

# The Design of Multi-Motor Synchronous Monitoring System Based on IOT

Li Hui<sup>1,2,\*</sup>, Li Jing<sup>1</sup>, Liu Xing-qiao<sup>2</sup>

<sup>1</sup>Faculty of Automation, Huaiyin Institute of Technology, Huaian, China

<sup>2</sup>School of Electrical and Information Engineering, Jiangsu University, Zhenjiang, China

## Email address:

13645234923@163.com (Li Hui), 15996199342@163.com (Li Jing)

\*Corresponding author

## To cite this article:

Li Hui, Li Jing, Liu Xing-qiao. The Design of Multi-Motor Synchronous Monitoring System Based on IOT. *Advances in Wireless Communications and Networks*. Vol. 4, No. 1, 2018, pp. 11-16. doi: 10.11648/j.awcn.20180401.13

**Received:** July 19, 2018; **Accepted:** August 17, 2018; **Published:** September 12, 2018

---

**Abstract:** In order to improve the quality of network monitoring, the monitoring design of the multi-motor synchronization control system based on IOT was present. In the paper IOT technology was adopted. The server part was completed on PC. And the client system was implemented on Android smart phone platform. And the motor control parameters were saved in text mode. Various parameters of multi motor synchronous system were monitored in text and graphics on Android smart phone. The parameter changes of the master motor speed and the tension were monitored on-line and shown in data curve mode. Users could browse in real time by mobile phone. Users could use manual / automatic two control methods to implement real-time control on the system. The communication parameters were stored in SQLite database. Socket network communication based on message notification was used in the system. More and more experiments showed that: The monitoring system was cost-effective, and good real-time performance. And the same time the system was quickly, low demand on the network, and easy to use.

**Keywords:** Android Smart Mobile, Monitoring, Motor Control, Database

---

## 1. Introduction

With the development of modern industrialization, the synchronous control of multi-Motor has been applied more and more widely, such as in textile, paper making, printing and dyeing industries [1-10]. With the development of the Internet of things technology, the intelligent monitoring of the water line was becoming more and more important. The production managers need to be able to master the operating conditions of the production line urgently. At present, the technology of the Internet of things in China started late, but the development momentum is very strong, and the synchronous monitoring of multi-motor based on Android smart phone has not yet been reported. The literature (11) proposed the development of Android smart phone in aquaculture, and Literature (12) proposed the application of Android smart phone in agriculture. The above literature showed the received data in text mode or direct video reception. Based on the experience of predecessors, the design scheme of synchronous monitoring was adopted. The receiving data was displayed in

graphic and textual way, and the operation is humanized. It provides first-hand information for the working status of multi motor production lines, and can directly control the operation of the production line after password verification.

## 2. Overall System Structure

The system consisted of several modules, such as the underlying control module, the local server, and the remote mobile phone user client. The system structure diagram was shown in Figure 1. the master motor speed was collected in real time through the incremental circular encoder by CPLD. The tension sensor combined with the transmitter to collect and process two sets of tension data for DSP processing. The DSP and CPLD communicated by the data bus directly. DSP and PC server communicated by RS232 serial connection. The data interaction between three inverters and three asynchronous motors was finished by RS485 bus built by state machine in CPLD. The optimized auto disturbance rejection control algorithm was used in the underlying control part.

