MODIFICATION OF DEAD BEAT ALGORITHM FOR CONTROL PROCESSES WIH TIME DELAY

Mikuláš Alexík

University of Žilina, department of Technical Cybernetics,Slovak Republic

Abstract: This paper describes extension of DB(n) control algorithm which improves the behaviour quality of control processes with time delay. This extension is called eDBd algorithm and it can function even if controller output is limited. The synthesis was derived for control loop with proportional third order behaviour of a controlled plant. However, the algorithm can be generalised for controlled plants with higher order and applied for plants with non-minimal phase, for plants with time delay and also for plants with combination of non-minimal phase with time delay. This is shown in simulation experiments, which are described in the paper. For experimental evaluation of the control algorithm in real time, hardware in loop simulation was used, which replaces analogue model of real plant with A/D and D/A converters and PC as a controller.

Keywords: algorithm design, time delay, sampling interval, dead beat control

1. INTRODUCTION

Discrete version of time optimal control or DB(n) algorithm is generally known as an elementary control algorithm. Its disadvantage is a great jump of actuating variable usually exceeding actuating variable limitation. The author derived three extensions of DB(n) algorithm where this limitation is respected. First extension is the anti windup extension of classical DB(n) algorithm - eDB algorithm. The second is the state space version of eDB algorithm (Alexik, 2002). The third extension is a version of eDB algorithm, which improves behaviour quality of control processes with time delay. This version is called eDBd algorithm. All described algorithms are the best ones from algorithms operating only with output from controlled process. The synthesis was derived for control loop with proportional third order controlled plant. However, the algorithm can be generalised for controlled plants with higher order, and applied for plants with non-minimal phase, for plants with time delay and also their combination, as is shown in simulation experiments. Real time continuous identification of plant parameters consecutiveness of control algorithm parameters synthesis also enabled us to realize adaptive control with eDB algorithm (Alexik, 2001). Verification of adaptive eDB and eDBd algorithms in real time was carried out by block the scheme depicted in Fig. 1.

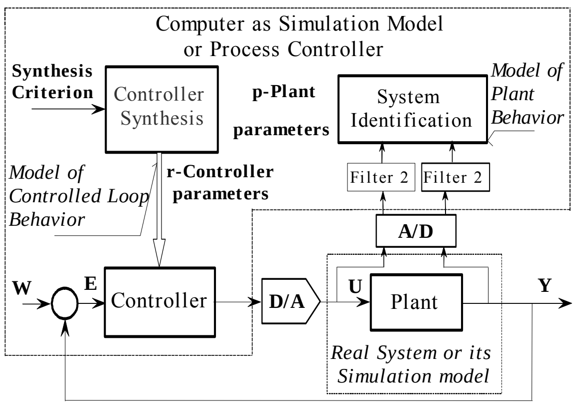

Fig 1.Block scheme in algorithms verification.

The paper is organised as follows. Section 2 describes extensions of both versions of eDB and eDBd algorithm, section 3 describes real time operating model of the controlled plant and section 4 digital and real time simulation experiments. The paper ends with a conclusion and an outlook in Section 5.

2. DB ALGORITHMS FOR PROCESS WITH DEAD TIME

Control of time-delayed processes is a known problem and many researchers work in this area. Block scheme in Fig, 2a shows a classical control loop with dead time process, and Fig 2b shows control loop where dead time process is between controlled process and control algorithm (e.g. control through internet). An interesting analytical approach to PID algorithm design for dead time processes were published in (Vitečková, 1999) and (Žáková, 2003). Both authors described a procedure to find such proportional gain of PID algorithm (smaller as for d=0), which provides stability as well as suitable quality of set point response. After every proceeding, which significantly decreases proportional gain in PID algorithm, the set point response quality is much worse than in performance that is only time delayed of previous response, or it is response from actuating variable computed for system without dead time but applied to system with dead time (Control loop with Schmidt predictor). The comparison of these proceedings is shown on Fig. 3, the controlled process for this example is from (Schlegel, *at all,* 2003). On set point responses on Fig.3 is documented a significantly better loop response quality for loop with Schmidt predictor in comparison with classical

