Maintaining Good Conditioning of Model Identification Task in Immune Inspired On-line Optimizer of an Industrial Process

K. Wojdan * K. Swirski † M. Warchol ‡ M. Maciorowski §

Abstract—**Methods which provide good conditioning of model identification task in immune inspired, steady-state controller SILO (Stochastic Immune Layer Optimizer) are presented in this paper. These methods are implemented in a model based optimization algorithm. The first method uses a safe model to assure that gains of the process's model can be estimated. The second method is responsible for elimination of potential linear dependences between columns of observation matrix. Moreover new results from one of SILO implementation in polish power plant are presented. They confirm high efficiency of the presented solution in solving technical problems.**

Keywords: steady-state control, model identification, adaptation, artificial immune system

1 Introduction

Control and optimization of the steady state of industrial processes have been taken in a numerous research works and implementations. Industrial processes which take place in a large scale plants, with high level of complexity, are characterized by large number of control inputs, outputs and disturbances, long time of process response for control change, essentially non-linear characteristics and impossible to omit cross transforms. Classic control systems based on PID (Proportional Integral Derivative) controllers are not able to perform optimal control of such processes.

Methods based on predictive control algorithms with receding horizon [3] are the most common solutions for advanced control of industrial process. In a multilayer structure of control system [3] an advanced control layer computes control vector, which is a set-point vector for a base control layer. An MPC (Model Predictive Control) controller, based on the knowledge stored in a dynamic model, computes control vectors in consecutive moments $x(k)$, $x(k+1)$, ..., $x(k+N_u-1)$. This control trajectory minimizes the difference between estimated process's outputs and demand output values in consecutive moments.

Despite numerous advantages of MPC controllers there are some important disadvantages. Implementation of MPC controllers is expensive. Long lasting and labour-consuming parametric tests have to be done to create a dynamic, mathematical model. Honeywell experts have calculated that the cost of creating a dynamic model vary from USD250 to USD1000 for each, single dependence between one input and one output of the process [2].

Another disadvantage of predictive control methods is an insufficient adaptation to process's characteristics changes. These changes are considered in a long-term horizon, like months and years, resulting from wearing or failure of devices, rebuilding of industrial system, changes of chemical properties of components used in a process or external conditions changes (e.g. seasonality). The easiest way to consider these changes is a manual, periodic update of the model based on most recent identification experiments of the process. However, this is not a satisfying solution. Implementation of an adaptation method in a MPC controller is a more desired approach. This solution is related with some difficulties:

- Estimation of model's parameters when there is an insufficient changeability of noised signals,
- Estimation of model's parameters in closed loop operation,
- High computer resources usage in on-line operation,
- Possibility of linear dependences in an observation matrix, which consists of process measurements.

The mentioned problems cause adaptation methods implemented in MPC controllers to be often insufficient. This motivates us to search for new solutions. In [4, 5] SILO has been presented – a solution for stead-state control and for on-line economic optimization of process operating point. This solution is inspired by operation of

*K. Wojdan is with the Warsaw University of Technology, Institute of Control and Computation Engineering, Poland, k.wojdan@tt.com.pl

†K. Swirski is with the Warsaw University of Technology, Institute of Heat Engineering, Poland, swirski@itc.pw.edu.pl

‡M. Warchol is with the Warsaw University of Technology, Institute of Control and Computation Engineering, Poland, M.Warchol@ia.pw.edu.pl

§M. Maciorowski is with Transition Technology, Poland, m.maciorowski@tt.com.pl

immune system of living creatures. SILO can be used to control processes, characterized by fast but rare disturbance changes, or processes where disturbances change continuously but the rate of changes is essentially slower than the dynamics of the process. In the first case control quality in transition states depends on a base control layer. In industrial plants which fulfill above-mentioned condition, SILO can be a low cost (in an economic meaning) and effective alternative for MPC controllers. It results from:

- There is no need to perform identification experiments of the process manually;
- There is no need to create *a priori* a dynamic, mathematical model of the process. It results in a significant reduction of work time of high qualified engineers;
- Consideration of higher number of process operating points. SILO automatically learns the process and adapts to current operating point. SILO can be forced to higher precision of adaptation by definition of more narrow ranges of process's signals, in which linear approximation is sufficient. In the case of MPC controllers number of such areas (i.e. fuzzied partitions) is limited by amount of labor related with parametric tests;
- Immune inspired efficient adaptation algorithm which is able to acquire knowledge about the process.

