

Unfalsified Adaptive Control with Online Optimization to a pH Neutralization Process

Tanet Wonghong *

Department of Electrical Engineering, Faculty of Engineering, Kasetsart University, Si Racha Campus, Si Racha, Chonburi 20230, Thailand

Abstract

This work presents a data-driven adaptive controller with an online optimization. The proposed algorithm computes a real-time controller parameter vector for a PID controller structure using only plant input-output measurement data. The controller can be adapted by the algorithm to handle a change of set-points with a minimum knowledge about a plant. Finally, we demonstrate the performance of the algorithm using a pH neutralization process with an adaptation of PID controllers.

Keywords: Data-driven Control; Adaptive Control; Unfalsified Control; Online Optimization; Automatic PID Controller Tuning; pH Neutralization Process

1. Introduction

Data-driven industrial process control has been actively researched in recent years [1]. In many industrial applications, deriving a model from the first principles that is both simple and reliable for controller design is both difficult and time-consuming. Nowadays, the progress of data-acquisition technology is easy and straightforward to collect a large amount of measurements from industrial processes. In [2], they claimed that the utilization of measurement data as an alternative to physical knowledge to design simple fixed-order controllers, e.g. a PID controller, has become something of increasing interest throughout the years because it is cheaper and less time-consuming.

The motivation for this paper has arisen from the problem of a traditional model-based approach. If we can derive an accurate model of a plant, it is sufficient to design a good performing controller, e.g. aerospace applications, it is straightforward

to design such a controller. However, model-based approaches may not be enough to cope with a situation when a model is difficult to obtain or even unavailable (worst case situation) [3]. For this reason, a data-based methodology should be alternatively considered to solve this issue. This is the case when we deal with industrial applications, e.g. a pH neutralization process which the modeling step may be hard to derive.

Unfalsified control is one of the data-driven techniques [3]. The main attractive point of this approach is that no plant model is required to design an adaptive controller. Using this concept, an adaptive controller is implemented by means of switching among many candidate controllers in a predefined set. The plant input signal and the plant output signal are observed while one candidate controller is active in the feedback loop, and they are used for the unfalsification procedure to decide on which candidate

controller in the set should be switched on as the next active controller.

The very first version of the unfalsified control concept was originally proposed by [4]. It was motivated by a limitation of traditional model-based theories. The unfalsified control influenced the adaptive control theory because it can classify the performance of a given set of candidate controllers using only the observed plant input-output data to meet a given closed-loop specification. The key issue is the computation of a fictitious reference signal and a fictitious error signal of a non-active candidate controller while another controller is active. In [4], they claimed that using this approach based on a given cost function, the observed plant input-output data are sufficient to falsify candidates from a set of controllers that need not actually be in the feedback loop to be falsified. This is an important issue because the switching of an active controller in the set of unfalsified controllers can give a better transient response than other adaptive controller algorithms available at that time, e.g. [5], [6], [7] which require all candidate controllers to be active one by one in the feedback loop to check whether or not they are suitable. This implies that it can take a long time to end up with the last switching to a stabilizing controller if the number of candidate controllers is large.

Further developments by [8] and [9] led to a concept of a cost-detectable cost function. Unfalsified adaptive controllers are not always safe using the algorithm proposed in [4]. Safe means that the closed-loop stability is always guaranteed. One example of a cost-detectable cost function was proposed in [8]. The cost-detectability means that instability of all candidate controllers can be detected by a cost function. Moreover, the switching method of an active controller proposed in [4] was modified in [9] to be implemented with a cost-detectable cost function to achieve a safe algorithm. A switching algorithm always compares the

performance of the active controller with that of all non-active controllers. If the active controller performs badly, a minimum-cost non-active controller in a predefined set is switched to. Otherwise, the active controller is always in the feedback loop. Such an algorithm was proposed by [10], so called ε -hysteresis switching algorithm.

The cost function proposed by [9] was reported by [11], [12], and [13] that there was a severe problem. It will only detect instability of a controller in the set of candidate controllers if this controller is active. In [11], they proved that the problem originated from the pole-zero cancellation in the transform of the fictitious error signal. This leads to failure of the cost function of [9] if it is used to the candidate controllers as shown by an example in [11]. A new fictitious error signal to solve this problem was also proposed by [11].