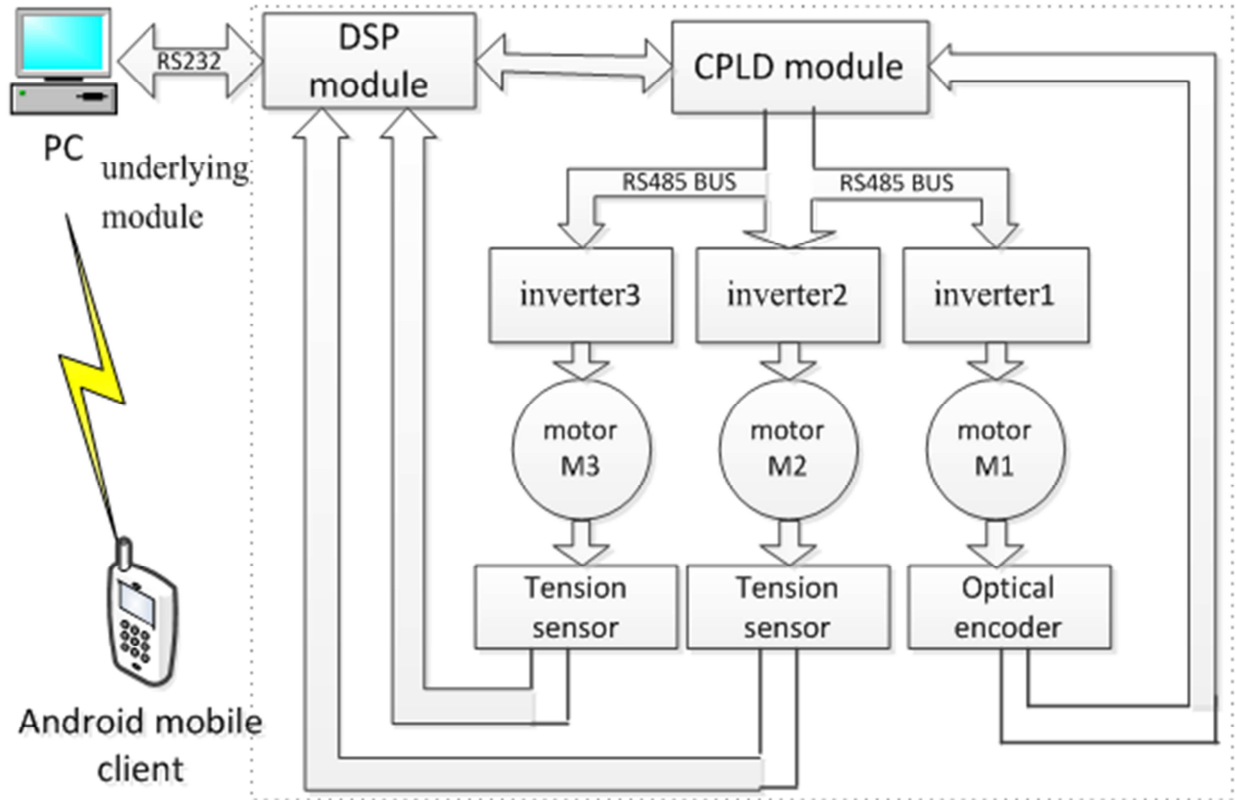


Figure 1. Whole framework of system.

### 3. Multi-motor Synchronization System Introduction

The control object of this subject was three asynchronous motors driven by three inverters. And the belts were used for hardware connection between the two motors. The specific physical diagram was shown in Figure 2.



Figure 2. Three-motor synchronous system physical map.

According to the literature (5), the mathematical model of the AC three-motor synchronous system in the d, q synchronous rotating coordinate system using rotor field orientation control could be obtained in formula(1).

$$\begin{cases} \dot{\omega}_1 = \frac{n_{p1}}{J_1} \left[ (\omega_1 - \omega_{r1}) \frac{n_{p1} T_{r1}}{L_{r1}} \Psi_{r1}^2 - (T_{L1} + r_1 F_{12}) \right] \\ \dot{\omega}_2 = \frac{n_{p2}}{J_2} \left[ (\omega_2 - \omega_{r2}) \frac{n_{p2} T_{r2}}{L_{r2}} \Psi_{r2}^2 - (T_{L2} + r_2 F_{23} - r_2 F_{12}) \right] \\ \dot{\omega}_3 = \frac{n_{p3}}{J_3} \left[ (\omega_3 - \omega_{r3}) \frac{n_{p3} T_{r3}}{L_{r3}} \Psi_{r3}^2 - (T_{L3} - r_3 F_{23}) \right] \\ \dot{F}_{12} = \frac{K_1}{T_1} \left( \frac{1}{n_{p1}} r_1 k_1 \omega_1 - \frac{1}{n_{p2}} r_2 k_2 \omega_2 \right) - \frac{F_{12}}{T_1} \\ \dot{F}_{23} = \frac{K_2}{T_2} \left( \frac{1}{n_{p2}} r_2 k_2 \omega_2 - \frac{1}{n_{p3}} r_3 k_3 \omega_3 \right) - \frac{F_{23}}{T_2} \end{cases} \quad (1)$$

