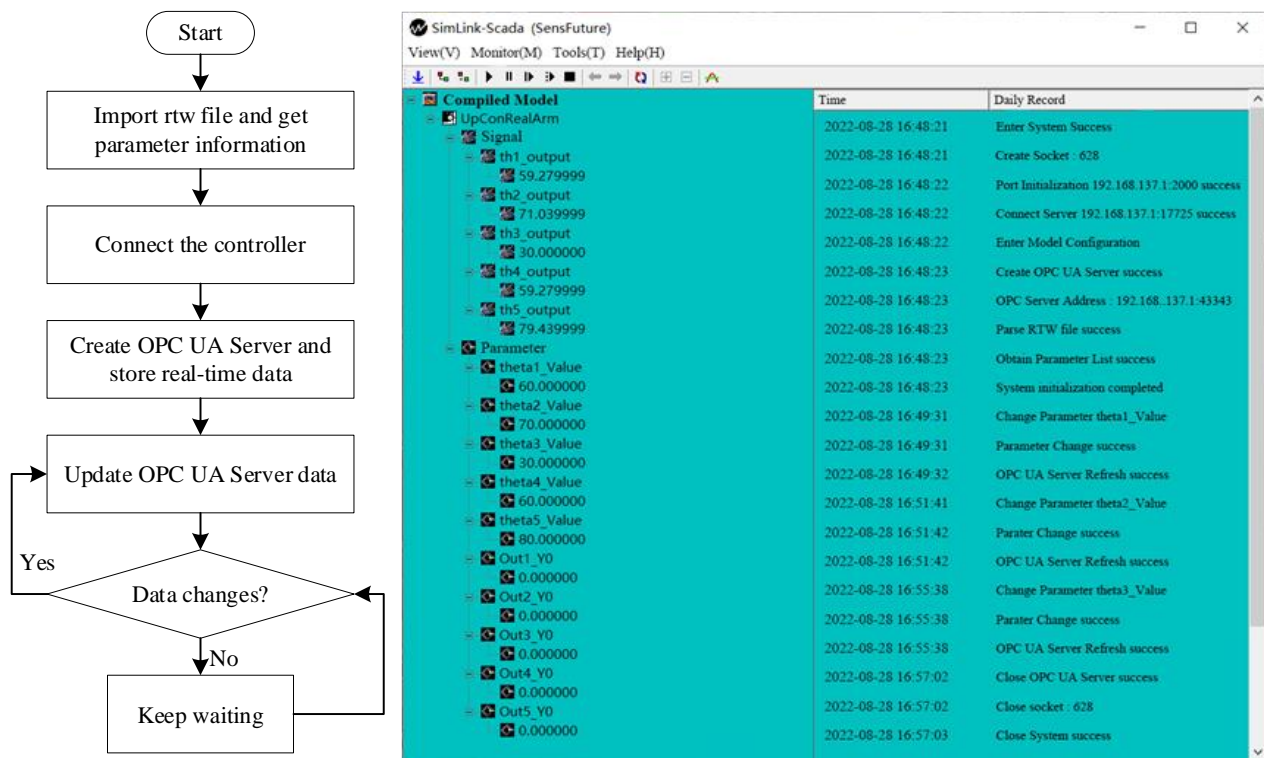


Figure 8. Data flow of OPC UA server and software interface.

The REP-View is a set of data visualization interface developed based on WEB. The interface adopts the flask framework. In the front-end interface, there are two monitoring methods for data. One is numerical display, and the other is line chart display. The back-end reads the data in the OPC UA server. When the user changes the data in the front-end interface, the back-end also updates the data to the server. The WEB-based monitoring interface is compatible with almost all browsers on the market. Figure 9 shows the visual interface of the REP-View.

3.4. Design and Implementation of Video Monitoring

This section contains the design and implementation of video monitoring and the network delay compensation scheme.