In [11], they also developed a new cost-detectable cost function that can detect instability of non-active candidate controllers. This cost function leads to a possibility to optimize a predefined set of candidate controllers. An evolutionary algorithm is used as an optimization technique because it is easy to handle and manipulate a set of candidate controllers. The simulations were performed using a third-order stable linear system with a time delay as an unknown plant. Using this method, an optimal performance could be achieved after the evolutionary algorithm was executed once. Further developments by [14] presented a way to deal with a nonlinear process model and measurement errors at output. The effectiveness of industrial automatic tuning for the adaptation of linear single-loop unfalsified control algorithm was demonstrated via a nonlinear non-minimum phase continuous stirred-tank reactor model.

Following on the concept presented in [14], this algorithm consists of 3 modules:

1. Fictitious reference signal computing module
2. Cost computing module and

3. Supervisor module.

These modules lead to an automatic controller tuning without physical knowledge about a plant. We will demonstrate how controller parameter can be adapted via a PID controller structure.

This paper is organized as follows: In section 2, a data-based adaptive control based on unfalsified control is presented. Then, in section 3, one application to a pH neutralization process is demonstrated with an evolution of PID controllers to handle changes of set-points. Finally, conclusions and future work are given.

2. Materials and Methods

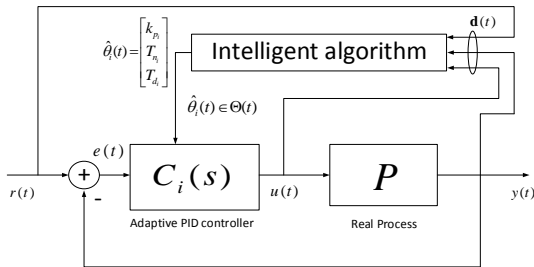


Fig.1. Data-based Adaptive Control System.

We consider a SISO system in the continuous time-domain as shown in Fig. 1.

A mapping $\Lambda(P, \hat{C}(s)): L_{2e} \rightarrow L_{2e}^2$ that transforms $r(t) \rightarrow [u(t), y(t)]^T$ is called an adaptive control system. We denote $r(t) \in L_{2e}$ as the reference signal or the external excitation. We assume that the real process P in Fig. 1 is completely unknown. A black-box mapping $P: L_{2e} \rightarrow L_{2e}$ that transforms $u(t) \rightarrow y(t)$ is called an unknown plant P .

Only information that we need to measure is the process output $y(t)$. The process input $u(t)$ is generated from one active PID controller by using the controller parameter vector $\hat{\theta}_i(t)$ which is computed by the proposed algorithm.

This algorithm requires only a data vector signal $\mathbf{d}(t)$ which consists of the reference signal $r(t)$, the control signal $u(t)$ and the observed output signal $y(t)$;

$$\mathbf{d}(t) = \begin{bmatrix} r(t) \\ u(t) \\ y(t) \end{bmatrix}$$

As shown in Fig. 2, we have three important components to build such an algorithm they are as follows: 1. Fictitious reference signal computing module 2. Cost computing module and 3. Supervisor module. We will discuss them later in details.

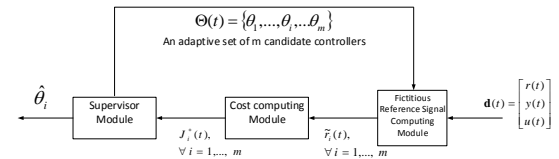


Fig.2. Structure of intelligent algorithm.

A set of candidate controllers is defined as

$$\Theta(t) = \{\theta_1, \dots, \theta_i, \dots, \theta_m\}$$

where θ_i is a controller parameter vector.

Since we aim this algorithm to deal with a nonlinear process, $\Theta(t)$ can be changed over time in order to adapt to new process dynamics due to a change of set-points.

