REMOTE PROCESS CONTROL USING MATLAB

Sysel, M., Pomykacz, I.

Tomas Bata University in Zlín
Faculty of Applied Informatics
Nad Stráněmi 4511, 760 05 Zlín, Czech Republic
e-mail: Sysel@fai.utb.cz

Abstract: This contribution presents remote process control and monitoring. The developed application MatlabLink enables to create MATLAB applications that use the capabilities of the World Wide Web to send data to MATLAB for computation and to display the running results in a Web browser. The Running results are accessible via Internet. Presented experiment demonstrates properties of delta adaptive controllers and recursive least square methods for the second order systems.

Keywords: Remote Control, Adaptive control, delta models

1. INTRODUCTION

Using program Matlab for control on the distant computer is the keynote of remote control of laboratory technological processes. Let's assume technological process controlled by Matlab program on the first computer (no matter what - simulation or real laboratory plant). It is possible to command and monitor this one by any computer in the Internet network. But just only one computer can control this process, monitoring is accessible to arbitrary number of computers. This system uses client/server architecture. Proposed solution considers contact server which allows communication with computers in private network.

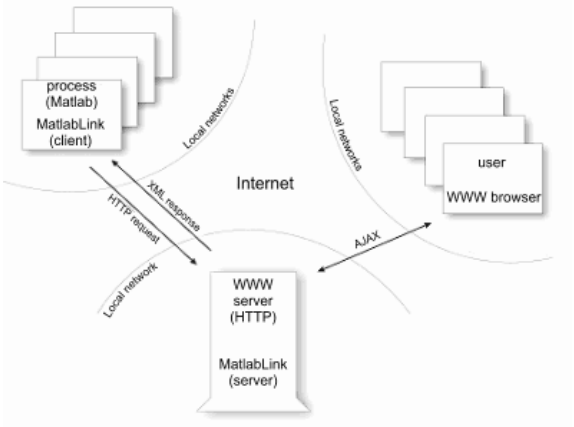

Fig. 1.: Complete common diagram

Diagram on figure 1 is generalized about three different networks, but can operate on just only one network. This architecture allows concurrent run of several technological processes in different local networks, users can access WWW server with public IP address from any network.

2. MATLABLINK

The whole application consists of several subsystems and makes use of several different technologies in this time.

- JMatLink – Library, which allows Matlab connection with Java (Mueller, S.). By the instance of a JMatLink class is possible to start up program Matlab, execute commands and obtain state of the Matlab workspace.
- MatlabLink – Application that runs on the same computer where program Matlab is installed. This part is written in Java and uses JMatLink library. MatlabLink periodically executes main control, reads variables from workspace and communicates with server.
- MatlabLink Server – Presents a server part of application MatlabLink. MatlabLink server communicates with MatlabLink clients and stores comunication on the server.
- MatlabLink Web - This part works together with MatlabLink server. HTML and JavaScript code generated by this part is sent to the user web browsers.

2.1 Design and structure of Matlab application

Matlab application commonly consists of several m-files. There is necessary initialize variables, define starting conditions and load drivers in the beginning. Next, process control proceeds in defined calculation loop. Finally, application should be proper finished (for example unloads drivers) otherwise we can expect undefined state or controlled system damage. Developed application for remote control - MatlabLink - requires specific structure of m-files which is based on above mentioned behavior. Matlab application should consist of three parts; each of parts can consist of arbitrary number of m-files. The name of the m-files is not strictly specified.

- The repetition rate should be defined in the first part and the name of this variable should be defined in the configuration file (XML format).
- The second part presents process control. MatlabLink provides periodic execution of this file. This file can call another m-files or functions. This file should be periodically executed by MatlabLink, because Matlab does not offer content of workspace during execution.
- The last part is executed after completing control loop and provides proper termination and disconnection of hardware.

2.2 MatlabLink Web

The figure 2 shows the part of the interface of the MatlabLink Web which is accessible via web browser. The running processes are listed in the table which is periodically refreshed. The time till the next refreshing is displayed on the top-right part of this table.