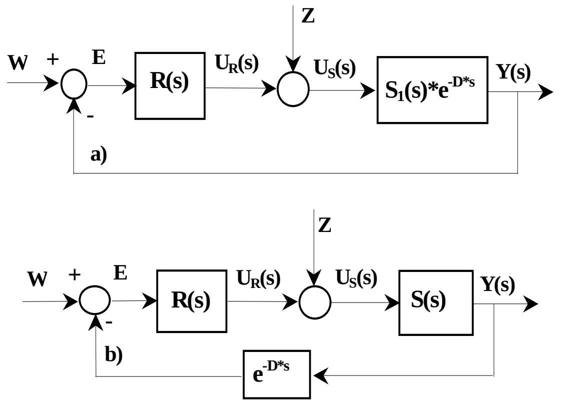

Fig. 2 Block scheme of control loops with dead time.

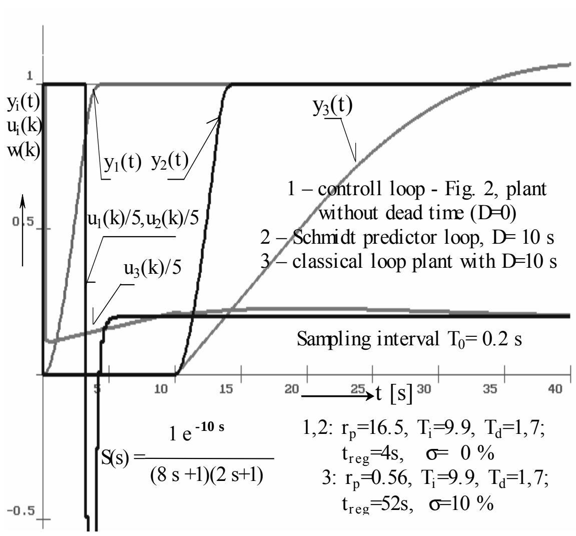

Fig. 3 Set point responses and control signals of dead time process controlled by PID algorithm.

control loop (see settling time „t_{reg}" and overshoot „σ" for output $y_2(t)$). The fact, that time behaviour of actuating variables $u_1(k)$ and $u_2(k)$ are the same is essential.

It will be appropriate to find a control algorithm which could provide set point responses as "$y_2(t)$" on Fig. 3, but for classical control loop structures from Fig. 2. Such an easy compensation is possible only (at discrete realization of dead time only) with algorithm DB(n) as is shown from following procedure. Consider classical control loop with dead time process and DB(n) algorithm (1). Set point response (2) in this case has the same performance as in loop without dead time process, however response in considered case (2) is time-delayed. Controller output u(k) (3) has the same performance, without time delay, in the considered case (dead time process) as well as in control loop with process without dead time.

$$S(s) = \frac{K*B(s)*e^{-D}}{A(s)} \cong \frac{B(z^{-1})*z^{-d}}{A(z^{-1})} \quad \Rightarrow$$

$$R(z) = \frac{Q(z^{-1})}{1\text{-}P(z^{-1})\ z^{-d}} = \frac{q_0\ A(z^{-1})}{1\text{-}\ q_0\ B(z^{-1})\ z^{-d}} \qquad (1)$$

and $\mathrm{d} = Int(D/T_0)$, T_0 – Sampling interval;

Transfer function of control loop is:

$$F_w(z^{-1}) = \frac{Y(z)}{W(z)} = \frac{\frac{B(z^{-1})z^{-d}}{A(z^{-1})} * \frac{q_0\ A(z^{-1})}{1\text{-}\ q_0\ B(z^{-1})z^{-d}}}{1 + \frac{B(z^{-1})z^{-d}}{A(z^{-1})} * \frac{q_0\ A(z^{-1})}{1\text{-}\ q_0\ B(z^{-1})z^{-d}}} =$$

$$= \frac{\frac{q_0\ B(z^{-1})\ z^{-d}}{1\text{-}\ q_0\ B(z^{-1})\ z^{-d}}}{\frac{1 + q_0 B(z^{-1})z^{-d}(1-1)}{1\text{-}\ q_0\ B(z^{-1})\ z^{-d}}} = q_0\ B(z^{-1})\ z^{-d} \qquad (2)$$

$$U(z^{-1}) = \frac{R(z^{-1})W(z^{-1})}{1 + S(z^{-1})R(z^{-1})} = q_0\ A(z^{-1})\ W(z^{-1}) \qquad (3)$$

Resulting set point responses for processes with dead time (2) should have the same quality, as ones without dead time and it would only be shifted in time of "d" sampling intervals. In the equation (2) it is considered linear transfer function R(z). In real control loops, transfer function of controller is not linear, because computed controller output is often out of controller limits. In such case, responses with overshoot are being produced.

