ADAPTIVE CONTROL OF A PRESSURE TANK SYSTEM WITH SATURATION NONLINEARITIES

Z. Ma [⊗], **A. Jutan** [◊] **and V. Bajic** [*]

[⊗] *Canadian Space Agency,*
6767 route de l'Aeroport St. Hubert,
Quebec J3Y 8Y9 Canada
e-mail:zi_ma@hotmail.com

[◊] *Department of Chemical & Biochemical Engineering,*
The University of Western Ontario,
London, Ontario, N6A 5B8, Canada
e_mail:ajutan@julian.uwo.ca

[*] *Computational Immunology Group*
KRDL 21 Hemg Mui Keng Terrace
Singapore 119613
e-mail:bajicv@krdl.org.sg

Abstract

The paper presents the design of a new nonlinear real-time adaptive control algorithm which is suitable the systems with saturation nonlinearity at the input. The linear part of the system can be non-minimum phase and/or unstable. The control algorithm is of the self-tuning type. The algorithm utilizes an auxiliary control parameter that can be chosen on-line to keep the controller output in the linear zone of the saturation nonlinearity. Stability of the algorithm is discussed. Several simulation experiments are performed to demonstrate effectiveness of the algorithm. Finally, the algorithm is applied to the real time pressure tank process control system successfully.

Keywords: Adaptive control, saturation, nonlinear plant, pressure tank.

1.Introduction

Due to a rapid increase in the power of computing technology, adaptive control algorithms have become ever more important for practical implementation. Research in adaptive control for linear systems has produced important developments in the last twenty years (see [1][2][3] and [4]). However, in practice, systems are often not linear and often have actuators at the input, which impose a natural saturation limit. A typical example of an actuator in the process industry can be a valve such as in the pressure tank pressure control system used here. The static input-output characteristic of the actuator can be approximated by a saturation type nonlinearity. The presence of such a nonlinearity may lead to windup problems or loss of control, if the control signal enters the saturation domain of the actuator. These problems are common in many types of real time controllers, thus it is of an interest to develop an adaptive control scheme that can handle this situation. Significant progress has been made in this direction using adaptive control systems (see [5] and [7]). A minimum variance adaptive control of discrete-time system with a saturating actuator was presented in [7]. These authores developed an adaptive algorithm for their system using Goodwin's approach [6]. Unfortunately, their results cannot be applied to systems whose linear part is unstable and/or of nonminimum phase. In this paper we resolved this problem and develop a new adaptive control algorithm for time-discrete systems that have a saturation type nonlinearity. Our algorithm is based on the generalized minimum variance adaptive control scheme and is applicable to systems whose linear part can be nonminimum phase and/or unstable. The stability characteristics of the algorithm are discussed. The algorithm performance is illustrated by simulation experiments. Finally, it is applied to a real time pressure tank control system. The successful real time control results show that the control algorithm can maintain the control signal in the linear control zone, so that the adaptive control algorithm can be directly applied without the need for the any `ad hoc' work around that are often used in engineering practice.

2. Controller design

We consider a linear plant with a nonlinear saturation component at its input. The model of such a system

$$\begin{gathered} Ay^*{}_{k+d} = Bu^*{}_k ; \left|u^*{}_k\right| \le u_s , \\ Ay^*{}_{k+d} = Bu^*_s sign(u^*{}_k); \left|u^*{}_k\right| > u_s \\ k = 0,1,2,... \end{gathered} \tag{1}$$

where $y^*{}_k$ and $u^*{}_k$ represent the values of the variables y^* and u^* at the moment k respectively; d is a positive integer representing the time delay in the system; A and B are polynomials of the backward shift operator z^{-1}. These polynomials are given by

$$A = 1 + a_1 z^{-1} +,..., a_{n_a} z^{-n_a}; \quad B = b_0 + b_1 z^{-1} +,..., b_{n_b} z^{-n_b} \tag{2}$$

To simplify notation we will write P for a polynomial $P(z^{-1})$, except when shorter notation is inadequate. In (1), $u_s > 0$ is the saturation value of the control output determined by the saturation nonlinearity at the plant input. When $\left|u^*{}_k\right| > u_s$, the control signal will be $u^*{}_k = u_s sign(u_k)$. In order to ensure that the control output $\left|u^*{}_k\right| \le u_s$, an auxiliary control mechanism characterized by the adjustable parameter π is introduced into the system. The complete control system is depicted in Fig. 1.