2.1 Fictitious Reference Signal Computing Module

The original fictitious reference signal for a candidate controller $C_i(s)$ is defined [4] in Laplace domain as

$$\tilde{R}_i(s) = C_i^{-1}(s)U(s) + Y(s) \tag{1}$$

and the original fictitious error signal for a candidate controller $C_i(s)$ can be computed from

$$\tilde{E}_i(s) = \tilde{R}_i(s) - Y(s) = C_i^{-1}(s)U(s) \tag{2}$$

Since we use a PID controller structure $C_i(s) = k_{p_i} (1 + \frac{1}{T_{n_i}s} + \frac{T_{d_i}s}{\zeta s + 1})$, thus we obtain

$$\tilde{R}_i(s) = \frac{\zeta \frac{T_{n_i}}{k_{p_i}} s^2 + \frac{T_{n_i}}{k_{p_i}} s}{(T_{d_i}T_{n_i} + \zeta T_{n_i})s^2 + (T_{n_i} + \zeta)s + 1} U(s) + Y(s)$$

Now we can compute the fictitious reference signals and the original fictitious error signals for all m candidate controllers to be the inputs for the cost computing module.

2.2 Cost Computing Module

The original cost function is defined in [9],

$$\tilde{J}_i(t) = \frac{\|\tilde{e}_i(t)\|^2 + \gamma \|u(t)\|^2}{\|\tilde{r}_i(t)\|^2} \tag{3}$$

The problem of (3) was proven in [11] that unstable dynamics of non-active destabilizing controller are not excited using (3) and thus it cannot be used for a controller parameter optimization. However, the cost computing module uses the modified cost function as defined in [11]

$$J_i^*(t) = \frac{\|e_i^*(t)\|^2 + \gamma \|u_i^*(t)\|^2}{\|r(t)\|^2} \tag{4}$$

where $e_i^*(t)$ is the new fictitious error signal and $u_i^*(t)$ is the new fictitious control input signal and γ is a positive small number. We consider

$$\tilde{E}_i(s) = \frac{1}{1 + C_i(s)P(s)} \tilde{R}_i(s) = S_i(s)\tilde{R}_i(s) \Rightarrow \tilde{S}_i(s) = \frac{\tilde{E}_i(s)}{\tilde{R}_i(s)}$$

Although the sensitivity function $S(s)$ is unknown because $P(s)$ is unknown, we still can estimate it using the deconvolution in time domain.

$$\tilde{e}_i(t) = s_i(t) \otimes \tilde{r}_i(t) \Rightarrow \tilde{s}_i(t) = \tilde{r}_i(t) \otimes^{-1} \tilde{e}_i(t)$$

and \otimes^{-1} is a deconvolution operator. A practical way to compute was shown in [11]. Then we can compute

$$e_i^*(t) = \tilde{s}_i(t) \otimes r(t)$$

and

$$u_i^*(t) = c_i(t) \otimes e_i^*(t)$$

where $c_i(t) = L^{-1}\{C_i(s)\}$. Finally, the computed cost values for all m-candidate controllers will be sent to the supervisor module.

2.3 Supervisor Module

The supervisor will perform two functions:

1. Selection of active controller

Since we work with an unknown plant, we arbitrarily choose the initial set of candidate controllers and choose one controller in that set to be active first. If the process is stable by itself, we should start with a low gain controller (small k_{p_i}) and then the supervisor will select the best available controller from the initial set. This criterion is based on the work of [3], ϵ -hysteresis switching algorithm. This algorithm scans performances of all candidate controllers. If the active controller perform poorly, it will switch to the best available controller in that set.

2. Adaptation of a whole set of candidate controllers (online optimization)

The supervisor has to monitor whether or not the process output has almost settled to the current set-point. If it does so, the supervisor will activate an evolutionary algorithm (EA) to obtain an optimized controller for this set-point and use it to generate a new set of candidate controllers $\Theta(t)$ for the next set-point. In this work, we use an evolution strategy (ES) [15] as the EA because it is easy to handle and manipulate the controller parameter vector.

Evolution Strategy

It is an optimization problem for a fitness function $f(\theta_i)$:

$$\theta^* = \min f(\theta_i)$$

θ_i denotes an n-dimensional object parameter vector, e.g. a PID controller parameter vector in a search space of candidate controllers,

$$\Theta_{EA}^{PID} = \mathbf{K}_p \times \mathbf{T}_n \times \mathbf{T}_d$$

and θ^* denotes an optimal controller parameter vector.

Representation

The ES works with a population $\mathbf{P}(q)$ of the size μ and λ . μ denotes the number of parent individuals and λ denotes the number of offspring individuals. q denotes the number of generations. An \mathbf{a}_i individual consists of a controller parameter vector θ_i and an internal (self-adaptation) n-dimensional strategy parameter vector \mathbf{s}_i , and its fitness value $f(\theta_i)$:

$$\mathbf{a}_i = (\theta_i, \mathbf{s}_i, f(\theta_i)).$$

Note that \mathbf{s}_i is not involved in the computation of the fitness of the individual but it is transmitted to the offspring.

