A Practical Approach To Combustion Process Optimization Using An Improved Immune Optimizer

K. Wojdan, K. Swirski, M. Warchol, J. Milewski, and A. Miller

Abstract—An improved version of the immune inspired optimizer SILO is presented in this paper. The new model identification method allows for utilization of model gains constraints. Moreover the operation of a new *Transition State* algorithm is analyzed based on a real-life example. The improved version of SILO was implemented in a real power boiler. Results from a real combustion process optimization are presented in this paper.

Keywords—About four key words or phrases in alphabetical order, separated by commas

I. Introduction

THE optimization of power boilers is an important topic in research and in the implementation projects in the power industry. The combustion process in a power boiler is a complex process, with a large number of control variables, disturbances and outputs. This is a dynamic non-linear process characterized by long response time caused by process inertia and transport delay. It is hard to control such a process using only standard SISO (Single Input Single Output) control algorithms. In this article we present an improved version of the immune inspired optimizer SILO. The task of SILO is to perform an on-line optimization of the current process operating point. The optimizer is implemented above the base control layer in a layered control structure (refer fig. 1). The SILO system calculates setpoints or setpoints corrections for controllers that operate in the base control layer. Control systems of power units in power plants are based on PI (Proportional-Integral) controllers. These controllers control sub-processes (e.g.: oxygen level in exhaust gases or windbox to furnace differential pressure in case of a power unit control system) that have an influence on a main optimized MIMO (Multi Input Multi Output) process (e.g. combustion process in case of a power unit control system).

The SILO system optimizes a process steady state. The output decision vector m^d is updated every optimization cycle period T_{opt}. This time period is not shorter than time needed to reach a new process steady state after a control vector change. Changes to the m^d vector (setpoints for base controllers) are limited due to the stable and safe operation of the base control system. In the case of detecting a significant process operating point transition (e.g.: essential modification of an industrial plant load), the SILO system activates a transition state mechanism. In such a case SILO tries to move the decision vector m^d to the neighborhood of an optimal solution related to the new process operating point. This m^d vector transition is safe to the plant and fast in comparison with standard, steady state optimization.

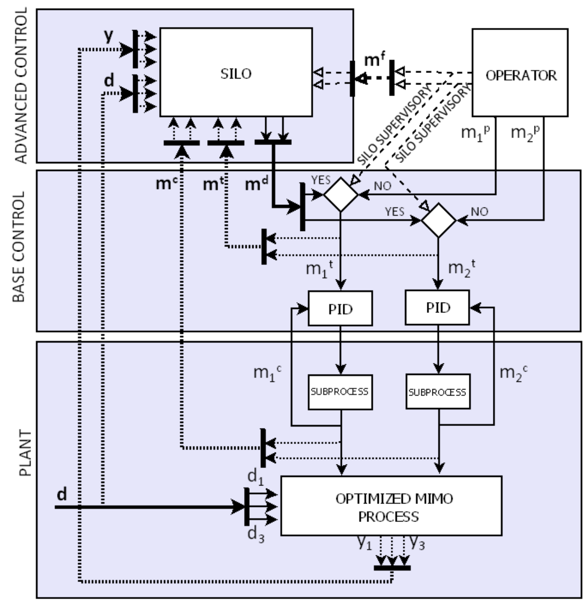

Fig. 1. SILO system in a layered control structure. Description: y - main process outputs vector, d - main process disturbances vector, m^d - decision vector (setpoints for low level controllers), m^t - traced setpoints for low level controllers, m^c - vector of measured sub-process outputs (inputs to the main optimized process), m^f - control variables availability vector, m^p - operator setpoints vector.

In each optimization cycle the SILO system calculates the decision vector increment Δm^d based on:

- measured output vector y,
- measured disturbances vector d that have an impact on the optimized process outputs,
- information about availability of control loops and devices (m^f vector) that can be influenced by SILO.

The optimizer calculates an output vector increment Δm^d and adds this increment to the vector m^t that represents traced decision variables (traced setpoints for low level controllers). The calculated sum of Δm^d and m^t is saved in the output m^d vector. Utilization of the m^t vector is caused by application of rate and range constraints for setpoints in the base control

K. Wojdan, K. Swirski, J. Milewski, A. Miller, Warsaw University of Technology, Institute of Heat Engineering, Poland.

M. Warchol, Warsaw University of Technology, Institute of Control and Computational Engineering, Poland.

layer. If the k-th low-level control loop is excluded from SILO supervisory ($m_k^f = 0$) then SILO passes the m_k^t value directly to the m_k^d output element. Thus inactive outputs trace operator settings. It allows bumbles switching between SILO and process operator.

