Slovak University of Technology in Bratislava
Institute of Information Engineering, Automation, and Mathematics

PROCEEDINGS

of the 18th International Conference on Process Control

Hotel Titris, Tatranská Lomnica, Slovakia, June 14 – 17, 2011

ISBN 978-80-227-3517-9

http://www.kirp.chtf.stuba.sk/pc11

Editors: M. Fikar and M. Kvasnica

Comparison of Supervisory and Networked Control in Remote Laboratories

P. Folvarčík

Slovak University of Technology,
Faculty of Electrical Engineering and Information Technology,
Institute of Control and Industrial Informatics,
Ilkovičova 3, 812 19 Bratislava, Slovak Republic
(e-mail: pavol.folvarcik@stuba.sk)

Abstract: This paper presents some problems of remote control of real systems. Firstly, it compares the quality of local control (controller are located on the server PC) and remote control (controller are located on client PC) for systems with small time constant. Consequently it deals with solving the problems in the remote control. Solution is realized by modification of communication between client and server and reduction of the quantity of transferred and processed data. After that, communication will be faster and the application will be usable for systems with a lower time constant.

Keywords: remote control, MATLAB, client-server, time delay

1.INTRODUCTION

The existence of transport delays is a normal feature of many technological processes in the input-output relations. Production devices with a time delay often can't be controlled using standard controller designed without considering the presence of transport delays. Action value generated by controller faces to destabilize the feedback loop. This paper is dedicated to analyze the influence on the quality of the control system and eliminate their effects.

2. TRANSPORT DELAY OF SYSTEM

Our control system consists of several parts, between which delays of data can occur and it can cause reduction of control quality. The whole system is composed of client-server application, the computer with running Matlab and the real system, which is connected to a computer.

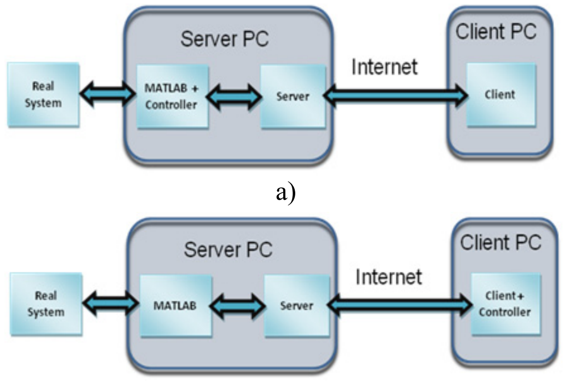


Fig.1. Block diagram of the system: a) local control; b) remote control

In the early stages of my work was control scheme created in Matlab, that was controlled the system (Fig. 1.a). The task of server has to been transfer the necessary data between client and Matlab, especially the parameters necessary to run the simulation. Consequently it sent the measured data from the system to the client that can visualize it for the users. This solution of telematics control was restrictive for the user, because they could not design their own controller. More appropriate solution is to move the controller from the scheme in Matlab directly into the client application, where user can freely modify it (Fig. 1.b). This solution has brought to the feedback loop a transfer delay that was before minimal. It is only an academic example and it can never be used in practice, because it does involve significant security risk and eliminates the possibility to achieve quality closed loop behavior. It was developed only to show an impact of transport delays on quality of control process for students. In the final version of the application the user can choose between server-side control (in Matlab scheme) and the client-side control (its own controller).

2.1Transport delay between client and server

Seeing that client and server run in simulations on the same computer, network delay is minimal. Network delay will increase when running a client application on another computer on the network. Its duration depends on your connection speed and also the network traffic. When the network is without traffic, the network delay is around several tens of millisecond. When the network is with traffic, the value of transport delay is increasing on the value of hundreds of milliseconds, which can lead to the instability of the system for systems with small time constant. Seeing that client can connect to the server from anywhere on the Internet, it is necessary to consider which type of control in simulation will be better.