The individuals \mathbf{a}_i construct a population, i.e. μ parent individuals $\mathbf{a}_i, i=1,2,\dots,\mu$ and λ offspring individuals $\hat{\mathbf{a}}_i, i=1,2,\dots,\lambda$. At generation, the population of the parent individuals and the offspring individuals can be defined as

$$P_\mu(q) = \{\mathbf{a}_1(q), \mathbf{a}_2(q), \dots, \mathbf{a}_\mu(q)\}$$

$$P_\lambda(q) = \{\hat{\mathbf{a}}_1(q), \hat{\mathbf{a}}_2(q), \dots, \hat{\mathbf{a}}_\lambda(q)\}.$$

Recombination

A basic recombination in the ES uses two parent individuals to create one child individual. To obtain λ children individuals, the recombination is performed λ times. There are two recombination variants: 1)

discrete recombination and 2) intermediate recombination. We assume that two parent vector \mathbf{x} and \mathbf{y} are uniformly randomly chosen from $P_\mu(q)$ to produce a child vector \mathbf{z}' :

$$z'_k = \begin{cases} x_k \text{ or } y_k & : \text{ discrete} \\ \frac{x_k + y_k}{2} & : \text{ intermediate} \end{cases}$$

where $k=1,2,\dots,n$ ($n=3$ for PID controller structure). The former is used for the controller parameter vectors and the latter is used for the strategy parameter vectors.

Mutation

Mutation is very important for the ES because it is the source of genetic variations. After the recombination, each child individual is mutated to an offspring individual. Mathematically, each controller parameter vector θ' is mutated using a normal distribution $N(0, \hat{s}_k)$:

$$\hat{\theta}_k = \theta'_k + \hat{s}_k \cdot N(0, \hat{s}_k).$$

According to the self-adaptation mechanism of the ES, each strategy parameter s'_j is modified log-normally:

$$\hat{s}_k = s'_k \cdot \exp(\delta \cdot N(0,1))$$

where δ is an external parameter. Usually it is inversely proportional to the square root of the problem size $\delta = \frac{1}{\sqrt{n}}$. Thus an

offspring individual is defined as

$$\hat{\mathbf{a}}_l = (\hat{\theta}_l, \hat{\mathbf{s}}_l, f(\hat{\theta}_l))$$

where $l=1,2,\dots,l$.

Selection

There exist two different selection for the evolution strategy, (μ, λ) and $(\mu + \lambda)$ selections. The difference between both selections is defined by the set of individuals involved in the selection. The former selects the μ best individuals out of the offspring,

while the latter selects the μ best individuals out of the union of parents and offspring to form the next population. [16] pointed out that μ has to be chosen larger than one and a ratio of $\frac{\mu}{\lambda} = \frac{1}{7}$ is optimal according to the accelerating effect of self-adaptation.

Termination

Convergence of the ES is defined in the sense of probabilistic convergence

$$\lim_{q \rightarrow \infty} \Pr(f(\theta_i(q)) = f^*) = 1$$

where f^* is a global minimum. In practice, if the location of the global minimum $f^* = f(\theta^*)$ is reached with an arbitrary small ε , the termination criterion is

$$\lim_{q \rightarrow \infty} \Pr(|f(\theta_i(q)) - f(\theta^*)| \leq \varepsilon) = 1.$$

Remark: after termination of ES, there is a problem, i.e. a diversity of candidate controllers is lost. Before it will be used for the next operating point, we need to diversify it as defined:

$$\Theta_j^F(t) = \left\{ \begin{array}{l} \theta_i(t^*) | \theta_i = [k_{p_i}^*, T_{n_i}^*, T_{d_i}^*]^T, \\ k_{p_i} \in \left\{ \dots, \frac{1}{2F} k_{p_i}^*, \frac{1}{F} k_{p_i}^*, k_{p_i}^*, Fk_{p_i}^*, 2Fk_{p_i}^*, \dots \right\} \\ T_{n_i} \in \left\{ \dots, \frac{1}{2F} T_{n_i}^*, \frac{1}{F} T_{n_i}^*, T_{n_i}^*, FT_{n_i}^*, 2FT_{n_i}^*, \dots \right\} \\ T_{d_i} \in \left\{ \dots, \frac{1}{2F} T_{d_i}^*, \frac{1}{F} T_{d_i}^*, T_{d_i}^*, FT_{d_i}^*, 2FT_{d_i}^*, \dots \right\} \end{array} \right\}$$

where F is a constant.