TABLE I
ANALOGY BETWEEN IMMUNE SYSTEM AND SILO.

Immune System	SILO
Pathogen	Measured and non-measured disturbances
B cell	A data unit that represents a static process response for a control change in a particular process operating point
Antibody: antigen binding side	Current process operating point
Antibody: effector part	Optimal control vector change Δm^d
T$_h$ cell	Algorithm which is responsible for selection of proper group of B cells during model creation process
Primary immune response	*Mixed Model Optimization* algorithm
Secondary immune response	*Quasi-Random Extremum Control* layer of the optimization algorithm

The SILO operation and optimizer structure are inspired by immune system – biological structures and processes within an organism [4]. To stay in compliance with previous papers the immune analogy is provided by table I in order to provide information concerning biological inspiration of our system.

II. Gathering knowledge about a plant

The goals of the SILO Knowledge Gathering module are listed below:

- Identification of the static relations between optimized process inputs m^c (measured input to a main, optimized process) and outputs y at different process operating points;
- Saving and updating of long term averages of the control vector m^c at different process operating points (e.g. different power unit loads and coal mill configurations in the case of a power boiler optimization). This information will be used to handle a significant process point transition in a safe and effective way.

Identification of the static relations between process inputs and outputs is based on an on-line analysis of a time window. This time window consists of the current and historical values of the y, d and m^c vectors. The Knowledge Gathering module only analyzes time windows that include an essential change of at least one element of the vector m^c and in which the process disturbances d are constant. Static process reaction for a control change is automatically identified. The computed increment Δy, Δm^c and information about the current process operating point are saved in memory in the form of a B cell (refer fig. 2). Each B cell has a timestamp. Immune memory in a real SILO implementation for combustion process optimization in a power plant consists of tens of thousands of B cells. The information stored in B cells is utilized in

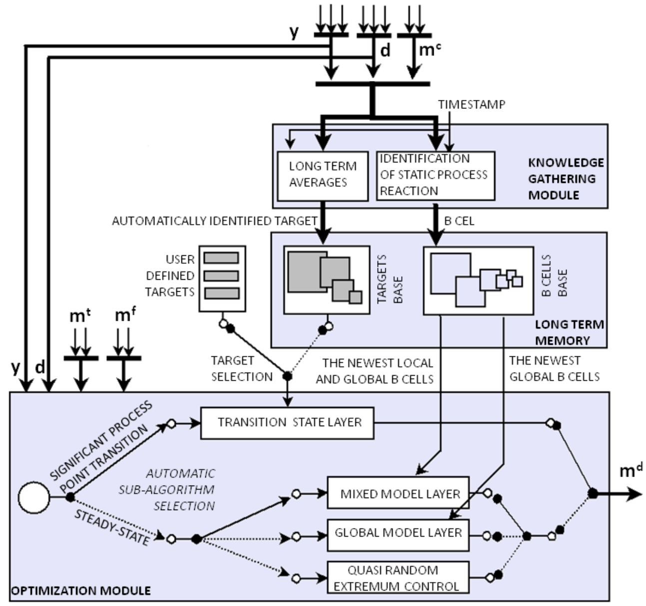

Fig. 2. Optimization and Knowledge Gathering module.

the model creation process that is automatically performed in each optimization cycle. More information about SILO's B cell structure can be found in [1].

The second goal of the Knowledge Gathering module is the saving and updating of long term averages of the m^c, d and y vectors at different process operating points. These long term averages are transformed into AIT (Automatically Identified Targets) objects that are used in the *Transition State* layer (a sub-algorithm of the Optimization module). Each AIT has a timestamp. When a process transition state is detected the system searches for the most recent AIT that fits the current or estimated process operating point. The AIT creation process is described in [3]

III. Quasi Random Extremum Control

In the initial phase of SILO operation the size of the immune memory is relatively small. In analogy to the immune system one can say that the body is often attacked by new, unknown pathogens. Early on, SILO does not have sufficient knowledge to create a mathematical model of the process and solve the optimization task based on this model. A special heuristic that is applied in the *Quasi Random Extremum Control* layer covers the following goals:

- Gathering knowledge about the process. This is done by modifying the m^d vector in such way that each modification can be treated as a standard identification experiment. New B cells are created based on these identification experiments;
- Decreasing the value of an optimized quality indicator at a long time horizon with the assumption that process disturbances are constant at a long time horizon. In analogy to the immune system one can say that a goal of the *Quasi Random Extremum Control* layer is the elimination of the pathogen at the long time horizon with the assumption that the body is attacked by one sort of

pathogen at the long time horizon (a primary immune response);
- maintaining the good conditioning of the model identification task [2].

A special heuristic applied in the *Quasi Random Extremum Control* layer changes only one element of m^d vector in each optimization cycle. In reaction to this change the process outputs y reach a new steady state. The Knowledge Gathering module automatically identifies such a static process reaction and creates B a cell. In a new optimization cycle a different element of m^d is modified and a new B cell is created. After a defined number of cycles the best m^d vector value is restored and applied. This value of the m^d vector is related to the lowest registered value of an optimized quality indicator.

The *Quasi Random Extremum Control* layer is executed if:

- There are not enough B cells in the immune memory to create a mathematical model of the process;
- The knowledge stored in the immune memory is not sufficient to improve the value of the quality indicator. The model is not accurate enough. The applied increment of the m^d vector calculated in the *Mixed Model Optimization* or the *Global Model Optimization* layer (refer fig. 2) is not able to decrease the value of the quality indicator.

IV. Steady state optimization

The SILO system performs an on-line, model based, steady state process optimization during most of its operation time. The optimized quality indicator is the sum of penalties related to the process outputs and selected elements of the m^c vector. The SILO system penalizes a difference between a demand value of a process output $\check{y}_k$ (optionally for a selected element of the $\check{m}^c$ vector) and the measured or estimated value for output $\tilde{y}_k$ (optionally a selected element of the $\tilde{m}^c$ vector). Each single penalty is the sum of a linear and square term, and each term takes insensitivity zones into account. No penalty is applied when an analyzed signal is within an insensitivity zone.

$$J = \sum_{k=1}^{n_m} \left[\alpha_k \left(|\check{m}_k^c - \tilde{m}_k^c| - \tau_k^{lm} \right)_+ + \right. \\ \left. + \beta_k \left(\left(|\check{m}_k^c - \tilde{m}_k^c| - \tau_k^{sm} \right)_+ \right)^2 \right] + \\ + \sum_{k=1}^{n_y} \left[\gamma_k \left(|\check{y}_k - \tilde{m}_k| - \tau_k^{ly} \right)_+ \right. \\ \left. + \delta_k \left(\left(|\check{y}_k - \tilde{y}_k| - \tau_k^{sy} \right)_+ \right)^2 \right] \tag{1}$$

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^{lm} – width of insensitivity zone for linear part of penalty for k-th control variable,
τ_k^{sm} – 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 output,
τ_k^{sy} – width of insensitivity zone for square part of penalty for k-th output,
$(\cdot)_+$ – "positive part" operator $(x)_+ = \frac{1}{2}(x + |x|)$,
$\tilde{m}_k^c$ – current process value for k-th control variable,
$\tilde{y}_k$ – current process value for k-th optimized output,
$\check{m}_k^c$ – demand value for k-th control variable,
$\check{y}_k$ – demand value for k-th optimized output.

Steady state, model based optimization is performed in the *Mixed Model Optimization* or the *Global Model Optimization* layer (refer fig. 2). In both layers a model is formulated in the following way

$$\Delta y = \Delta m^d K \tag{2}$$

In case of mixed model based optimization, elements of the matrix K are estimated based on information stored in the local observation matrices ΔM_L and ΔY_L as well as the global observation matrices ΔM_G and ΔY_G, where

$$\Delta M_L = \begin{bmatrix} \Delta m_{1,1}^c & \Delta m_{1,2}^c & \dots & \Delta m_{1,nm}^c \\ \Delta m_{2,1}^c & \Delta m_{2,2}^c & \dots & \Delta m_{2,nm}^c \\ \vdots & \vdots & \ddots & \vdots \\ \Delta m_{l,1}^c & \Delta m_{l,2}^c & \dots & \Delta m_{l,nm}^c \end{bmatrix},$$

$$\Delta Y_L = \begin{bmatrix} \Delta y_{1,1} & \Delta y_{1,2} & \dots & \Delta y_{1,ny} \\ \Delta y_{2,1} & \Delta y_{2,2} & \dots & \Delta y_{2,ny} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta y_{l,1} & \Delta y_{l,2} & \dots & \Delta y_{l,ny} \end{bmatrix}.$$