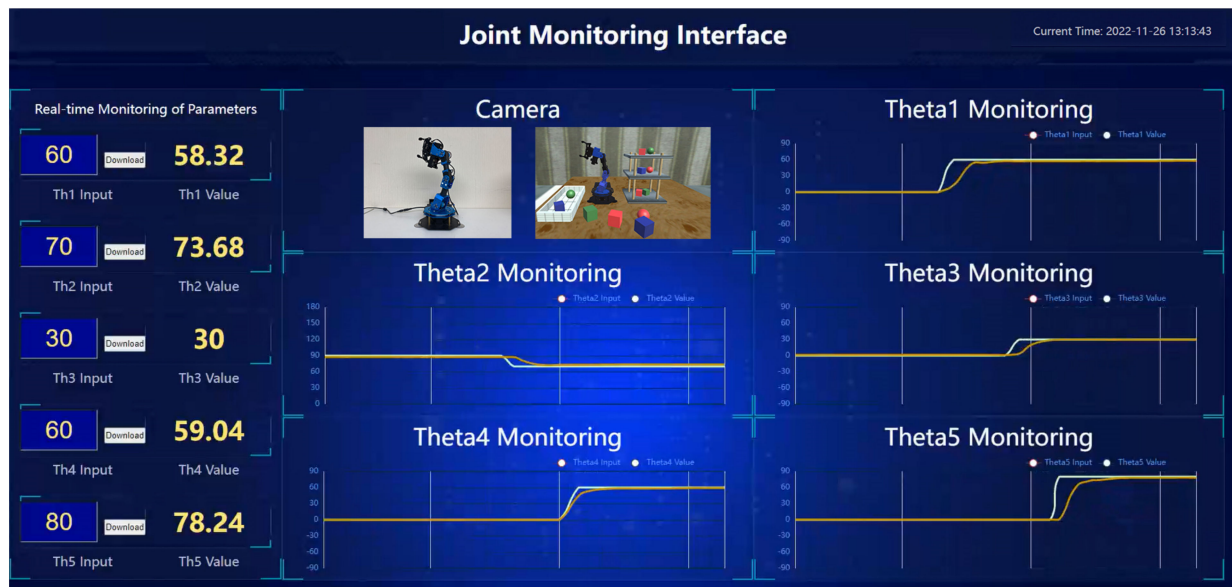


Figure 9. Graphic user interface of REP-View.

The development of this experimental platform aimed to realize remote experiments based on actual objects. To enhance the immersion of the experiments, this experimental system deployed a live-action camera and a virtual simulation camera to observe the experimental effects of physical objects and its digital twins in real time. In the physical space, the live-action camera was deployed in the laboratory where the controlled object was located. The model of the camera used was HIKVISION E12, which was connected to the server via USB. On the web server side, video data were received and provided to the front-end interface. Then, the video of the actual object could be seen through the REP-View. In the digital space, the rotation angle value of each joint of the REP-DT model was fed back to the user side through the network. The REP-3D received the pose data of the REP-DT and reconstructed the 3D scene of the virtual manipulator. Finally, a virtual camera was deployed in the 3D scene with an appropriate viewing point, and the video data of the virtual camera were transmitted to the REP-View interface to realize virtual simulation monitoring, seen as the camera part of Figure 9.

Since the experiment was carried out remotely, the real-time picture of the camera would be affected by network delay, jitter, packet loss, etc. This could cause video lag, freezing, and other phenomena, which would degrade the user experience. Therefore, this paper tested the network performance of remote experiments, which can be divided into two cases, including LAN and WAN. The network performance test used Ping software with the protocol type of the Internet Control Message Protocol, implemented in the form of request and response messages. LAN and WAN tests of network RTT latency and data loss were carried out separately. The local IP of the LAN test was 10.220.12.18 and the target IP was 10.32.246.38; both nodes were in the campus network of the North China University of Technology, Beijing. The local IP of the WAN test was 58.128.2.39 (Beijing) and the target IP was 212.64.62.186 (Shanghai). The sample data of the test were selected from both LAN and WAN for a 24 h continuous test, and a 64-bit data packet was sent every 1 min. The test results are shown in Figure 10, and the statistical results of network RTT latency and data loss are shown in Table 2.

Since the sampling period of the manipulator control system was 10 ms, the frame rate of the video was 24 fps. The LAN's average RTT latency was less than one control sampling period and frame period of the video. Thus, the video transmission within the LAN was minimally affected by the network. However, for the WAN experiment, the average and maximum RTT latency reached 29.9 ms and 96.6 ms, respectively, both of which exceeded one sampling period, and the packet loss rate reached 0.35%, which significantly reduced

the user experience. In order to compensate for the unfavorable situation caused by the wide area network, this paper designed a network compensation strategy to improve the experimental experience.

