Real-time simulation environment for control loop performance monitoring using Matlab

Riku Pöllänen*, Ari Koponen, Manu Huttunen, Olli Pyrhönen
Department of Electrical Engineering
Lappeenranta University of Technology
P.O. Box 20, 53851 Lappeenranta, Finland

Abstract

Evaluating control loop performance has attracted considerable interest in recent years. Different kinds of performance indices have been developed and tested with simulation models and offline process data before implementing them in real automation system or performance monitoring tool. For developing and simulation stages of process control and control performance monitoring, Matlab is often regarded as an efficient tool among both academic researchers and industry. As a major drawback, however, there have been only reduced possibilities to connect it directly to the distributed control system (DCS) and programmable logic controllers (PLCs) of the plant for testing the designed algorithms with the real application. An easy way to establish a link between DCS or PLC and plant or process simulator running in an ordinary PC hardware is provided by OPC (OLE for Process Control) data access specification. Now, the same data exchange principle is emerging to be applied also with Matlab-based control engineering tools. In this paper, a simulation environment consisted of Matlab-based process simulation, control and monitoring tools and connected to the commercial PC automation system by the aid of OPC is demonstrated. A summary of typical performance indices and case studies of real-time control performance assessment are given.

1 Introduction

While the number of control loops in industries is continuously increasing, the supervision and maintenance of these control loops has become very time consuming and challenging task. The demands for the process operators are not, however, set by the complexity of the controller structures because typically over 90% of the control loops in a process-control plant are implemented using traditional PI-controller. Nevertheless, surveys carried out in 90s have revealed that only about 20% of the control loops were found to work well and decrease process variability [1]. As a motivation to maintain and improve the performance of the control loops, the improvement of the product quality, increased productivity, reduction of energy and raw material consumption and longer life span of the process actuators can be mentioned.

Evaluating control loop performance has attracted considerable interest in recent years. Different kinds of performance indices have been proposed and reported by several researchers. The evaluation methods can be divided into two categories: stochastic and deterministic [2]. Many reported industrial applications of control performance monitoring is mainly based on the principle of the Minimum Variance (MV), a stochastic method suggested by Harris [3]. In this method the actual process output variance is related to the minimal achievable variable, which is estimated using only the measurement of the process output and the process time-delay. Another loop monitoring system based on similar ideas have been proposed in [4]. A modified version of Harris' method has been proposed by Horch [5].

Deterministic methods are useful especially when the transient behaviour of the control loops is to be evaluated but they can be also applied to detect the steady-state performance. Various dimensionless indices have been presented in the literature, see e.g. [6], [7] and [8]. Oscillations in the control system can be identified by the methods proposed by Thornhill [9] and Hägglund [10]. Hägglund [11] presented also a measure, the Idle index, to describe the sluggishness of the control loop.

The recent development of information technology in automation has been shifting from DCSs towards PC-

---

*Corresponding author. E-mail: `Riku.Pollanen@lut.fi` based systems and the specifications for open connectivity. These together with the development of simulation tools and the increase of computing power have made it possible to extend the use of simulation-aided methods in process control engineering. Applications of dynamic process simulators connected with DCS using OPC standard for operator training, automation design and testing purposes have already been reported e.g. in [12–14]. Lately, the OPC connectivity has been released also for Matlab by several vendors facilitating thus the integration of the widely-used development environment with practically any automation system.

This paper presents a real-time simulation environment consisted of Matlab-based process simulator, controller and monitoring tool connected to the commercial PC automation system by the aid of OPC. The environment is used for demonstration, testing and evaluation of different control performance monitoring indices. The presentation is organized as follows. In chapter two the indices for monitoring the control loop performance are briefly summarized. The architecture and implementation of the real-time simulation environment is presented in chapter three. Experimental and numerical results are given in chapter four, and conclusions are drawn in chapter five.

2 Control loop performance monitoring

The assessment of the performance of the process under feedback control, shown in Fig. 1, is usually considered in two different states: in a transient situation due to a set point change or an occurrence of a load disturbance and in normal steady-state operation. Steady-state indices may be calculated continuously, whereas the set point change and load disturbance rejection indicators are typically evaluated only for a certain time period after the detection of the set point change or the disturbance. Different indices can be selected for monitoring and to assess the control performance in these two states.

2.1 Steady-state indices

The purpose of the steady-state indices is to evaluate the control performance in the case of a non-varying set point. The following steady-state indices were programmed in the process monitoring tool discussed in chapter 3.2.

