Research Article

Haifeng Ma*, Anton Pljonkin, and Pradeep Kumar Singh

Design and implementation of Internet-of-Things software monitoring and early warning system based on nonlinear technology

https://doi.org/10.1515/nleng-2022-0036
received October 29, 2021; accepted May 31, 2022

Abstract: In order to realize and design a software monitoring and early warning system for the Internet of Things (IoT), this paper establishes a “trinity” control platform integrating PLC, WINCC, and MATLAB based on nonlinear technology and realizes the proportion integration differentiation (PID) control based on the RBF neural network tuning on this platform. Based on the framework of the trinity control platform, the PID control system set by the radial basis function (RBF) neural network and the STEP7 virtual object programming of the control platform are designed and realized. The experimental data update cycle is 0.5 s to record 1,000 data item objects, \( U \) is the control quantity, which is associated with the \( U \) communication driver variable in WINCC, and the corresponding storage address in the PLC is MD200; \( Yout \) is the controlled quantity, which is related to the \( Yout \) communication driver variable in WINCC, and the corresponding storage address in the PLC is MD100; start is the control switch, associated with the start communication driver variable in WINCC, corresponding to the storage address in the PLC of M0.3; reset is the reset control switch, It is associated with the reset communication driver variable in WINCC, and corresponds to the storage address in the PLC as M0.0. KP, KI, KD, and TIME correspond to three real-time PID parameters and are the cycle time in MATLAB (used for the X-axis of trend graphing), and are the variables of the communication driver. The addresses in the PLC are MD20, MD24, MD28, and MD32. It shows that for these three software programs, the update cycle of the data in the respective storage areas must be consistent, the program control cycles in MATLAB and PLC need to be consistent, and the transmission of parameters must be correctly implemented in a control cycle according to the programming logic sequence, in order to realize the design of an IoT software monitoring and early warning system.

Keywords: RBF neural network, OPC, WINCC

1 Introduction

With the popularization of computer networks and the development of information and intelligence, the development trend of automation software has become the key to integration and factory automation. Configuration software, also known as monitoring configuration software or industrial control configuration software, is accompanied by the emergence of the majority of production process automation technicians who are familiar with it. Due to its typical characteristics of data collection and output, data processing, data display, and preservation, it shows a broad application prospect in the process of industrial production control [1]. However, the current configuration software occupies the majority of the market share, and some small companies mainly develop Chinese products. Due to the limitations of human and material resources, there is a large gap between the key technologies and foreign products; hence, they can only be applied to very limited occasions [2]. Therefore, it is necessary to continue to analyze the general structures and development methods of industrial data acquisition and monitoring software and design a set of configuration tools to develop a reliable monitoring system quickly and stably. After entering user information, the main interface diagram of the monitoring software is entered, as shown in Figure 1. The main monitoring interface displays real-time data by default. In this monitoring interface, a function can be
chosen not only from the menu bar but also from the list of functions, which is convenient for user operation.

The proposal of nonlinear technology has completely changed the traditional development mode. As a nonlinear data client, the upper-layer application software does not contain any hardware communication interface program and does not care about the underlying hardware [3]. As long as the nonlinear data interface protocol is followed, data can be obtained from data servers provided by different hardware vendors, thus realizing the flexible configuration of data and real integration of various systems. In this paper, the precise control of nonlinear objects and the visual output of virtual objects are realized through precise time settings. Such a control platform connected by real-time communication is beneficial to designing and implementing of complex control algorithms and to building a control system with complex algorithms to adapt to the increasingly complex control objects and high control requirements.

The design ideas proposed in this paper have the advantages of openness and simplicity. The designed monitoring software can be put into practical application to monitor the data of PLC and other industrial field equipment. Data access with multiple field devices was successfully achieved in a unified manner, avoiding the reproducibility of driver development. Not only does it save costs but also it protects the hardware provider's technology. Practice has proved that this design idea is in line with the development direction of software, so it has broad application prospects.