Since even small time delay will also cause response overshoot, control signal rising to limitation, or flips from one extreme to the other. Therefore a control algorithm can be realized only when an anti-windup version of algorithm is provided, which is documented on Figure 4 in set point w_2 and w_3 responses. In assumed DB(n) algorithm it is needed to realize such computation, which will secure anti –windup performance of step response. Classical DB(n) algorithm does not have this performance, therefore in the next part of paper will be describe an extension of classical DB(n) algorithm to the form, which works well also under the controller output limitation.

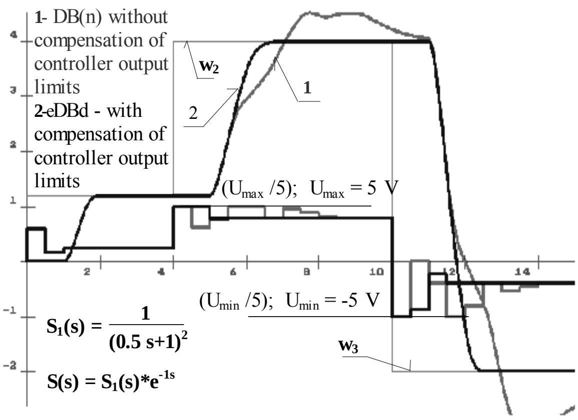

Fig. 4 Comparison of classical and extended DB(n) control on closed loop with time delay.

2.1. Extension of DB Algorithm

Synthesis of parameters for known structure of DB(n) algorithm comes out from Z transform function (1) of controlled plant. The disadvantage of DB algorithm is a great jump of actuating variable. These jumps are computed from demand of discrete time optimal transient response behaviour of control loop. Actuating variable (equation (6)) cannot be realized in any case, considering D/A converter output limitation and this limitation is cause of follow-up controlled output overflow as depicted on Fig. 5. The first part of Fig. 5 (control loop response for set point w_1) is without overshoot, because the actuating variable is under limitation. In the second part of Fig. 5 there is overshoot for classical DB(n) algorithm. Actuating variable for classical DB(n+1) algorithm is under D/A converter output limitation. In great change of set point (third part of Fig. 5 – set point w_3) only eDB algorithm provides response without overflow respecting actuating variable limitation.

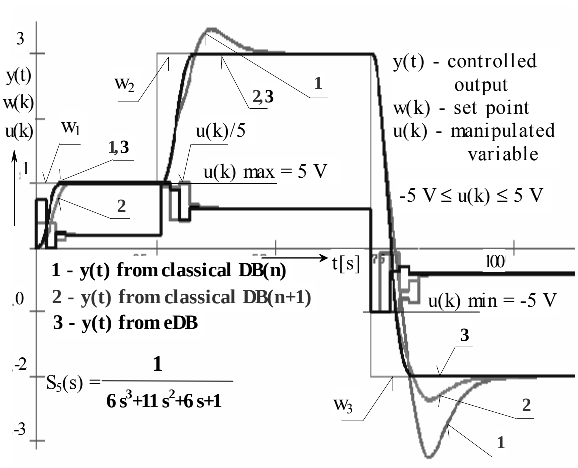

Fig. 5 Comparison of classical and extended DB(n) algorithm.