Oscillation index [10] can be used to detect the oscillations of the process output around the set point. The index is calculated by monitoring the values of Integral of Absolute Error (IAE) between consecutive zero crossings of the control error

$$IAE_i = \int_{t_{i-1}}^{t_i} |e(t)|\,dt = \int_{t_{i-1}}^{t_i} |r(t) - y(t)|\,dt \qquad (1)$$

where t_{i-1} and t_i are two successive instances of zero crossings. The oscillation index is calculated recursively by using the forgetting factor γ as

$$OSC_i = \gamma OSC_{i-1} + (1-\gamma)\,DIST_i \qquad (2)$$

where the value of $DIST_i$ depends on the IAE_i and the predefined limit $IAE_{\lim}$ as

$$DIST_i = \begin{cases} 1, IAE_i \geq IAE_{\lim} \\ 0, IAE_i < IAE_{\lim} \end{cases} \qquad (3)$$

The forgetting factor is defined as

$$\gamma = 1 - \frac{1}{5\tau} \qquad (4)$$

where τ is an estimate of the time constant of the process [2].

Integral of the Squared Error (ISE) can be used to monitor the stochastic variations around the set point value, which are too short-term to be observed by the oscillation index. The on-line calculation of the ISE index can be done recursively as

$$ISE_i = \gamma ISE_{i-1} + (1-\gamma)\,e(t)^2 \qquad (5)$$

Contrary to the classical ISE the index evaluated by Eq. (5) converges to zero if the control error is zero. Large control deviations are emphasized by the ISE index.

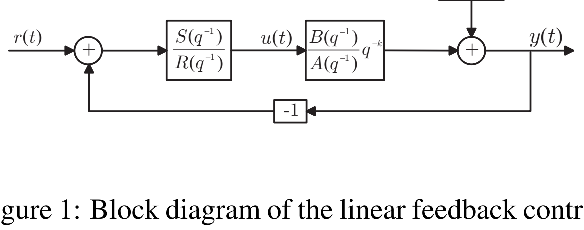


Figure 1: Block diagram of the linear feedback control system.

To measure the amount of control effort an index denoted as ISU can be used. It is calculated similar to ISE index

$$ISU_i = \gamma ISU_{i-1} + (1-\gamma)\left[u_i - u_{i-1}\right]^2 \tag{6}$$

Steady-state error or Permanent Error (PE) between set point and controlled process output can be monitored with index

$$PE_i = \gamma PE_{i-1} + (1-\gamma)\, p_i \tag{7}$$

where

$$p_i = \begin{cases} -1,\, e_i < -e_{\mathrm{lim}} \\ 0,\, -e_{\mathrm{lim}} < e_i < e_{\mathrm{lim}} \\ 1,\, e_i > e_{\mathrm{lim}} \end{cases} \tag{8}$$

e_{lim} is the user-definable largest acceptable value of the control error. Zero value of the PE index will denote that the control error is inside the tolerance band, but when the value approaches either +1 or -1 it means that permanent error between set point and control variable exists. The sign of the index denotes whether the error is positive or negative.

Another useful indicator for evaluating the steady-state error is to calculate the expected or the mean value of the measured process output. Recursively this can be performed by

$$\mu_{y,i} = \frac{i-1}{i}\mu_{y,i-1} + \frac{1}{i} y_i \tag{9}$$

which actually represents an adaptive low-pass filter, the pole of which approaches to one but never reaches it.

Last two steady-state indices calculated in the process monitoring tool are the variance of the process output and the Minimum Variance (MV) index [3]. An equation suitable for on-line calculation of the process output variance can be formulated as

$$\sigma^2_{y,i} = \frac{i-1}{i}\sigma^2_{y,i-1} + \frac{1}{i-1}\left[y_i - \mu_{y,i}\right]^2 \tag{10}$$

The convergence of the variance close to the final value takes a while, because of the settling time of the mean value calculation. The Minimum Variance index is defined by

$$MV = \frac{\sigma^2_{\mathrm{mv}}}{\sigma^2_{y,i}} \tag{11}$$

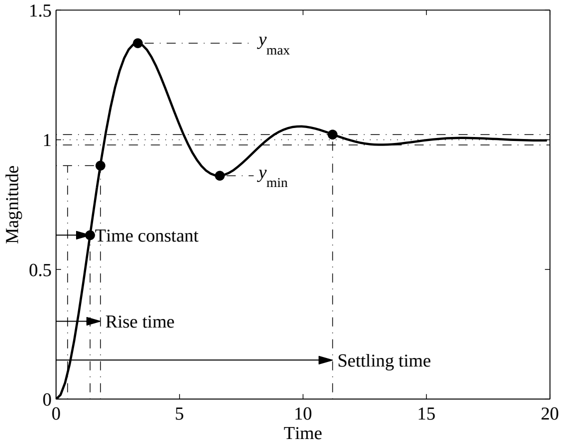


Figure 2: Set point step response and key values to be determined for performance evaluation.

where σ^2_{mv} is the estimate of the minimum achievable output variance. Harris [3] showed that the minimum achievable variance can be calculated from

$$\sigma^2_{\mathrm{mv}} = \left(h_0^2 + h_1^2 + \ldots + h_{k-1}^2\right)\sigma_v^2 \tag{12}$$

where h_i are the parameters of the impulse response of the closed loop transfer function $G_{vy}(q^{-1})$ from the zero mean white noise $v(t)$ with variance σ_v^2 to process output $y(t)$ and k is the time delay of the process (see Fig. 1). An estimate of σ_v^2 is obtained from the identification of $G_{vy}(q^{-1})$ using a prediction error method as the final value of the loss function [15].