Among them:  $F_{12}$  was the tension between No. 1 motor and No. 2 motor,  $F_{23}$  was the tension between No. 2 motor and No. 3 motor,  $\omega_1, \omega_2, \omega_3$  were the synchronous angular velocity of the three motors;  $\omega_{r1}, \omega_{r2}, \omega_{r3}$  were the rotor angular velocity;  $\Psi_{r1}, \Psi_{r2}, \Psi_{r3}$  were rotor flux linkages;  $J_1, J_2, J_3$  were moments of inertia;  $T_{L1}, T_{L2}, T_{L3}$  were load torques;  $T_{r1}, T_{r2}, T_{r3}$  were motor time constants;  $L_{r1}, L_{r2}, L_{r3}$  were rotor inductances;  $L_{m1}, L_{m2}, L_{m3}$  were rotor and stator mutual inductance; and  $n_{p1}, n_{p2}, n_{p3}$  were pole pairs.

### 4. Android Mobile Phone Monitoring Platform Design

Android is a newly developed open source mobile phone operating system based on Linux platform in recent years[14].

Android's architecture is divided into three parts: the underlying Linux system kernel, the middle management layer and the application layer [15]. This design uses JAVA as the development language. The Lenovo A288t smartphone was used as test machine with Android 2.3.5 version and kernel 2.6.35.7. The system is developed under the Android SDK+JAVA JDK6+Eclipse3.5 environment.

#### 4.1. Android Monitoring System Process Design

After the system was powered on, it entered the main control interface which showed many monitoring parameters

of the master motor speed and tension  $F_{12}$  and tension  $F_{23}$ . In this control interface, the motor running parameter could be controlled, the initialization of communication port could be finished. In the master motor speed and tension parameter monitoring interface, the network was required to be unblocked; in the motor operation parameter control interface, the control security was considered, and the security key was required. The communication IP address and the port number were set in the communication port initialization interface. The specific design process was shown in Figure 3.

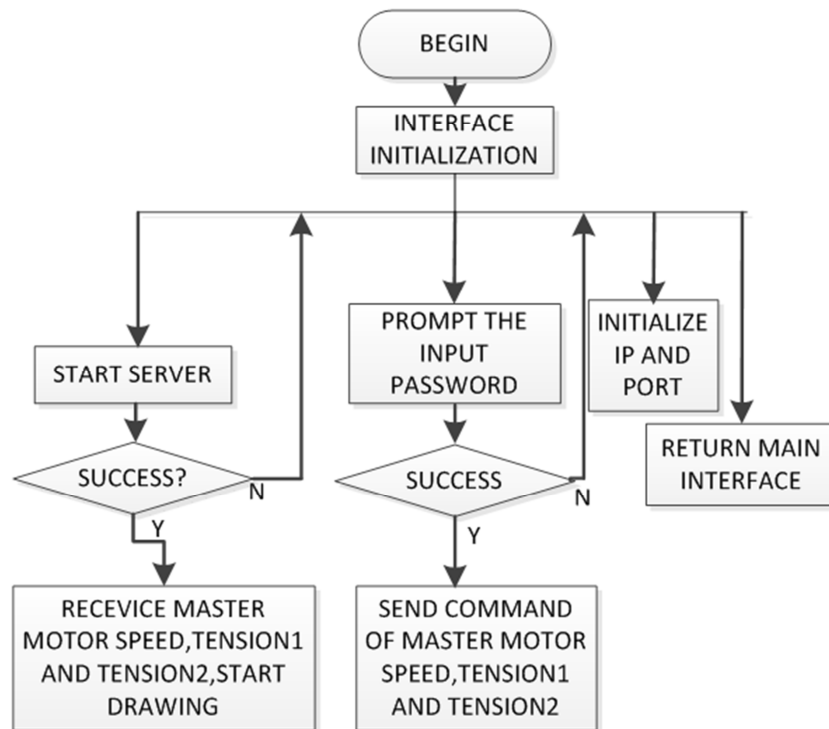


Figure 3. Android flow chart of the system.