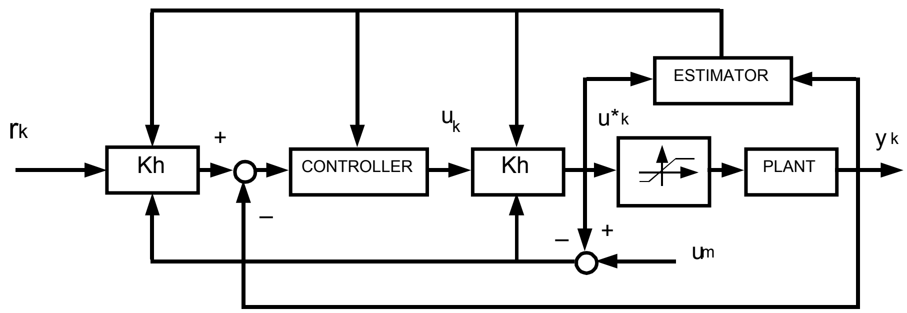


Fig. 1 Control system structure

In Fig.1, u_m is upper bound of the control signal, u_m may be chosen as $u_m < |u_s|$, and $k_h = 1/\pi$. The estimator determines the values of the parameters of the linear part of the plant, as well as the value of the adjustable parameter π .

To ensure that the system operates in the linear zone of the actuator, the parameter π will be selected so that

$$|u_k / \pi| \le u_s \tag{2}$$

If we define

$$u_k / \pi = u^*{}_k ; \quad y_{k+d} / \pi = y^*{}_{k+d} \tag{3}$$

then (1) implies (for $\pi \ne 0$) that

$$Ay_{k+d} / \pi = Bu_k / \pi \tag{4}$$

The adaptive control system in Fig.1 will be designed based on (1)-(4). This model includes the auxiliary control parameter π . If π is obtained to satisfy (2) this will make $|u^*{}_{k\,s}| \le u_s$ and eliminate the saturation component.

We use the following generalized error *e* in the design process

$$e_{k+d} = Py^*{}_{k+d} - Sr_k / \pi + Qu^*{}_k$$

and define the performance cost function of the control system by

$$J = e^2{}_{k+d} = \left[Py^*{}_{k+d} - Sr_k / \pi + Qu^*{}_k\right]^2 \tag{5}$$

Here *P, Q* and *S* are, in general, polynomials introduced to improve the system dynamics and steady-state characteristics of the overall control system. By selecting *P* and *Q* on-line we can make the closed-loop system stable, while selection of S will control system behavior when $k \to \infty$

We define an auxiliary output $\Phi_{k+d} = Py^*{}_{k+d}$, which we obtain by a predictor of the form

$$\Phi^*{}_{(k+d)/k} = Gy^*{}_k + EBu^*{}_k \tag{6}$$

where G and E are polynomials of z^{-1} with orders $n_e = d-1$ and $n_g = n_a - 1$, respectively, i.e.

$$E = e_0 + e_1 z^{-1} +,...,+ e_{n_e} z^{-n_e}; \quad G = g_0 + g_1 z^{-1} +,..., g_{n_g} z^{-n_g}$$

and which satisfy the following Diophantine equation

$$P = EA + z^{-d} G \tag{7}$$

Now we can state the following

Theorem1: *The control law minimizing (5) is described by*

$$\Phi^*{}_{(k+d)/k} = Sr_k / \pi - Qu^*{}_k \tag{8}$$

and it ensures that $y_k \to r_k$ *as* $k \to \infty$

The closed-loop system equations are given by

$$(BP + QA) y^*{}_{k+d} = BSr_k / \pi \tag{9}$$

$$(BP + QA) u^*{}_k = ASr_k / \pi \tag{10}$$

Remark 1: *The reason for having two equations that characterize the system in the closed loop is that (9) determines the dynamics of system from the 'reference input - plant output' point of view, while (10) determines its behavior from the viewpoint of the reference input - plant input' relation. Since both have the same characteristic polynomial BP+QA, then if this polynomial is made stable, we will achieve stable process output, as well plant input. In this manner stable behavior of both the plant output and the plant input is ensured.*