2.2 Set point change indices

Four indices have been selected for evaluating the control performance in a set point change. Calculation of these indices is started when a set point change is observed and it is performed for a predetermined time period defined in the monitoring tool parameters by the user. Usually a multiple of the process time constant estimate is recommended.

The key values to be determined from the measured process response to a set point change are the rise time t_{r}, the settling time t_{s}, approximation of the closed-loop time constant τ_{cl} and the maximum and minimum values of the process output after the rise time as illustrated in Fig. 2.

Performance indices evaluated for a set point change are the dimensionless indices for rise time and settling time expressed as follows

$$SPD = \frac{t_r}{\tau} \tag{13}$$

$$TIME = \frac{t_s}{\tau} \tag{14}$$

and an index describing the relative size of the overshoot

$$AMP = \frac{y_{\max} - y_{\min}}{\Delta r} \tag{15}$$

where Δr is the magnitude of the set point change. It should be noticed that the time constant used in calculation of the dimensionless indices SPD and $TIME$ is the open-loop time constant τ of the process, not the estimated one τ_{cl} for the closed-loop system.

Process monitor program calculates also an index, denoted here as a saturation index SAT, which describes the time t_{act} that the actuator utilisation u_{act} is greater than 90 % or smaller than 10 % of the actuator dynamic range u_{dr} with respect to the time needed to carry out the set point change. Here the estimated closed-loop time constant τ_{cl} is used as this reference time value. In [2] the approximation of the time constant τ was applied as a reference time and the index was referred to as a valve capacity index. The saturation index is therefore calculated as

$$SAT = \frac{\int_0^{\tau_{cl}} t_{act} dt}{\tau_{cl}} \tag{16}$$

where

$$t_{act} = \begin{cases} 0, u_{act} \in [0.1u_{dr} \ldots 0.9u_{dr}] \\ 1, u_{act} < 0.1u_{dr} \vee u_{act} > 0.9u_{dr} \end{cases} \tag{17}$$

Values close to one indicate improper actuator sizing while the values near zero show a correct dimensioning.

3 Real-time simulation environment

3.1 System architecture

The real-time simulation environment consists of four main parts: Matlab/Simulink software, TwinCAT PLC and OPC Server and IPCOS OPC interface for Matlab (OPC client). These components and required subcomponents are installed in one RedHat 6.2/RTLinux 2.0 and two Windows 2000 workstations (WS) as described in Fig. 3. WS1 operates as a host PC for the software-based PLC while the process simulator is running on WS2. OPC server is located on the WS1 while both WS1 and WS3 contains the OPC clients. Process simulator is based on the Simulinux, a proprietary program from Quanser that allows development of RTLinux code using Matlab. The process simulator is connected to the I/Os of the PLC through the Quanser's 8 channel hardware in the loop I/O board and twisted pair cabling. The I/O system of the PLC is based on the PROFIBUS-DP fieldbus. The OPC server and the client of the WS3 are networked through Ethernet and TCP/IP protocol.

3.2 Process monitoring tool

Process monitoring tool in WS3 is implemented in Matlab using its own programming language and the Graphical User Interface Development Environment (GUIDE). The program has the following three main functionalities: 1) initialization and creation of the OPC connection, 2) Minimum Variance (MV) Tool and 3) the process monitoring, the last of which includes e.g. functions for the data request and calculation of selected performance indices. The GUI of the process monitoring tool consists of the main view shown in Fig. 4 and five other subviews such as OPC parameter view and MV Tool view presented in Fig. 5. Process monitoring can be performed either online using the OPC connection or off-line for the presampled data. Trends of the selected performance indices are plotted for the user-selectable period of time. Index values are also indicated as numerical values next to the corresponding check boxes and index titles. Monitoring results can be saved into a file for further analysis.

MV Tool is a group of functions needed to estimate the minimum achievable process variance from the process measurements. It may also be used for time series modelling of the processes and for estimating the process delay.