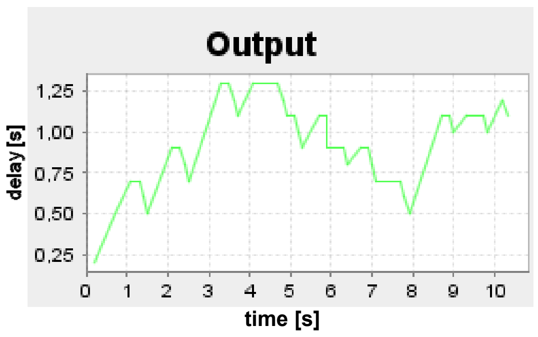
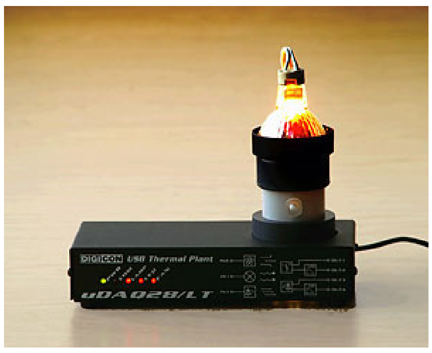


Fig.2. Transport delay between client and server

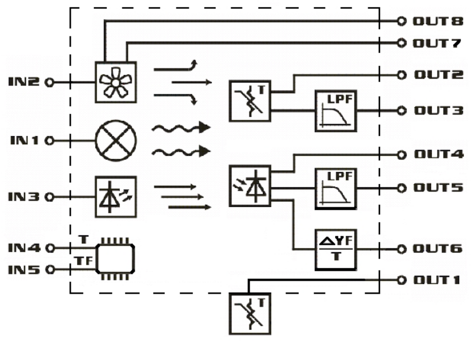
In Fig. 2 you can see evolving of time delay between running the simulation and the time at which the sample was created. The values are in second. Time was measured by creating timer and run it on the client side at a time when it was confirmed to launch simulation on the server side. Their difference was the transfer delay with which the sample was received from the server.

These delays are total, and it includes itself all parts of the delays which occurring during the simulation. Highest part of delay takes the delay of the communication between server and Matlab. Seeing that client and server run in simulations on the same computer, network delay is minimal. Network delay will increase when running a client application on another computer on the network using a video stream from the camera. Delays in some points are more than 1 second. It is necessary to reduce this delay or to use it only to systems with large time constant.

In Fig. 4 is shown the delay if the network is loaded. To made a traffic, we used a video stream, that transmits data over the network at 6Mbps. The picture shows that delays in some samples adding up to less than 3 seconds. This delay leads to system instability. We will show the impact of this delays on the control quality on real thermo-optical plant uDAQ28/LT (Fig. 3) (Huba, 2008).



a)



b)

Fig.3. a) Thermo-optical plant uDAQ28/LT b) Basic electric diagram of thermo-optical plant uDAQ28/LT

This system was designed to support of education of process control. System has three manipulated inputs: bulb voltage (0-5V) which represents heater and light source, fan voltage (0-5V) which can be used for temperature decreasing and voltage of led diode (0-5V) which represents another source of light. On the output is possible to measure seven variables: temperature insight the system (direct or filtrated), outsight temperature, light intensity (direct or filtrated), fan velocity and fan current. In the next two figures (Fig. 5. and Fig. 6.) is a comparison of the output of optical channel with control on the server side (without transport delay) and control on the client side (with transport delay). The network was with traffic 6 Mbps.

After elimination of network traffic delay was reduce (Fig. 2.) and the quality of regulation was better (Fig. 7.). Values of transfer delays are stil quite high and the system we can not regulate.

Control process for this delay may not work properly because it is a system with time constant less than 2 second. In Fig. 8 and Fig. 9 are shown traces of the outputs of the system with time constant of more than 20 seconds. It's unfiltered thermal channel of thermal-optical plant.