The presented eDB algorithm (8) has the most important attribute – the ability to assure discrete time optimal control in despite of manipulated variable limitation. This is secured by extended – an anti windup part of the algorithm. Derivation of PID version was described by an author in (Zentko *and* Alexík 1986) and derivation of eDB version in (Alexik, 2002).

$$S(p) = \frac{K}{(T_1 p+1)(T_2 p+1)(T_3 p+1)} = $$
$$= \frac{b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}} \qquad (4)$$

$$q_0 = (1/\sum b_i); \quad q_i = q_0 * a_i; \quad p_i = q_0 * b_i \qquad (5)$$

Classical DB(n) algorithm has the form:

$$u(k) = q_0\, e(k) + q_1\, e(k-1) + q_2\, e(k-2) + q_3\, e(k-3) +$$
$$+ \left[p_1\, u(k-1) + p_2\, u(k-2) + p_3\, u(k-3)\right] \qquad (6)$$

Derivation of extended form of DB algorithm (7) is outgoing from demand on equality of state space variables from linear part of controller output (***x(k)*** – see Fig. 6) and non-linear (***x^o(k)***) state variables in the steady state.

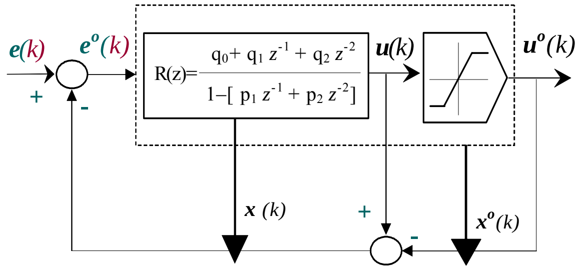

Fig. 6 Block scheme of extended DB(n) algorithm.

$$u(k) = q_0\, e(k) + q_1\, e(k-1) + q_2\, e(k-2) + q_3\, e(k-3) +$$
$$+ \left[p_1\, u(k-1) + p_2\, u(k-2) + p_3\, u(k-3)\right] -$$
$$- \left[q_{d1}\, u_d^o(k-1) + q_{d2} u_d^o(k-2) + q_{d3}\, u_d^o(k-3)\right\} \quad (7)$$

$$u_d^o(k\text{-}1) = [u(k\text{-}1)\text{-}u^o(k\text{-}1)]; \quad q_{di} = p_i + \frac{q_i}{q_0} \qquad (8)$$

$u^o(k)$ - D/A converter constraints;

State space form of extended eDN algorithm

$$x_1(k) = p_1 x_1(k-1) + p_2 x_2(k-1) + p_3 x_3(k-1) + e(k)$$
$$u(k) = q_1 x_1(k-1) + q_2 x_2(k-1) + q_3 x_3(k-1) + q_0 x_1(k)$$
$$x_1(k-1) = x_1(k) - (1/q_0)[u(k) - u^o(k)] \qquad (9)$$

A modal state space algorithm can obtain similar behaviour of feedback responses, however eDB algorithm (7) does not need a state estimator. State space form (9) of eDB algorithm is easier (7) because there is less arithmetic operation.

$$u(k) = q_0\, e(k) + q_1\, e(k-1) + q_2\, e(k-2) + q_3\, e(k-3) +$$
$$+ \left[p_1\, u^o(k-1) + p_2\, u^o(k-2) + p_3\, u^o(k-3)\right] -$$
$$- \left[q_{q1}\, u_d^o(k-1) + q_{q2} u_d^o(k-2) + q_{q3}\, u_d^o(k-3)\right\} \quad (10)$$

$$u_d^o(k\text{-}1) = [u(k\text{-}1)\text{-}u^o(k\text{-}1)]; \qquad q_{qi} = \frac{q_i}{q_0} \qquad (11)$$

$u^o(k)$ - D/A converter limitations;

Simplification of equation (7) to the form (10) is better before is executed eDBd extension.

2.2 .Robustness of eDB Algorithm

There is an important question. What effect does plant parameters change have on control loop performance quality for untouched parameters of control algorithm? Or in other words, what is the robustness of the control algorithm? From Fig. 7 it can be seen, that the quality of the control loop response is not affected by controlled plant parameter changing. Algorithm synthesis has been done for the nominal model. The picture figures that after changing plant parameters within plus or minus 20 % (not only time constant but also gain) but with controller parameters for nominal model, the quality of the loop response is not changing substantially. Consequently there is potential presumption that if synthesis of DB(n) algorithm parameters are done by identified parameters of controlled plant, although identification errors are + - 20%, the quality of control loop response won't be changing substantially. It comes to this, that it is appropriate to execute identification of a higher order plant as controlled plant with third order for which was eDB algorithm derived. The conjunction of continuous identification with eDB algorithm synthesis on this presumption enables a higher order controlled plant adaptive control. Extension eDB synthesis ability for more structure of controlled plants is enabled by Z - transform behaviour as is documented on equation (12) to (14). Controlled plant, which is described in continuous domain by third order transfer function without or with one or two positive or negative zeros (13), is described with the same transfer function structure (14) in discrete domain.