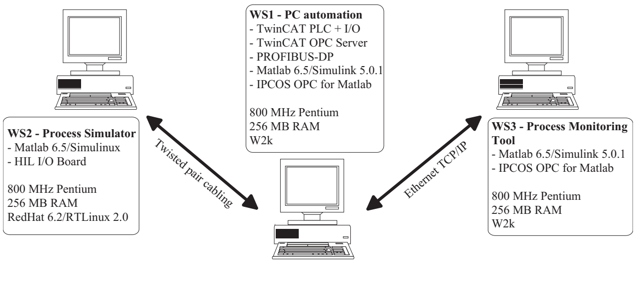
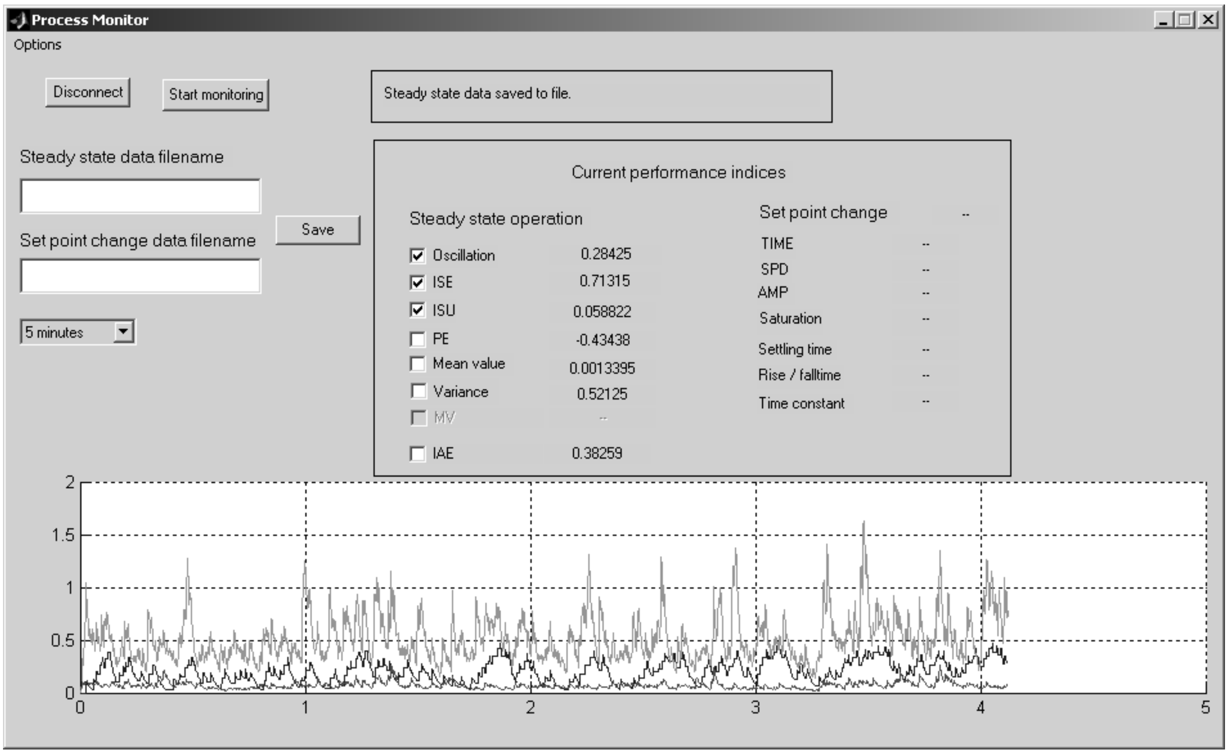


Figure 3: Schematic of simulation system architecture.



Steady state data saved to file.

5 minutes

Current performance indices

Steady state operation

☑ Oscillation 0.28425
☑ ISE 0.71315
☑ ISU 0.058822
☐ PE -0.43438
☐ Mean value 0.0013395
☐ Variance 0.52125
☐ MV --
☐ IAE 0.38259

Set point change --
TIME --
SPD --
AMP --
Saturation --
Settling time --
Rise / falltime --
Time constant --

Figure 4: Main view of the process monitoring tool.

4 Simulations and results

Two different process models were used to test the indices described in chapter 2. For steady-state operation a process model described in [16] was used. The process is described by a discrete time transfer function

$$Y\left(q^{-1}\right) = \frac{0.33}{1-0.67q^{-1}} q^{-4} U\left(q^{-1}\right) + D\left(q^{-1}\right) \quad (18)$$

where U is the input and the disturbance transfer function D is described by the transfer function

$$D\left(q^{-1}\right) = \frac{1-0.4q^{-1}}{1-0.67q^{-1}} v\left(q^{-1}\right) \quad (19)$$