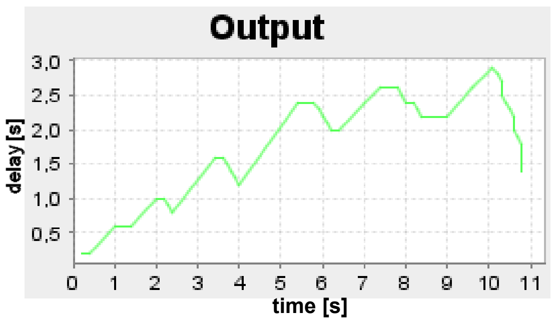


Fig.4. Transfer delay on the network with traffic

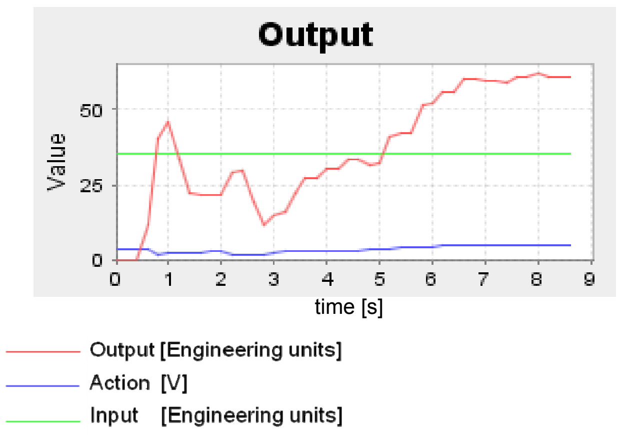


Fig.5. Simulation on real system with client control

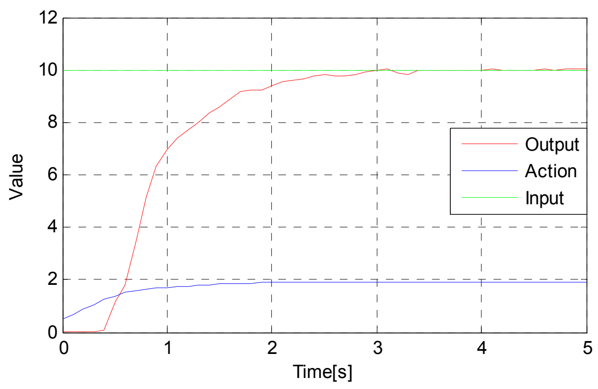


Fig.6. Simulation on real system with server control

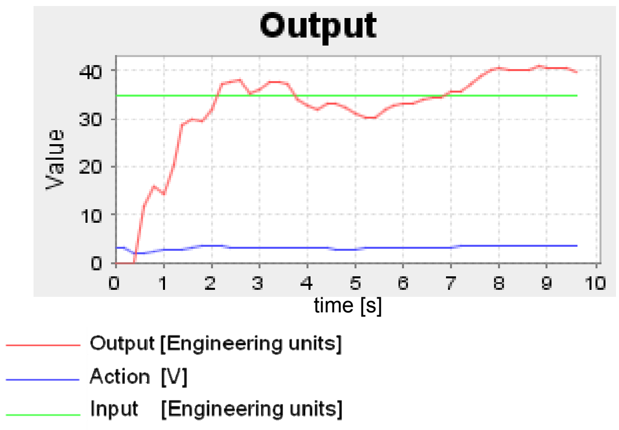


Fig. 7. Output value on network without trafic

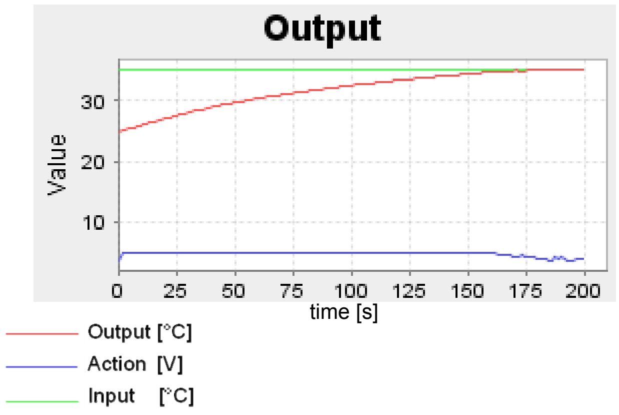


Fig. 8. Output value with server control

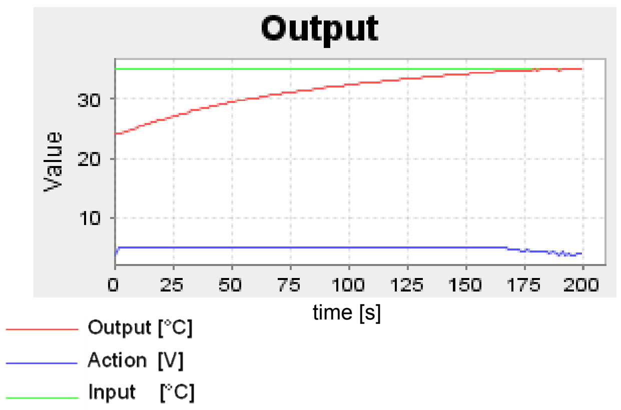


Fig. 9. Output value with client control

In the chart we can see that delays caused small overshoot on output values. The course of action values is different for both graphs, which is caused by transfer delay. As this is a system with high time constant, did not influence the course of output variables. Our application works reliable on systems with high time constant.

2.2 Transfer delay betweet Matlab and server

In communication between Matlab and the server it causes delays in transmission in both directions. Communication has been made by using COM objects (Beránek, 2006). Calculation of the action value is performed on the client side and therefore it is necessary send required data to the client to calculate. Server application receives by the method *getRealtimeData()* measured values of Matlab, that it sends to the client.

Data are obtained from the *scopeData* variable in that Matlab Scope object stores data during simulation. This is the only way to get the measured values during simulation. Its disadvantage, however, is that Matlab sends to the server all previously stored data. Data is sent as a matrix whose number of rows is equal to the number of samples and number of columns is equal to the number of sent variables. During the simulation it's changing the number of rows in the matrix. The server then must recognize what is already sent to the client and what is not his yet. For this task we have *upperBorder* and *lowerBorder* variables. In to the *lowerBorder* variable is written number of samples received in the previous matrix. In to the *upperBorder* variable is written number of samples in new matrix. Client receives only the samples with indexes between *lowerBorder* and *upperBorder*. This method of sending was designed for the application with server side control and it was sufficient for the user, because the delay, that it caused, has no effect on the quality of control. However, when we wanted used the same method to the client side control, we encountered a few problems.