3. Results and Discussion

We consider the nonlinear process model for neutralization processes in a continuous stirred tank reactor (CSTR) [17] as shown in Fig. 3. The process dynamics can be described as

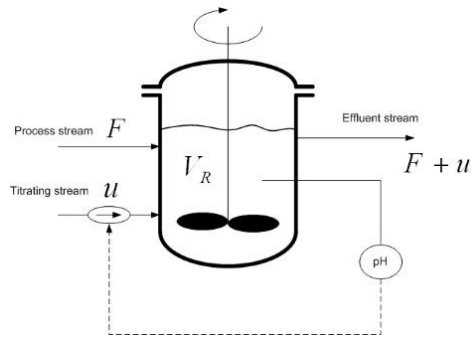


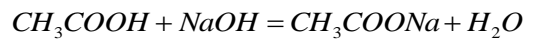
Fig.3. pH neutralization process.

$$\dot{x}_1 = \frac{(1 + I_T(x_1))(u - I_T(x_1)F)}{V_R(\partial I_T(x_1) / \partial x_1)} \tag{4}$$

$$\dot{x}_2 = \frac{1}{\tau}(x_1 - x_2) \tag{5}$$

$$y = x_2 \tag{6}$$

where $y = x_2$ is the measured output. In this work, we assume a perfect measurement. The feasibility of the proposed nonlinear control scheme for neutralization process where acetic acid (CH_3COOH) is neutralized by sodium hydroxide ($NaOH$) in a CSTR with a volume of $V_R = 5.5L$. The stoichiometry of this reaction is described by



The pH-value of the reactor effluent has to be adjusted without steady state offset in a range between pH 5 and pH 10 in the presence of varying process streams and deviations from the standard titration stream is $c_{B,in} = 0.01$ mol/L. The nominal concentration of acetic acid in the process stream is $c_{A,in} = 0.007$ mol/L and the nominal flow rate of the process stream is $F = 20$ L/h. Due to the incomplete dissociation of acetic in water and its reaction with sodium acetate, the system behaves like a buffer solution for low pH-values. For this reaction system the inverse of the standard titration curve is explicitly given by

$$I_T(pH) = -\frac{10^{-pH} - 10^{pH} K_w - [C_{A,in} / (1 + 10^{(pK_s - pH)})]}{c_{B,in} + 10^{-pH} - 10^{pH} K_w}$$

where $pK_s = 4.75$ and $K_w = 10^{-14} \text{ mol}^2/\text{L}^2$ for room temperature. Because of the low concentration of the reactants and the small reaction enthalpy of the neutralization process, we can assume constant temperature within the reactor. The flow rate of peristaltic pump which supplies the titrating stream is limited to the range $0 \leq u \leq 45 \text{ L/h}$. The sampling period is of 1 second and the additional time delay $\tau = 10$ seconds for pH sensor.

3.1 Simulation Setup

The simulations are carried out under the following assumptions:

- 1) The pH neutralization process model is used as the unknown plant P .
- 2) Operation scheme:

$$r_j(t) = \begin{cases} pH = 7(p_1) & : 0s \leq t < 300s; \\ pH = 9.5(p_2) & : 300s \leq t < 600s; \\ pH = 6(p_3) & : 600s \leq t < 900s; \\ pH = 8.5(p_4) & : 900s \leq t < 1200s; \\ pH = 7.5(p_5) & : 1200s \leq t < 1500s; \\ pH = 6.5(p_6) & : 1500s \leq t < 1800s. \end{cases}$$

Note that we first move the process to the standard operating point $y = 7$ from the origin (start-up). Then we move to the second operating after $t \geq 300s$ etc.