2 Immune structure of SILO

The structure of immune optimizer was described in [4, 5]. In this chapter the immune structure of SILO is briefly reminded. Vector x represents control inputs, vector y represents process outputs (y) and vector z represents measured disturbances. An analogy between an immune system and SILO is presented below:

Pathogen – Measured and non-measured disturbances,

B cell – Historical static process responce to a control change,

Antibody - effector part – Optimal control vector change,

Antibody - antigen binding side – Current process state (current values of x, y and z vectors), stored in a B cell.

In the SILO system the B cell represents values of x, y and z vectors before and after a control change. Thus B cell represents static process responce to a control change.

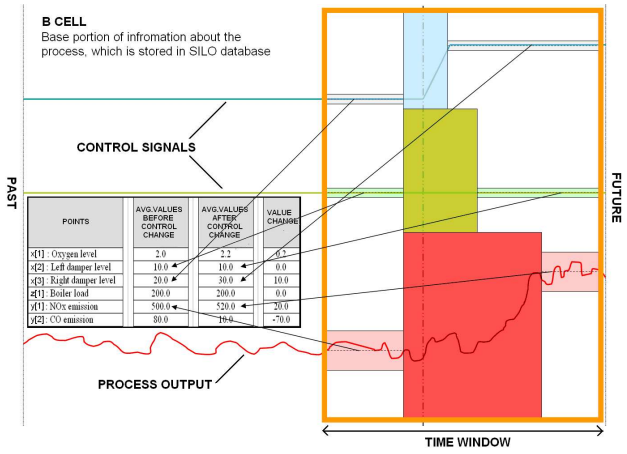

Figure 1: Time window of a B cell.

One should notice that the B cell represents only such process states, in which measured disturbances z are constant. The example of a B cell is presented in Fig. 1. The formal definition of the k-th B cell is presented below:

$$L_k = \left[{}^b\bar{x}^k, {}^p\bar{x}^k, {}^b\bar{y}^k, {}^p\bar{y}^k, \bar{z}^k \right],$$

where:

${}^b\bar{x}^k$ – average values of control signals measured before control change,

${}^p\bar{x}^k$ – average values of control signals measured after control change,

${}^b\bar{y}^k$ – average values of process outputs measured before control change,

${}^p\bar{y}^k$ – average values of process outputs measured after control change,

$\bar{z}^k$ – average values of measured disturbances.

In later considerations the following increases of control points and process outputs will be useful

$$\Delta x^k = {}^p\bar{x}^k - {}^b\bar{x}^k,$$

$$\Delta y^k = {}^p\bar{y}^k - {}^b\bar{y}^k.$$

In a SILO system, an antigen which is located on a pathogen's surface, is represented by current process state vector $A = [x^a, y^a, z^a]$. Antibody (historical process state stored in a B cell) binds the antigen only when a current process state is similar to the process state stored in a B cell that has created the antibody. Affinity between B cell L_k and antigen A is defined in the following way:

$$\mu\left(L_k, A\right) = \left(\prod_{i=1}^{nx} g_i^{x^b} \left({}^b\bar{x}_i^k, x_i^a \right) \right) \times \left(\prod_{i=1}^{nx} g_i^{x^p} \left({}^p\bar{x}_i^k, x_i^a \right) \right) \times$$

$$\times \left(\prod_{i=1}^{ny} g_i^{y^b} \left({}^b\bar{y}_i^k, y_i^a \right) \right) \times \left(\prod_{i=1}^{ny} g_i^{y^p} \left({}^p\bar{y}_i^k, y_i^a \right) \right) \times \times \left(\prod_{i=1}^{nz} g_i^z \left(\bar{z}_i^k, z_i^a \right) \right)$$

where $\forall x_1, x_2 \in \Re \;\; g(x_1, x_2) \in \{0, 1\}$. The antibody binds the antigen only when $\mu(L_k, A) = 1$. Examples of $g(x_1, x_2)$ function are presented below:

$$g(x_1, x_2) = \begin{cases} 0 & \text{if} \quad |x_1 - x_2| > \varepsilon \\ 1 & \text{if} \quad |x_1 - x_2| \leq \varepsilon \end{cases}$$

$$g(x_1, x_2) = \begin{cases} 1 & \text{if} \quad x_1 < 10 \wedge x_2 < 10 \\ 1 & \text{if} \quad x_1 \geq 10 \wedge x_2 \geq 10 \\ 0 & \text{otherwise} \end{cases}$$

$$g(x_1, x_2) = 1$$

3 Optimization algorithm

SILO consists of two main, independent modules – a learning and an optimization module. The operation of both modules is wider discussed in [4, 6]. The learning module performs on-line analysis of current and historical values of elements of x, y and z vectors and searches for *time windows* which fulfill the criteria of being a B cell. When the module finds a time window which meets those requirements, then the recognized B cell is saved in a database, which represents the knowledge about the process. At least one control signal must change, to transform a time window into a B cell. During normal operation of a plant there are frequent changes of control inputs. Thus, the process of creating B cells is continuous. SILO updates its knowledge about the process all the time. It ensures constant adaptation to changeable operation conditions. Control changes can be caused by an operator, a base control layer or by an optimization module of SILO.

The optimization module operates in closed-loop. Based on current process state, B cells from immune memory and quality indicator coefficients, the optimization module computes an optimal increment of the control vector. The quality indicator's value depends on x and y vectors. Equation 1 defines a *control form* of quality indicator (refer to Fig. 2)

$$J = \sum_{k=1}^{nx} \Big[\alpha_k \left(|x_k^a - x_k^s| - \tau_k^{lx} \right)_+ + \\ + \beta_k \left(|x_k^a - x_k^s| - \tau_k^{sx} \right)_+^2 \Big] + \\ + \sum_{k=1}^{ny} \Big[\gamma_k \left(|y_k^a - y_k^s| - \tau_k^{ly} \right)_+ + \\ + \delta_k \left(|y_k^a - y_k^s| - \tau_k^{sy} \right)_+^2 \Big] \tag{1}$$

where:

α_k – linear penalty coefficient for k-th control variable

β_k – square penalty coefficient for k-th control variable

γ_k – linear penalty coefficient for k-th optimized output

δ_k – square penalty coefficient for k-th optimized output

τ_k^{lx} – width of insensitivity zone for linear part of penalty for k-th control variable

τ_k^{sx} – width of insensitivity zone for square part of penalty for k-th control variable

τ_k^{ly} – width of insensitivity zone for linear part of penalty for k-th optimized output

τ_k^{sy} – width of insensitivity zone for square part of penalty for k-th optimized output

$(\cdot)_+$ – "positive part" operator $(x)_+ = \frac{1}{2}(x + |x|)$

x_k^s – demand value for k-th control variable

y_k^s – demand value for k-th optimized output

One can easy transform this equation to an *economic form* of quality indicator

$$J(c, y) = \sum_{k=1}^{nc} p_j^c c_j - \sum_{k=1}^{ny} p_j^y y_j$$

where:

c_j – j-th element of decision variable vector in optimization layer,

p^y – vector of output product prices,

p^c – vector of prices of input streams.

In such case optimization module will perform on-line economic optimization of process operating point [6].

Optimization method is discussed in [5, 6]. The optimization period is defined as a the time range between control vector changes. This time is not shorter than time needed to reach steady-state of process outputs, after control change. In case of combustion process in large

scale power boiler, this time period vary from 5 to 15 minutes.

Optimization algorithm operates in one of three optimization layers. In each layer a different optimization algorithm is used to compute an optimal control vector increment that minimizes quality indicator. An algorithm that lets SILO switch between layers is a key algorithm of this solution (refer to Fig. 3). Activation of particular layer depends on SILO's knowledge and current process's state.