$$S(p)=\frac{K}{(T_3 * s+1)(T^2 * s^2+2aT * s+1)} \qquad (12)$$
$$a<1 \; or \; a\geq 1$$

$$S(p)=\frac{K(\pm T_5 * s+1)(\pm T_4 * s+1)}{(T_3 * s+1)(T^2 * s^2+2aT * s+1)} \qquad (13)$$

$$S(z)=\frac{b_1 * z^{-1}+b_2 * z^{-2}+b_3 * z^{-3}}{1+a_1 * z^{-1}+a_2 * z^{-2}+a_3 * z^{-3}} \qquad (14)$$

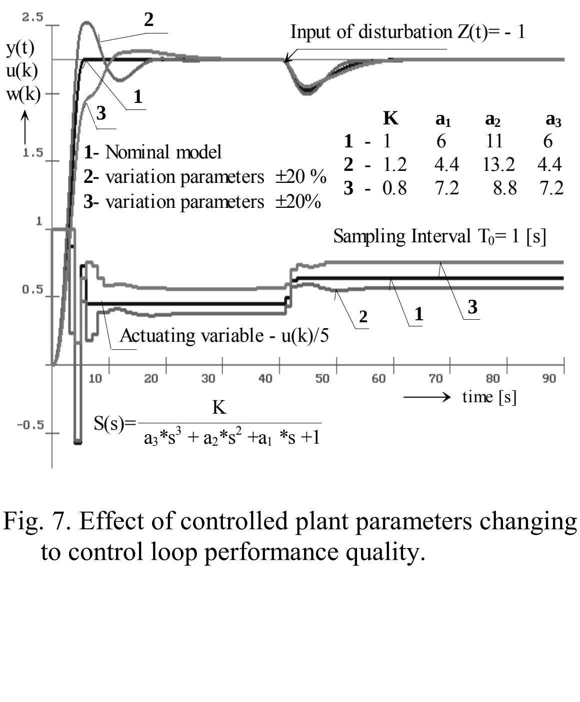

Fig. 7. Effect of controlled plant parameters changing to control loop performance quality.

2.3 .Extension of eDB with Dead Time

By time delay compensation we apply the same transfer function structure of discrete closed loop as in structure from Fig. 2 for continuous closed loop. This simple compensation is working only by discrete realization of dead time and only by DB(n) algorithm as was shown in equation (1) and (2). As it was commented in part 2, controller (1) for processes without dead time can be realized only when an anti-windup version of algorithm is provided. In part 2.1 this version of algorithm –eDB was commented and formulate as equation (7). Calculation of dead time ergo "z^{-d}" (for d > 0) in equation (3) represents shift of controller output "u(k-i)" and modification of equation (7). Adapted algorithm is represented by the equation (15).

$$\begin{aligned} u(k) &= q_0\, e(k) + q_1\, e(k-1) + q_2\, e(k-2) + q_3\, e(k-3) + \\ &+ \left[p_1\, u^o(k\text{-}d-1) + p_2\, u^o(k\text{-}d-2) + p_3\, u^o(k-\text{d}-3)\right] - \\ &- \left[q_{q1}\, u_d^o(k-1) + q_{q2} u_d^o(k-2) + q_{q3}\, u_d^o(k-3) + \right] \end{aligned} \qquad (15)$$

$$q_{qi} = q_i / q_0 \qquad (16)$$

$u_d^o(k\text{-}i) = [u(k\text{-}i)\text{-}u^o(k\text{-}i)]$; $u^o(k)$ - D/A limitation

Derived algorithm (15) is appropriate considered as a new modification of DB algorithms, called eDBd, because it can be applied on control time-delayed process. Initialization resetting of parameters, variables, shifting of error and output in every sampling interval is assumed. Algorithm (15) is formulated for a third order controlled processes because it is possible to model a higher order processes with the third order by adaptive control with continuing identification, as it is commented in (Alexík, 2001). Also state version of eDBd algorithm, as well as (9) for eDB, can be formulated.