- 3) The PID controller structure is set up as follows
 - a) The initial controller set of PID controllers consists of 27 candidate controllers:

$$\Theta_0 = \left. \begin{cases} \theta_i(t=0) | \theta_i = [k_{p_i}, T_{n_i}, T_{d_i}]^T, \\ k_{p_i} \in \{1, 5, 10\}, \\ T_{n_i} \in \{1, 5, 10\} \\ T_{d_i} \in \{0.01, 1, 10\} \end{cases} \right\} = \Theta_1$$

We choose the first active controller as $\hat{\theta}(0) = [1, 1, 0.01]^T$.

- b) The search space for PID candidate controllers is

$$\Theta_{EA}^{PID} = \mathbf{K}_p \times \mathbf{T}_n \times \mathbf{T}_d \\ = [-1000, 1000] \times [0.00001, 1000] \times [0.00001, 1000].$$

- c) We define the diversity for 27 candidate controllers as

$$\Theta_j^F(t) = \left. \begin{cases} \theta_i(t^*) | \theta_i = [k_{p_i}^*, T_{n_i}^*, T_{d_i}^*]^T, \\ k_{p_i} \in \left\{ \frac{1}{F} k_{p_i}^*, k_{p_i}^*, F k_{p_i}^* \right\} \\ T_{n_i} \in \left\{ \frac{1}{F} T_{n_i}^*, T_{n_i}^*, F T_{n_i}^* \right\} \\ T_{d_i} \in \left\{ \frac{1}{F} T_{d_i}^*, T_{d_i}^*, F T_{d_i}^* \right\} \end{cases} \right\}$$

where $F = 2$ is chosen.

- 4) $\varepsilon = 0.01$ for the switching of an active controller.
- 5) $\zeta = 0.01$ to smoothen the derivative term
- 6) The cost function is used:

$$J_i^*(t) = \frac{\|e_i^*(t)\|_{\tau=[t_j, t_j^*]}^2 + \gamma \|u_i^*(t)\|_{\tau=[t_j, t_j^*]}^2}{\|r_j(t)\|_{\tau=[t_j, t_j^*]}^2}$$

where $\gamma = 0.0005$ is a control weighting.

- 7) The controller tuning points (t_j^*) are every 100s after a change of set-points.
- 8) The $(\mu + \lambda)$ -selection is used for ES.

3.2 Simulation Results

We first run the algorithm with the low gain controller $\hat{\theta}(0) = [1, 1, 0.01]^T$ that we choose from the initial controller set Θ_1 to start up the process to the first operating point p_1 . The supervisor will keep this controller up to 100 seconds because there is no a better controller available. At time 100 seconds, the supervisor activates the ES as shown in Fig. 5. Note that after the termination, the ES returns almost the same controller (Upper). Thus we need to diversify it before it uses for the second set-point (Lower). This will be kept as the new set of candidate controllers Θ_2 for the second operating point. The optimized controller

will be used as the new active controller during time 100 seconds to time 300 seconds. In addition, the supervisor will assign it as the first active controller for the second set-point and allowing the switching between candidate controllers occurs in the set Θ_2 . The same procedure is repeated again at time 400 seconds, at time 700 seconds, at time 1000 seconds, at time 1300 seconds, and finally at time 1600 seconds. This leads to an automatic PID controller tuning as shown in Fig. 11. A good output performance can be achieved as shown in Fig. 12.

4. Conclusion

We proposed the intelligent algorithm based on only measurement data. This algorithm can be considered as an economical method that can deal with a situation when a plant model is difficult to derive or even unavailable to design a controller. This is a case that we encounter industrial problems. In this paper, we demonstrated that the algorithm can work well with a pH neutralization process model which the process dynamics are highly nonlinear. In the future, we will investigate for a real single-input single output pH neutralization process, e.g. a pH control for hydroponics.