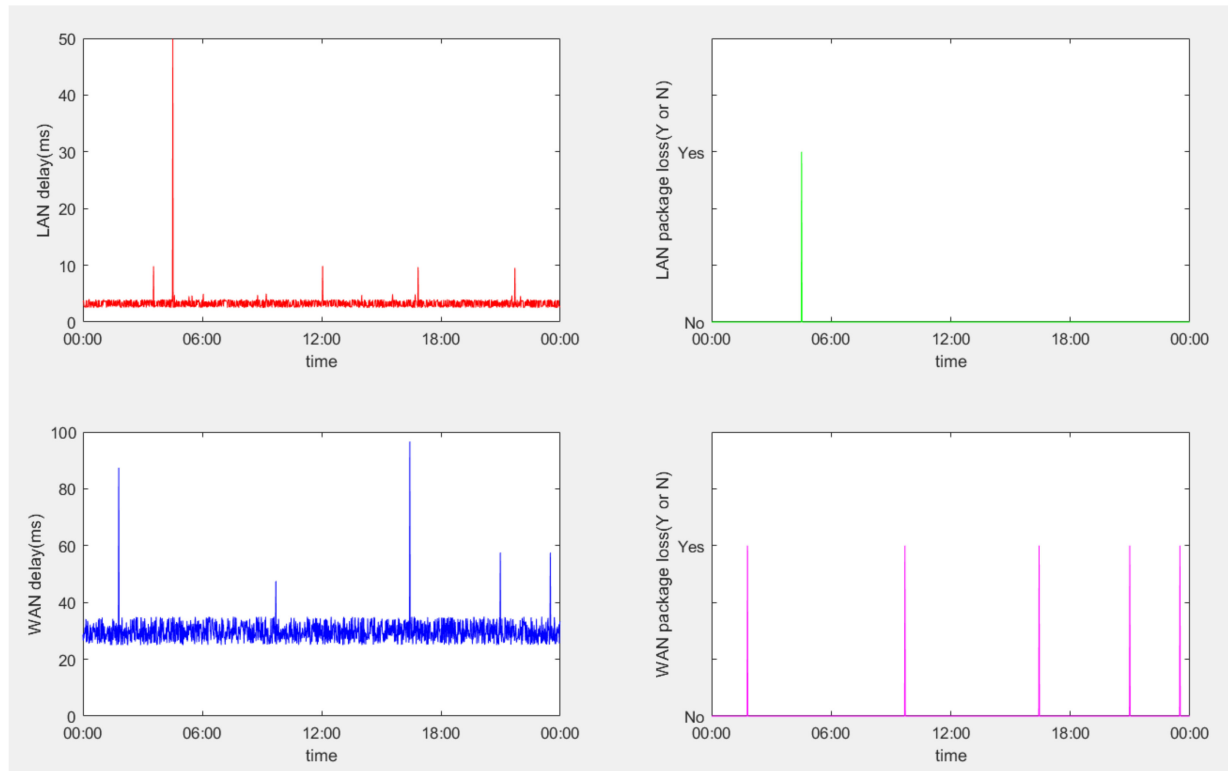


Figure 10. Network RTT latency and packet loss test of LAN and WAN.

Table 2. Statistical results of network RTT latency and packet loss test.

	Min RTT Latency (ms)	Max RTT Latency (ms)	Average RTT Latency (ms)	Loss Rate (%)
LAN	2.5	49.9	3.3	0.07%
WAN	25.0	96.6	29.9	0.35%

As shown in Figure 11, the real manipulator and its digital twin were controlled by the same controller, whose control value $u(k)$ acts on both the physical and digital objects to obtain outputs $y(k)$ and $y'(k)$, respectively. The $y(k)$ of the actual manipulator and its live video were transmitted to the REP-VIEW via the network and displayed in the graphic user interface. The delay and jitter of actual data and video cannot be compensated. Since the digital twin of this experimental platform contained an accurate model of an actual manipulator, this paper designed a network delay compensation strategy by predicting the future output of the REP-DT and reconstructing the virtual 3D scene of manipulator by the predicted value.

Considering that the manipulator control system is a single-input, single-output discrete-time device, it can be defined as

$$y'(k+1) = \frac{B(z^{-1})}{A(z^{-1})}u(k) \quad (3)$$

where $A(z^{-1}) = 1 + a_1z^{-1} + a_2z^{-2} + \dots + a_{a_n}z^{-a_n}$, $B(z^{-1}) = 1 + b_1z^{-1} + b_2z^{-2} + \dots + b_{b_n}z^{-b_n}$ is the coefficient polynomial, $u(k)$ is output of the controller, and $y'(k)$ is output of the mathematical model of manipulator.

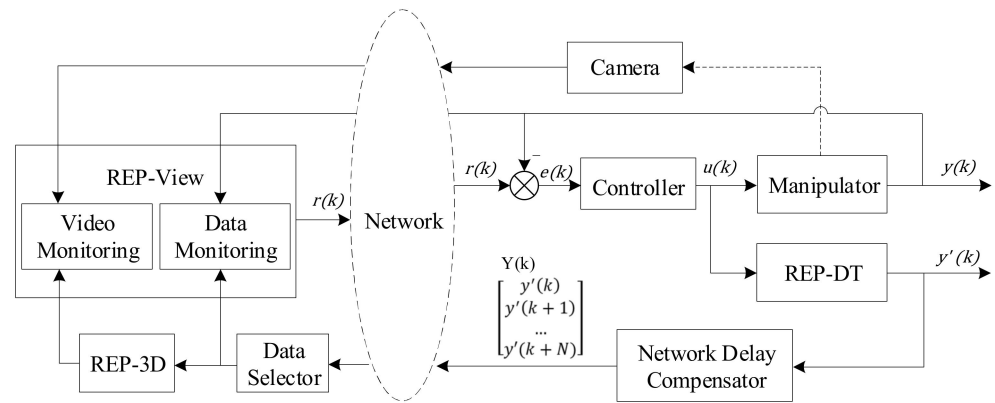


Figure 11. Video monitoring scheme and network delay compensation.

