Level Control System – Virtual Experiment

Ion Matei, Dumitru Popescu, Ciprian Lupu, Ilie Luican
Politehnica University of Bucharest – Control and Computer Science Faculty
Splaiul Independentei nr. 313, sector 6, Bucuresti, Romania
email: mion@home.ro

Abstract – *This paper presents the construction of a distant level control system. There are presented theoretical concepts for digital control system design and the hardware and software architecture that allow external access to the experimental platform using a simple Internet browser. The application is build as a virtual laboratory but it may be extended with some consideration to industrial process control.*

Keywords: high education, client-server application, communication, internet, control algorithms, level control.

1. Introduction

Today the Internet is largely used in education, commercial, industry and personal activities. It represents a gate to a lot of useful information. From its birth, in the 1960^{th} till our days, the Internet has evolved from a structure that used to offer only static information (from server to client) to a structure that allow real-time information, an user being able to connect to an equipment or a process from any place in the world. This type of architecture facilitates the distant learning technologies.

The possibility of interaction with a didactical platform by Internet, allow free access to laboratory activities because students will be able to perform their practical work without time limitation. Distant access reform the traditional way of education by reducing the time spent in universities and increasing the role of self education.

Distant control does not limit itself only to education. While individuals become more and more active because of economic dynamic, distance access represents an interesting option for scientists, who will be able to access equipments spread over a large area. The advantages of distance access have lead to a great number of research activities. An example of a distributed laboratory built around object oriented protocol (TelRIP) is described in [1]. Another example is a distributed monitoring and control system created by a university in Singapore [2].

This paper presents the aspects of building of a level control system with the possibility a distance access using the Internet. The Internet is used as a communication medium because it is largely spread around the world so that from any place on the globe with an Internet connection the platform can be accessed using a simple Internet browser.

The paper is structured in two parts. The first part presents some theoretic notions about the digital algorithms used for level control. The second part describes the client-server architecture for distant access.

2. Level control system

The practical activity of this virtual experiment is the level control of fluid from a tank with consumer. The next figure presents the schema of the level control system.

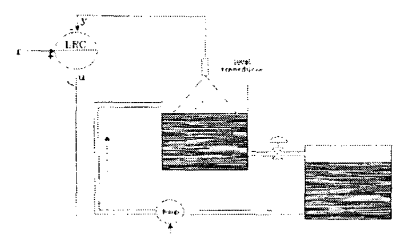

Fig. 1 – Process diagram

The fluid flow is controlled by an electric pomp. The process input is a voltage with values between 0 and 10 V and the output is the fluid level measured by an ultrasounds transducer. The process is nonlinear but it is approximated with linear mathematical model around a representative point of functioning. The approximated mathematical model which was obtained using the sample period Te = 5s, has the form:

$$\frac{B(q^{-1})}{A(q^{-1})}=\frac{0.184q^{-1}+0.173q^{-2}}{1-0.325q^{-1}-0.335q^{-2}} \qquad (1)$$

It is important to mention that for designing a numerical regulator three steps must be followed:

- *computing the discrete model;*
- *performances specification;*
- *computing the adequate control algorithm*

In this case for the level control were used the next numerical control algorithms: RST and PID with its two variants, PID1 and PID2. These algorithms have different performances so that the students will have the opportunity to compare them. First it will be presented the RST algorithm and than the paper will continue with PID algorithms which will be structured analyzed and compared to the first one.

2.1 RST numerical algorithm

For any control system there are defined two important objectives:

1. *Reference's tracking;*
2. *Rejection of the perturbations;*

A classical control structure, with only one liberty degree (fig. 2) has the great disadvantage that it can not fulfill the two objective defined above.

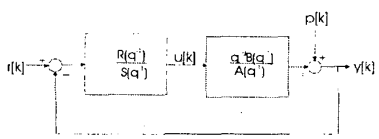

Fig.2 – One freedom degree control structure

The input control $u[k]$ given by such an algorithm is:

$$u[k]=\frac{R(q^{-1})}{S(q^{-1})}(r[k]-y[k])=\frac{R(q^{-1})}{S(q^{-1})}r[k]-\frac{R(q^{-1})}{S(q^{-1})}y[k] \qquad (2)$$

As it may be noticed the command is not pondered differentially by reference and measure. This means that some performances defined for perturbations rejection could restrict the reference tracking. If the polynomial $R(q^{-1})$, which filter the reference, is replaced by another polynomial $T(q^{-1})$ it will lead to a RST structure with two liberty degree that accepts different performances in tracking and rejection [3][4].