The most significant was that the server sent at once 3 samples on average and client calculate an action variable for each of them in the order they were received. First it calculated an action value for the sample which was calculated at time t-2Ts, then t-Ts and finally for the sample at time t. In calculating the control value it calculated value from the sample, which was no longer current. The most accurate calculation was only at the last received sample. On the control process that had the most impact in systems with small time constants. The impacts of this delay, we have analyzed in section 2.1, where we showed that delay is relatively high also without the delays in transfer over the network. Delays in some places are more than 1 second. That is why it is necessary to reduce this delay or to use it only for systems with high time constant. Because we want a system with the widest possible use, we looked for a way to reduce this delay.

3.MINIMIZING THE TRANSFER DELAY

The biggest time delay was identified when server has sent more measured samples at the same time. Because we can not control the speed of taking samples of Matlab, the only possibility to speed up the transfer is sent only the most current sample to the client. This ensures that the client will not receive samples that are not current and it will not calculate an action value. At the same time we reduce the number of transferred data and communication will be faster. Previous server sent measured samples by the index value from *lowerBorder* to *upperBorder*. After modifying the servers code, server sends only the value with index *upperBorder*, consequently client receive only the most current value. Client calculates action value and sent it to the server. Server sent the value to Matlab and wait for next sample. In the equation for calculating the action value acts parameter Ts. It is sampling time. Client, but does not receive all of the the samples, but only some of them. In practice this means that is sending approximately every third sample, so the actual sampling time is approximately 3Ts.So that client calculate an action value always at the correct sampling time it must calculate sampling time at each step itself. Sampling time is thus variable. This function provides the following code:

```
multiple=(t[0]-t[1])/Ts;
if(multiple==0) multiple=3;
t[1]=t[0];
```

Action value is then calculates:

```
u[0]=u[1]+P*e[0]+multiple*Ts*I*e[1]-P*e[1];
```

`u[0]`	– actual control action
`u[1]`	– control action in time t-Ts
`e[0]`	– actual control error
`e[1]`	– control error in time t-Ts

After these modifications, we minimize the delays that arose in our application. In Fig.10 we can see time delay of the sample. It is seen that the delay is significantly reduced. Before the modifications it fluctuated between 0.5 and 1.25 seconds. After the modifications it fluctuated between 0.2 and 0.4 seconds.