The designed controller of the manipulator can be expressed as the discrete transfer function:

$$u(k) = \frac{D(z^{-1})}{C(z^{-1})} e(k) \quad (4)$$

where $C(z^{-1}) = 1 + c_1 z^{-1} + c_2 z^{-2} + \dots + c_{c_n} z^{-c_n}$, $D(z^{-1}) = 1 + d_1 z^{-1} + d_2 z^{-2} + \dots + d_{d_n} z^{-d_n}$ is the coefficient polynomial, $u(k)$ is the controller output, and $e(k)$ is the error. Where $e(k) = r(k) - y(k)$ and $r(k)$ are the reference inputs. Assuming the model of the manipulator is sufficiently accurate, $y'(k)$ can be considered to converge to $y(k)$ so that we have

$$e(k) \approx r(k) - y'(k) \quad (5)$$

By combining Equations (3)–(5), it follows that

$$y'(k+1) = \frac{B(z^{-1})D(z^{-1})}{A(z^{-1})C(z^{-1})} [r(k) - y'(k)] \quad (6)$$

and then

$$y'(k+2) = \frac{B(z^{-1})D(z^{-1})}{A(z^{-1})C(z^{-1})} [r(k+1) - y'(k+1)] \quad (7)$$

Since $r(k)$ is the reference value, which in this experiment represents the position of the manipulator, which is generally a step signal, it is assumed that $r(k+1) = r(k)$. Therefore, it is not difficult to introduce:

$$y'(k+n) = \frac{B(z^{-1})D(z^{-1})}{A(z^{-1})C(z^{-1})} [r(k) - y'(k+n-1)] \quad (8)$$

where $n = 1, 2, \dots, N$. Based on (8), the output prediction sequence, defined as $Y(k) = \{y'(k|k), y'(k+1|k), \dots, y'(k+N|k)\}$, can be iteratively computed, where $(\bullet|k)$ is the timestamp. The output prediction sequence contains the output at the current moment and the predicted output for the next N moments, where N is the predictive horizon.

Assumption 1. The RTT latency has an upper bound of $2d_{Max}$; the number of consecutive packet losses have an upper bound of L_{Max} .

Let T be the sampling period of the system, then N is decided by

$$N \geq \text{Max}(d_{Max}/T, L_{Max})$$

$$\text{Max}(a, b) = \begin{cases} a, & a \geq b \\ b, & a < b \end{cases} \quad (9)$$

After the predicted sequence $Y(k)$ has been generated by network delay compensator, it is sent to the user side over the network.

At the user side, the data selector is set up to cache the received sequence $Y(k)$ and to make selections from $Y(k)$. In this system, every node in the network adopted the NTP protocol for clock synchronization. Assume that, at moment k on the user side, the data selector receives packets as below:

$$[Y(k - M_0), Y(k - M_1), Y(k - M_2), \dots, Y(k - M_a)] \quad (10)$$

where $M_0, M_1, \dots, M_a \in \{0, 1, 2, \dots, N\}$. In Equation (10), $Y(k - M_0)$ is the prediction sequence retained in the last moment; $Y(k - M_1), Y(k - M_2), \dots, Y(k - M_a)$ are the newly received prediction sequences in this moment, where $a = \{0, 1, 2, \dots, N + 1\}$. When $a = 0$, it means that no prediction sequence is received in this moment. The data selector's strategy is to reserve $Y(k - \text{Min}(M_0, M_1, \dots, M_a))$, discard the rest of the sequences, then output $y(k|k - \text{Min}(M_0, M_1, \dots, M_a))$ to the REP-3D and REP-View, and store $Y(k - M_0) = Y(k|k - \text{Min}(M_0, M_1, \dots, M_a))$ for the next moment. Its schematic diagram is shown in Figure 12.

Object side

Moment	k-7	k-6	k-5	k-4	k-3	k-2	k-1	k
Prediction sequences	$Y(k-7)$	$Y(k-6)$	$Y(k-5)$	$Y(k-4)$	$Y(k-3)$	$Y(k-2)$	$Y(k-1)$	$Y(k)$
One way latency(step)	7	lost	4	3	1	2	1	2