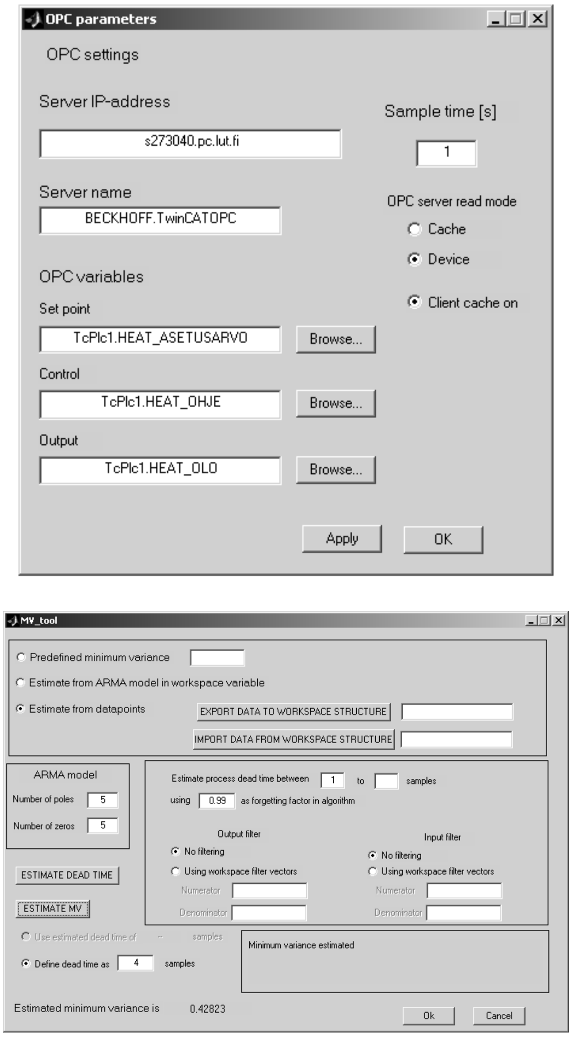
where v is independent white noise sequence with variance $\sigma_v^2 = 0.36$. Because of the architecture of the simulation environment the sampling of the OPC server cannot be synchronized with the process simulator computer. This behaviour poses some problems if a pure discrete time process model were used in the process simulator. The samples produced by the simulator cannot be transferred through the network immediately when they have become available and therefore the signal properties would suffer badly and consequently the information given by the samples would be inaccurate. This would in turn affect for example the minimum variance estimate of the process. To overcome the synchronizing problem the process model (18) was transformed back to the Laplace domain by Matlab's function *d2c* with zero-order-hold as the method and 100 ms as the sample time. This yielded a continuous time transfer function

$$Y(s) = \frac{4.005}{s+4.005} e^{-0.4s} U(s) \quad (20)$$

The PC automation system was then set to sample this process at much faster sample rate (1 kHz) compared to what was used in back-transformation. Now the sampling synchronisation is not an issue anymore because the OPC server can take its samples whenever it wishes and the samples are still representing the current state of the process accurately. The noise transfer function does not have to be transformed. The controller used in the simulations was a PI-controller of the form

$$U(s) = K_{\mathrm{p}} \frac{T_{\mathrm{i}}s+1}{T_{\mathrm{i}}s} E(s) \quad (21)$$

where $E(s) = R(s) - Y(s)$.



Server IP-address: s273040.pc.lut.fi

Sample time [s]: 1

Server name: BECKHOFF.TwinCATOPC

OPC server read mode: ○ Cache ● Device

● Client cache on

OPC variables

Set point: TcPlc1.HEAT_ASETUSARVO

Control: TcPlc1.HEAT_OHJE

Output: TcPlc1.HEAT_OLO

○ Predefined minimum variance

○ Estimate from ARMA model in workspace variable

● Estimate from datapoints

ARMA model

Number of poles: 5

Number of zeros: 5

Estimate process dead time between 1 to samples

using 0.99 as forgetting factor in algorithm

Output filter: ● No filtering ○ Using workspace filter vectors

Numerator

Denominator

Input filter: ● No filtering ○ Using workspace filter vectors

Numerator

Denominator

○ Use estimated dead time of -- samples

● Define dead time as 4 samples

Minimum variance estimated

Estimated minimum variance is 0.42823

Figure 5: Captures of the OPC parameter and MV Tool views.

The reference signal r was set to zero as the model was used to evaluate steady state indices. Parameters of the PI-controller were adjusted using the λ-method [17].

The process model and the controller used to evaluate the setpoint change indices was completely different. This time the process simulator computer carried out only the process modeling. The controllers were programmed into Matlab that resided on the same computer as the OPC server and PC automation system. Control and feedback signals were transferred over the Ethernet network. The reference signal r was directly set from the controller interface and it was also sent to the OPC server for the process monitoring purposes.