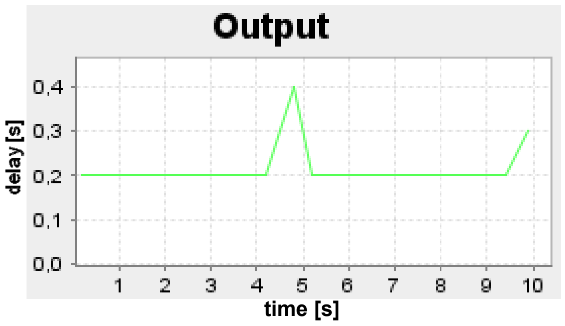


Fig. 10. Time delay of sample

This delay was measured without network load. After adding the traffic like in section 2.1, thus the video stream with speed of 6Mbps, we measured delay between 0.2 and 0.8 seconds (Fig. 11). In the previous server application it was between 1 and 3 seconds. In Fig. 12 and Fig. 13 is a comparison of control on the client side with control on the server side for unfiltered optical channel of thermal-optical system (system with small time constant).

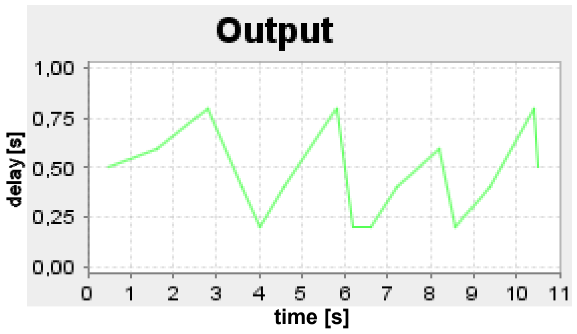


Fig. 11. Delay of sample on network with traffic

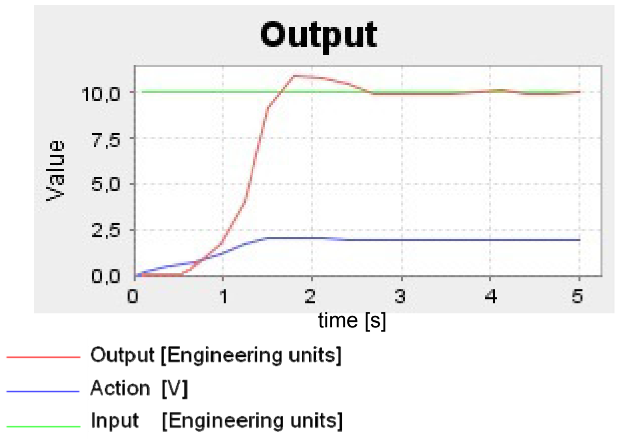


Fig. 12. Output value with client control

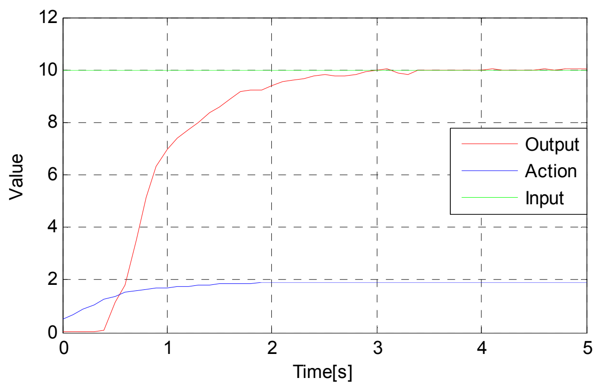


Fig. 13. Output values with server control

The chart shows that the delay of the sample has an impact on the quality of control. Transport delays caused overshoot in our case. Controller, despite delay, regulates the output value to the desired value. In the previous server application was system unstable. On the quality of the regulatory process had influence two basic components. First was the delay and the second was sampling time. On the server side we used Ts = 0.2 seconds. But n the client side not. Although Matlab counted with sample time Ts = 0.2 seconds, but the client receives every second or third sample. Thus, sampling time was at the client two to three times greater. Thus, if we want to compare most accurate, we must set the sampling time for the client minimal to the value Ts / 2. Fig. 14 shows the course of output variables with Ts = 0.1 seconds.