#### 4.2. Database Module Design

The SQLite database that comes with Android was used. SQLite was a very popular embedded database, which supported SQL language, and had high execution efficiency. It was open source and could be used by any developer. SQLite consisted of several components, including the SQL compiler, kernel, backend, and attachments. SQLite adopted virtual machines and virtual database engines (VDBE) to make it easier to debug, modify, and extend the kernel of SQLite. The biggest difference between SQLite and other databases was the support of data types. When creating a table, you could specify the data type of a column in the CREATE TABLE statement in advance, but if the data type of a column did not match the pre-specified data type, it would Try to automatically convert the data type to system-pre-specified data type. If it could not be converted, the value would be stored as its own type. If you used SQLite in an Android application, you must create your own database firstly, and then created tables, indexes, and populated the data. A

database was mainly designed to store communication IP addresses and communication ports in this design. Firstly, the communication parameter database ip\_config could be created by SQLiteOpenHelper in the onCreate event, and then the tables could be created. The command named creat table could be executed by calling the execSQL() method of SQLiteDatabase. The creation table was shown in Table 1. Then the execSQL () method could be used to execute INSERT, UPDATE, DELETE to update the table data.

Table 1. Login User Data Table.

Field Name	Type of data
User	Varchar
Key	Varchar

After the database statement was executed, the Close method was called to release the connection to the database.

#### 4.3. Dynamic Curve Generation Module Design

The dynamic curve was implemented by the AChartEngine

API. The AChartEngine API mainly used the data set to implement dynamic drawing. First initialized the drawing and associated the canvas. Then added a graph to the canvas, created a dynamic dataset with the buildDataset function, and created a chart chart from the dynamic dataset. In the chart initialization, it was mainly to complete the setting of the chart style, coordinate axis scale, color and other information. The specific design procedure was as follows:

```
layoutGraph=(LinearLayout)findViewById(R.id.pic);
//Associate canvas
Context = getApplicationContext();
// Create a dataset that will be used to create the map table
mDataset = buildDataset(titles, x, y);
Int[] colors = new int[] { Color.BLUE, Color.BLACK,
Color.RED};
// Set the color of the three curves: blue, black, //red
PointStyle[] styles = new PointStyle[] {PointStyle.CIRCLE,
PointStyle.DIAMOND, PointStyle.SQUARE };
// Set the style of the three curves
Renderer = buildRenderer(colors, styles, true);
// Create a renderer
// For the rendering of the chart, see the //related api
documentation.
setChartSettings(renderer, "Multi-motor Synchronous
System Status Display", "Time/s", "", 0, 2000, 0, 500,
Color.GRAY, Color.RED);
Chart=ChartFactory.getLineChartView(context, mDataset,
renderer);
// Set the style of the chart, the scale of the x, y axis, the xy
axis color, the xy label and the title color
layoutGraph.addView(chart);
// Generate a chart and add a layout box to //display
The dynamic drawing of the data curve was completed in
the receiving program. The specific design scheme is as
follows.
```

After receiving the data, the communication program set the values of the horizontal and vertical coordinates to form a two-dimensional data, added and updated to the dynamic data set, and called the invalidate graphic function of the chart to display the dynamic association.

#### 4.4. Data Communication Module Design

The data communication part was mainly implemented by socket communication. This paper mainly included three parts: initial communication connection module, parameter transmission module and online data receiving module.

In the initialization communication connection module, the IP address and the communication port in the communication parameter database were first called, and then a new socket function connection was created according to the communication parameters, the receiving thread was started, and the data of the server was waiting to be received in the thread.

In the parameter transmission module, the master motor speed, tension 1 and tension 2 are mainly transmitted. Firstly, the identity of the user was verified. After the identity was obtained, the user entered the sending page. After the

parameter input was completed, the data to be sent was sent to the data buffer, and the data was encoded and converted, and the parameter command was sent in the manner of data flow.

The receiving module mainly received the master motor speed in real time, the tension 1 and the tension 2 between the two motors. The message mechanism was used to complete the online reception of the data in the thread. After receiving the data, the split function was used to segment the data. The communication format between the Android mobile client and the server is: AA: master motor speed, tension 1, tension 2: BB. (starting with AA, ending with BB).

#### 4.5. Text Online Storage Module Design

Monitoring window started, connected to the server, obtained the system time, took the system time (accurate to the second) as the main element, used the FileOutputStream function to create a new text file in the user SD in file stream mode, automatically opened, and obtained online data, used write combination the flush function for writing online data to a text file. When the user exited the watch window, the connection to the server was automatically cut off, the text file was saved, and the resource was released.