Each of the l rows of the ΔM_L matrix consists of increments Δm_i^c of elements of the m^c vector. These increments are stored in a local B cell. This local B cell belongs to the set of l youngest local B cells. The local B cell is a selected B cell that is related to the current process operating point. Such B cell was created when a historical process operating point (e.g. unit load in case of a combustion process optimization in a power boiler) was similar to the current process operating point. By analogy matrix ΔM_G consists of m^c vector increments that are stored in the set of g youngest global B cells. Global B cell selection is based only on a time criterion. Each of the l rows of the ΔY_L matrix consist of y vector elements increments Δy_i that are stored in a local B cell. By analogy matrix ΔY_G consists of y vector increments that are stored in the set of g youngest global B cells.

In the case of a MISO (Multi Input Single Output) model, a weighed sum of squared residuals can be used to estimate a vector k value that is related to a selected column of the matrix K. This sum is expressed in the following way

$$S(k) = \eta e_L^T e_L + \vartheta e_G^T e_G = \\ = \eta \left(\Delta y_L - \Delta M_L k \right)^T \left(\Delta y_L - \Delta M_L k \right) + \vartheta \\ \left(\Delta y_G - \Delta M_G k \right)^T \left(\Delta y_G - \Delta M_G k \right) = \\ = \eta \left(\Delta y_L^T \Delta y_L - 2k^T \Delta M_L^T \Delta y_L + k^T \Delta M_L^T \Delta M_L k \right) + \\ + \vartheta \left(\Delta y_G^T \Delta y_G - 2k^T \Delta M_G^T \Delta y_G + k^T \Delta M_G^T \Delta M_G k \right)$$

The relation between an influence of knowledge from the local and global B cells on the elements of the final gain matrix can be expressed by a relation between values of the η and ϑ weights.

In the new version of SILO an additional optimization task is solved to estimate elements of the gain matrix

$$\min_{k}\{k^T\left(\eta\Delta M_L^T\Delta M_L+\vartheta\Delta M_G^T\Delta M_G\right)k$$
$$-2k^T\left(\eta\Delta M_L^T\Delta y_L+\vartheta\Delta M_G^T\Delta y_G\right)\}$$

with constraints

$$k^l\le k\le k^u$$

The additional optimization task allows for the utilization of constraints related with automatically identified model gains. In most SILO implementations these gains are unbounded. In such case a Least Square Method can be used to estimate elements of the matrix K. Thanks to the additional optimization task the system can use some expert knowledge about the range of gain values for selected dependences between process inputs and outputs. One of the advantages of the new identification method is that the SILO system still has information about the matrix W

$$W=\eta\Delta M_L^T\Delta M_L+\vartheta\Delta M_G^T\Delta M_G$$

This information can be used to compute a conditioning level of the model identification task based on a spectral norm

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

If the conditioning of the model identification task is bad, the system executes actions that can improve the situation. These actions were presented in [3].

An optimal increment Δm^d of the m^d vector is computed based on the identified model. This increment minimizes the value of a quality indicator. It also fulfills constraints for a maximal absolute increment of the m^d vector in one optimization cycle. The following optimization task is solved in each optimization cycle

$$\min_{\Delta m^d,m^{dlp},m^{dln},m^{ds},y^{dlp},y^{dln},y^{ds}}\left\{\sum_{k=1}^{n_m}\left[\alpha_k\left(m_k^{dlp}+m_k^{dln}\right)+\right.\right.$$
$$\left.+\beta_k\left(m_k^t+\Delta m_k^d-\check{m}_k^c-m_k^{ds}\right)^2\right]+\sum_{k=1}^{n_y}\left[\gamma_k\left(y_k^{dlp}+y_k^{dln}\right)+\right.$$
$$\left.\left.+\delta_k\left(\tilde{y}_k+\Delta m^dK_k-\check{y}_k-y_k^{ds}\right)^2\right]\right\}$$