The block structure of the RST numerical algorithm is presented in the next figure:

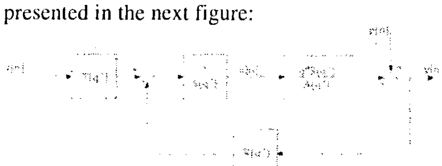

Fig.3 – RST block structure

The closed loop transfer function is:

$$H_0(q^{-1})=\frac{B(q^{-1})\cdot T(q^{-1})}{A(q^{-1})\cdot S(q^{-1})+q^{-d}B(q^{-1})\cdot R(q^{-1})}=\frac{B(q^{-1})\cdot T(q^{-1})}{P(q^{-1})} \qquad (3)$$

The performances for rejection are established by characteristic polynomial $P(q^{-1})$ obtained by sampling a second order continuous system. The polynomials $R(q^{-1})$ and $S(q^{-1})$ results from solving the next polynomial equation:

$$A(q^{-1})\cdot S(q^{-1})+q^{-d}B(q^{-1})\cdot R(q^{-1})=P(q^{-1}) \qquad (4)$$

The tracking performances are controlled by $T(q^{-1})$ and reference filtering. The polynomial $T(q^{-1})$ is chosen so that the close loop transfer is unitary (the output will follow the imposed reference y*).

$$T(q^{-1})=P(q^{-1}) \qquad (5)$$

Tracking dynamics can be modify by polynomials $B_m(q^{-1})$ and $A_m(q^{-1})$ which filter the reference:

$$y^*[k]=\frac{B_m(q^{-1})}{A_m(q^{-1})}r[k] \qquad (6)$$

Polynomials B_m and A_m result also from sampling a second order continuous system. Therefore choosing the parameters $_n$ and will determine the tracking performances [4].

The block diagram of the RST control structure with filtered reference is presented below:

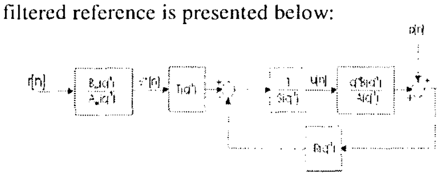

Fig. 4 – RST structure with filter reference

The system tracking performances are chosen by a second order system with $_n = 0.25$, $= 0.99$ which results in:

$$\frac{B_m(q^{-1})}{A_m(q^{-1})}=\frac{0.0071\cdot q^{-1}+0.0066\cdot q^{-2}}{1-1.7669\cdot q^{-1}+0.7807\cdot q^{-2}} \qquad (7)$$

The rejection performances are also given by a second order system with: $_n = 2.5$, $= 0.8$.

The characteristic polynomial for the poles allocation method, will become:

$$P(q^{-1})=1-1.2807\cdot q^{-1}+0.4493\cdot q^{-2} \qquad (8)$$

Solving the system (4) and applying (5) results in:

$$R(q^{-1})=5.1232-6.5\cdot q^{-1}+2.0516\cdot q^{-2} \qquad (9)$$

$$S(q^{-1})=1-11.432\cdot q^{-1}+0.4321\cdot q^{-2} \qquad (10)$$

$$T(q^{-1})=6.3423-9.281\cdot q^{-1}+3.6137\cdot q^{-2} \qquad (11)$$

System response to step and perturbations rejection are presented in the figure below:

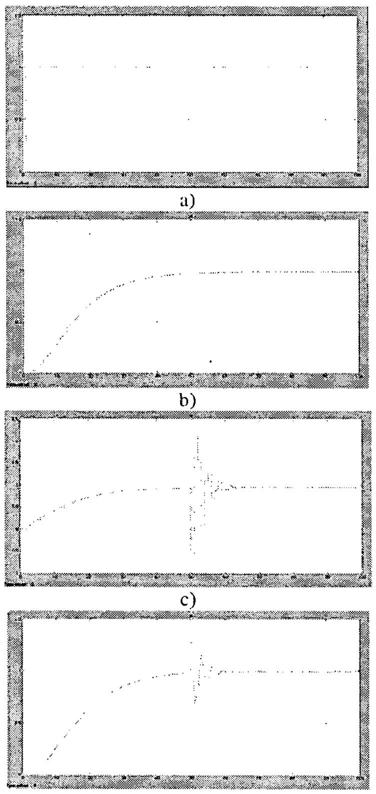

Fig.5 a) reference - r; b) filtered reference - y^*
c) command - u; d) output - y

2.2 PID algorithm

Numerical PID algorithm comes from sampling the continuous PID algorithm with an appropriate sampling period.
Numerical PID controllers are suitable only for processes modeled by a second order transfer function with or without pure delay.
It is worth to mention that some numerical PID controllers do not have a continuous equivalent.