The process model used in transient situation was a fictitious heating process depicted in Fig. 6. The control objective of the cascaded control structure was to regulate the outgoing liquid temperature. The transfer function models for the valve and the heater were

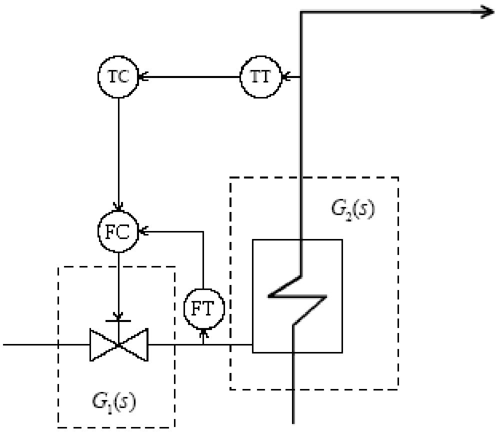


Figure 6: Instrumentation chart for a process to test setpoint change indices

$$G_1(s) = \frac{1}{(s+1)^2(10s+1)} \tag{22}$$

and

$$G_2(s) = \frac{1}{(3s+1)(30s+1)} \tag{23}$$

respectively.

4.1 Steady-state indices

First, the feasibility of the estimation of the minimum achievable variance $\sigma^2_{\mathrm{mv}}$from the data points collected from the OPC server was evaluated. The controller was tuned for $\lambda = 0.1$. The process output variable was observed with four different sample lengths 100, 500, 1000 and 5000 using the sample rate of 100 ms. Each sampling was repeated 100 times. As described in [16] the theoretically calculated minimum variance for the process is $\sigma^2_{\mathrm{mv}} = 0.4033$. The developed MV-tool was used to fit an ARMA time-series model for each of the samples and then the minimum variance estimates $\hat{\sigma}^2_{\mathrm{mv}}$were calculated using the impulse response of the model. Fig. 7 shows the histograms of the estimated minimum variances $\hat{\sigma}^2_{\mathrm{mv}}$ and noise variances $\hat{\sigma}^2_{\mathrm{v}}$ along with the mean values and the standard deviations of the estimates. It can be seen that the mean values are very close to the theoretical one. An even more important matter, however, is that deviation of the estimates decreases as the sample length increases. This leads to a conclusion that OPC can be reliably used in data acquisition for process monitoring purposes.

Next, the rest of the steady-state indices were evaluated. Indices were calculated for the controllers tuned with λ values of 0.1, 1.3, 3 and 7. Fig. 8 shows the simulated closed loop responses for a step change in reference signal. The process monitor application was used to collect data at 10 Hz sample rate from the OPC server and to calculate the indices online. After the experiment, all indices were saved into a file for further analysis. Other user-definable parameters used in calculations were: $\tau = 2\,\mathrm{s}$, $IAE_{lim} = 0.1$ and $e_{lim} = 0.05$. Figs. 9-12 show the indices and some summary statistics for different controllers. All the indices for this particular process seem to share the same behaviour: They decrease in average as the closed loop dynamics are slowed down. This observation might not be so evident by plotting and examining only the time domain trends.

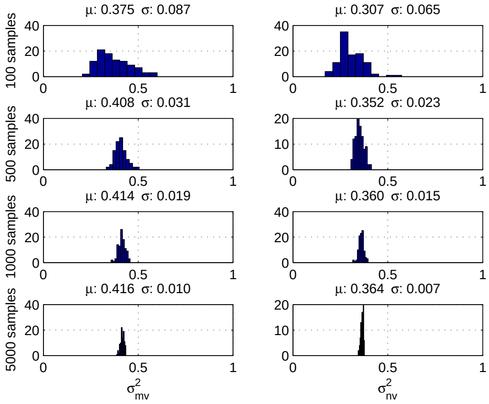


Figure 7: Histograms for the estimated minimum variances in the first column and noise variances in the second column for different sample lengths.

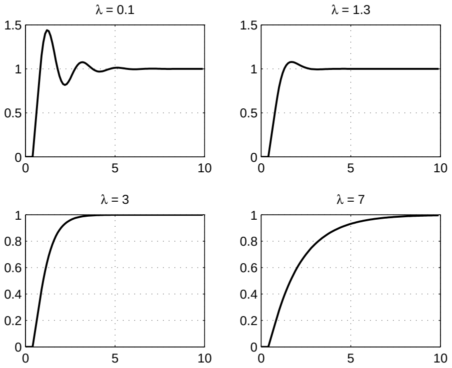


Figure 8: Step responses for each tuning parameters.