with constraints

$$m_k^{dlp}\ge m_k^t+\Delta m_k^d-\check{m}_k^c-\tau_k^{lx},m_k^{dlp}\ge 0,$$
$$m_k^{dln}\ge\check{m}_k^c-x_k^t+\Delta m_k^d-\tau_k^{lx},m_k^{dln}\ge 0,$$
$$-\tau_k^{sx}\le m_k^{ds}\le\tau_k^{sx},-\tau_k^{sy}\le y_k^{ds}\le\tau_k^{sy},$$
$$y_k^{dlp}\ge\tilde{y}_k+\Delta m^dK_k-\check{y}_k-\tau_k^{ly},y_k^{dlp}\ge 0,$$
$$y_k^{dln}\ge\check{y}_k-\tilde{y}_k-\Delta m^dK_k-\tau_k^{ly},y_k^{dln}\ge 0,$$
$$z_{low}\le\Delta m^d\le z_{hi},u_{low}\le m^t+\Delta m^d\le u_{hi}$$

where:

- $\check{m}_k^c$ – demand value for k-th control variable,
- $\check{y}_k$ – demand value for k-th optimized output,
- $\tilde{y}_k$ – current process value of k-th optimized output,
- m_k^{dlp}, m_k^{dln} – additional variables representing the distance from $m_k^t+\Delta m_k^d$ to the neighborhood of $\check{m}_k^c$ with radius τ_k^{lx},
- y_k^{dlp}, y_k^{dln} – additional variables representing the distance from $\tilde{y}_k+\Delta m^dK_k$ to the neighborhood of $\check{y}_k$ with radius τ_k^{ly},
- m_k^{ds} – additional variable representing the current part the of insensitivity zone around x_k^t used in the square part of performance index,
- y_k^{ds} – additional variable representing the current part of the insensitivity zone around $\check{y}_k$ used in the square part of performance index.

The computed increment Δm^d is added to the current value of the m^t vector. This sum is saved as the optimizer output m^d

$$m^d=m^t+\Delta m^d$$

The increment of inactive elements of the m^d vector (defined by the m^f vector – refer fig. 1) is set to zero.

The *Mixed Model Optimization* layer is activated when SILO has sufficient knowledge about static process dependencies in the close neighborhood of the current process operating point. If there are not enough local B cells in memory, then only global B cells will be used to create a global model. However if there are not enough global B cells in memory (initial phase of SILO operation) or further improvement of a quality indicator value is not possible based on the model, then SILO switches to the *Quasi Random Extremum Control* layer.

By analogy to the immune system the operation of the *Mixed Model Optimization* layer can be compared to a secondary immune response. The SILO system uses the knowledge stored in the B cells to provide for the fast and effective elimination of pathogens (process disturbances compensation).

V. Transition of process operating point

The newest version of SILO has a new algorithm that is able to handle a significant process transition in an effective way. This algorithm is implemented in the *Transition State* layer in the Optimization module of SILO (refer fig. 2). In the case of a combustion process optimization this new mechanism allows for optimization of relatively small power units characterized by frequent transitions of unit load.

When an essential disturbance change is detected the operation of the *Quasi Random Extremum Control* layer or one of two model based layers are suspended. At the same time the *Transition State* layer is activated (refer fig. 2). The memory is searched for AIT (Automatically Identified Targets) and for UDT (User Defined Targets). AIT and UDT objects store information about a process operating point pattern and a related target value of the m^d vector. In each execution cycle of the *Transition State* layer (e.g. every 30 seconds in the case of a combustion process optimization) a special algorithm searches for AIT and UDT characterized by patterns

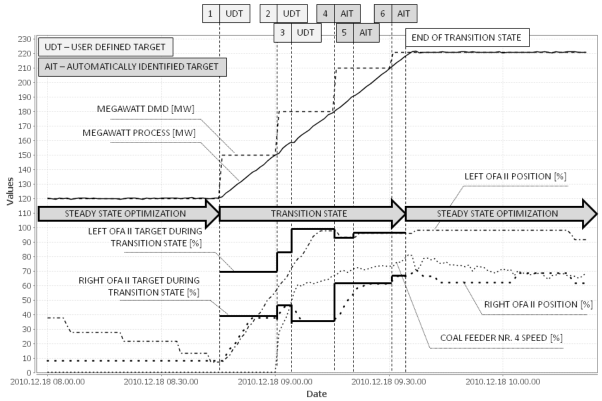

Fig. 3. Operation of the *Transition State* layer.