## 5. Server Part Program Design

The PC server part mainly used DELPHI language combined with SOCKET communication and RS232 serial communication to complete the design. RS232 serial communication between the server and DSP, used MSCOMM control to complete the design. The data obtained from the lower computer would be stored into the SQL database by the server and simultaneously displayed online. And then socket network communication was started. The data was sent to the Android client. The receiving of the control command of the client and the sending of the control command to the underlying computer were also completed in the server program. Considering that this article was based on Android mobile client design, the specific design of the server side will not be described here.

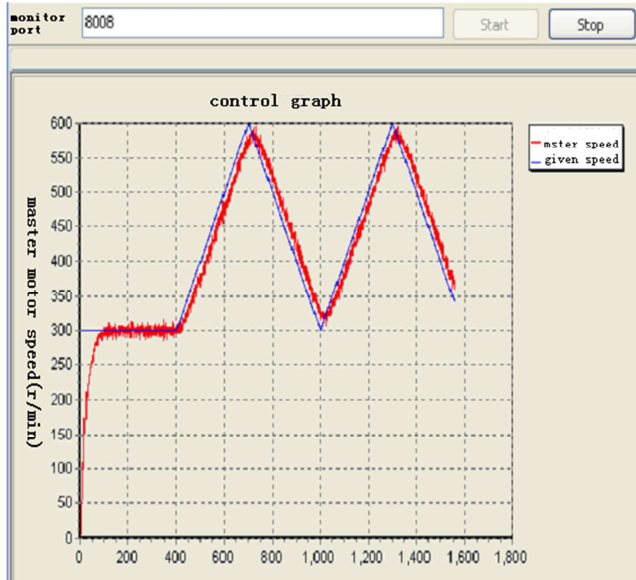
## 6. Experiment and Analysis

Three three-phase AC asynchronous motors were used as the control object. The rated speed of the motor was 1470r/min, the inverter was Siemens inverter, the processor selected DSP2812 microprocessor, CPLD selected MAXII chip EPM1270TC5, and the speed sensor adopted incremental circular grating. The encoder and tension sensor used the SL-100 sensor. Took the traditional PID control strategy as an example to test the quality of the online waveform of the mobile client.

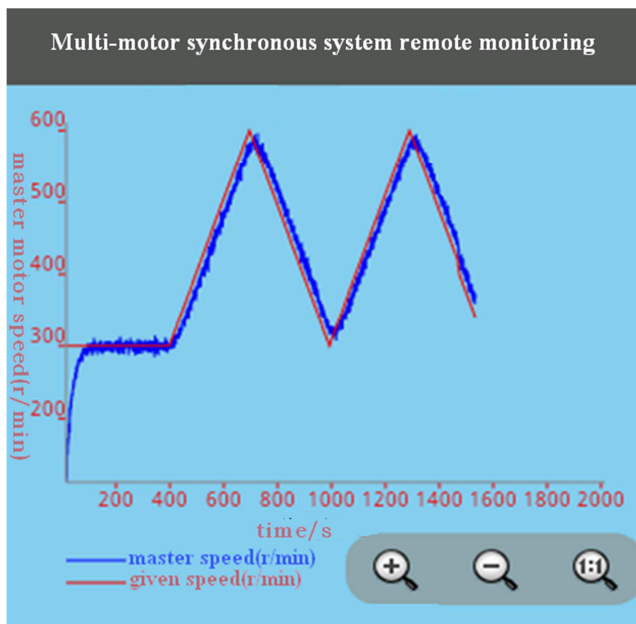
### 6.1. PID Tracking Performance Test

The reference input was run at 300r/min for 400 seconds after the initial start, and started with a triangular wave input with a peak value of 600r/min and a period of 600 seconds.

The master motor started the tracking reference signal and adopted the PID control mode. After the DSP collected the master motor speed signal, it sent it to the server through RS232. After receiving the message, the server uploaded it to the mobile phone client. The PC server and the Android mobile phone client received the graph as shown in Figure 4(a), (b).