3. MODEL OF THE PROCESS.

For evaluation of control algorithms, the hardware in loop simulation in comparison with the digital one is more suitable. In our case it is the continuous process modelled with the discrete/continuous model working in real time. The technical design of the model on Fig. 9 enables us to model the controlled plants of the first to the nine orders and also to change some parameters and the structure of the controlled system, which is suitable for the evaluation of robustness of the control algorithm. Fig. 8 shows and describes patching panel of model for controlled processes with time delay. The dynamics of the process realised with the model under Fig. 8 and Fig. 9 is described with continuous transfer functions and then is re-calculated to parameters of Z transfer function. Micro controller on Fig. 9 calculates differential equation, which represents continuous controlled plants. A/D and D/A conversion, differential equation computation and output signal shift in the time delay buffer "D" are executed with sampling interval 10 [ms]. It is also possible to connect a model such as the MIMO system with 2 inputs and 2 outputs.

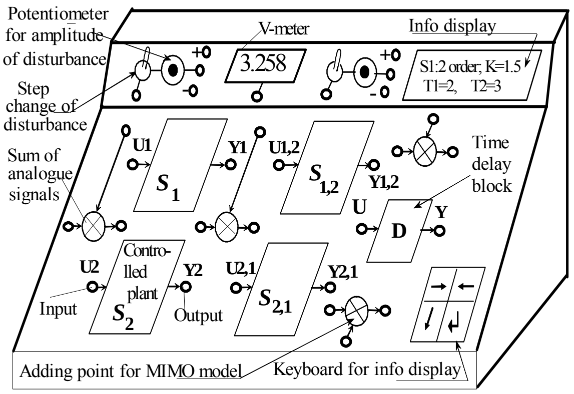

Fig. 8. Patching panel of the model.

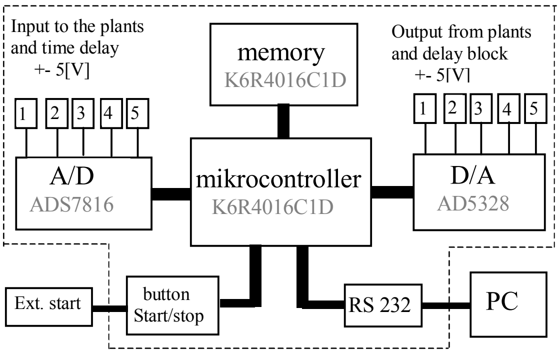

Fig. 9. Electronic scheme of plants model.

4. REAL TIME SIMULATION EXPERIMENTS DESCRIPTION

Real time simulation experiments are also called hardware in loop simulation (HIL) or hybrid simulation, because the controlled variable is scanned in real time by a A/D converter from hardware realized model of controlled plant described in previous section. If we can to identify parameters along with dead time in time delayed processes then it will be possible to control processes with variable dead time.

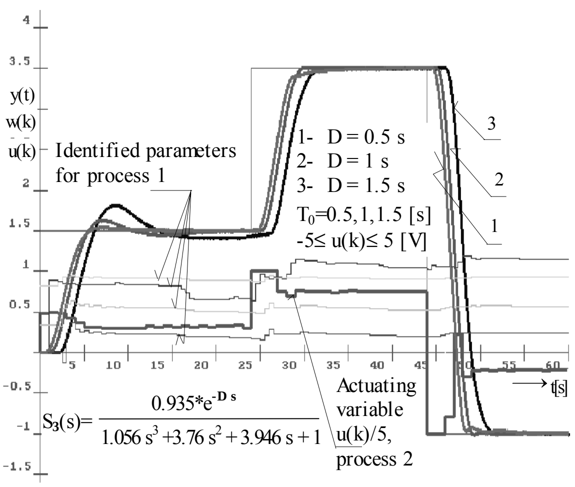

Fig. 10. Adaptive control of dead time processes

At present time author can identify up to three sampling intervals of dead time and also parameters of third order process. Then adaptive control for this kind of time-delayed processes was verified. Illustration of this is given in Fig. 10. There is HIL simulation of a closed loop system with time-delayed processes where the adaptive controller is designed by an eDBd method.