Proof: First we will prove that the closed-loop system obeys (9) and (10). From the predictor equation (6) and the control law (8), system input $u^*{}_k$ can be written as

$$u^*{}_k = \frac{Sr_k/\pi - Gy^*{}_k}{EB+Q} \tag{11}$$

Substituting this $u^*{}_k$ in (4) and using Diophantine equation (7), the closed-loop equation (9) can be obtained as follows

$$Ay^*{}_{k+d} = Bu^*{}_k = B\frac{Sr_k/\pi - Gy^*{}_k}{EB+Q}$$

$$(EB+Q)Ay^*{}_{k+d} = BSr_k/\pi - BGy^*{}_k$$

$$(EB+Q)Ay^*{}_{k+d} = BSr_k/\pi - Bz^{-d}Gy^*{}_{k+d}$$

$$[B(EA+z^{-d}G)+QA]y^*{}_{k+d} = BSr_k/\pi$$

$$(BP+QA)y^*{}_{k+d} = BSr_k/\pi$$

Also, from (11) one gets

$$y^*{}_k = \frac{Sr_k/\pi - (Q+EB)u^*{}_k}{G} \tag{12}$$

Substituting this $y^*{}_k$ in (3) and (4), and using Diophantine equation (7), the closed-loop equation (10) can be obtained in the following way

$$Ay^*{}_{k+d} = Bu^*{}_k$$

$$Ay^*{}_k = z^{-d}Bu^*{}_k$$

$$A\frac{Sr_k/\pi - (BE+Q)u^*{}_k}{G} = z^{-d}Bu^*{}_k$$

$$ASr_k/\pi = B(EA+z^{-d}G)u^*{}_k + AQu^*{}_k$$

$$ASr_k/\pi = (BP+AQ)u^*{}_k$$

Now we need to prove that the control (8) minimizes (5). In equations (9) and (10) of the closed-loop system, orders of weight polynomials P and Q can be determined based on the orders of polynomials B and A as $n_p = n_a - 1$ and $n_q = n_b - 1$, respectively, so that

$$P = p_0 + p_1 z^{-1} +,...,+ p_{n_p} z^{-n_p} \;;\quad Q = q_0 + q_1 z^{-1} +,..., q_{n_q} z^{-n_q};$$

Eqn. (11) implies

$$(EB+Q)u^*{}_k = Sr_k/\pi - Gy^*{}_k$$

Since $y^*{}_{k+d} = z^{-d}y^*{}_k$, then from (9) one obtains

$$y^*{}_k = \frac{z^{-d} BSr_k / \pi}{BP + QA} \tag{13}$$

To obtain a stable system, we select a stable polynomial T in advance, so that the polynomials P and Q are determined from the following Diophantine equation

$$BP + QA = T \tag{14}$$

From (13) it follows that if the following closed-loop transfer function is equal to one, i.e. if

$$\frac{Y(z^{-1})}{R(z^{-1})} = \frac{BSz^{-d}}{BP + QA} = 1$$

where Y and R are the z-transforms of y and r, we get $y^*{}_k = r_k / \pi$ and $y_k = r_k$ for all *k=0,1,2,...* Therefore, we need to select S to satisfy the following relation

$$S \equiv \frac{T}{Bz^{-d}} \tag{15}$$

However, in general, (15) cannot be satisfied for all allowed values of z but it can be satisfied for $z=1$. Note that the case $z=1$ corresponds to the behavior of system when $k \to \infty$. Thus S is selected as

$$S = \frac{T(1)}{B(1)} \tag{16}$$

Hence, for $k \to \infty$, the control mechanism given by (8) implies that $y_k = r_k$ is achieved. This also makes $J=0$ it is minimum value.

From (10), we have

$$u^*{}_k = \frac{ASr_k / \pi}{BP + QA} \tag{17}$$

Q.E.D.

Remark 2: *For practical purposes, in order to keep the input* $u^*{}_k$ *within suitable limits, we select an allowed (prespecified) upper bound for* $\left|u^*{}_k\right|$ *as* u_m *where*

$$0 < u_m \le u_s$$

Then the parameter π *can be determined on-line at the moment k by*

$$\pi_k > |u_k / u_m|; \quad \pi_k > 0,$$

where π_k is the value of π at the moment *k*. By the above rule, π is chosen sufficiently large so that $\left|u^*{}_k\right| = \left|\frac{u_k}{\pi_k}\right| < \left|\frac{u_k}{u_k / u_m}\right| = u_m$. Therefore, when $k \to \infty$ (i.e. $z=1$) one has from (14), (16) and (17) that

$$u^*{}_k = \frac{A(1)}{B(1)} r_k / \pi_k .$$

Here, since

$$\left|u^*{}_k\right| < u_m$$

then π_k can be chosen as

$$\pi_k > \left|\frac{A(1) r_k}{B(1) u_m}\right| \tag{18}$$

where $B(1) \neq 0$ is assumed.