that fit the current process operating point (e.g. similar unit load and configuration of active pulverizers in the case of a combustion process optimization). In the case that there are AIT and UDT that meet the pattern requirements the system skips a set of UDT objects and selects an AIT object with the youngest timestamp. If there are no AIT objects and only UDT objects fit the current process operating point then the system selects a UDT object with the highest priority. Thanks to different priority levels an expert can define more general and more specific UDT objects depending on their knowledge. The algorithm was described in [3].

Real time plots of disturbances (d vector) and control signals (m^c vector) are presented in figure 3. These trends were recorded on a real 220 MW power unit. The middle section of the plot presents a significant process operating point transition. Unit load increases from the minimal acceptable level of 120 MW to the maximal acceptable level of 220 MW. This transition is done with the highest possible unit load rate. In fig. 3 only two out of seven total elements of the vector m^c are presented – positions of the left and right OFA (Over Fire Air) damper in the second section of OFA dampers. One should note that SILO also modified elements of the m^c vector other than those presented. Disturbances are represented by three signals: demand megawatt production (MEGAWATT DMD in fig. 3), current megawatt production (MEGAWATT PROCESS in fig. 3) and the speed of the coal feeder related with pulverizer number 4. This pulverizer was activated during the presented unit load transition.

The first time period presented in fig. 3 is related with a steady state optimization. SILO updates the value of the m^d vector every 8 to 15 minutes (please refer to the left OFA II position trend). Based on a mathematical model of the process the optimizer tries to maintain process outputs at desired levels. Load transition process starts at 8:45. Modification of the demand megawatt production is automatically identified (based on comparison with current megawatt production). The *Transition State* layer algorithm finds and applies a UDT object related with a new demand megawatt level and current active pulverizers configuration. SILO moves values of the m^d vector elements to the target values stored in the UDT. The speed of the m^d vector transition is limited by the rate limits. One can see that the left OFA II damper was not able to reach the demand target value due to a rate constraint. After the next demand megawatt level change a new UDT vector is applied. When the speed of coal feeder number 4 was essentially increased, a third UDT was applied. Further modifications of the demand megawatt level and coal feeder speed caused application of AIT objects from the optimizer memory (refer fig. 3). The presented power unit often operates in the 190-220 MW load range. The Knowledge Gathering module had enough time to create and save AIT objects related to this load range. One should notice that SILO applies the youngest AIT. Thanks to these feature the system is able to adapt to variable process characteristics (e.g. changes related with seasonality or devices wear).

After a unit load transition is over, the regular steady state optimization is activated. One can see that the solution is only slightly modified. There is no need for an essential and slow (due to rate constrains) m^d vector modification. Thus the *Transition State* layer was able to move the control vector value to a point that lies in a close neighborhood of an optimal solution related with the new process operating point.

VI. Real combustion process optimization

Combustion process optimization is a task of all SILO real life applications. So far SILO has been implemented in 10 power units in the USA, 7 power units in Europe and 5 power units in Asia. The smallest unit is 135 MW and the largest is 890 MW. The newest implementation of the improved version of SILO in 220 MW power unit in Poland is presented in this chapter.

SILO goals are as follow:

- exhaust gases temperature has to be lower than 140 C degrees,
- CO emission has to be lower than 250 mg/Nm^3,
- NO_x emission has to be lower than 300 mg/Nm^3,
- superheat and reheat steam temperature should be maintained at the 535 C degree level,
- LOI (Loss Of Ignition) indicator should be less than 5 %.

The output decision vector m^d consists of seven elements: demand oxygen level in exhausted gases and six OFA (Over Fire Air) dampers. The disturbance vector d consists of six elements: current unit load, demand unit load, and the current speed of each coal feeder related with one of four pulverizes.

TABLE II
Exhaust gases temperature [degrees C].

Load range	SILO OFF	SILO ON
Average exhaust gases temperature [degrees C]		
low load 120-125 MW	120.64	113.32
load transition range 125-215 MW	132.11	124.68
high load 215-225 MW	137.17	132.55
Relative time of exceeding the 140 C degrees level [%]		
low load 120-125 MW	1.25	0.00
load transition range 125-215 MW	29.87	0.00
high load 215-225 MW	42.49	2.36

TABLE III
CO EMISSION [MG/NM^3].