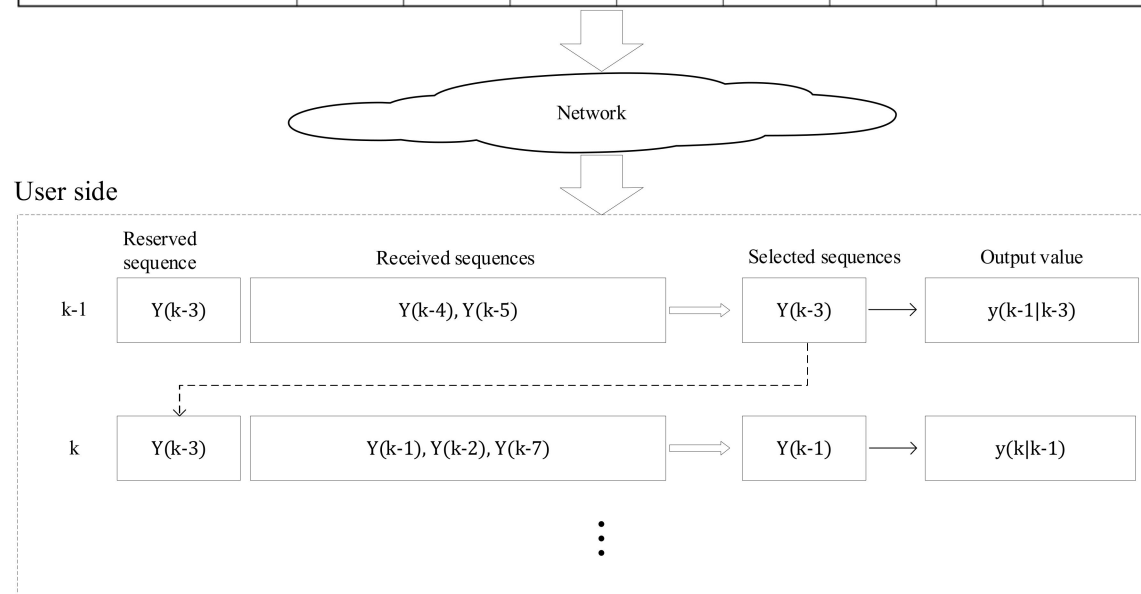


Figure 12. Scheme of the data selection at user side.

As a result, once Assumption 1 is satisfied, there is a certain value for the manipulator's pose at any moment. The movement of the virtual manipulator in the 3D scene was driven by the data selector's output value, providing a continuous and jitter-free virtual manipulator operation scene. Students could simultaneously observe the compensated virtual scene and the live scene sent back by the live camera. The former was consistent with the real manipulator in time, and there was a deviation in pose (depending on the model accuracy). The latter was a reflection of the real pose of the manipulator but suffered from image delay and freezing. The two compensated each other.

4. RCP Experimental Method and Case Study of PID Control

4.1. RCP Method Compared with Traditional Methods

In the traditional experimental platform, implementing a manipulator algorithm or application is relatively complex. First, you need to write the source code and integrate the code. Then, you need to write the driver code, compile the entire project, and finally, manually download the executable file to the target. If the experiment is not satisfactory, going back to the first step and re-adjusting the code is inevitable. In order to achieve the ideal experimental effect, not only is it time-consuming but the experimental period is also long. In terms of experiments, most of the traditional experiments are in the form of offline simulations; only a small number of experiments are carried out using physical manipulators. It is difficult to give students an intuitive experience of a manipulator by only using offline simulation, and only using a physical manipulator brings some uncontrollable problems. In terms of data monitoring, the monitoring interface of traditional experiments is limited. Sometimes it cannot meet the intuitive needs of monitoring, and the form of monitoring is single. Therefore, in the traditional experimental platform, the effect obtained in the experiments of the manipulator was not very satisfactory.

The experimental platform of this manipulator adopted the RCP to design the algorithm and application. The driver was encapsulated in the form of a module. It could be used by simply dragging in the connection. After completing the algorithm in Simulink, you only needed one click to generate code from the model and compile it to generate an executable file. Then, the file would be automatically /manually downloaded to the target. When you needed to modify the algorithm parameters or set values, you only needed to connect to the file running in the target through the external mode. You could realize the functions of online parameter adjustment and set the value. This allowed students to focus more on the algorithm level of the experiment, and the implementation of the algorithm could be completed through the module. This greatly realizes the learning cycle of the manipulator and improves the experimental efficiency. In terms of experiments, the use of digital twin technology can not only control the virtual and real manipulator independently but also realize the interaction between virtuality and reality. Achieving the effect of real control of the virtual and virtual control of the real. In terms of data monitoring, by uploading data to the OPC UA server, the monitoring interface could be customized for data monitoring to meet various monitoring needs. The comparison with the traditional experimental platform is shown in Figure 13.

4.2. PID Control Experiment with the Platform

In this part, an experimental case of this platform was introduced. Control of manipulator joints allowed the students to have a basic understanding of the entire platform. At the same time, the platform was generally explained, and the application of the PID algorithm in the joint control of the manipulator was introduced in detail. The control methods of the experiment can be divided into controlling the virtual manipulator only, controlling the real manipulator only, and controlling the virtual–real interaction of the manipulator. The last method was adopted here.

First of all, building a control model in Simulink and dragging and selecting the corresponding control object module was needed. After the model was finished, the build was clicked on to generate an executable file. Then, the file in the target was run. The model was controlled and the data monitored using the WEB monitoring interface.

This experiment adopted the closed-loop control method. As shown in Figure 14, in the case of only K_p , the desired position of the motor was set. After clicking download, the response curve followed the step curve, and there was almost no overshoot. However, there was a certain error from the expected value after the system was stable. Then, K_i was introduced, and the accumulated error was corrected for the system. It can be seen from Figure 15 that the response curve slightly oscillated around the expected value, and then tended to be stable. By modifying the parameters of the PID, the rotation effect of the joint motor of the manipulator could be controlled. When students carry out experiments,

they first adjust the proportional coefficient, then adjust the integral coefficient, and finally adjust the differential coefficient. If the proportional coefficient is too large, it causes the motor to rotate too fast and make the system unstable. After adjusting the proportional coefficient, there was a certain error between the actual position and the expected position. At this time, students could accumulate the error through the integral coefficient, and finally, reduce the error slowly to the allowable range.