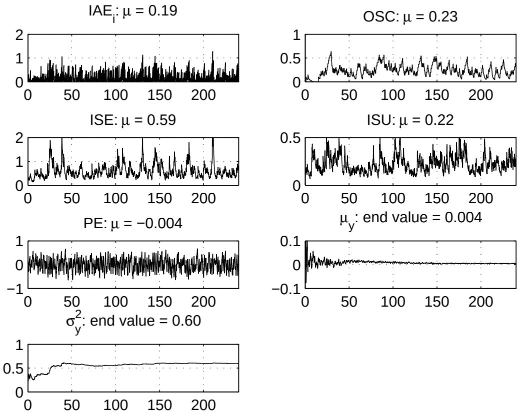


Figure 9: Steady state indices for $\lambda = 0.1$.

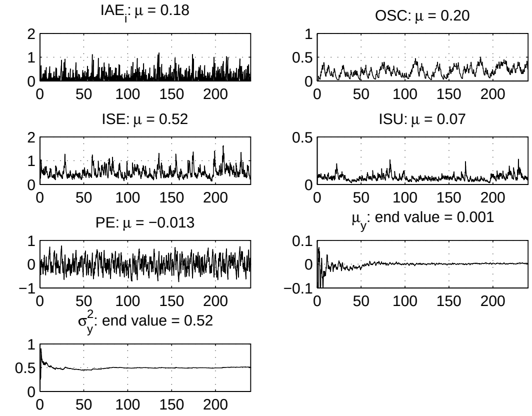


Figure 10: Steady state indices for $\lambda = 1.3$.

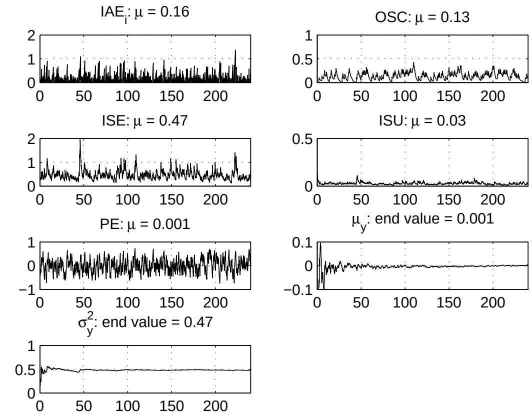


Figure 11: Steady state indices for $\lambda = 3$.

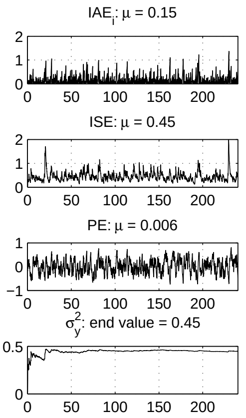


Figure 12: Steady state indices for $\lambda = 7$.

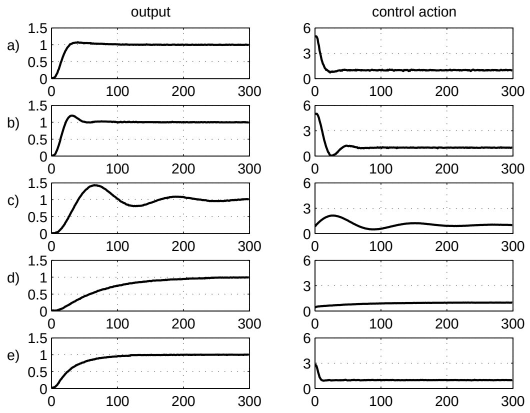


Figure 13: Setpoint change responses for different controller tunings.

4.2 Setpoint change indices

The second process given by Eqs. (22) and (23) and illustrated in Fig. 6 was controlled by two cascaded PI controllers that were tuned to yield different kind of step responses. Different tunings were obtained by varying the proportional gains of the controllers, while the integration time constants were kept the same. For digital implementation the controllers were discretized using the bilinear approximation with sampling time of 0.5 s. During the experiment the process monitoring tool was used to collect the process data with sampling time of 1 s, after which the setpoint change indices were determined. The step responses of the manipulated variable and controlled variable are presented in Fig. 13 and index values are given in Table 1. The indices were calculated using the following parameters: $\tau = 60$ s, $e_{\mathrm{lim}} = 0.05$ and the actuator dynamic range was defined to be $0 \le u_{\mathrm{dr}} \le 5$. According to the results the case b) seems to give the best overall performance due to the fastest response times. It also exhibits an acceptably small oscillating behaviour (AMP), though not the smallest. Control action requirements (SAT index) are seen to be in tolerable range in all the cases.

Although it is a bit questionable how to set the index calculation parameters (e.g. τ) for complex processes having more than one pole or when the process transfer function is unknown, these indices seem to be able to give supporting quantitative measures for the phenomena that can be also seen from the step response figures. These measures might help the control engineer to compare different tuning parameters and select the most appropriate.