3. Adaptive control algorithm

The system (1) may be written in vector notation as

$$y_k = \Phi_k \theta^T \tag{19}$$

where

$$\Phi_k = [-y^*{}_{k-1}, \ldots, -y^*{}_{k-n_a} \quad u^*{}_{k-1}, \ldots, u^*{}_{k-n_b}] \tag{20}$$

$$\theta = [a_1, \ldots, a_{n_a} \ b_0, \ldots, b_{n_b}] \tag{21}$$

The following least squares (RLS) algorithm can be used to identify the parameters θ

$$\hat{\theta}_k = \hat{\theta}_{k-1} + L_k \varepsilon_k \tag{22}$$

where

$$\hat{\theta} = [\hat{a}_{1k}, \ldots, \hat{a}_{n_a k} \ \hat{b}_{0k}, \ldots, \hat{b}_{n_b k}] \tag{23}$$

$$L_k = P_k \Phi_k{}^T \tag{24}$$

$$P_k = [P_{k-1} - \frac{P_{k-1}\Phi^T{}_k \Phi_k P_{k-1}}{\lambda_k + \Phi_k P_{k-1} \Phi^T{}_k}] / \lambda_k \tag{25}$$

$$\lambda_k = \lambda_0 \lambda_{k-1} + (1 - \lambda_0) \tag{26}$$

$$\varepsilon_k = y^*{}_k - \Phi_k \hat{\theta}(k-1) \tag{27}$$

where λ_k is a variable forgetting factor [6].

Based on the parameter estimation $\hat{\theta}_k$, the adaptive control algorithm is defined by the following steps

Step 1. According to the saturation value u_s of the system, determine the upper bound of controller output u_m.

Step 2. Select a stable polynomial T to be used in connection with (14).

Step 3. Establish the initial value π_0 of the auxiliary control parameter π and initial vectors values of the parameters Φ_0 and θ_0.

Step 4. Using RLS algorithm from (22) to (27), calculate the updated system parameters $\hat{A}_k = \hat{a}_{1k}, ..., \hat{a}_{n_a k}, \hat{B}_k = \hat{b}_{0k}, ..., \hat{b}_{n_b k}$

Step 5. Based on Diophantine equation (14), P_k and Q_k are calculated from

$$\hat{B}_k P_k + Q_k \hat{A}_k = T \tag{28}$$

Step 6. Using A_k and P_k, the values of E_k and G_k are calculated based on Diophantine equation (7)

$$P_k = E_k \hat{A}_k + z^{-d} G_k \tag{29}$$

Step 7. Based on (16), the value S_k is calculated by

$$S_k = \frac{T(1)}{B(1)} \tag{30}$$

Step 8. The control parameter π is determined by (18).

Step 9. Based on (11), the controller output is calculated by

$$u^*_k = \frac{S_k r_k / \pi - G_k y^*_k - D_k u^*_k}{e_{k0}\hat{b}_{k0} + q_{k0}} \tag{31}$$

where

$$D_k = \hat{B}_k E_k + Q_k - e_{k0}\hat{b}_{k0} - q_{k0} \tag{32}$$

The above steps, except Steps 1, 2 and 3, are repeated at each step *k*.

4.Stability analysis of the algorithm

The system stability analyses is based on the following assumptions

Assumption A: The system structure is known, i.e. the order n_a and n_b are known.
Assumption B: Polynomial *A* and *B* are coprime.

The well known, the recursive least squares algorithm has the following properties,

$$\lim_{k\to\infty}\left\|\hat{\theta}_k\right\| \le M < \infty \tag{33}$$

$$\lim_{k\to\infty}\sum_{k=1}^{N}\left\|\hat{\theta}_k - \hat{\theta}_{k-1}\right\|^2 < \infty; 0 < l < \infty \tag{34}$$

$$\lim_{k\to\infty}\left\|\hat{\theta}_k - \hat{\theta}_{k-l}\right\| = 0 \tag{35}$$

The following lemma concerning input-output dynamics is developed.

