Enhancement of Model Predictive Control Implementation on a DCS PCS7

Hanieh Asadipooya
Department of Power and Control Engineering
School of Electrical and Computer Engineering
Shiraz University
Shiraz, Iran, 7134851154
Email: Anipooya@yahoo.com

Ali Akbar Safavi
Department of Power and Control Engineering
School of Electrical and Computer Engineering
Shiraz University
Shiraz, Iran, 7134851154
Email: Safavi@shirazu.ac.ir

Abstract- **It is well-known that the implementation of Advanced Process Control (APC) techniques such as Model Predictive Control (MPC) can enhance the performance of the plant operations in comparison to the common PID controller. However, the limited capabilities of most Distributed Control Systems (DCS) and their simple available controller libraries do not allow flexible and effective implementations of many advanced control approaches. The main purpose of this paper is to focus on simple and effective implementation of MPC on a Siemens DCS called "PCS7". While PCS7 allows a limited version of MPC called Dynamic Matrix Control (DMC) (i.e. available in its library), our aim is to provide some new tools on PCS7 for simpler and faster implementations of more general versions of MPC with better performances. First, the available DMC block is invoked for a case study. Then, some new blocks for more advanced MPC implementations using SCL (Structured Control Language) programming language are developed and provided. The simulation results show the effectiveness of the newly developed blocks. Besides, it is shown that the new blocks has lower hardware and software limitations.**

Keywords: Model Predictive Control (MPC); Distributed Control System (DCS); PCS7; Structured Control Language (SCL)

I. INTRODUCTION

Distributed control systems (DCS) have become the industry standard of automatic control especially for large-scale and complex operations. Each DCS contains a data highway connecting control modules to other hardware units such as operator stations, printers, and gateways to local control networks [1]. The main advantage of DCS lies in the functional distribution of control in a way that each control module performs one control task with a redundancy measure. The control module is similar to a mini-computer except that it is pre-programmed with a set of standard algorithms of which PID series are mostly utilized [2]. However, PID control performance is limited and PID design for a multi-input multi-output process is complicated and cannot guarantee the best performance [3].

Other controllers rather than PID such as artificial neural network control, fuzzy control, and identification- free adaptive control have difficulties in implementations and time-critical control with DCS for their complicated control laws and massive computation involved [4-6].

The above-mentioned issues trigger the development of MPC which is one of APC techniques [7]. APC methods are practical tools to improve plant performance with respect to productivity and economics, operability and availability, product quality, agility, and safety and environmental issues [8]. A major obstacle in successful implementation of advanced controllers in the DCS is the limited support in terms of hardware, software, and personnel training [9].

While the knowledge of MPC has advanced and now the technology makes it easier to apply, there are still significant limitations on hardware and software for the implementation of many MPC algorithms. The success rate of MPC across the industry is uneven. Some companies are consistently successful in deploying MPC, whereas others are not [10].

Meanwhile for many customers (i.e. engineering companies, system integrators and facility operators) it is of high importance that MPC techniques are increasingly offered as embedded functions in many DCSs [11] (e.g. Siemens in SIMATIC PCS7). PCS7 with its integrated data storage, communication and configuration offers an open basis for modern, future-oriented and economical automation solutions in all sectors of the processing industry.

In PCS 7, there is a block allocated for MPC, named ModPreCon, which implements the DMC algorithm. However, this block does not allow the implementation of more complete methods than aforesaid MPC. Fortunately, in PCS7 software, some functions can be created for each control operation which can be written in SCL programming language [12]. Hence, the implementation of an MPC controller requires creating some functions in SCL without using PCS7 library blocks, and consequently define them as new function blocks which draws some pitfalls in turn. For instance, as MPC controller is concerned, matrix calculations are necessarily basic building blocks but the SCL generally recognizes and processes floating point variables, and matrix variables are not defined in this language. Besides, there is no possibility for matrix calculations in PCS7 library blocks. Another technical difficulty is that the implementation of transfer functions along with the controlling of processes with complex poles are not feasible [13].