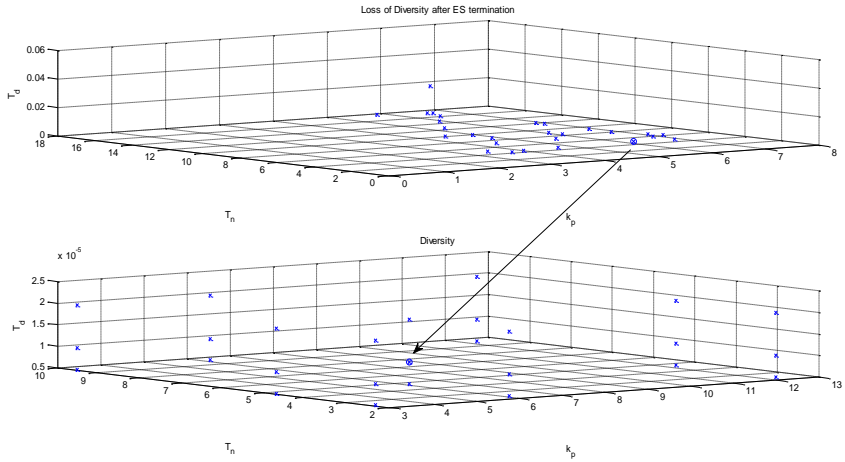


Fig.5. ES activation at 100 seconds to generate the new set of candidate controller Θ_2 .

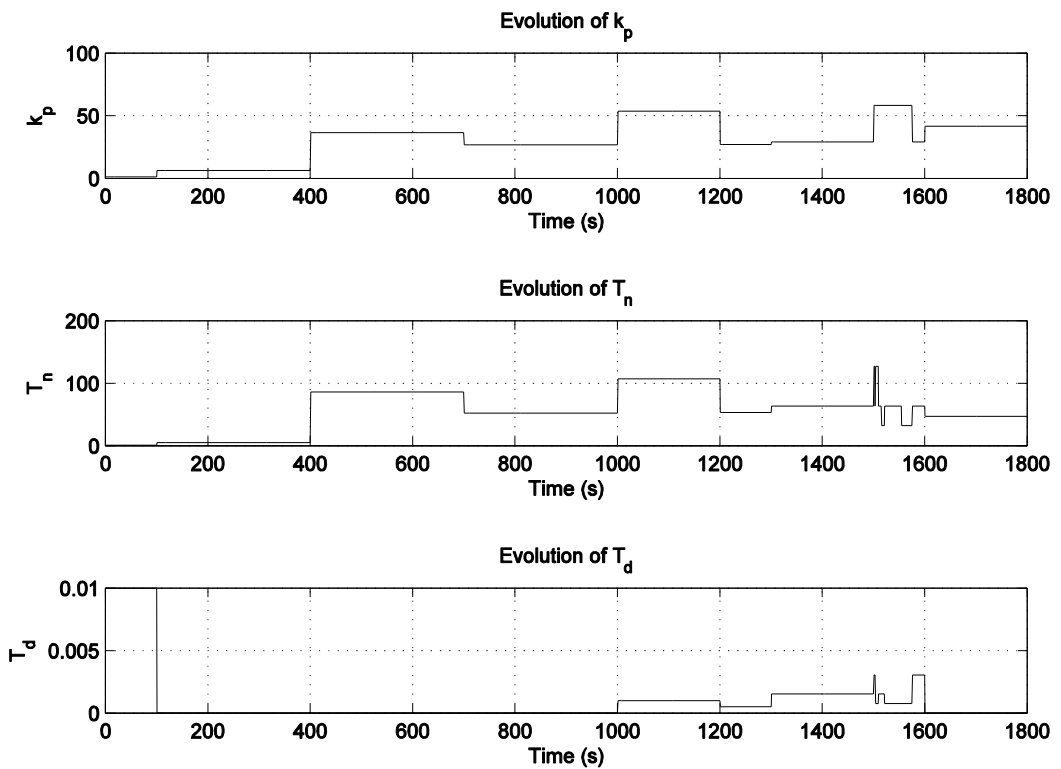


Fig.6. Adaptive PID controller.