Table 1: Setpoint change indices for different controller tunings presented in Fig.13. The smaller the the index the faster (TIME,SPD, etc) or less oscillating (AMP) is the process.

case	TIME	SPD	AMP	SAT	t_{s}	t_{r}	τ_{cl}
a)	0.91	0.28	0.16	0.24	55	17	17
b)	0.74	0.22	0.28	0.33	45	13	15
c)	3.5	0.42	0.62	0	209	25	30
d)	3.5	2.27	0.10	0.03	208	136	76
e)	1.7	1.05	0.10	0	100	63	33

5 Conclusions

Demand for real-time process simulation environments and analysis tools, with which process models, control algorithms and maintenance supporting systems, such as process monitoring applications, can be interconnected and networked with DCS or PLCs has clearly increased among automation system and process control engineers. PC workstation based technologies together with OPC connectivity has turned out to be one competitive answer to this development. In this paper an alternative for a real-time simulation environment was presented. The system consisted of Matlab-based process simulator, controller and performance monitoring tool connected to the commercial PC automation system by the aid of OPC. Case studies of the real-time control performance assessment with developed simulation environment were demonstrated. It was concluded that OPC can be reliably used in data acquisition for process monitoring purposes and that investigated performance indices provide at least supporting quantitative measures for control loop assessment.

References

[1] W. L. Bialkowski, “Dreams versus reality: A view from both sides of the gap,” *Pulp and Paper Canada*, vol. 94, pp. 19–27, 1993.

[2] S.-L. Jämsä-Jounela, R. Poikonen, Z. Georgiev, U. Zuehlke, and K. Halmevaara, “Evaluation of control performance: Methods and applications,” in *Proceedings of the IEEE International Conference on Control Applications*, 18-20 September 2002, pp. 681–686.

[3] T. J. Harris, “Assessment of control loop performance,” *The Canadian Journal of Chemical Engineering*, vol. 67, pp. 856–861, Oct. 1989.

[4] C. B. Lynch and G. A. Dumont, “Control loop performance monitoring,” vol. 4, pp. 185–192, Mar. 1996.

[5] A. Horch and A. J. Isaksson, “A modified index for control performance assessment,” *Journal of Process Control*, vol. 9, pp. 475–483, 1999.

[6] K. Åström, C. Hang, P. Persson, and W. Ho, “Towards intelligent pid control,” *Automatica*, vol. 28, pp. 1–9, 1992.

[7] A. P. Swanda and D. E. Seborg, “Controller performance assessment based on setpoint response data,” in *Proceedings of American Control Conference*, October 1999.

[8] H.-P. Huang and J.-C. Jeng, “Monitoring and assessment of control performance for single loop systems,” *Industrial and Engineering Chemistry Research*, vol. 41, pp. 1297–1309, 2002.

[9] N. F. Thornhill and T. Hägglund, “Detection and diagnosis of oscillation in control loops,” *Control Engineering Practice*, vol. 5, pp. 1343–1354, 1997.

[10] T. Hägglund, “A control-loop performance monitor,” *Control Engineering Practice*, vol. 3, pp. 1543–1551, 1995.

[11] T. Hägglund, “Automatic detection of sluggish control loops,” *Control Engineering Practice*, vol. 7, pp. 1505–1511, 1999.

[12] T. Karhela, M. Paljakka, P. Laakso, T. Mätäsniemi, J. Ylijoki, and J. Kurki, “Connecting dynamic process simulator with distributed control system using OPC standard,” in *Proceedings of TAPPI Process Control, Electrical and Information Conference*, March 1999, pp. 329–337.

[13] J. Lappalainen, S. Tuuri, T. Karhela, J. Hankimäki, P. Tervola, S. Peltonen, T. Leinonen, E. Karppanen, J. Rinne, and K. Juslin, “Direct connection of simulator and dcs enhances testing and operator training,” in *Proceedings of TAPPI Engineering/Process and Product Quality Conference*, September 1999, pp. 495–502.

[14] J. Rinta-Valkama, M. Välisuo, T. Karhela, P. Laakso, and M. Paljakka, “Simulation aided process automation testing,” in *Proceedings of IFAC Conference on Computer Aided Control System Design CACSD*, September 2000, pp. 277–280.

[15] P.-G. Eriksson and A. J. Isaksson, “Some aspects of control loop performance monitoring,” in *Proceedings of the Third IEEE Conference on Control Applications*, 24-26 August 1994, pp. 1029–1034.

[16] T. J. Harris, C. T. Seppala, and L. D. Desborough, “A review of performance monitoring and assessment techniques for univariate and multivariate control systems,” *Journal of Process Control*, vol. 9, pp. 1–17, 1999.

[17] K. J. Åström and T. Hägglund, *PID Controllers: Theory, Design and Tuning*, 2nd ed. Instrument Society of America, 1995.