In the present study, we aim to add several new blocks to PCS7 using SCL which is more complete and efficient methods than ModPreCon block and thus more flexibility can be utilized. Therefore, in order to overcome the first problem, matrix calculations are added to the software by using SCL. Considering the second problem, since the implementation of complex variables calculations besides MPC controller imposes too much load on the CPU, creating a new block for state-space system model is feasible via matrix calculations which are possible in this study due to MPC requirement. This makes the process with complex poles applicable and hampers the execution of load on the CPU.

In many cases, economic objectives are achieved in reduction of controlled variable variances and the time reaching the setpoint by improved process control which is one of the goals of the new blocks in this paper. In the context of setpoint step changes, the new MPC block will have much stronger advantages over the blocks in the PCS7 library in a way that the former is able to react faster in a process with real poles without oscillation, control process with complex poles, and to impose less processing load on the CPU in contrast to available blocks in the PCS7 library.

The remainder of this paper is organized as follows. MPC and DCS are briefly introduced Section II. This paper proposes an implementation of predictive control using SCL which is well suited to run on standard DCS hardware of the type that already exists in most process plants. A detailed implementation method of MPC on DCS along with hardware and software limitations are presented in Section III comparison of simulation results and the impact of changing control parameters are discussed in Section IV and finally, the concluding remarks are included in Section V respectively.

II. PRELIMINARIES

A. Model Predictive Control

This section briefly presents the elements common in all MPC controllers, state space formulation of MPC with embedded integrator and reaching control law.

1. MPC Elements

All MPC algorithms possess various elements in common, and different options which can be selected for each element giving rise to different algorithms. These elements fall in three different categories as follows:

- prediction model,
- performance index, and
- obtaining the control law [14]

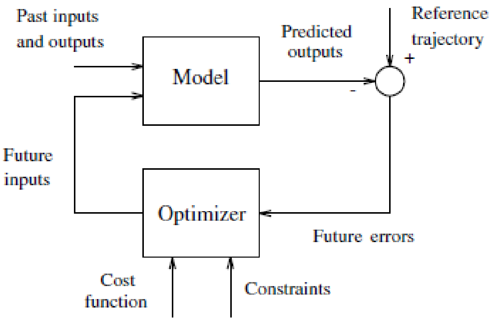

Figure 1. Basic structure of MPC [15]

2. State Space Formulation

MPC systems are designed according to a mathematical model of the plant. State space models can also be used to formulate the predictive control problem. The main theoretical results of MPC related to stability come from a state space formulation, which can be utilized for both mono-variable and multivariable processes and can definitely be extended to nonlinear processes. The following equations are used in the linear case to capture process dynamics [14-16]:

$$x_m(k+1) = A_m x_m(k) + B_m u(k) \quad (1)$$

$$y(k) = C_m x_m(k) \quad (2)$$

where u is the manipulated variable; y is the process output; and x_m is the state variable vector with assumed dimension $n1$. This model allows the process response to be predicted over a defined period in the future.

Since the detailed derivation has been well documented in the literature [14-16], only a summary of the contents is shown below:

According to the predicted state variables, the predicted output variables are defined as follows:

$$Y = Fx(k_i) + \phi \Delta U \quad (3)$$

$$Y = \left[y(k_i+1|k_i)\, y(k_i+2|k_i)\, y(k_i+3|k_i) \ldots y(k_i+N_p|k_i) \right]^T$$

$$\Delta u = \left[\Delta u(k_i)\, \Delta u(k_i+1)\, \Delta u(k_i+2) \ldots \Delta u(k_i+N_C-1) \right]^T$$

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ . \\ . \\ . \\ CA^{N_p} \end{bmatrix} \qquad \phi = \begin{bmatrix} CB & 0 & 0 & ... & 0 \\ CAB & CB & 0 & ... & 0 \\ CA^2B & CAB & CB & ... & 0 \\ . & . & . & ... & . \\ . & . & . & ... & . \\ . & . & . & ... & . \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & ... & CA^{N_p-N_c}B \end{bmatrix}$$

3. Optimization

Based on the prediction in (3), a performance index is minimized as follows:

$$J = (R_s - Y)^T W (R_s - Y) + \Delta U^T \bar{R} \Delta U \quad (4)$$

Where;

- R_S contains the time series of the future setpoints,
- Y contains the vector of the outputs in the future,
- ΔU contains the future changes to the manipulated variable.
- $\bar{R}$ and W are diagonal matrices that consider the future behavior and usually have constant values.