Stochastic optimization layer corresponds with stochastic exploration of the solutions' space. It is being executed when SILO has no knowledge about the process or when the knowledge about the process is insufficient. In this layer SILO changes only one control variable (one element of control vector) in one optimization step. The main goal of this layer is to gather knowledge about the process. Special heuristic used in this layer is responsible for:

- Gathering knowledge about the process, by performing automatic parametric tests in the neighborhood of current process state. Learning module creates new B cells based on this tests;
- Decreasing quality indicator value in long time horizon, assuming that disturbances do not essentially change in a long time horizon. By analogy to immune system it can be said that the goal of this layer is elimination of pathogen in long time horizon, assuming that the system is attacked frequently by the same pathogen;
- Assuring that observation matrices, which are based on B cells from immune memory (X_{L} and X_{G} matrices), are nonsingular and model identification task (3) is good conditioned. This model identification task is performed in model based layers (refer 3.1)

The solution found in this layer is a starting point for optimization in layers that use a process's model. By

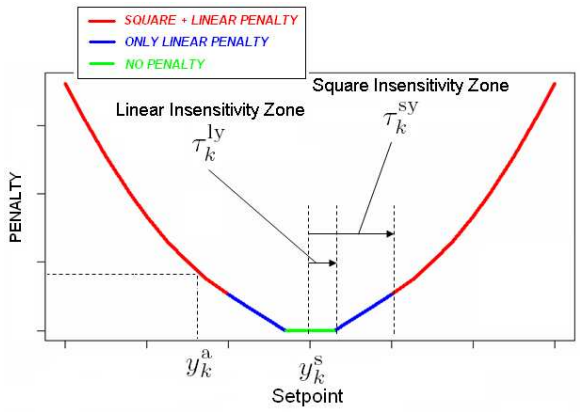

Figure 2: Penalty function.

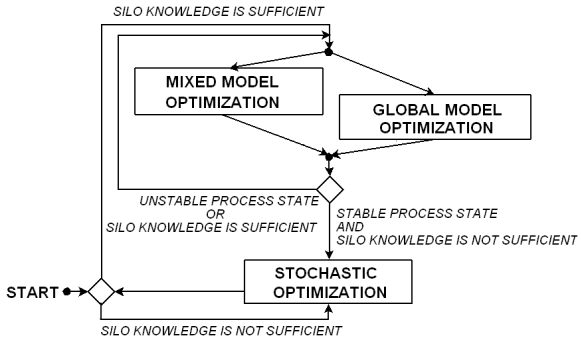

Figure 3: Layers of optimization algorithm.

analogy to an immune system this layer represents the primary response of the immune system [1].

Optimization on the global model layer – this layer uses general knowledge about an optimizing process to compute an optimal increment of control vector. An automatically created steady-state model (2), used in this layer, aggregates knowledge about basic process dependences. This layer is being executed when SILO does not have sufficient knowledge about the process and thus SILO is not able to create a mathematical model that represents static process dependences in the neighborhood of the current process's operating point. The newest portions of knowledge (B cells) from different process's states are being used to create a global model.

Optimization on the mixed model layer – this layer uses information from immune memory to create steady-state model (2) that represents static process dependences in the neighborhood of current process operating point. Model created in this layer is based on knowledge stored in the neighborhood of the current process's operating point so the model is more accurate than global model. This layer corresponds with exploitation of solutions' space. By analogy to immune system this layer represents secondary immune response [1].

Methods which provide good conditioning of model identification task, are implemented in global and mixed model based optimization layers. These methods are responsible for maintenance of a reliable model of the process.

3.1 Model Identification in mixed and global model layers

In optimization on the mixed model layer a mathematical, linear model of the process is constructed. Information stored in g youngest *global* B cells and l youngest *local* B cells are used to construct this model. The set of global B cells represents all B cells stored in immune memory. The set of local B cells represents B cells which fulfill affinity conditions. For those B cells L_k, it holds:

$$\mu\left(L_k, A\right) = 1,$$

where $A = [x^a, y^a, z^a]$ is a current process state.