Lemma 1: *When the control algorithm is applied to the plant (1), the input-output dynamics of the system is given by*

$$\begin{bmatrix} B_k P_k + A_k Q_k & 0 \\ 0 & B_k P_k + A_k Q_k \end{bmatrix}\begin{bmatrix} y^*_{k+d} \\ u^*_k \end{bmatrix} = \begin{bmatrix} (B_l.S_k - B_k S_k) + B_k S_k \\ (A_l.S_t - A_k S_k) + A_k S_k \end{bmatrix} r_k / \pi +$$
$$\begin{bmatrix} B_t.(E_k A_k - E_k.A_l) + (B_k P_k - B_l.P_k) & B_l.(E_k B_k - E_k.B_l) + (B_k Q_k - B_l.Q_k) \\ A_l.(E_k A_k - E_k.A_l) + (A_k P_k - A_l.P_k) & A_l.(E_k B_k - E_k.B_l) + (A_k Q_k - A_l.Q_k) \end{bmatrix}\begin{bmatrix} y^*_{k+d} \\ u^*_k \end{bmatrix} +$$
$$\begin{bmatrix} B_l.E_k(A_l - A_k) & B_l.E_k(B_k - B_l) \\ A_l.E_k(A_l - A_k) & A_l.E_k(B_k - B_t) \end{bmatrix}\begin{bmatrix} y^*_{k+d} \\ u^*_k \end{bmatrix} \tag{36}$$

where A_k and A_l denote the estimate values of parameter A at the k and k+l (l is a finite constant) moments respectively.

$$A_k = A(k, z^{-1}), A_l(k+l, z^{-1}) \tag{37}$$

$$A_k.B_l = A(k, z^{-1})B(k+l, z^{-1}) \neq B_l.A_k \tag{38}$$

$$A_k B_k = A(k, z^{-1})B(k, z^{-1}) = B_k A_k \tag{39}$$

Proof: From (4), we have

$$A_l y^*_{k+d} = B_l u^*_k + (A_l - A_k) y^*_{k+d} + (B_k - B_l) u^*_k \tag{40}$$

eqn. (40) can also be written

$$A_l y^*_{k+d} = B_l u^*_k + \nabla; \nabla = (A_l - A_k) y^*_{k+d} + (B_k - B_l) u^*_k \tag{41}$$

Multiplying (41) by E_k, one has

$$E_k.A_l y^*_{k+d} = E_k.B_l u^*_k + E_k \nabla \tag{42}$$

from engs. (29), (38) and (39), the following equation can be obtained

$$P_k y^*_{k+d} - G_k y^*_k - E_k B_k u^*_k = (E_k.B_l - E_k B_k) u^*_k + (E_k A_k - E_k.A_l) y^*_{k+d} + E_k \nabla \tag{43}$$

Using eqns. (6), (8) and (43), we have

$$P_k y^*_{k+d} - S_k r_k / \pi - Q_k u^*_k = (E_k.B_l - E_k B_k) u^*_k + (E_k A_k - E_k.A_l) y^*_{k+d} + E_k \nabla \tag{44}$$

Multiplying eqn. (44) by B_l, gives us

$$\begin{aligned} &B_l.P_k y^*_{k+d} - B_l.S_k r_k / \pi - B_l.Q_k u^*_k \\ &= B_l.(E_k.B_l - E_k B_k) u^*_k + B_l.(E_k A_k - E_k.A_l) y^*_{k+d} + B_l.E_k \nabla \end{aligned} \tag{45}$$

using the system open-loop equation (4) and eqns. (38), (39), we obtain

$$\begin{aligned} (B_k P_k + A_k Q_k) y^*_{k+d} = {} & [(B_l.S_k - B_k S_k) + B_k S] r_k / \pi + \\ & [B_t.(E_k A_k - E_k.A_l) + (B_k P_k - B_l.P_k)] y^*_{k+d} + \\ & [B_l.(E_k B_k - E_k.B_l) + (B_k Q_k - B_l.Q_k)] u^*_k + B_l.E_k \nabla \end{aligned} \tag{46}$$

Multiplying eqn. (44) by A_l, follows can be obtained

$$\begin{aligned} & A_l.P_k y^*{}_{k+d} - A_l.S_k r_k / \pi - A_l.Q_k u^*{}_k \\ & = A_l.(E_k.B_l - E_k B_k) u^*{}_k + A_l.(E_k A_k - E_k.A_l) y^*{}_{k+d} + A_l.E_k \nabla \end{aligned} \tag{47}$$

using the system open-loop equation (4) and eqns. (38), (39), we obtain

$$\begin{aligned} (B_k P_k + A_k Q_k) u^*{}_k = & [(A_l.S_k - A_k S_k) + A_k S_k] r_k / \pi + \\ & [A_l.(E_k A_k - E_k.A_l) + (A_k P_k - A_l.P_k)] y^*{}_{k+d} + \\ & [A_l.(E_k B_k - E_k.B_l) + (A_k Q_k - A_l.Q_k)] u^*{}_k + A_l.E_k \nabla \end{aligned} \tag{48}$$

at this point, Lemma 1 has been proven.