The optimal ΔU that will minimize J is found by substituting the equation No. 3, from the first derivative of the cost function J:

$$\frac{\partial J}{\partial \Delta U} = 0 \Rightarrow \Delta U = \left(\phi^T \phi + \bar{R}\right)^{-1} \phi^T W \left(R_s - Fx(k_i)\right)$$

The matrix $\left(\phi^T \phi + \bar{R}\right)^{-1}$ is called the Hessian matrix in the optimization literature.

The optimal solution of the control signal is associated with the setpoint signal $r(k_i)$ and the state variable $x(k_i)$ via the following equation:

$$\Delta u = \left(\phi^T \phi + \bar{R}\right)^{-1} \phi^T W \left(\bar{R}_s r(k_i) - Fx(k_i)\right) \quad (5)$$

$$R_s^T = \overbrace{\begin{bmatrix} 1 & 1 & . & . & . & 1 \end{bmatrix}}^{N_P} r(k_i) = \bar{R}_s r(k_i)$$

III. IMPLEMENTATION

Due to the principle on which MPC works, the runtime for MPC controller is significantly longer than that for PID controller, because the matrix calculations in the algorithm are much more complex. This is why the MPC controller is unqualified for rapid control and is usually used for slow but complex control tasks. Despite the benefits of MPC, the calculation time load caused by the MPC block is also too much, so CPUs with high processing speed are required [17].

A. Model Predictive Control Implementation using PCS7 blocks

In order to implement MPC, the blocks have to be inserted from PCS7 library into the Control Function Chart (CFC), and be configured there. CFC is a feature to configure continuous processes in a plant which exists in PCS7.

In this paper, several categories of blocks are used. First, in order to receive signal from the process, the analog input driver PCS7AnIn is connected to the controlled variable. Second, manipulated variable has to be connected to the periphery via the analog output driver PCS7AnOu. Third, the AutoExcitation block is interconnected with ModPreCon block to generate suitable excitation signals for the identification of dynamic processes. The ModPreCon block is an MPC controller. In this section, the transfer function from input to output, which can be implemented by PCS7 library blocks is used. In the following, the ModPreCon block will be explained in detail.

1. Description of ModPreCon Block

The ModPreCon block is an MPC controller. The ModPreCon algorithm just works for stable processes with a step response that settles to a fixed value in a finite time. This block is derived from the familiar DMC algorithm, and contains the analytic solution of the optimization problem.

Future changes to the manipulated variable within the control horizon are calculated according to the formula:

$$\Delta \vec{\underline{u}} = \underline{C}.(\vec{\underline{s}} - \vec{f}) \qquad (6)$$

- s contains the time series of the future setpoints
- f contains the predicted free movement of the controlled variables (with constant manipulated variables) in the future
- C is the constant controller matrix calculated by the MPC controller. C includes both the process model and the weighting of the manipulated variable changes and the controlled variables from the objective function of the optimization.

B. MPC implementation via creating new blocks

In this section, the program is subdivided into a number of blocks, each responsible for a specific subtask by using a modular design. The source codes are in SCL. SCL is a PASCAL-oriented language for creating your own user blocks in accordance with IEC 61131-3 for SIMATIC PCS7. SCL is particularly suitable for programming complex algorithms and mathematical functions [13].

SCL is basically capable of processing basic mathematical functions for floating-point variables, neither for matrix variables nor complex variables. In this paper, first, matrix variables are defined in SCL, then functions and blocks for some matrix calculations such as matrix summation and multiplication, moving the rows and columns of a matrix, trace, transpose, matrix inversion and etc. which are necessary in the implication of MPC controller are designed using linear algebra and numerical calculations.

SCL can be used for any possible complex algorithm, if it is broken into small feasible tasks through basic instructions. One should be careful about designing a complex algorithm in order not to impose too much computational load on the CPU. According to this problem and the calculation load caused by the MPC, the state-space model for process control with complex poles is recommended here to satisfy our goal.