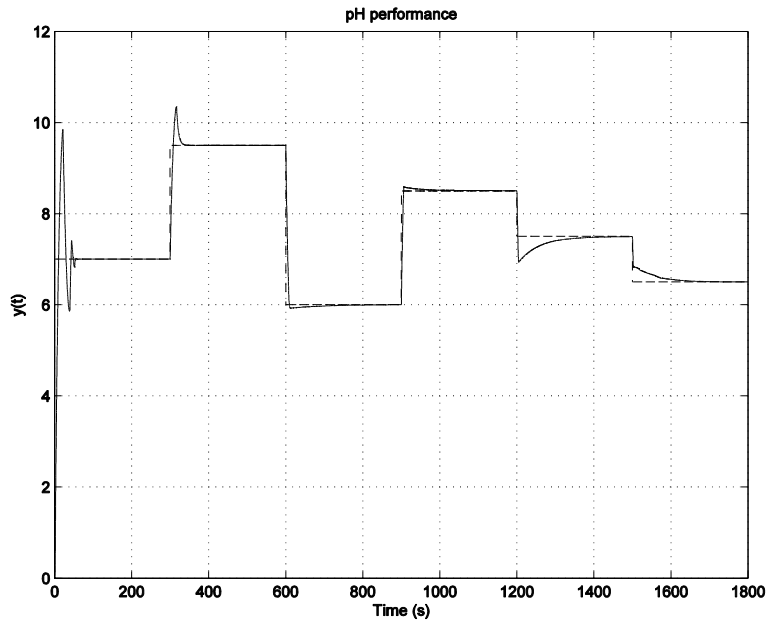


Fig.7. Output performance.

5. References

- [1] S. Yin, X. Li, H. Gao, and O. Kaynak, Data-Based Techniques Focused on Modern Industry: An Overview, *IEEE Transactions on Industrial Electronics*, Vol.62, pp. 657-667, 2015.
- [2] S. Formentin, K.V. Heusden, and A. Karimi, A Comparison of Model-based and Data-driven Controller Tuning. *Int. J. Adapt. Control Signal Process*, 00:1-16, 2012.
- [3] Z.S. Hou and Z. Wang, From Model-Based Control to Data-driven Control: Survey, Classification and Perspective, *Information Sciences*, Vol.253, pp. 3-35, 2013.
- [4] M.G. Safonov and T. Tsao, The Unfalsified Control Concept and Learning, *IEEE Transactions on Automatic Control*, Vol.42, No.6, pp. 843-847, 1997.
- [5] M. Fu and B.R. Barmish, Adaptive Stabilization of Linear Systems via Switching Control, *IEEE Transactions on Automatic Control*, Vol.31, pp. 1097-1103, 1986.
- [6] D.E. Miller and E.J. Davison, An Adaptive Controller which Provides an Arbitrary Good Transient and Steady-state Response, *IEEE Transactions on Automatic Control*, Vol.36, No.1 pp. 68-81, 1991.
- [7] S.R. Weller and G.C. Goodwin, Hysteresis Switching Adaptive Control of Linear Multivariable Systems, *IEEE Transactions on Automatic Control*, Vol.39, pp.1360-1375, 1994.
- [8] M. Stefanovic, R. Wang and M.G. Safonov, Stability and Convergence in Adaptive Systems, *Proc. American Control Conf.*, Boston, pp.1923-1928, 2004.
- [9] R. Wang, A. Paul, M. Stefanovic and M.G. Safonov, Cost-detectability and Stability of Adaptive Control Systems. *Proc. IEEE CDC-ECC*, Seville, pp. 3584-3589, 2005.

- [10] A.S. Morse and D.Q. Mayne and G.C. Goodwin, Application of Hysteresis Switching in Parameter Adaptive Control, IEEE Transactions on Automatic Control, Vol.37, pp.1343-1354, 1992.
- [11] A. Dehghani, B.D.O. Anderson, and A. Lanzon, Unfalsified Adaptive Control: A New Controller Implementation and Some Remarks, Proc. European Control Conf. Kos, pp. 709-716, 2007.
- [12] C. Manuelli, S.G. Cheong, E. Mosca, and M.G. Safonov, Stability of Unfalsified Adaptive Control with Non SCLI Controllers and Related Performance under Different Prior Knowledge, Proc. European Control Conf., Kos, pp.702-708, 2007.
- [13] T. Wonghong and S. Engell, Automatic Controller Tuning via Unfalsified Control. Journal of Process Control, Vol.22, pp. 2008-2025, 2012.
- [14] A. Eiben and J. Smith, Introduction to Evolution Computing. Springer-Verlag, Berlin, 2003.
- [15] H.P. Schwefel, Collective Phenomena in Evolutionary Systems. Preprints of the 31st Annual Meeting of the International Society for General System Research, volume 2, Budapest, pp.1025 - 1033.
- [16] K.U. Klatt and S. Engell, Nonlinear Control of Neutralization Process by Gain-scheduling Trajectory Control. Industrial & Engineering Chemistry Research, Vol.35, pp.3511-3518, 1996.