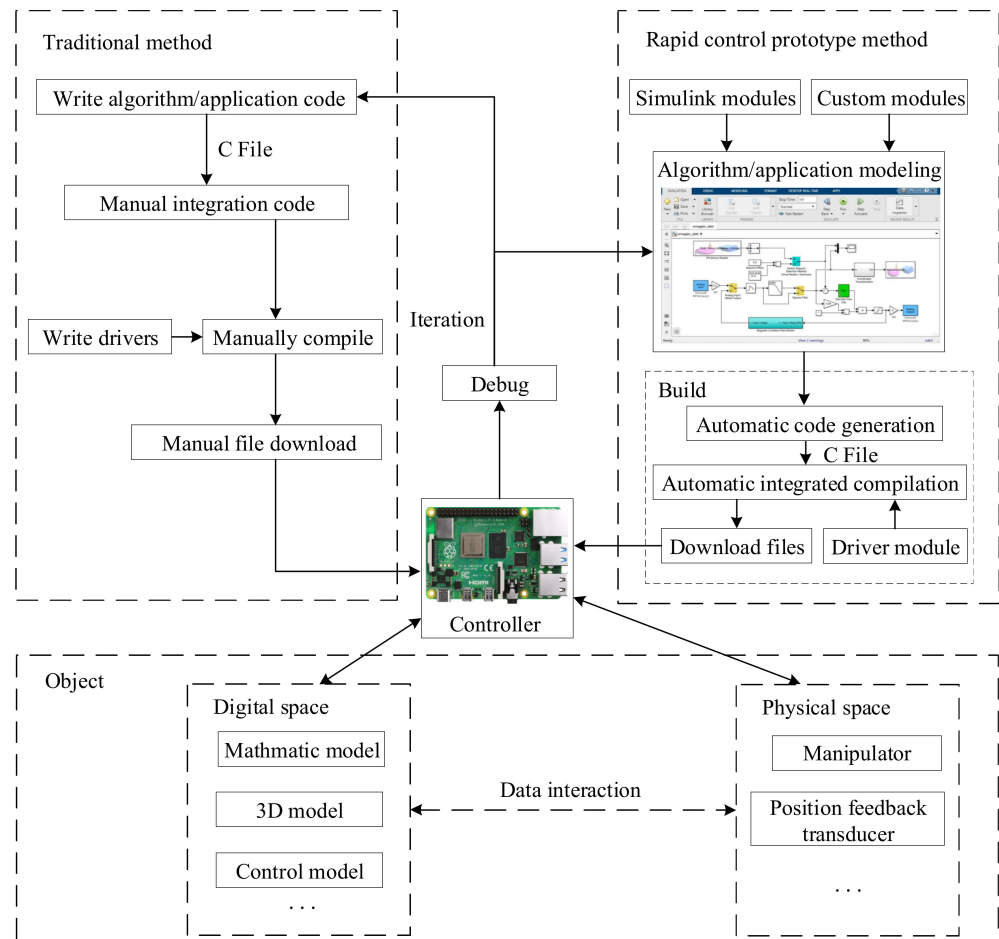


Figure 13. Rapid control prototype experiment compared with traditional methods.

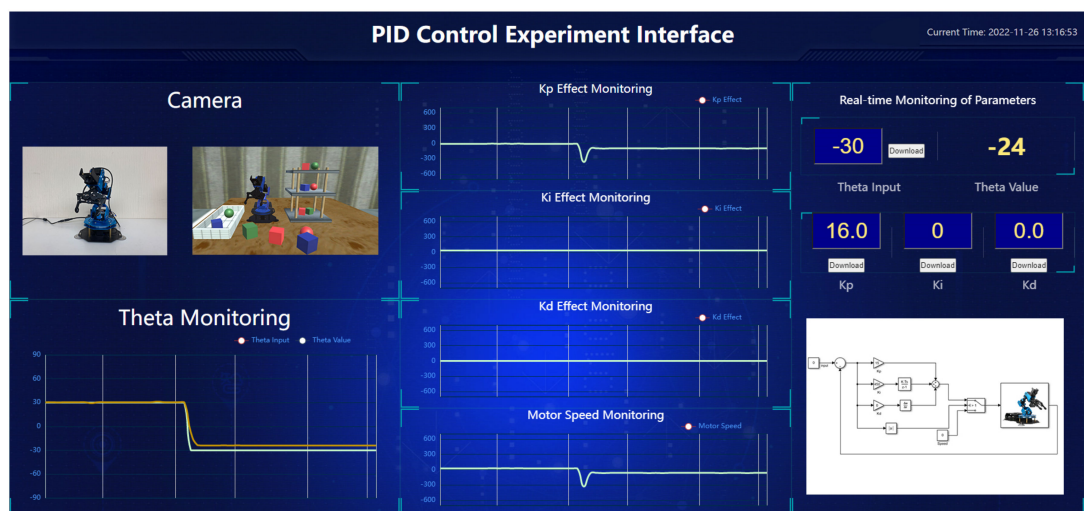


Figure 14. Experimental result with P controller.