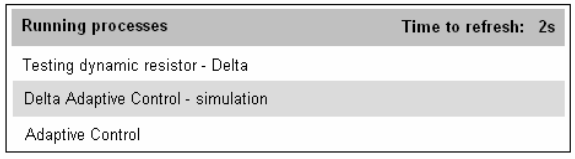

Figure 2.: MatlabLink Web.

User can select one of the running processes, after it, the next window will require access password defined in the configuration file. Each of the running process is possible protect with a unique password to guarantee the process is controlled by the authorized user. The password is required just only for the remote control; the monitoring is accessible without any password. If the authorized user is log-in, system does not require the password and the user is automatically redirected to the monitoring page.

The user can see just only permitted Matlab outputs in the monitoring page. The variables (or figures) are displayed in the individual window, they are periodically refreshed. The authorized user moreover gets possibility of command this process (start, stop, set-up variables, change rate and terminate process).

3. EXPERIMENT

The main idea of an STC is based on a recursive identification procedure and a selected control synthesis.

3.1 Identification

For the identification is used a stochastic second order delta model with discrete equation

$$\delta^2 y(k) = -\bar{a}_1 \delta y(k) - \bar{a}_2 y(k) + \bar{b}_1 \delta u(k) + \bar{b}_2 u(k) \quad (1)$$

Using substitution

$$\delta = (q-1)/T_0 \quad (2)$$

where q is shift operator, can be written

$$\begin{aligned}
&\delta^2 y(k) = y_\delta(k) = \frac{y(k) - 2y(k-1) + y(k-2)}{T_0^2};\\
&\delta y(k) = y_\delta(k-1) = \frac{y(k-1) - y(k-2)}{T_0};\\
&y_\delta(k-2) = y(k-2);\\
&\delta u(k) = u_\delta(k-1) = \frac{u(k-1) - u(k-2)}{T_0};\\
&u_\delta(k-2) = u(k-2)
\end{aligned} \quad (3)$$

The vector of delta parameters has the form

$$\Theta_\delta^T(k) = \left[\bar{a}_1, \quad \bar{a}_2, \quad \bar{b}_1, \quad \bar{b}_2\right] \quad (4)$$

and the regression vector is

$$\phi_\delta^T(k-1) = \left[y_\delta(k-1), -y_\delta(k-2), u_\delta(k-1), u(k-2)\right] \quad (5)$$

The recursive least squares method is utilized for calculation of the parameter estimates $\hat{\boldsymbol{\Theta}}_\delta(k)$ and adaptation is supported by directional forgetting. The value of the directional forgetting factor $\varphi(k)$ basically depends on the level of conformity

achieved between the model and the real behavior of the system.
The actualization of the parameter estimates vector is calculated by

$$\hat{\boldsymbol{\Theta}}_{\delta}(k) = \hat{\boldsymbol{\Theta}}_{\delta}(k-1) + \frac{C(k-1)\boldsymbol{\phi}_{\delta}(k-1)}{1+\xi(k-1)}\hat{e}(k-1) \quad (6)$$

The start - up conditions for the most commonly used identification methods are the initial parameter estimates and their covariance matrix. The importance of the covariance matrix is often neglected and is difficult to design. It is a good idea to choose the following conditions for the start of the algorithm: the elements of the main diagonal of the covariance matrix should be $C_{ii}(0) = 10^3$, start-up value for the directional forgetting factor $\varphi(0) = 1$, $\lambda(0) = 0.001$, $\upsilon(0) = 10^{-6}$, $\rho = 0.99$. The start estimates for the parameter estimates vector $\hat{\boldsymbol{\Theta}}_{\delta}(0)$ is chosen according to a priori information.
Calculating auxiliary variables from the following relations (Bobal, 2000)

$$\xi(k-1) = \boldsymbol{\phi}_{\delta}^{T}(k-1)C(k-1)\boldsymbol{\phi}_{\delta}(k-1);$$

$$\upsilon(k) = \varphi(k)\left[(\upsilon(k-1)+1\right]; \quad \eta(k) = \frac{\hat{e}^2(k)}{\lambda(k)};$$

$$\lambda(k) = \varphi(k)\left[\lambda(k-1) + \frac{\hat{e}^2(k-1)}{1+\xi(k-1)}\right] \quad (7)$$