2 Literature review

At present, according to the needs of different users in various fields, in addition to further improving the large-scale decentralized control system, most manufacturers have focused on and increased investment to improve the small-scale decentralized control system to meet the growing requirements of small and medium sized continuous or intermittent operation control [4]. Colombelli et al. argue that componentized software design is conducive to system integration, where components are applications running separately. Componentization allows users to freely choose the required component objects without purchasing the entire software system, which reduces the cost. At the same time, it allows users to add third-party applications to the system, which greatly improves the flexibility of software [5]. Yang et al. believed that to realize communication between upper computer software and lower computer, hardware developers should provide server software with interface so that any client program can access the server software with an interface. The development trend of software is openness, expansibility, networking, increasingly centralized management, and increasingly decentralized control [6]. The development of various technologies has provided a more effective means for the control software to be highly versatile, integrated with high-level management and control, and highly reliable. In the current research, this paper proposes to build a “trinity” control platform integrating PLC, WINCC and MATLAB based on nonlinear technology [7]. A PM sensing system consisting of off-the-shelf Rolola-based wireless hardware boards and low-cost PM sensors was introduced by Yun et al. By leveraging a software platform compatible with Internet-of-Things (IoT) standard called em2m, PM datasets can be collected and accessed in a standardized manner by transmitting programmable application interface via a representational state defined by em2m. Reliable point monitoring, a short-term (i.e., at 2 h) point prediction method based on an autoregressive integrated moving mean and vector autoregressive moving average (VARMA) model, is presented, and a 30-day evaluation point dataset collected from 15 LoRa-based point sensor nodes is installed in a university campus. Experimental results show that the overall RMSE error and correlation coefficients were improved by 7.77 and 3.7%, respectively, when compared to the single-node-based prediction model [8]. Zhang et al. have focused on the simulation technology of sensing and monitoring livestock and poultry breeding emissions, mainly carrying out the monitoring of the main components of livestock and poultry breeding pollutants based on the IoT, and using the foul-smelling gas air diffusion software “ModOdor” to simulate the diffusion of pollutants. The aim is to determine the diffusion characteristics of pollutants in typical farms and provide a decision reference for establishing an odor hygiene buffer zone and a minimum shelter distance to achieve the ecological safety objectives of the farms [9]. Network is used to connect many devices and communicate with each other, with a greater impact on animal health care systems. IoT devices appearing as wearable devices have been used to track human activity. Karthick et al.
used the integration of existing medical sensors to create a connected digital platform with enhanced connectivity with pets and livestock and improved efficiency. The range of biosensing, computing, communication, and wearable technologies available to the animals is described. The main objective is to review recent developments in the field of animal care, including livestock, farms, and wildlife. Intelligent technologies that can be used for different species of animals are reviewed. The results are expected to improve the future research and development of animal welfare systems [10].

3 Methods

The main function of the monitoring software is to monitor the real-time data and realize the query of the historical data. The OPC clients can access different OPC servers in a unified way to obtain the data from the field equipment and store these data in the real-time database, so as to conduct the real-time data display and real-time alarm function [11,12]. The data information and replacement situation at the last moment are stored in the historical database as the data source of historical reports, historical curves, and historical replacement inquiry.

3.1 Trinity control platform framework

The establishment of the trinity experimental control platform is mainly used for the optimization and self-tuning design and implementation of proportion integration differentiation (PID) control parameters of nonlinear objects using the radial basis function (RBF) neural network [13,14]. First, the trinity control platform MATLAB, as the OPC client, accesses the OPC server WINCC to obtain the field data and carries on the online adjustment of PID parameters and the calculation of the control quantity [15,16]. Finally, the calculated control quantity is reverentially transmitted to PLC, and the PLC output module gives the control quantity to the actuator to realize the whole control process, and its structure frame is shown in Figure 2. On the Trinity experimental platform, virtual experimental objects can also be used to complete the analog output of the actual control object. The simulation process is shown in Figure 3.

By this way, the simulation experiment of the whole actual control process can effectively prove the feasibility of the “trinity” control platform. The establishment of this platform maximizes the advantages of field equipment, monitoring configuration software, and control programming software [17,18]. The control system built on this platform can be very convenient to modify the control algorithm program, control fault diagnosis, and is very beneficial to the design and implementation of a complex algorithm control system.

3.2 PID control system based on RBF neural network tuning

For complex nonlinear control objects, a controller based on RBF neural network tuning PID parameters is designed.