Illustration of the classical dead time process control is given in Fig. 11, which is a PC simulation of a closed loop system with time-delayed processes where the controller is designed by an eDBd method and processes are the test batch from (Astrom, *and* Häglund 1995) as typical industrial processes. In this case we find that the sampling interval T_0= 0.25, 0.5, and 1 [s] is suitable for dead time but too small for process time constant (T= 5 s), but against expectation set point responses are with quality behaviour only a big change of the set point produces a small overshoot. The reason for this is that the controller output is limited from both maximum and minimum limitation, but responses are with good time behaviour. New modification of state space eDBd algorithm could solve this problem.

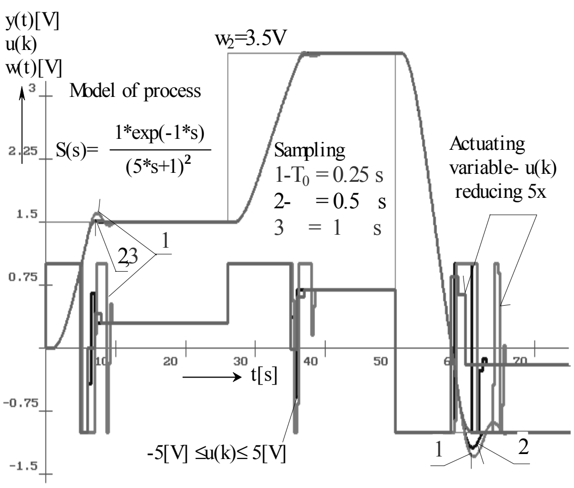

Fig. 11. Control of classical time delayed process.

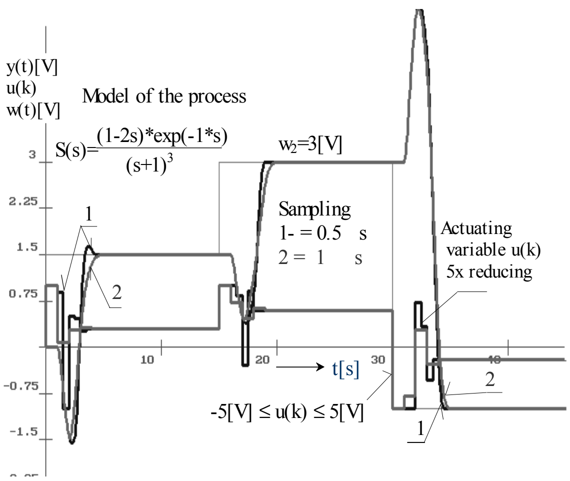

Fig. 12. Control of process with non-minimal phase and time delay by eDBd algorithm.

On Fig. 12, there is a PC simulation of a closed loop system with non-minimal and time-delayed processes where the controller is designed by an eDBd method. Denominator of transfer function of the process is the test batch from (Astrom *and* Häglund 1995) as typical industrial processes and nominator is added as typical non-minimal phase. In this case we find that the sampling interval T_0 = 0.5, and 1 [s] is suitable for dead time and also for process time constant (T= 1 s), and as is expected set point responses are with good quality behaviour. In part 2.1 of article (transfer function (13), (14)) was commented the possibility and a reason why eDB algorithm is able to control non-minimal phase processes.

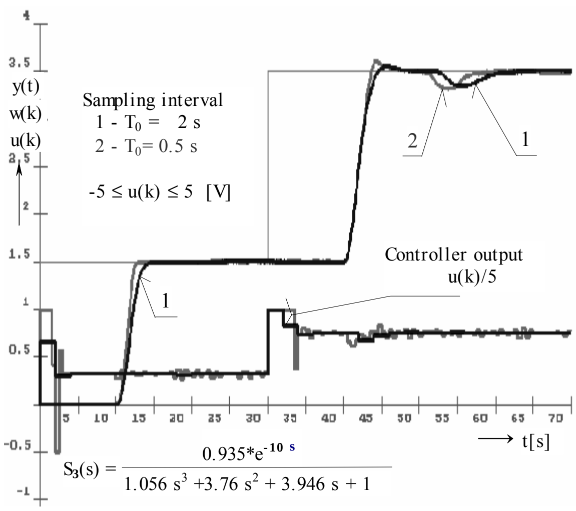

Fig. 13. Control of process with dominating dead time.