Load range	SILO OFF	SILO ON
Average CO emission [mg/Nm^3]		
low load 120-125 MW	53.57	18.40
load transition range 125-215 MW	93.50	34.60
high load 215-225 MW	148.29	55.90
Relative time of exceeding the 250 mg/Nm^3 level [%]		
low load 120-125 MW	3.60	0.12
load transition range 125-215 MW	8.62	2.22
high load 215-225 MW	17.63	2.36

TABLE IV
NO_x EMISSION [MG/NM^3].

Load range	SILO OFF	SILO ON
Average NO_x emission [mg/Nm^3]		
low load 120-125 MW	261.27	262.16
load transition range 125-215 MW	321.21	275.23
high load 215-225 MW	363.41	328.50
Relative time of exceeding the 300 mg/Nm^3 level [%]		
low load 120-125 MW	13.30	11.35
load transition range 125-215 MW	61.45	20.11
high load 215-225 MW	96.50	64.98

Average exhaust gases temperature is essentially lower when SILO optimization is enabled (refer table II). Relative time of exceeding the 140 degrees C level is essentially reduced. It results in a higher boiler efficiency. Average CO emission is essentially reduced when SILO optimization is enabled (refer table III). Relative time of exceeding the 250 mg/Nm^3 level is essentially reduced. It results in a higher boiler efficiency and significantly lower air pollution. SILO reduces the NO_x emission (refer table IV). Relative time of exceeding the 300 mg/Nm^3 level is reduced, especially when the unit load is high and when there is a unit load transition (positive effect of the new *Transition State* layer). It results in lower air pollution. Average LOI (Loss Of Ignition) is reduced when SILO optimization is enabled (refer table V). Relative time of exceeding the 250 mg/Nm^3 level is reduced by 7 % in case of low unit load and by 26.5 % in case of high unit load. It results in a higher boiler efficiency. Moreover a power plant can sell a bottom ash from the boiler only if the LOI is lower than 5 %. Superheat and reheat stem temperatures are maintained on a similar level when SILO is enabled and when optimization is disabled. SILO slightly improves superheat and reheat stem temperature when the unit operates with low load. In such case it is hard to maintain a stem temperature at the desired 535 degrees C level. SILO increases these temperatures by 1.5 degrees C.

VII. SUMMARY

Observations and results from a real life SILO implementation for combustion process optimization show that the optimizer is able to perform an effective steady state optimization. Values of all optimized process outputs were improved when power unit was operating with the low and high load. Significant improvement can also be observed during a unit load transition. This is a result of the new *Transition State* layer application.

TABLE V
LOI – LOSS OF IGNITION [%].

Load range	SILO OFF	SILO ON
Average LOI [%]		
low load 120-125 MW	4.5318	4.1760
load transition range 125-215 MW	4.3231	3.9833
high load 215-225 MW	5.7711	5.3074
Relative time of exceeding the 5 % level [%]		
low load 120-125 MW	30.93	23.78
load transition range 125-215 MW	21.77	16.98
high load 215-225 MW	82.04	55.51

VIII. ACKNOWLEDGMENTS

This work is financed from polish Scientific Funds in 2010-2011 years as a research project.

REFERENCES

[1] Wojdan, K., Swirski, K., Warchol, M., Chomiak, T., Jarmoszewicz, G.: Bio-inspired process control, chapter of a book: Bio-inspired Computing and Networking, CRC Press, (2011)

[2] Wojdan, K., Swirski, K., Warchol, M., Maciorowski, M.: Maintaining Good Conditioning of Model Identification Task in Immune Inspired On-line Optimizer of an Industrial Process. Engineering Letters, vol. 17 issue 2, pp 93-100 (2009)

[3] Wojdan, K., Swirski, K., Warchol, M.: Transition States Handling in Self-Adaptive Steady State Optimizer of Industrial Processes, Proceedings of AsiaMIC2010, ACTA Pres, (2010)

[4] Eldra P. Solomon and Linda R. Berg and Diana W. Martin and Claude A. Villee: Biology. Saunders College Publishing, Fort Worth, Texas, USA (1993)