The stability and convergence of the control algorithm are given by the following theorem

Theorem 2: The control algorithm has the following properties

$$\lim_{k\to\infty} |y_k| < \infty; \lim_{k\to\infty} |u_k| < \infty \tag{49}$$

$$\lim_{k\to\infty} |e(k+d)|^2 = 0 \tag{50}$$

Proof: From the properties of least squares algorithm (34) and (35), it is shown that the parentheses terms in (36) tend to zero as $k \to \infty$. According to the rule of determining π

$$\pi > \left| \frac{r_k \hat{A}_k(1)}{\hat{B}_k(1) u_m} \right| \neq 0$$

and from the boundedness of r_k the property of the least squares algorithm (33), it can be seen that

$$\lim_{k\to\infty} |\pi| < \infty \tag{51}$$

using eqn. (28), the boundedness of control signal $u^*{}_k$ and $y^*{}_k$ cab be obtained . From the boundedness of π, the boundedness of system output y_k and u_k cab be obtained. From the definition of generalized error e_{k+d} (see eqn. (5))

$$e_{k+d} = P_k y^*{}_{k+d} - S_k r_k / \pi_k + Q_k u^*{}_k \tag{52}$$

using eqn. (44), we have

$$|e_{k+d}|^2 = \left|(E_k.B_l - E_kB_k)u_k^* + (E_kA_k - E_k.A_l)y^*_{k+d} + E_k\nabla\right|^2 \tag{53}$$

From the convergence of the parameter estimation term (34) and (35), it can be shown that eqn. (50) is satisfied.

5. Simulation results

Simulation 1: The following unstable and nonminimum phase discrete model is considered

$$y_k = 1.1y_{k-1} + 0.2u_{k-1} + 0.5u_{k-2}$$

It is assumed that if the system control input $|u^*_k| > 10$, the system will enter the saturation zone (i.e. the saturation value is $u_s = 10$), the upper bound u_m is chosen as *7.* The initial data vector for the parameter identification algorithm is chosen as the zero vector, the system parameter initial values are chosen as a set random numbers in the rang $0 \to 1$, the ideal system closed-loop characteristic equation *T* is chosen as $1 - 0.2z^{-1}$ and the initial value of the auxiliary parameter chosen as $\pi_0 = 1$ initially. The reference input signal is a square wave with amplitude ± 10 up to 200 time steps, then it is a sinusoidal wave. Fig. 2-Fig.5 shows that the control result is very good. From Fig.3, with no auxiliary parameter π, the control signal u_k is much larger than saturation value. After the auxiliary parameter is used, the control signal is limited to within the linear zone.

Simulation 2: In order to demonstrate the algorithm effectiveness for the plants with large delay, the simulation model is chosen as the following high order discrete model with delay $d = 12$

$$y_k = -0.1y_{k-1} - 0.13y_{k-2} + 0.04y_{k-3} + 0.015y_{k-5} + 0.125y_{k-6} + 0.023y_{k-7} + 0.2u_{k-12} + 0.01u_{k-13}$$

The saturation value of the control signal is chosen as $, u_s = 15$ to ensure that the control signal is in linear zone, the upper bound u_m is chosen as *10.* Similar to Simulation 1, the initial data vector for the system parameter identification algorithm is chosen as the zero vector, the system parameter initial values are chosen as random numbers in range $0 \to 1$, the system closed-loop characteristic equation *T* is chosen as $1 - 0.1z^{-1}$, and the initial auxiliary parameter value is chosen as $\pi_0 = 1$. The reference input signal is a square wave with amplitude ± 10, Fig.6-Fig.9 shows the results. Again very good control is achieved even for this high order large delay system with saturation nonlinearities.