![Figure 2: Trinity control platform structure diagram.](image1)

![Figure 3: Simulation process of the trinity control platform.](image2)
It mainly consists of two parts: (i) the classical PID controller, by adjusting the incremental PID parameter to calculate the control quantity, to achieve incremental closed-loop PID control of the controlled quantity [19,20]; (ii) the RBF neural network – RBF main basis is accused of output, through the study of the gradient descent method to calculate the hidden layer to output layer weights, center of hidden layer nodes and node width, identification of jacobian information of the object, and then according to the gradient descent method to calculate the incremental PID parameters, adjust PID controller parameters online, improve the control performance of the control system [21,22]. The RBF neural network is mainly used to identify jacobian matrix information. For the RBF neural network structure, after collecting a large number of samples and normalizing the samples, the network structure with the RBF neural network structure of 4 is selected through offline training. The three nodes in the input layer are \( x(1), x(2), \) and \( x(3) \), respectively, corresponding to the output of the KTH cycle, the output of the \( k - 1 \) cycle, and the control quantity of the KTH cycle. There are six nodes in the hidden layer and one node in the output layer. In the control system, the incremental PID control algorithm is adopted in the master controller: the control quantity \( U(K) \) is calculated according to the deviation as follows:

\[
U(K) = U(K - 1) + K_p \cdot e(K) + K_i \cdot e(K) + K_d \cdot e_d(K) \\
\]

\( e_d(K) = e(K) - e(K - 1) \),

\( e(K) = r(K) - y(K), e_d(K) = 2e(K - 1) + e(K - 2). \)

The auxiliary controller adopts the gradient descent method as follows:

\[
w(k) = w(k - 1) + \eta \left[ \frac{\partial f}{\partial w(k)} \right] + \delta[w(k - 1) - w(k - 2)],
\]

\[
b(k) = b(k - 1) + \eta \left[ \frac{\partial f}{\partial b(k)} \right] + \delta[b(k - 1) - b(k - 2)],
\]

\[
c(k) = c(k - 1) + \eta \left[ \frac{\partial f}{\partial c(k)} \right] + \delta[c(k - 1) - c(k - 2)].
\]

By learning the structural parameters of the RBF network, PID parameters can be adjusted by using the Jacobi information of the controlled object identified by the RBF neural network.

### 3.3 Control platform STEP7 virtual object program design and implementation

In the actual engineering control of the platform, the PLC completes the sampling of the field data and stores the sampled data into the input impression storage area through the input module, and then the data exchange is realized through the CONNECTION with WINCC through the MPI communication protocol [23,24]. MATLAB uses OPC as the client to access the WINCC server and read the controlled data collected by the PLC LAB control quantity calculation, and then control quantity reverse transfer to PLC, through the output module will control quantity to the actuator, complete the whole control. The virtual control object will discretize the process control object model. Firstly, the object function of discrete model is written by PLC; Then calculate the control quantity and controlled quantity based on MATLAB, and simulate the process that the control is transferred to the actuator; Transfer the control quantity data to WinCC through MPI, and then access the WinCC server through MATLAB to read the control quantity data. According to the data, through the RBF neural network tuning the PID parameter control algorithm program, calculate the next cycle control quantity, reverse transfer to PLC, calculate the next output, realize the whole communication control process, complete the whole simulation control experiment [25,26]. The platform uses CPU314 and OB35 periodic interrupt organization block. In the hardware configuration of 300 sites, THE CPU attribute setting is opened and the cycle interrupt time of OB35 periodic interrupt organization block was set to 500 ms, which is the same as the control cycle time in WINCC and MATLAB. The OB35 program mainly calls subroutine FBJ to realize the periodic interrupt loop. Therefore, before programming OB35, FBJ needs to be programmed first. Before FBJ programming, variables are defined first. After the declaration of input and output variables, they are automatically generated in background data blocks. After the variable declaration is completed, the program is written for FBJ [27,28]. FBJ mainly realizes the calculation of the discrete mathematical model function of the controlled quantity, and the discrete model function of the controlled object is shown as follows:

\[
y(k) = \frac{u(k - 1) - 0.1 \cdot y(k - 1)}{1 + y(k - 1)^2}.
\]

After the functional block FBJ is programmed, OB35 is programmed and subfunction FB1 is called, and then
the corresponding configuration is carried out. Among them, the control quantity $U$ is stored in the present storage area MD200, the controlled quantity Yout is stored in the present storage area MD100, and the simulation switch is controlled by the bit storage M1.0.