The proposed MPC block uses a quadratic cost function and state-space model of the system for calculating control law. To implement the new MPC block on the desired process and to apply the control signal to the process in order to achieve the desired output, the blocks related to state-space model of the system, updating the states and new MPC block should be placed in the CFC. Each of these blocks in SCL uses several functions which had been written formerly. For ease of use, the required functions are called in several specific blocks and for any purpose, it is just necessary to put the designed block into the CFC. Controller design is performed automatically by the created software tools. The user has to only specify N_P prediction horizon and W weight parameter in the performance index as described in Section II. The new MPC block has monitoring capabilities and its trends are displayed in PCS7 OS.

C. CPU Resources

The free memory space in SIMATIC CPU should be verified before loading the MPC block into target system and make sure that the cycle load of the CPU is not close to the constraints before inserting the MPC block. In this study, a new block is designed in SCL, which calculates the occupied memory with a program. By using this block and the prior knowledge of the capacity CPU memory, the free memory can be computed.

In the new blocks, the programs are written in a way that the amount of occupied memory, the calculation time and the computational load imposed on the CPU is as low as possible. This results in accelerating the output time to reach the setpoint with less oscillations in input and output which prevents from system failure compared with common MPC block in the PCS7 library. Because of the increased speed of running program, the proposed MPC block can control both fast and slow processes, whereas the available MPC block in the PCS7 library can only control slow processes.

IV. RESULTS AND DISCUSSIONS

A. *The simulated controller experiment of the study*

In this section, the control performance of the designated predictive controller is to be checked. Two closed-loop control simulations are used to demonstrate the new MPC block approach. The first simulation will show that the new MPC block has better control performance than ModPreCon block in a process with real poles. The second example is a process with complex poles. The overall process is more complicated because of the complex poles and available blocks in PCS7 cannot implement and control this process. Simulation results are displayed graphically as trends in PCS7 OS in Figs. 3, 4, 5 and 6 which are explained in detail in the following. In all figures, the horizontal axis is related to time, and the vertical axis in the upper half of the figure show the setpoint and controlled variable, while in the lower half of the figure show manipulated variable. For the ease of comparison, the time length of the results displayed is equal and the vertical axes range is the same.

1. *Simulation 1: Process with real poles*

This example is based on an example in [14]. Part of a paper-making machine is shown in Fig. 2. The variables in table I are also involved.

TABLE I. VARIABLES OF PAPER MACHINE HEADBOX

Input (u)	States (x)	Controlled variable (y)
Stock flowrate G_P	Feed tank level H_1 Headbox level H_2 Feed tank consistency N_1 Headbox consistency N_2	N_2

A linear model of this machine has the following state-space matrices:

$$A=\begin{bmatrix} -1.93 & 0 & 0 & 0 \\ 0.394 & -0.426 & 0 & 0 \\ 0 & 0 & -0.63 & 0 \\ 0.82 & -0.784 & 0.413 & -0.426 \end{bmatrix} \quad B=\begin{bmatrix} 1.274 \\ 0 \\ 1.34 \\ 0 \end{bmatrix} \quad C=\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$$

Predictive control is to be applied to this plant.

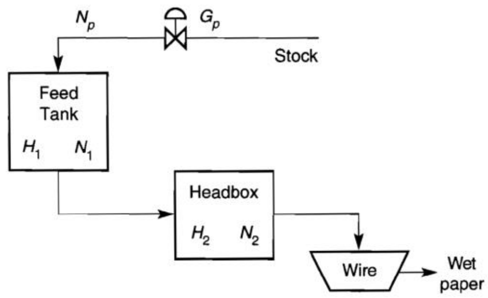

Figure 2. Paper machine with headbox [14]

Both versions of MPC are compared in two different simulations with the same plant and consequently, the changes in the setpoint show changes in the control performance.

a) *Control via Available blocks in PCS7*

The results of step changes in the setpoint with available MPC controller in the PCS7 library can be seen in Fig. 3. This scenario is controlled slowly by ModPreCon block and noticeable deviations from the setpoint arise in output. These deviations are compensated to zero in steady state.