2.2.1 PID1 numerical algorithm

Let us consider the algorithm:

The transfer functions of a continuous PID controller:

$$H_{PID}(s) = K \cdot \left(1 + \frac{1}{T_i s} + \frac{T_d s}{\alpha T_d s + 1}\right) \tag{12}$$

This algorithm is specified by four parameters:

K – proportional action;
T_i – integral action;
T_d – derivative action;;
T_d – filtered derivative action.

There are many methods that can be used to sample this algorithm resulting in a numerical controller of the same form.

The sampling rectangular method consists in approximating $1/s$ with $T_e/(1-q^{-1})$ which leads to the next numerical algorithm:

$$H_{PID}(q^{-1}) = \frac{R(q^{-1})}{S(q^{-1})} = \frac{r_0 + r_1 q^{-1} + r_2 q^{-2}}{(1-q^{-1})(1+s_1 q^{-1})} \tag{14}$$

with the parameters r_0, r_1, r_2 si s_1 obtained from:

$$s_1 = -\frac{1}{1 + \frac{T_e}{\alpha T_d}},\ r_0 = K\left(1 + \frac{T_e}{T_i} - \frac{s_1}{\alpha}\right),$$

$$r_1 = K\left[s_1\left(1 + \frac{T_e}{T_i} + \frac{2}{\alpha}\right) - 1\right],\ r_2 = -K s_1\left(1 + \frac{1}{\alpha}\right) \tag{15}$$

As it may be seen, PID1 numerical regulator has also four parameters (r_0, r_1, r_2, s_1) as its continuous form.
This algorithm can be structured in a RST form if $T(q^{-1})=R(q^{-1})$ as it is shown in figure 6.

Fig. 6 - PID1 structure with three braches

It is easy to notice that PID1 is one freedom degree algorithm. The transfer function in closed loop which join the reference $r(t)$ with the output $y(t)$ is:

$$H_0(q^{-1}) = \frac{B(q^{-1}) \cdot R(q^{-1})}{A(q^{-1}) \cdot S(q^{-1}) + B(q^{-1}) \cdot R(q^{-1})} = \frac{B(q^{-1}) \cdot R(q^{-1})}{P(q^{-1})} \tag{16}$$

where $P(q^{-1})$ defines the poles wanted in closed loop that determine the performances for perturbations rejection. PID1 algorithm introduces new zeros in closed loop by polynomial $R(q^{-1})$ that alter the performances. This negative effect will be partially eliminated by the PID2 algorithm. Having already determined the RST algorithms it is now easy to calculate the PID1 parameters (if the rejection performances define by $P(q^{-1})$ are not changed) because the polynomials R(q-1) and $S(q^{-1})$ will be the same [3].
As it may be observed in the next graphics the performances of the PID1 algorithm are inferior to the RST algorithm:

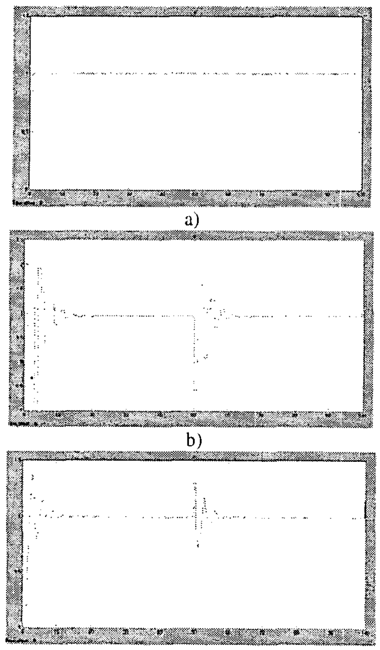

Fig. 7 a) reference PID1 – r b) command PID1 – u c) output PID1 – y

2.2.2 PID2 numerical algorithm

As it was noticed above the PID1 regulator introduces supplementary zeros in closed loop by polynomial $T(q^{-1})=R(q^{-1})$. Those zeros affect specially the control performances as it was observed in figure 7. This disadvantage can be partially eliminated if the polynomial $T(q^{-1})$ is chosen as follow:

$$T(q^{-1})=\frac{P(1)}{B(1)}=R(1) \tag{17}$$

assuring in this manner a unitary transfer function in stationary regime [3].
The block schematics of the PID2 numerical algorithm structured as RST is presented bellow:

Fig. 8 - PID2 numerical controller

This algorithm assures superior performances compare to PID1 as it may be seen in the next graphics:

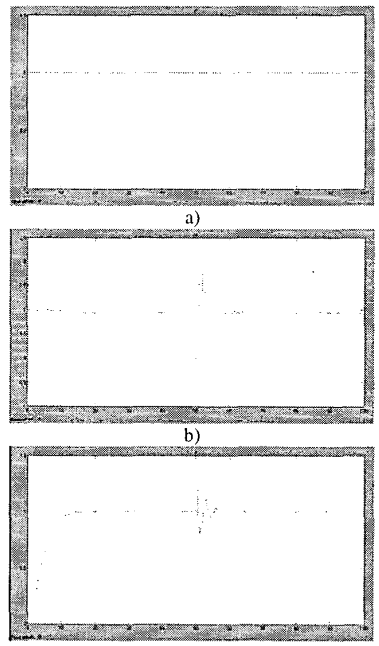

Fig. 9 a) refernce PID2 – r b) command PID2 – u c) output PID2 – y

3. Virtual experiment

The experiment presented in this paper is a part of the laboratory activity of "Identification and Control Systems", course dedicated to the forth year students from the UPB-Control and Computer Science Faculty. The laboratory purpose is to train students in practical activity of system identification and numerical process control. During this part of the practical activity the students have the possibility to control the level fluid from a tank by testing different type of numerical algorithms. In the figure 10 is presented an image of the level control didactical platform.

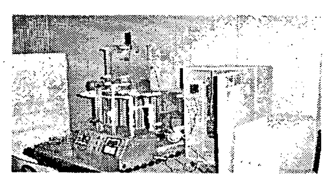

Fig.10 – Didactical platform

The lab activity, in the virtual version proposed, is structures in two parts. In the first part students practice the open loop identification of the studied process, the presence of the students in the laboratory being necessary. The second part of the lab consists in numeric control of the identified process by testing various algorithms. For this section of the lab the presence of the students in the laboratory is not compulsory because the activity could be undertaken from any place with an internet connection. The virtual experiment is formed from two components: a hardware and o software structure.

3.1 Hardware structure

The base of this experiment is a hardware structure composed from:

- an acquisition board with analog to digital and digital to analog converters which acquires and send data to the process;
- a PC compatible IBM with a Pentium II – 350 MHz processor and 64 Mbytes RAM, which contain the acquisition board;
- a video camera for real-time images broadcasting;
- a HP LH3 server for Internet connection.

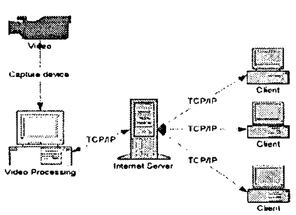

Fig. 11 – Hardware structure

3.2 Software structure

The software structure is formed by to subsystems which allow access to the didactical platform local as well as distant – using the Internet as it may be observe in figure 10:

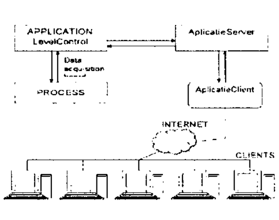

Fig. 12 – Software structure

The first subsystem represents a program written in Lab Windows/CVI. It runs on the computer connected to the process and allows direct local access to the platform. This application represents a gate for the data that leave and come to the process.

The second subsystem is a client-server application which assures distant interaction with the process.

3.2.1 Local access

The program for local access was created using Lab Windows/CVI and it is an independent application that allows local access to the platform independent towards the distant access.

This program makes the connection with the process acquiring and sending data from respectively to the process. The data is sent further on to the clients applications via the Internet.

The local application assures a few operations over the platform such as:

- reference changing;
- automat/manual regime changing;
- manual command;
- algorithms configuration.

For an easy use, the local program has a graphical user interface (GUI) presented bellow.