A linear static model of the process is defined below:

$$\Delta y = \Delta x K, \tag{2}$$

where $\Delta x = [\Delta x_1, \Delta x_2, \ldots, \Delta x_{nx}]$, $\Delta y = [\Delta y_1, \Delta y_2, \ldots, \Delta y_{ny}]$ and K is a gain matrix. Coefficients of a K matrix are estimated using the least square method. Analytical solution (4) of normal equation (3) is presented below:

$$(W + \mu I)\, K = (V + \mu M) \tag{3}$$

$$K = (W + \mu I)^{-1} (V + \mu M) \tag{4}$$

where:

$V = \eta \Delta X_L^T \Delta Y_L + \vartheta \Delta X_G^T \Delta Y_G$,

$W = \eta \Delta X_L^T \Delta X_L + \vartheta \Delta X_G^T \Delta X_G$,

ΔX_L, ΔX_G – observation matrices with increases of control variables. Each of l rows of matrix ΔX_L contains an increase of control variable vector Δx stored in local B cell (from the set of l youngest B cells)

$$\Delta X_{\mathrm{L}} = \begin{bmatrix} \Delta x_{1,1} & \Delta x_{1,2} & \ldots & \Delta x_{1,nx} \\ \Delta x_{2,1} & \Delta x_{2,2} & \ldots & \Delta x_{2,nx} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta x_{l,1} & \Delta x_{l,2} & \ldots & \Delta x_{l,nx} \end{bmatrix}.$$

Analogously, matrix ΔX_G contains increases of control variables from g youngest global B cells,

ΔY_L, ΔY_G – observation matrices with increases of optimized outputs. Each of l rows of matrix ΔY_L contains an increase of output vector Δy stored in local B cell (from the set of l youngest B cells)

$$\Delta Y_{\mathrm{L}} = \begin{bmatrix} \Delta y_{1,1} & \Delta y_{1,2} & \ldots & \Delta y_{1,ny} \\ \Delta y_{2,1} & \Delta y_{2,2} & \ldots & \Delta y_{2,ny} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta y_{l,1} & \Delta y_{l,2} & \ldots & \Delta y_{l,ny} \end{bmatrix}.$$

Analogically, matrix ΔY_G contains increases of optimized outputs from g youngest global B cells,

M – matrix which represents *safe model*,

μ – weight of safe model,

η – weight of local B cells,

ϑ – weight of global B cells.

In a global model optimization layer local B cells are not used. Equations (2), (3), (4) are the same. The only differnce is that $V = \Delta X_G^T \Delta Y_G$ and $W = \Delta X_G^T \Delta X_G$.

One should noticed that matrix $W + \mu I$ is symmetric. Moreover, this matrix is nonsingular. Thus it is more efficient to compute elements of matrix K using specialized numerical methods, such as QR factorization of matrix $W + \mu I$, in comparison with analytical solution (4).

Global B cells are used to improve the quality of mixed model when none of local B cells contain information about an impact of particular element of control vector on process's outputs. In such case, without using global B cells, matrix $\Delta X_L^T \Delta X_L$ would be singular. Global B cells, used even with small weight ϑ, would cause those gains of the model to be similar to real values. Only the gain of control input, for which there was no increment recorded in the set of local B cells, can be inaccurate in the current process operating point. Global B cells can be useful in such process's states, when modification of some control inputs is impossible. For example in case of combustion process in power boiler some coal mills (vector's x elements) are turned off when unit's load (measured disturbance) is low.

Elements μI and μM in equation (3) ensure good conditioning of equation, even if matrix W is singular. In such case, utilization of safe matrix M, generally modifies the smallest eigenvalues of matrix $W + \mu I$. Coefficient μ is small (about 10^{-4}), thus in case of well conditioned equation (3) solution is not essentially modified. Elements of matrix M represent estimated process gains. This estimation is based on *a priori* knowledge about the process. Elements of matrix M can be changed manually. If there is no *a priori* knowledge about the process, elements of matrix M can be set to zero.

The quality indicator (1) is minimized based on the linear model (2) with respect to Δx subject to constraints

$$z_{low} \le \Delta x \le z_{hi},\ u_{low} \le x^a + \Delta x \le u_{hi}.$$

The minimization problem can be formulated as an LQ problem after introducing additional variables

$$\min_{\Delta x, x^{dlp}, x^{dln}, x^{ds}, y^{dlp}, y^{dln}, y^{ds}} \left\{ \sum_{k=1}^{nx} \left[\alpha_k \left(x_k^{dlp} + x_k^{dln} \right) + \right.\right.$$

$$+\beta_k \left(x_k^a + \Delta x_k - x_k^s - x_k^{ds} \right)^2 \Big] + \sum_{k=1}^{ny} \Big[\gamma_k \left(y_k^{dlp} + y_k^{dln} \right) +$$