4 Results and analysis

4.1 MATLAB program design of control platform

In the communication between MATLAB and WINCC based on OPC technology, WINCC acts as the server and MATLAB as the client, and MATLAB needs to write the relevant client program to access the server and read and write the data of the server [29, 30]. On the premise of activating the WINCC screen, there are two ways to realize the communication between the two. The first is to use the MATLAB programming language to write the client program, and the other is the OPC toolbox provided by MATLAB, which can realize the visual client program programming. The system uses MATLAB programming language to write OPC client program to access the OPC server, and the specific program to establish communication connection is shown in Table 1.

Table 1: OPC client programs accessing the OPC server

<table>
<thead>
<tr>
<th>Start</th>
<th>Additem</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U$</td>
<td>grp.$U$</td>
</tr>
<tr>
<td>Yout</td>
<td>grp.Yout</td>
</tr>
<tr>
<td>KP</td>
<td>grp(KP)</td>
</tr>
<tr>
<td>KI</td>
<td>grp(KI)</td>
</tr>
<tr>
<td>KD</td>
<td>grp(KD)</td>
</tr>
<tr>
<td>KP_0</td>
<td>grp(KP_0)</td>
</tr>
<tr>
<td>KI_0</td>
<td>grp(KI_0)</td>
</tr>
<tr>
<td>KD_0</td>
<td>grp(KD_0)</td>
</tr>
<tr>
<td>RIN</td>
<td>grp.RIN</td>
</tr>
<tr>
<td>TS</td>
<td>grp.Ts</td>
</tr>
<tr>
<td>TIME</td>
<td>grp.TIME</td>
</tr>
<tr>
<td>$m$</td>
<td>grp.$m$</td>
</tr>
</tbody>
</table>

Record time when data change occurs once when data changes once set (1,000); The data update period is set as 0.5 s, and 1,000 data item objects are recorded. $U$ is the control quantity, which is associated with the $U$ communication driver variable in WINCC and corresponds to the storage address MD200 in PLC. Yout is the controlled quantity, which is associated with the VARIABLE of Yout communication driver in WINCC and corresponds to the storage address of MD100 in PLC. Start is the control switch, associated with the WINCC start communication driver variable, corresponding to the PLC storage address M0.1; Reset is the reset control switch, associated with the Reset communication driver variable in WINCC, corresponding to the STORAGE address M0.0 in PLC. KP, KI, KD, and TIME respectively, correspond to three real-time PID parameters and are the cycle TIME in MATLAB (used for the X-axis of trend drawing), are the communication driver variables, and the addresses in PLC are MD20, MD24, MD28, and MD32. KP_0, KI_0, and KD_0, respectively, correspond to three initial PID parameters, TS is the sampling time, $m$ is the control quantity selected by conventional PID and RBF neural network PID, and these variables are associated with internal variables of the same name in WINCC. In addition, because the final control and monitoring process is implemented on the upper computer WINCC, there is no screen switch in the control process, that is, all control operations can be carried out only in the WINCC running screen. Due to the use of three software, to achieve this function, it is necessary to ensure that WINCC click on the start button, PLC and MATLAB to automatically run. For PLC, it only needs to open its simulation module, run the CPU in advance, put the CPU in the RUN state, and use the control button of M1.0 to associate with WINCC through the MPI protocol to realize program running [31].

4.2 Step7–300 time sequence setting of communication connection of WINCC and MATLAB

The control platform adopts STEP7300, WINCC, and MATLAB to realize PID control based on RBF online tuning parameters. In order to achieve correct and effective control, it is necessary to ensure the correct communication of data connection between the three software. Therefore, for these three software, the update cycle of the data in the respective storage area must be consistent, MATLAB and PLC in the program control cycle need to be consistent, and in a control cycle, the transmission of parameters must be correct according to the programming logic order. Because the controller adopts an incremental algorithm, if the above three requirements cannot be guaranteed, it is easy to have delayed parameter updates, wrong and chaotic data transmission control cycle, and wrong control cycle. After each control cycle, the control value will be wrong due to the error of the control quantity, which will
lead to the failure of the whole control process. Therefore, ensuring the consistency of data update cycle and the correctness of data transfer logic sequence within the control cycle is the key to realize the online PID parameter tuning in this example. In this example, the data update cycle time is 500 ms. Therefore, the same update cycle time should be set for all the data that need to be exchanged and shared through communication in PLC, WINCC, and MATLAB. During WINCC screen configuration, the update cycle of all communication driver variables is set to 500 ms. In MATLAB, the program statement set 0.5 is used to set the data update time to 0.5 s, that is, 500 ms, so as to ensure the consistency of the data update cycle. In STEP7300CPU, set the interrupt time of OB35 cycle interrupt organization block to 500 ms, that is, let each control cycle complete within 500 ms. Correspondingly, two pause statements are used in the control cycle (i.e., for statement loop) in MATLAB, so that the time of each cycle is 0.5 s, that is, 500 ms, ensuring the consistency of the control cycle. For the logic order of data transfer, mainly MATLAB and PLC data transfer logic order problem, the control logic order of the program is as follows: PLC first calculates the value of the controlled quantity, through the “trinity” control platform communication mechanism; Set PID parameters through MATLAB, calculate the control quantity, and then transfer the control quantity back to PLC to calculate the control value of the next control cycle. Repeat the above control process. The logical sequence transfer diagram is shown in Figure 4.