(a) the data graph from PC Sever



(b) The data curve from Android mobile client

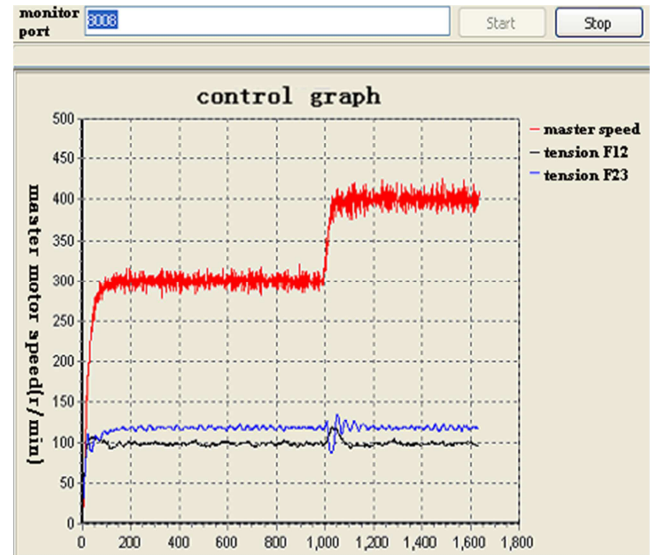
**Figure 4.** The curve of tracking triangle wave.

Through this test, the real-time tracking status under the PID control strategy could be obtained online, and remote control could be implemented if necessary.

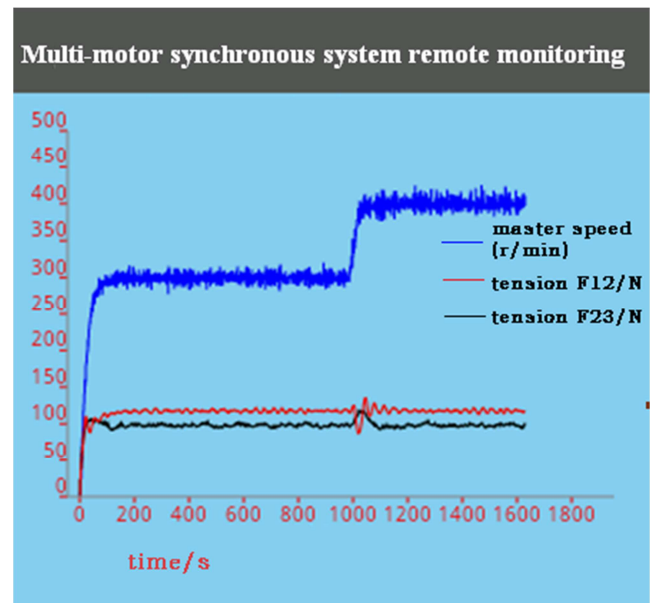
## 6.2. PID Decoupling Performance Test

The main motor initially ran at a constant speed of 300r/min,

and the speed suddenly increased to 400r/min at 1000 seconds. The effect of sudden speed changes on the tension between the two motors was investigated. The curves sent by the server and the specific graphs received from the Android mobile client were shown in Figure 5(a) and (b), respectively.



(a). The data curve from PC Sever



(b). The data curve from Android mobile client

**Figure 5.** The curve of PID decoupling test.

Through this test, the decoupling between the master motor speed and the two-two tension under the PID control strategy could be obtained in real time and remotely controlled if necessary. Based on the above two tests, it could be known that the Android mobile phone client could completely reflected the motor waveform of the PC server on the line, and the waveforms of the mobile client and the server were basically the same, the real-time performance was good, and the communication efficiency was high.

### 6.3. Communication Packet Loss Rate Test

Socket communication often had the problem of packet loss. The main reason was usually that the data was sent too fast, the amount of data sent was large, and the buffer setting was unreasonable. In this paper, the transmission data was truncated within the range of accuracy requirements. The master motor speed was accurate to 0.001. The tension was accurate to 0.01. And the buffer is set to 512. The packet loss test was performed on the system in terms of the timing interval. After the Android mobile phone client and the PC server were connected, the PC server sent 100 sets of master motor speed, tension 1 data, and tension 2 data at regular intervals. The specific test results were shown in Table 2.