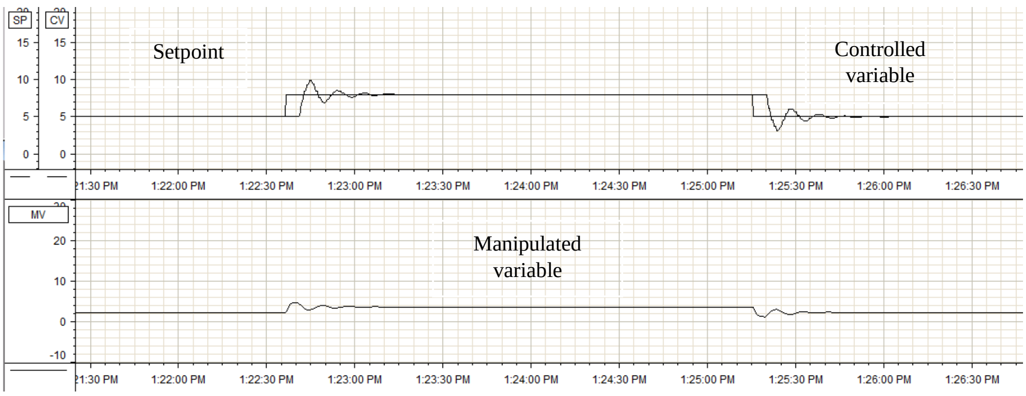

Figure 3. Setpoint step $5 \leftrightarrow 8$% with available MPC controller in PCS7 library

b) *Control via the New Blocks in PCS7*

In the following, control via the new MPC block is considered. The same step changes in the setpoint as before using the available blocks in PCS7 library is done with the new MPC block. Fig. 4 evidently displays that the new MPC block masters this task successfully. The output is adjusted rapidly and without oscillation.

The advantages of the new MPC block compared to the available MPC controller can be seen clearly. Setpoint step changes are controlled by both controllers, but the new MPC block is reacting faster and without oscillation.

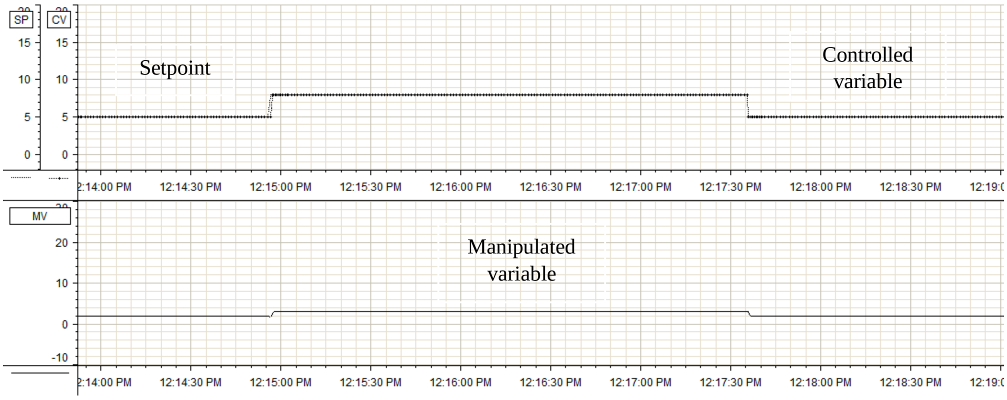

Figure 4. Setpoint step $5 \leftrightarrow 8$% using new MPC block

2. *Simulation 2: Process with Complex Poles*

This section demonstrates the action of the new predictive control block by applying it to a plant with complex poles. The state space model of the plant is given by:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & -1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & -2 & -2 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad C = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}$$

poles: $-1.309 \pm 0.9511j, -0.191 \pm 0.5878j$

As far as the poles are close to $j\omega$ axis, the overall plant is oscillatory.

Fig. 5 shows the new MPC block behavior in setpoint step changes. The control performance is not satisfactory when N_P=4. There are temporary deviations of output despite of step changes in the setpoint.