The sampling time is 0.001 s; the initialization parameters are set as follows: the initial value of the implied layer center vector is 30; the width vector is 40; the weight vector of the hidden layer to the output layer is 10; the RBFNN learning rate is 0.25; the momentum factor is 0.05; and the PID parameter learning rate is 0.2. The experimental results are shown in Figure 5.

As can be seen from Figures 5 and 6, the system adjustment time is short and oversized, and the PID parameters can converge to the optimum quickly. The system platform realizes the communication connection of the three software, and the trend chart of the accused quantity output reproduces the original operation results in the MATLAB. The test results show that the communication of each control cycle is normal, with no control timing disorder, and the PID control based on the RBF neural network rectification is well realized. The success of this experiment proves that it is possible to write complex control algorithm programs using MATLAB instead of PLC for the practical control process, which is very

![Figure 4: Logical sequence of data transfer between MATLAB and Step 7-300.](attachment:image4)

![Figure 5: The PID control results of the RBF neural network under the “Trinity” control platform.](attachment:image5)
5 Conclusion

Based on nonlinear technology, this paper builds a "trinity" control platform integrating PLC, WINCC, and MATLAB and implements PID control based on RBF neural network tuning on this platform. Based on the framework of the trinity control platform, a PID control system set by the RBF neural network and the STEP7 virtual object programming of the control platform is designed and realized. According to the experimental results, the data update period is 0.5 s to record 1,000 data items, $U$ is the control quantity, which is associated with the $U$ communication driver variable in WINCC, and the corresponding storage address in the PLC is MD200; $Y_{out}$ is the controlled quantity, which is related to $Y_{out}$ communication driver variable in WINCC, and the corresponding storage address in the PLC is MD100; start is the control switch, which is associated with the start communication driver variable in WINCC, and corresponds to the storage address in the PLC as MD0.1; reset is the reset control switch, which is associated with the reset communication driver variable in WINCC, and corresponds to the storage address in the PLC of MD0.0. KP, KI, KD, and TIME correspond to three real-time PID parameters respectively, and are the cycle time in MATLAB (used for the X-axis of trend graphing), and are the variables of the communication driver. The addresses in the PLC are MD20, MD24, MD28, and MD32. It shows that for these three softwares, the update cycle of the data in the respective storage areas must be consistent, the program control cycles in MATLAB and PLC need to be consistent, and the transmission of parameters must be correctly implemented in a control cycle according to the programming logic sequence to realize the design of IoT software monitoring and early warning system. Non-linear technology is an emerging standard in the field of industrial control. The specification separates the hardware supplier and the application software developer, which greatly improves the work efficiency of both parties. Software developers do not need to understand the essence and operation process of hardware, and they can access the data in the server as long as they follow the specifications of development. It is very suitable for process control, and developers can write software programs in high-level languages, which greatly simplifies the complex process of data transmission from equipment in the past. Nonlinear technology is in the stage of continuous development; it has strong application potential, and has achieved success in many fields. It provides effective solutions and tools for building open automation systems and promotes the development of industrial control systems in the direction.

Figure 6: Routine PID control results under the "Trinity control platform."
of openness. Of course, nonlinear technology will make improvements in the realization of unified management of plug-and-play functions. At the same time, nonlinearity will integrate more new technologies, the application field will be further expanded, and there will be more extensive application prospects.

**Funding information:** The authors state no funding involved.

**Author contributions:** All authors have accepted responsibility for the entire content of this manuscript and approved its submission.

**Conflict of interest:** The authors state no conflict of interest.

### References