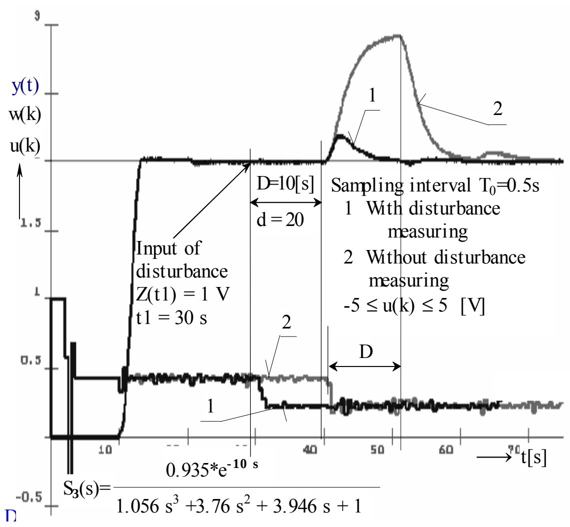

Fig. 14. Disturbance compensation by eDBd control.

Real time HIL simulation of a closed loop system with processes whose dynamics is dominated by dead time is shown on Fig. 13 and Fig. 14. Times constant of the process and dead time are the same. Fig. 14 shows that the load-disturbance response is excellent in case the disturbance is measured. Without disturbance being measured, dead time produces load-disturbance responses with big overshoots and settling time. It is needed to verify presented eDBd algorithm in detail and revise some problems such as overshoot on small sampling interval and its compensation by state space form of eDBd algorithm.

5. CONCLUSIONS AND OUTLOOK

Based on present experience, the laboratory verification of eDBd control algorithm for time-delayed processes on described hardware in loop simulation provides good results. In this paper descriptions of adaptive eDB and eDBd algorithms have been given. Simulation experiments were realised in program environment ADAPTLAB, developed and realised by author. It is suitable for developing and verification of classical as well as adaptive control algorithms for SISO and MIMO control loops. The program can be used in two basic modes: simulation and measurement. The simulation mode works with continuous transfer function set by operator. In the measurement mode the output (input) from model of the plant described in Section 3 is measured with A/D (D/A) converter with sampling interval controlled by real time clock or interrupt from A/D converter. Verification of eDBd algorithm documented excellent quality of set point and load-disturbance responses in closed-loop system with time-delayed processes. However it is needed to verify eDBd algorithm in detail and revise the problem with overshoot by small sampling interval. To solve this problem, different compensation with state variables of eDBd algorithm is needed.

REFERENCES

Alexík, M. (2001). Usage of Real Time Hybrid Simulation for Verification of Control Algorithms In: *Proceedings of the 4th International Eurosim 2001 Congress "SHAPING FUTURE WITH SIMULATION"*, TU Delft, The Netherlands, 26-29 June 2001, paper. number. 052, Delft University of Technology.

Alexík, M. (2002)- Adaptive DB(n) Algorithm with Antiwindup. In: *Proceedings of "3rd International Carpathian Control Conference*", Ostrava - Beskydy, Czech Republic, May 27-30, 2002, pp.677-684.

Astrom, K. J. and T. Häglund (1995) *PID Controllers, 2nd Edition.* Instrument Society of Amerika, Research Triangle Park, North Carolina.

Schlegel, M.; P. Balda, and M. Štětina (2003). Robust PID autotuner –moments method. (In Czech). In: *Automatizace*, **4/2003**. pp.242-246,.

Vítečková, M. (1999). Digital and Analog Controller Tuning for Processes with Time Delay (In Czech). In: *Automatizace*, **2/1999**. pp 106-112.

Zentko P. and M. Alexík (1986). Drying Kiln Parameters Adaptive Control. In *International IFAC Symposium Lignoatomatica,* Bratislava, Slovakia, October 1986, pp. 107-110.

Žáková, K. (2003). *Design of Controllers for Time-Delayed Systems*. PhD. Work :FEI STU Bratislava, 02. 09., 2003.

This contribution is prepared under the projects VEGA 1/2058/05 "Adaptive and Learning Algorithms for Automatic Control".