$$\left. \left. +\delta_k \left(y_k^a + \Delta x K_k - y_k^s - y_k^{ds} \right)^2 \right] \right\}$$

with constrains

$$x_k^{dlp} \ge x_k^a + \Delta x_k - x_k^s - \tau_k^{lx}, x_k^{dlp} \ge 0,$$

$$x_k^{dln} \ge x_k^s - x_k^a + \Delta x_k - \tau_k^{lx}, x_k^{dln} \ge 0,$$

$$-\tau_k^{sx} \le x_k^{ds} \le \tau_k^{sx}, -\tau_k^{sy} \le y_k^{ds} \le \tau_k^{sy},$$

$$y_k^{dlp} \ge y_k^a + \Delta x K_k - y_k^s - \tau_k^{ly}, y_k^{dlp} \ge 0,$$

$$y_k^{dln} \geq y_k^s - y_k^a - \Delta x K_k - \tau_k^{ly}, y_k^{dln} \geq 0,$$
$$z_{low} \leq \Delta x \leq z_{hi}, u_{low} \leq x^a + \Delta x \leq u_{hi}$$

where:

x_k^{dlp}, x_k^{dln} – additional variables, representing distance from $x_k^a + \Delta x_k$ to the neighborhood of x_k^s with radius τ_k^{lx},

y_k^{dlp}, y_k^{dln} – additional variables, representing distance from $y_k^a + \Delta x K_k$ to the neighborhood of y_k^s with radius τ_k^{ly},

x_k^{ds} – additional variable, representing current part of insensitivity zone around x_k^s used in the square part of performance index,

y_k^{ds} – additional variable, representing current part of insensitivity zone around y_k^s used in the square part of performance index.

3.2 Blocking Algorithm

Utilization of a *safe model* in a normal equation (3) assures that this equation is always well conditioned. However if the matrix W is singular, then a K matrix may not represents real process gains in the neighborhood of current operating point. The *blocking algorithm* tries to assure that matrix W is non-singular and thus the equation (3) is well conditioned.

Before each optimization in the mixed or global model layer, equation

$$WK = V \tag{5}$$

is checked for its conditioning. This verification is done based on proportion between the largest and the smallest singular values. Conditioning of a model identification task can be expressed by coefficient k

$$k = \left\|W^{-1}\right\|_2 \|W\|_2 = \frac{\sigma_{max}}{\sigma_{min}}$$

where σ_{max} is is the largest singular value and σ_{min} is the smallest singular value. One should noticed that matrix W is symmetric and positive semidefinite. Thus a relation between singular values and eigenvalues λ is defined in the following way:

$$\sigma_i = \sqrt{\lambda_i},\ i = 1, 2, ..., nx$$

If the coefficient k is equal 1 then the equation (5) is good conditioned. If the coefficient k is infinite then equation (5) is bad conditioned. One of optimization module's parameters is a boundary value for the k coefficient. Above this value, a special variable blocking algorithm is activated. If k coefficient value is above defined limit, before each optimization some of control variables are blocked

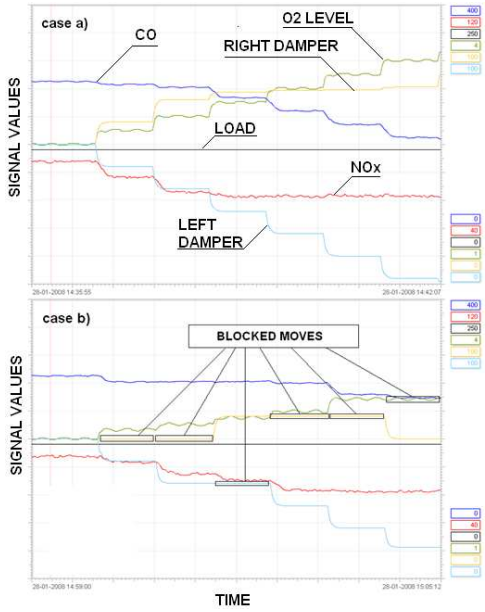

Figure 4: Example of control trajectory when: a) blocking algorithm is disabled, b) blocking algorithm is activated in each optimization step.

with a certain distribution of probability. Thus a dimension of optimization task is reduced. It causes a mutation in new B cells. The main goal of this algorithm is to eliminate potential linear dependences between columns of observation matrices (ΔX_L and ΔX_G). Elimination of these dependences causes that matrix W is not singular. Utilization of pseudo inverses instead of a blocking algorithm is not sufficient, because some of identified coefficients of a gain matrix K in equation (2) may not be correct.