Calculating the directional forgetting factor

$$\varphi(k) = \{1 + (1+\rho)\left[\ln\left(+\xi(k-1)\right)\right] + \left[\frac{\left(\upsilon(k-1)+1\right)\eta(k-1)}{1+\xi(k-1)+\eta(k-1)} - 1\right]\frac{\xi(k-1)}{1+\xi(k-1)}\}^{-1} \quad (8)$$

Calculating the auxiliary variable

$$\varepsilon(k) = \varphi(k) - \frac{1-\varphi(k)}{\xi(k-1)} \quad (9)$$

If $\xi(k-1) > 0$ then the covariance matrix is actualized using expression

$$C(k) = C(k-1) - \frac{C(k-1)\boldsymbol{\phi}_{\delta}(k-1)\boldsymbol{\phi}_{\delta}^{T}(k-1)C(k-1)}{\varepsilon^{-1}(k) + \xi(k-1)}$$

$$\text{if } \xi(k-1) = 0, \text{ then } C(k) = C(k-1) \quad (10)$$

Other approaches of recursive identification of delta models are described by Goodwin and Sang Sin (1984).

3.2 2DOF Control Loop

Design of controller is derived from general block diagram of closed - loop in Figure 3.

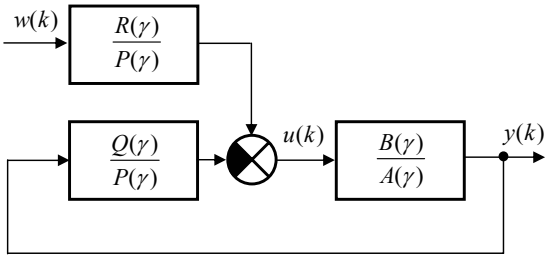

Fig.3.: 2DOF closed control loop

The controller operator equation takes the form

$$U(\gamma) = [R(\gamma)W(\gamma) - Q(\gamma)Y(\gamma)]\frac{1}{P(\gamma)} \quad (11)$$

Polynomials of controller have next forms

$$P(\gamma) = p_1\gamma + p_0$$
$$Q(\gamma) = q_1\gamma + q_0 \quad (12)$$
$$R(\gamma) = r_0$$

Polynomials of the discrete transfer function of the controlled system are

$$A(\gamma) = \gamma^2 + \bar{a}_1\gamma + \bar{a}_0$$
$$B(\gamma) = \bar{b}_1\gamma + \bar{b}_0 \quad (13)$$

Controller synthesis based on solving of the pair of equations

$$A(\gamma)P(\gamma) + B(\gamma)Q(\gamma) = D(\gamma)$$
$$F(\gamma)T(\gamma) + B(\gamma)R(\gamma) = D(\gamma) \quad (14)$$

The second polynomial of (14) is not used for FBFW control loop. Choice of polynomial $D(\gamma)$ determines process behavior of a closed control loop. The suitable selection of $D(\gamma)$ give us various modifications of the controller.

3.3 Controller Synthesis

One of the approaches is a pole placement controller. Let the characteristic polynomial be described in form

$$D(\gamma) = (\gamma-\alpha)^2[\gamma-(\alpha+j\omega)][\gamma-(\alpha-j\omega)] \quad (15)$$

The characteristic polynomial (15) has a double real pole $\gamma_{1,2} = \alpha$ within interval $0 \le \alpha < -1/T_0$ and a pair of complex conjugate poles $\gamma_{1,2} = \alpha \pm j\omega$. The parameter α influences the speed of the control-loop transient response and influences controller output changes, too. By changing the parameter ω the desired overshoot can be influence.
Controller synthesis is based on a solving of the pair of equations (13) and then it is possible to write

$$\begin{aligned}&\left(\gamma^2 + \bar{a}_1\gamma + \bar{a}_0\right)\left(p_1\gamma + p_0\right) + \left(\bar{b}_1\gamma + \bar{b}_0\right)\left(q_1\gamma + q_0\right) = \\ &\quad = (\gamma - \alpha)^2[\gamma - (\alpha + j\omega)][\gamma - (\alpha - j\omega)] \\ &\gamma\left(t_2\gamma^2 + t_1\gamma + t_0\right) + \left(\bar{b}_1\gamma + \bar{b}_0\right)\left(r_0\right) = \\ &\quad = (\gamma - \alpha)^2[\gamma - (\alpha + j\omega)][\gamma - (\alpha - j\omega)]\end{aligned} \tag{16}$$