As shown in Fig. 6 this problem is also controlled by the new MPC block by means of increasing N_P and the overall control performances satisfactorily when N_P=8.

As it can be seen, the output responds more accurately and faster and without fluctuation with larger N_P. This results from the larger dimension of output vector Y which contains predicted outputs. The larger dimension of output vector has influenced the objective function and triggers the larger magnitude of elements in ΔU. As a result, the output outperforms with larger prediction horizon. However, too much computational load would be imposed on the CPU if the prediction horizon experiences a drastic increase. Moreover, it can result in large condition number in the Hessian matrix which endangers the numerical stability of the algorithms in DCS [18].

Furthermore, with extremely large N_C, the dimension of parameters in Eq. (5) becomes larger. This can lead to longer time in calculating control input, due to the limited performance DCS when dealing with large algorithm. Therefore, the sampling time can be longer with a large N_C which in turn, causes the increasing settling time and poor performance.

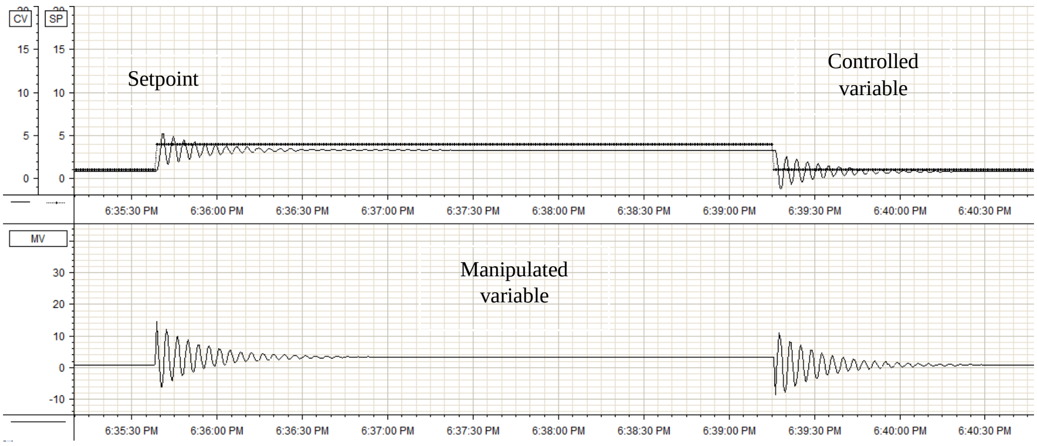

Figure 5. Simulation of process response and control input when setpoint step changes $1 \leftrightarrow 4$ and N_P=4

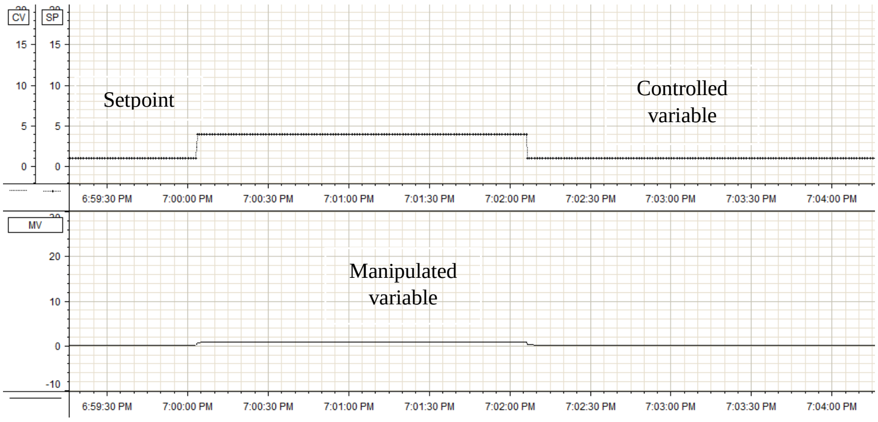

Figure 6. Simulation of process response and control input when setpoint step changes $1 \leftrightarrow 4$ and N_P=8

A comparison of the available MPC controller and the new MPC block can be found in table II.