**Table 2.** Data transmission interval test.

Timing interval (milliseconds)	Packet loss rate
50	75%
100	58%
200	28%
300	0%
500	0%

As can be seen from Table 2, the server sent data too fast and was prone to packet loss. When the timing was 300 milliseconds, the packet loss rate was 0%. Therefore, the time interval used for the PC server sent data to the client were controlled more than 300 milliseconds.

## 7. Conclusion

Through the multiple performance test of the monitoring system, the system could obtain the motor working curve remotely and in real time online, understood the working condition of the motor. The tracking and decoupling performance of the system could be shown on the mobile phone client and the PC server in real time. The packet loss rate of the system could reach 0%. The system ran stably, had low network requirements, was not limited by time and region, and was flexible and convenient to operate. The system could be operated on Android mobile. The system was cost-effective, and good real-time performance. Finally, this paper provided a new solution for remote monitoring of multi-motor synchronous control, with strong practicality and promotion value.

## References

- [1] FENG Jian-xiu, FENG Jing-yi. Control Strategy and Realization for Multi-motor Synchronous Transmission System, 2012, 42(7):64 ~68.
- [2] Chen Z, He J, Zheng Y, et al. An optimized feed forward decoupling PD register control method of roll-to-roll web printing systems [J]. IEEE Transactions on Automation Science and Engineering, 2016, 13(1):274-283.
- [3] Qing Zheng, Zhiqiang Gao. An energy saving, factory-validated disturbance decoupling control design for extrusion processes[C]. Intelligent Control and Automation (WCICA), 2012: 2891-2896.
- [4] Pagilla P R, Siraskar N B, Dwivedula R V. Decentralized control of web processing lines [J]. IEEE Transactions on Control Systems Technology, 2007, 15(1):106-117.
- [5] Wang Qinglong, Zhang Xing, Zhang Chongwei. Double sliding-mode model reference adaptive system speed identification for vector control of permanent magnet synchronous motors [J]. Proceedings of the CSEE, 2014, 34(6):897-902(in Chinese).
- [6] TANG Hongyu, CHEN Fei, WANG Cuijun. Design and Implementation of the Multi-motor Synchronous Control System [J]. ELECTRIC DRIVE, 2017, 47(6):45-49(in Chinese).
- [7] HAN Renyin, GUO Yangkuan, ZHU Lianqing, et al. Review of Multi-Motor Synchronization Control [J]. EMCA, 2017, 44(6):8-12(in Chinese).
- [8] MA Zhi-gang, ZHAO Zhi-qiang, WANG Bao-yun. Multi-Motor Synchronous Control of Improved Adjacent Cross Coupling Structure [J]. PACKAGING ENGINEERING, 2017, 38(3):114-118(in Chinese).
- [9] SHI Tingna, XIN Xiong, XIA Changliang. Multi-Motor Speed Synchronous Control Based on Improved Relative Coupling Structure With a Virtual Motor [J]. Proceedings of the CSEE, 2017, 37(23):7004-7014(in Chinese).
- [10] MAO Haijie, LI Wei, WANG Kehong, et al. Sensor fault tolerant sw itch strategy for multi-motorsynchronous system based on ADRC [J]. JOURNAL OF SHANDONG UNIVERSITY (ENGINEERING SCIENCE), 2017, 47(5):64-70 (in Chinese).
- [11] Li Hui, Liu Xingqiao, Li Jing, et al. Aquiculture remote monitoring systembased on IOT Android platform [J]. Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE), 2013, 29(13): 175~181.
- [12] Shang Minghua, Qin Leilei, Wang Fengyun, et al. Information collection systemof wheat production risk based on Android smartphone[J]. Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE), 2011, 27(5): 178~182.
- [13] LIU Xing-qiao, HU Jian-qun, ZHOU Li. Active Disturbance Rejection Control of Three-motor Synchronous Control System[J]. Proceedings of the CSEE, 2010, 30(12):81~85.
- [14] ZHAN Chengguo, ZHU Wei, XU Min. Man-machine interface design and implementation based on Android for measuring and control unit[J]. Electric Power Automation Equipment, 2012, 32(1):113~122.
- [15] LIU Chang-ping, FAN Ming-wei, WANG Guang-wei, et al. Light-weight access control oriented low and Android [J]. Application Research of Computers, 2010, 27(7):2611~2628.