By comparing the coefficients with the same powers of γ we obtain a pair of matrixes

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ \bar{a}_1 & 1 & \bar{b}_1 & 0 \\ \bar{a}_0 & \bar{a}_1 & \bar{b}_0 & \bar{b}_1 \\ 0 & \bar{a}_0 & 0 & \bar{b}_0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_0 \\ q_1 \\ q_0 \end{bmatrix} = \begin{bmatrix} -4\alpha \\ 6\alpha^2 + \omega^2 \\ -2\alpha\left(2\alpha^2 + \omega^2\right) \\ \alpha^2\left(\alpha^2 + \omega^2\right) \end{bmatrix} \tag{17}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \bar{b}_1 \\ 0 & 0 & 0 & \bar{b}_0 \end{bmatrix} \begin{bmatrix} t_2 \\ t_1 \\ t_0 \\ r_0 \end{bmatrix} = \begin{bmatrix} -4\alpha \\ 6\alpha^2 + \omega^2 \\ -2\alpha\left(2\alpha^2 + \omega^2\right) \\ \alpha^2\left(\alpha^2 + \omega^2\right) \end{bmatrix}$$

The result of the calculation gives us the parameters of controller which are used in the final form of the controller (18).

$$\begin{aligned} u(k) &= u_1(k) - u_2(k) \\ u_1(k) &= \frac{T_0 r_0 w(k) + p_1 u_1(k-1)}{p_1 + T_0 p_0} \\ u_2(k) &= \frac{q_1 y(k) - q_1 y(k-1) + q_0 T_0 y(k) + u_2(k-1) p_1}{p_1 + T_0 p_0} \end{aligned} \tag{18}$$

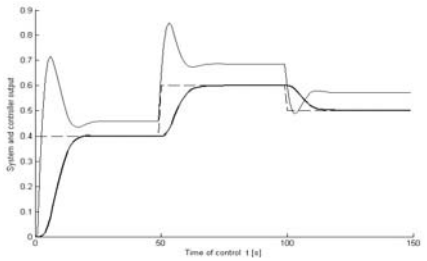

Fig. 4.: Final result for sampling period $T_0 = 1s$

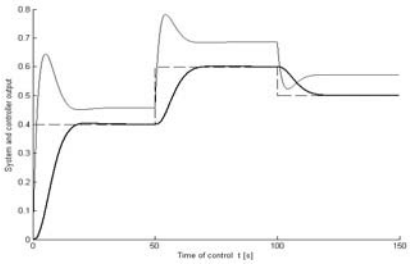

Fig. 5.: Final result for sampling period $T_0 = 0.01s$

4. CONCLUSIONS

This contribution presents application MatlabLink which make accessible and derive benefit from the computational program Matlab via internet. This application is useable for the remote process control of real laboratory plants. The authorized users can monitor or command the remote process control, the others can just only monitor. Although the functionality of this application was presented in connection with delta adaptive control, It is possible to use it generally. Delta adaptive control is presented in the example for wide range of sampling period (Fig. 4 and Fig. 5). Presented example uses controller PID-ZP2 (18) and RLSM with directional forgetting.

ACKNOWLEDGEMENTS

This work was supported by the GAČR under grant No. 102/05/0271, and by the Ministry of Education of the Czech Republic under grant No. MSM 7088352101.

REFERENCES

Middleton, R. H. and G. C. Goodwin (1990). *Control and Estimation - A Unified Approach,* Prentice Hall, Englewood Cliffs, New Jersey.

Feuer, J. and G. C. Goodwin (1984). *Sampling in Digital Signal Processing and Control,* Birkhäuser, Boston.

Goodwin, G. C. and Kwai Sang Sin (1984). *Adaptive filtering prediction and control,* Prentice Hall, Englewood Cliffs, New Jersey.

Bobál, V., J. Böhm, J. Prokop and J. Fessl (2000). *Practical Aspects of Self-Tuning Controllers: Algorithms and Implementation,* VUTIUM Press, Brno University of Technology.