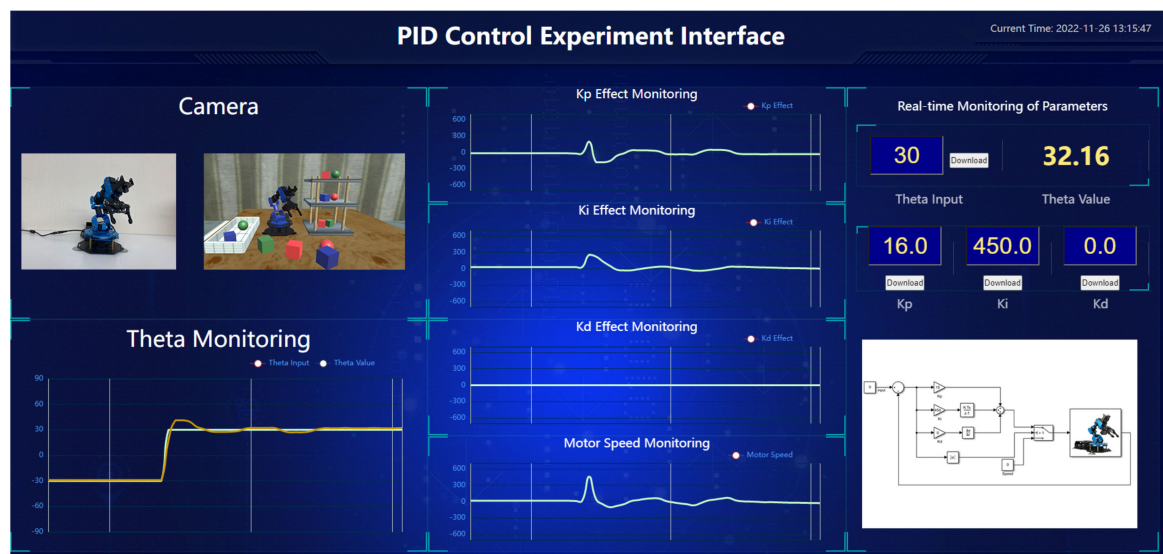


Figure 15. Experimental result with PI controller.

By learning the control method of this experiment, students can understand the whole process of control. Through the digital twin manipulator, the experimental effect is more intuitive. In addition, there are various manipulator algorithms that can be developed using this platform.

5. Conclusions

The REP based on RCP designed in this paper provides students with an experimental system that can quickly verify the manipulator algorithm. By using digital twin technology, the experimental results become more intuitive. The design of the experimental platform is not limited by time and space, and this solves the problems related to the solidification of teaching content in traditional teaching and the insufficient combination of theory and practice. These characteristics have played an important role during the COVID-19 online education period. The platform used Simulink under MATLAB to design the control algorithm of the manipulator and realized the hardware-in-the-loop simulation through RCP, using Unity3D to build virtual scenes, virtual manipulators, and more. Our experiments have shown that the system can provide students with a control strategy to quickly design and implement algorithms.

In terms of future research, the experimental platform will be continuously improved according to feedback. First of all, the objects used were relatively simple, and there were relatively few experiments carried out, which will continue to be enriched. Second, there was still a small difference between the current digital twin and the real manipulator, and the online algorithm will be considered to optimize the model parameters in the future.