TABLE II: COMPARISON OF THE AVAILABLE MPC CONTROLLER AND THE NEW MPC BLOCK

Available MPC Controller in PCS7		**Created New MPC block**
Process control with complex poles	-	Very good
Process control with real poles	Slow, With fluctuation	Fast, Without fluctuation
Engineering-effort	High	High
CPU resource-consumption (memory, calculation time)	High	Low
Control performance	Medium	Very good
Processes control with medium rate	-	Very good

V. CONCLUDING REMARKS

This paper concentrated on formulation of MPC design with DCS application. The aim was to enhance the DCS performance in industrial applications through the exploitation of the new MPC features of PCS7. In order to design a new MPC block, the tool for matrix calculations was created in PCS7 software using SCL. The proposed MPC block program was written in SCL in such a way that the calculation time, unlike the existing MPC block in the PCS7 library, is not very long. Therefore, it can not only be used in very slow process but also for fast processes. At the presented study, a good memory management leaded to impose less load on the CPU. This new block can also control the processes with complex poles as well as processes with real poles to reach the setpoint in less time and fluctuation compared to the existing block in the PCS7 library. The discrepancy of control parameters was also simulated and was discussed.

REFERENCE

[1] G. Liang, W. Li, and Z. Li, "Control of superheated steam temperature in large-capacity generation units based on active disturbance rejection method and distributed control system," *Control Engineering Practice,* vol. 21, pp. 268-285, 2013.

[2] T. Chai, S. J. Qin, and H. Wang, "Optimal operational control for complex industrial processes," *Annual Reviews in Control,* vol. 38, pp. 81-92, 2014.

[3] I. Jovanović and I. Miljanović, "Contemporary advanced control techniques for flotation plants with mechanical flotation cells–A review," *Minerals Engineering,* vol. 70, pp. 228-249, 2015.

[4] X. Luo and Y. Song, "Adaptive Predictive Control: A Data-Driven Closed-Loop Subspace Identification Approach," in *Abstract and Applied Analysis*, 2014.

[5] M. Davanipoor, M. Zekri, and F. Sheikholeslam, "Fuzzy wavelet neural network with an accelerated hybrid learning algorithm," *Fuzzy Systems, IEEE Transactions on,* vol. 20, pp. 463-470, 2012.

[6] S. H. Mousavi, N. Noroozi, A. A. Safavi, and A. Ebadat, "Modeling and control of nonlinear systems using novel fuzzy wavelet networks: The output adaptive control approach," *Communications in Nonlinear Science and Numerical Simulation,* vol. 16, pp. 3798-3814, 2011.

[7] L. Obregón, L. Quiñones, and C. Velázquez, "Model predictive control of a fluidized bed dryer with an inline NIR as moisture sensor," *Control Engineering Practice,* vol. 21, pp. 509-517, 2013.

[8] M. Bauer and I. K. Craig, "Economic assessment of advanced process control–a survey and framework," *Journal of Process Control,* vol. 18, pp. 2-18, 2008.

[9] Y.-G. Xi, D.-W. Li, and S. Lin, "Model Predictive Control — Status and Challenges," *Acta Automatica Sinica,* vol. 39, pp. 222-236, 3// 2013.

[10] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control engineering practice,* vol. 11, pp. 733-764, 2003.

[11] M. L. Darby and M. Nikolaou, "MPC: Current practice and challenges," *Control Engineering Practice,* vol. 20, pp. 328-342, 2012.

[12] *The SIMATIC PCS7 Process Control System*, Siemens, February 2012.

[13] *Structured Control Language (SCL) for S7-300/S7-400 Programming*, Siemens, 2009.

[14] J. M. Maciejowski, *Predictive control: with constraints*: Pearson education, 2002.

[15] E. F. Camacho and C. B. Alba, *Model Predictive Control*, second ed.: Springer-Verlag London, 2007.

[16] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB®*, first ed.: Springer-Verlag London, 2009.

[17] *Process Control System PCS7 Advanced Process Library (V8.1)*, Siemens, July 2014.

[18] F. Borrelli, *Constrained optimal control of linear and hybrid systems* vol. 290: Springer, 2003.