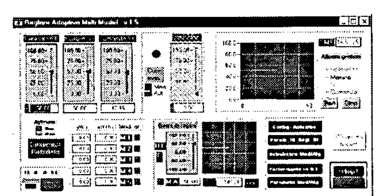

Fig. 13 – Interfata aplicatie locala

3.2.2 Distance access

The base of the distant communication is a ***client-server application***. There are some key words that define such an architecture which will be described as follows:

Server – entity that accepts requests and offers services to the clients connected to him [5]. For this experiment the server is a program written in Java which runs on HP LH3 server machine. The server connects itself to the local application and become a proxy for the clients that connect to him. The server is a multithreading program that runs in background and sends periodically data about the state of the process to the clients. Theoretically it can accept an infinite number of client connections but the number is limited by software to maintain a certain level of performances.

Client – software entity which sends requests to the server and process the data received as answer [5].

For this experiment the client is a Java applet incorporated in a web page. Next to the applet there are presented live images with the process evolution. For security reasons imposed by Java language the applet and the server must resides on the same machine. The client has a graphical interface which presents naturally information (received from server) about the state of the process and send data to this one. The access to the web page is supervised by an Apache web server and it is done using a URL address.
Communication protocol – connection between the clients and the server and between the server and the local application it is done using TCP/IP protocol. This protocol has a few important characteristics such as: allow multiple connections and has a mechanism that verify the integrity of the packets an altered one being send again.
Socket – a socket it is an abstract object that represents a communication gate between processes in a network.
The client application offers the same facilities as the local application. The user can modify the reference, change the automate/manual regime and generate manual commands as it is presented in figure 13.

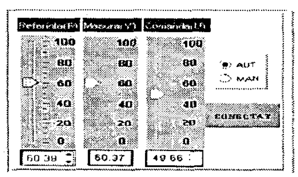

Fig. 14 – Interface for reference and command changing

In the same time the user can see the time evolution graphic of reference, measure and command.

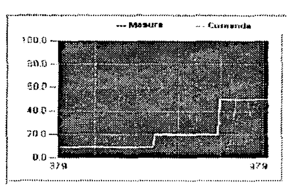

Fig. 15 – System input, system output
and control output evolution

Also, to increase the interaction with platform, the user is able to design its own control algorithms and send them to platform for testing.

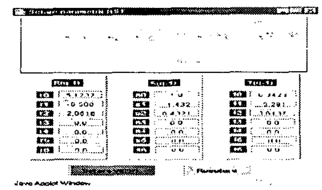

Fig. 16 – GUI for RST parameters change

For security reason, in case of bad algorithms the local application will take over and replace the bad algorithm with a default one.

An important aspect is the study of delay due to network congestion. This delay can affect the update of the parameters for the client program or the receiving of the new data by the platform. However, because the data are sent at Te = 5s rate, only a connection interruption could seriously affect the data transmission. The practice has shown that a 56 Kb/s Internet connection assures good performances.

By the distant connection to the platform, using the Internet, a user is able to know and change the state of the platform from any place in the world accessing a web page. This manner of distant access transforms the traditional way of education into a more flexible and more personal learning experience.

4. Conclusions

In this paper a level control system with distant access option that allows monitoring and updating of important parameters of the system is presented.

There are presented the base theory for level control and the hardware and software structures for distant access via the Internet.

A comparative study for performance evolution of different control algorithms can be realized.

The system has been created as an educational tool through the students can experiment distant access to a technological platform.

References:

[1] G. V. Kondraske, R. A. Voltz, D. H. Johnson, D Tesar, J. C. Trinkle and C.R. Price, "Network based infrastructure for distributed remote operations and roboticsresearch", *IEEE Trans. Robot. Automat, vol. 9, pp. 702-704, Oct 1993.*

[2] K. K. Tan, T. H. Lee, C. Y. Sog, "Internet-Based Monitoring of Distributed Control System-An Undergraduated Experiment", *IEEE Trans. Education, vol. 45, pp. 128-134, May2002.*

[3] I. D. Landau, "Identification et Commande des Systemes", *Ed.Hermes, Paris,1995.*

[4] D. Popescu, S. Gentil, "Commande Numerique et Intelligence Artificielle en Automatique", *Ed. Tehnica, Bucuresti, 1998*

[5] A. Tannenbaum, "Retele de calculatoare", *Ed.Agora, 1999.*