Author Contributions: Conceptualization, Z.D.; Methodology, X.H. and Y.S.; Software, X.H. and Y.S.; Validation, Z.D. and X.H.; Writing—original draft, X.H. and W.Z.; Writing—review & editing, X.H. and S.L.; Supervision, Z.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (grant number 61873006).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Li, Z. The Reform and Innovation of Chemistry Classroom Teaching Mode in the Era of Big Data. In Proceedings of the 2017 9th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Changsha, China, 14–15 January 2017; pp. 417–421.
- Tang, C.; Lin, H.; Zhang, L.; Bao, T.; Zhang, Y.; Liu, Z. Electrical and Electronic Experiment Platform Teaching Method with Remote Control. In Proceedings of the 2020 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Macau, China, 21–24 August 2020; pp. 1–6.
- Wang, S.; Zhang, F.; Tang, Q.; Zhang, X.; Zhao, R. A Take-Home Motor Control Teaching Experiment Platform for Control Engineering-Related Courses. *IEEE Trans. Educ.* **2022**, *65*, 115–123. [\[CrossRef\]](#)
- Zhai, J.M.; Xu, X. Development of a web-based remote experiment system for engineering curricula. In Proceedings of the 2011 IEEE International Symposium on IT in Medicine and Education, Guangzhou, China, 9–11 December 2011; Volume 1, pp. 469–472.
- Li, G. Research on the Importance of the Basic Experiment Based on the Experimental Platform. In Proceedings of the 2016 International Conference on Robots & Intelligent System (ICRIS), Zhangjiajie, China, 27–28 August 2016; pp. 384–386.
- Megalingam, R.K.; Vivek, G.V.; Bandyopadhyay, S.; Rahi, M.J. Robotic arm design, development and control for agriculture applications. In Proceedings of the 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 January 2017; pp. 1–7.
- Lee, J.D.; Li, W.C.; Shen, J.H.; Chuang, C.W. Multi-robotic arms automated production line. In Proceedings of the 2018 4th International Conference on Control, Automation and Robotics (ICCAR), Auckland, New Zealand, 20–23 April 2018; pp. 26–30.
- Krasňanský, P.; Tóth, F.; Huertas, V.V.; Rohal’-Ilkiv, B. Basic laboratory experiments with an educational robotic arm. In Proceedings of the 2013 International Conference on Process Control (PC), Strbske Pleso, Slovakia, 18–21 June 2013; pp. 510–515.
- Bing, L.I.A.O.; Hongbin, Z.A.N.G.; Nana, Z.H.U.; Dongsheng, L.I.U.; Jiangmin, T.U.O.; Tao, H.E.; Lei, H.U.; Zheng, Y.A.N.G. System Design and Experiment of Bionics Robotic Arm with Humanoid Characteristics. In Proceedings of the 2018 WRC Symposium on Advanced Robotics and Automation (WRC SARA), Beijing, China, 16 August 2018; pp. 90–95.
- Jahnavi, K.; Sivraj, P. Teaching and learning robotic arm model. In Proceedings of the 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT), Kannur, Kerala, India, 6–7 July 2017; pp. 1570–1575.
- Rosen, R.; von Wichert, G.; Lo, G.; Bettenhausen, K.D. About The Importance of Autonomy and Digital Twins for the Future of Manufacturing. *IFAC PapersOnLine* **2015**, *48*, 567–572. [\[CrossRef\]](#)
- Wang, Z.; Gupta, R.; Han, K.; Wang, H.; Ganlath, A.; Ammar, N.; Tiwari, P. Mobility Digital Twin: Concept, Architecture, Case Study, and Future Challenges. *IEEE Internet Things J.* **2022**, *9*, 17452–17467. [\[CrossRef\]](#)
- Groshev, M.; Guimarães, C.; Martín-Pérez, J.; Oliva, A.D.L. Toward Intelligent Cyber-Physical Systems: Digital Twin Meets Artificial Intelligence. *IEEE Commun. Mag.* **2021**, *59*, 14–20. [\[CrossRef\]](#)
- Yin, Z.H.; Wang, L. Application and Development Prospect of Digital Twin Technology in Aerospace. *IFAC PapersOnLine* **2020**, *53*, 732–737. [\[CrossRef\]](#)
- Bratchikov, S.; Abdullin, A.; Demidova, G.L.; Lukichev, D.V. Development of Digital Twin for Robotic Arm. In Proceedings of the 2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC), Gliwice, Poland, 25–29 April 2021; pp. 717–723.
- Avila, E.A.; Chapa, D.P.; Arenas, I.D.; Hurtado, C.V. A Digital Twin implementation for Mobile and collaborative robot scenarios for teaching robotics based on Robot Operating System. In Proceedings of the 2022 IEEE Global Engineering Education Conference (EDUCON), Tunis, Tunisia, 28–31 March 2022; pp. 559–564.
- Hercog, D.; Curkovic, M.; Jezernik, K. DSP based rapid control prototyping systems for engineering education and research. In Proceedings of the 2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, Munich, Germany, 4–6 October 2006; pp. 2292–2297.
- Werth, W.; Faller, L.; Liechtenecker, H.; Ungermanns, C. Low Cost Rapid Control Prototyping—A useful method in Control Engineering Education. In Proceedings of the 2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, 28 September–2 October 2020; pp. 711–715.
- Dong, W.; Zhang, X.; Jing, L.; Wu, W. Design of Rapid-control-prototype Platform for Modular Multilevel Converter Based on RT-lab. In Proceedings of the 2019 IEEE 10th International Symposium on Power Electronics for Distributed Generation Systems (PEDG), Xi’an, China, 3–6 June 2019; pp. 94–98.
- Michael, T.; Reynolds, S.; Woolford, T. Designing a Generic, Software-Defined Multimode Radar Simulator for FPGAs Using Simulink® HDL Coder and Speedgoat Real-Time Hardware. In Proceedings of the 2018 International Conference on Radar (RADAR), Brisbane, Australia, 27–31 August 2018; pp. 1–4.
- Milam, W.P. A distributed computing approach to rapid prototyping for embedded controllers. In Proceedings of the 36th Midwest Symposium on Circuits and Systems, Detroit, MI, USA, 16–18 August 1993; Volume 2, pp. 1155–1158.

-
22. Zhong, Q.C.; Matthews, C.; Nguyen, P.L.; Clarke, S. Low-cost Rapid Control Prototyping paradigm. In Proceedings of the UKACC International Conference on Control 2010, Coventry, UK, 7–10 September 2010; pp. 1–5.
 23. Zhao, Y.; Liu, Z.; Yang, W.; Cai, L. Rapid Control Prototyping of an Automated Clutch Using dSPACE and Matlab/Simulink. In Proceedings of the 2010 International Conference on Computing, Control and Industrial Engineering, Wuhan, China, 5–6 June 2010; Volume 1, pp. 221–226.