Potential linear dependences can occur, if i-th and j-th element of a control vector is changing in the following way in each optimization step (rule R^1 or rule R^2):

$$R^1 = \begin{cases} x_i^{new} = x_i^{old} + \Delta^{max} x_i; \\ x_j^{new} = x_j^{old} - \Delta^{max} x_j. \end{cases}$$

$$R^2 = \begin{cases} x_i^{new} = x_i^{old} - \Delta^{max} x_i; \\ x_j^{new} = x_j^{old} + \Delta^{max} x_j. \end{cases}$$

where $\Delta^{max} x_k$ is a maximal, absolute increment of k-th element of a vector x in one optimization step. In such case in ΔX_L and ΔX_G matrices there can be some linear dependences between i-th column, representing i-th element of a control vector, and j-th column, representing j-th element of a control vector. Mentioned situation is often observed in real implementations of advanced control solutions, when defined constrains for increment of i-th and j-th control vector's element are too narrow. Presented method causes that possible linear dependences are eliminated, thus accuracy of a model gains estimation is higher.

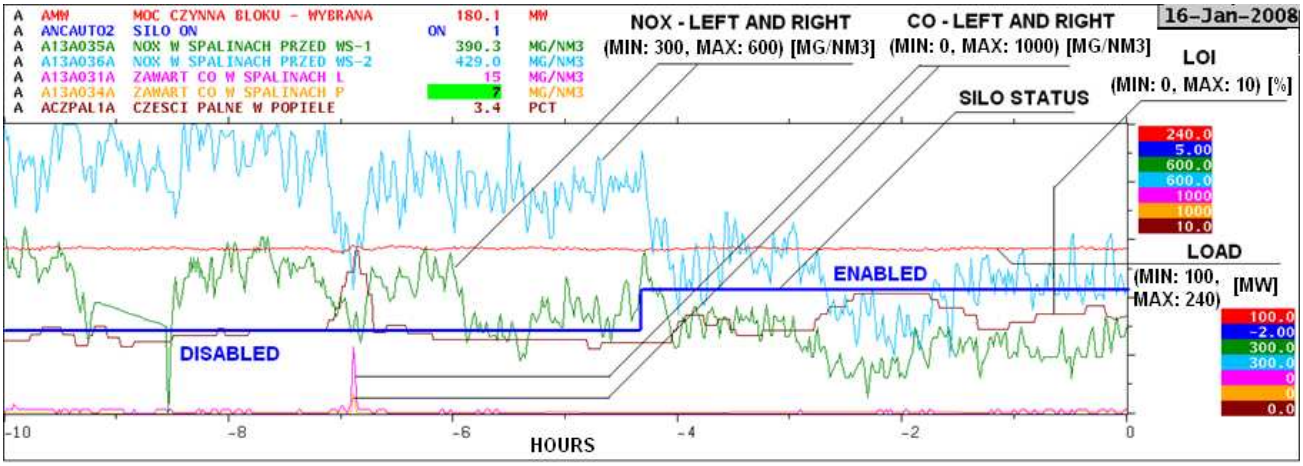

Figure 5: SILO influence on NO_x, CO and LOI signals.

Figure 4a presents a simulation in which control variables representing an oxygen level (O2 LEVEL) and a position of left damper (DAMPER LEFT) are correlated. It can lead to a situation when the matrix W is singular. Figure 4b presents, that a blocking algorithm is able to eliminate a linear dependence between mentioned signals. However, it results in worst quality of the solution. It is shown that one of minimized outputs (CO emission) has reached higher level in comparison to a situation when a blocking algorithm was turned off. Temporary acceptance of suboptimal solution is needed, to provide a good conditioning of equation (3).

In addition to the presented methods, a heuristic applied in a stochastic optimization layer [4], eliminates potential linear dependences between columns of matrix ΔX_L as well as ΔX_G. In this heuristic only one randomly chosen control variable is changing at a time.