The output is similar to the output of server-side control. This showed that the sampling time in this system had a greater impact to quality of control them the transfer delay. This simulation showed that the server side control is working correctly.

Server and client are modified so that we can send to the server information, which type of control we want to use (client side or server side). The user can choose this parameter before running simulations directly in the window of client application. Hereby he can set parameters of PI controller that is used in simulation with client-side control.

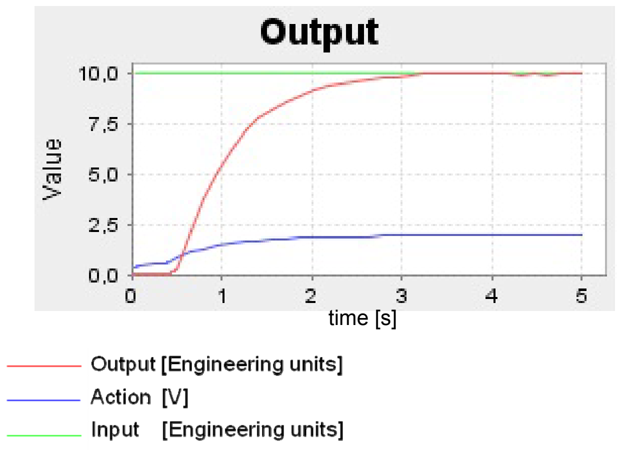


Fig.14. Output of system with Ts= 0.1 second

Each of these options uses a different control scheme, it is therefore necessary to choose the correct startup scheme. Otherwise the simulation will not run and Matlab sends an error message.

4. CONCLUSION

Simulations proved that an application for remote control works correctly also for systems with small time constant (approximately 2 second). The theory, about instability of feedback loop with high transfer delays, was confirmed. The quality of control process was much better after the identification of transfer delays and minimization of them. This solution will enable greater use of application, especially in educational process. It also allows users to create their own controller and simulate its behavior on different real systems. Subject of next research would be to eliminate any delays in communication on the server side.

ACKNOWLEDGMENTS

The work has been partially supported by the Grant KEGA No. 3/7245/09, by the Grant VEGA No. 1/0656/09 and by the Grant VEGA No. 1/0369/10. It was also supported by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

REFERENCES

Beránek, M. (2006). Client-server application for remote control of real. *Diploma thesis*. STU Bratislava.

Bisták, P. (2008) Remote Laboratory Java Server for Data Exchange with Matlab Automation Server, *Proceedings of International Conference REV 2008*, Düsseldorf, Germany.

Bistá, P. (2009). Matlab and Java based virtual and remote laboratories for control engineering. In: *17th Mediterranean Conference on Control & Automation*.

Gomes L., S. Bogosyan (2009). Current Trends in Remote Laboratories, *IEEE Trans. Industrial Electronics*, vol. 56, No. 12, pp. 4744- 4756.

Huba, M. (2008) Thermo-Optical Laboratory Plant uDAQ28/LT, Technical and Users Guide, http://www.eas.sk/members/2/file/070212%20opticko_tepelna_sustava.pdf

Müller S., H. Waller (1999). Efficient integration of real-time hardware and Web based services into MATLAB, *Proceedings of 11th European Simulation Symposium*, Erlangen, Germany.

Restivo, M. T. et al. (2009). A Remote Laboratory in Engineering Measurement. In: *IEEE transactions on industrial electronics*, Vol. 56, No.12 [online].

Safaric, R., S. Uran, M. Trunic, I. Hedrih (2004). Remote controlled mechatronics device via internet using Matlab, *Proceedings of 1st Int. REV Symposium*, Villach, Austria.

Sheng Y., W. Wang, J. Wang, J. Chen (2008). A Virtual Laboratory Platform Based on Integration of Java and Matlab, *Li, F. et al (Eds.) ICWL 2008*, LNCS, vol. 5145, Springer-Verlag Berlin Heidelberg.