4 Results

SILO has been implemented in six units in U.S. power plants, four units in Polish power plants and one unit in Taiwan power plant. Results of a SILO implementation in one of units in Polish power plants is presented in this chapter. The primary SILO goal was to:

- Keep NO_x (nitrogen oxides) emission (one hour average) below 500 mg/Nm^3;
- Keep CO (carbon monoxide) emission (5 minute average) below 250 mg/Nm^3.

The secondary SILO goal was to:

- Keep LOI (loss of ignition) below 5 %;
- Keep SH (super heat) temperature on 540 °C;
- Keep flue gas temperature below 140 °C (flue gas desulphurization process requirement).

SILO optimized nine output signals: CO emission (left and right side), NO_x emission (left and right side), estimated SH temperature (left and right side), flue gas temperature (left and right side) and LOI signal. Six disturbance signals were chosen from the set of all process's signals: unit load, coal calorific value and status of each of four coal mills. Control vector consisted of eleven signals: O2 level, eight secondary air dampers and two OFA dampers.

Table 1: Results of SILO operation

	SILO OFF	SILO ON
Analyzed hours	822	163
Load range [MW]	106.0 – 195.3	107.8 – 203.0
NO_x exceeding [%]	10.14	0.0
CO exceeding [%]	1.15	0.19
Avg. SH [°C]	532.24	536.58
LOI exceeding [%]	59.0	24.14
Avg. Flue Gas [°C]	117.76	120.54

Figure 5 presents a situation, when SILO is disabled and compares it with the situation when SILO is enabled. SILO essentially reduces NO_x emission and maintains CO emission and LOI below defined limit. Table 1 presents one month summary of SILO operation results. One can see that NO_x emission limit (500 mg/Nm^3) has not been exceeded even once, when SILO was enabled. When SILO was turned off, 10.41 % of analyzed one hour averages was above this limit. By 99.81 % of SILO operation time, CO emission has been kept below 250 mg/Nm^3. When SILO was disabled CO emission limit has been kept below 250 mg/Nm^3 by 98.85 % of time. SH temperature has been increased by 4.4 °C, so the efficiency of the process was higher, when SILO was enabled. LOI was reduced to 3.92 % (from 5.16 % level). SILO has

kept LOI below 5 % by 75.9 % of its operation time. In comparison, when SILO was disabled, LOI was below the limit only by 41 % of time. When SILO was enabled, Flue Gas Temperature was kept below 140 °C all the time, so flue gas desulphurization process has not been disturbed.

5 Summary

This paper presents some drawbacks connected with MPC controllers. MPC approach was compared with SILO. Furthermore two new methods, which provide good conditioning of model identification task, were presented. These methods cause that a steady-state process's model created by SILO represents a real derivative of a static process characteristic. Moreover results from one of SILO implementations in Polish power plant were presented. They confirm high efficiency of presented solution in solving real technical problems.

References

[1] L. N. De Castro and F.J. Von Zuben. Artificial immune systems: Part i – basic theory and applications. Technical Report RT DCA 01/99, Department of Computer Engineering and Industrial Automation, School of Electrical and Computer Engineering, State University of Campinas, Campinas, SP, Brazil, December 1999.

[2] L. Desborough and R. Miller. Increasing customer value of industrial control performance monitoring - honeywell's experience. In *Proc. of 2004 American Control Conference*, Boston, USA, June 2004.

[3] Piotr Tatjewski. *Advanced Control of Industrial Processes : Structures and Aglorithms*. Springer Verlag, London, 2007.

[4] K. Wojdan and K. Swirski. Immune inspired optimizer of combustion process in power boiler. In *Proc. of the 20th International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems*, Kyoto, Japan, June 2007.

[5] K. Wojdan, K. Swirski, M. Warchol, and T. Chomiak. New improvements of immune inspired optimizer SILO. In *Proc. of the 19th IEEE International Conference on Tools with Artificial Intelligence*, Patras, Greece, October 2007.

[6] K. Wojdan, K. Swirski, M. Warchol, G. Jarmoszewicz, and T. Chomiak. Bio-inspired process control. In *Bio-inspired computing and communication networks*, chapter 9. Taylor and Francis Group, 2009. approved for printing.