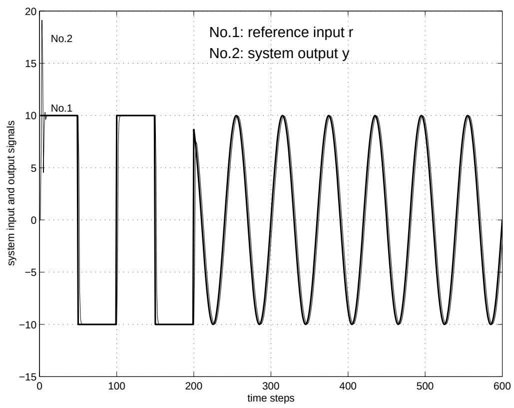


Fig.2 System tracking trajectories (simulation 1)

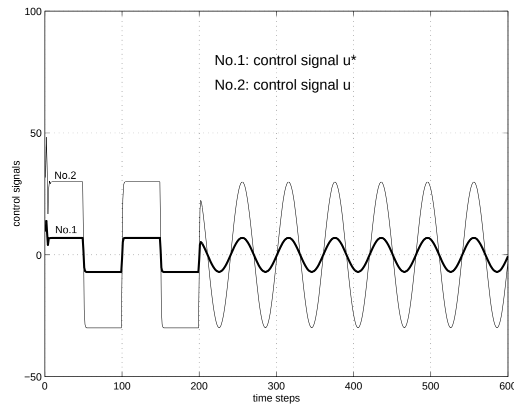


Fig. 3 Control signals (simulation 1)

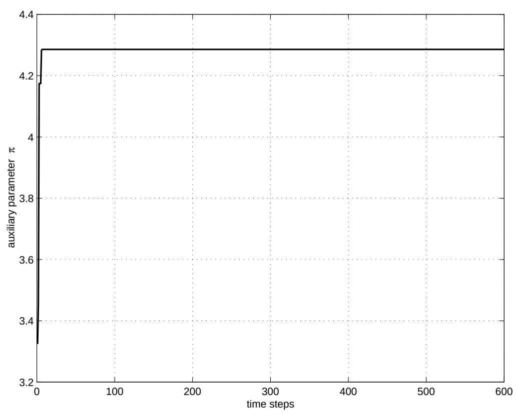


Fig.4 Auxiliary parameter π (simulation 1)

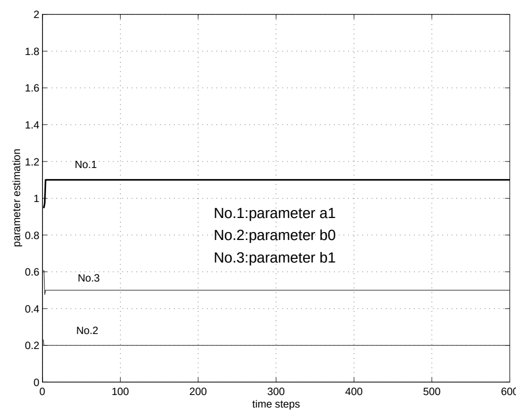


Fig.5 Identified parameter curves (simulation 1)

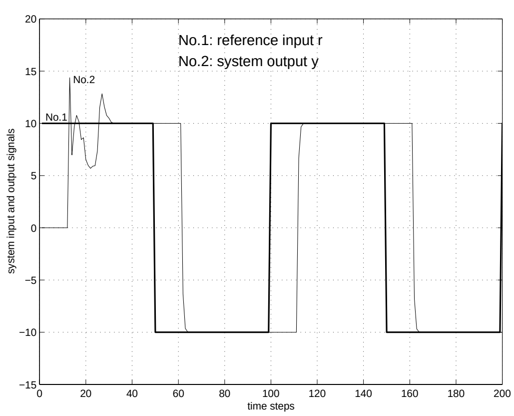


Fig.6 System tracking trajectories (simulation 2)

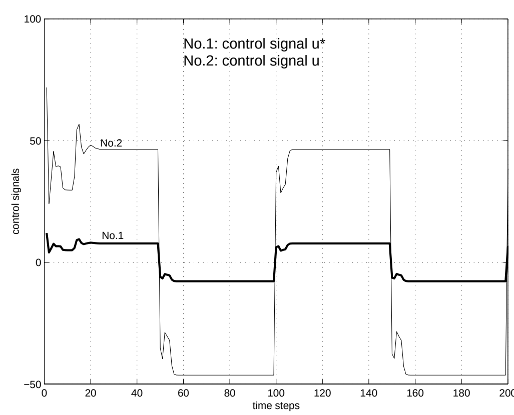
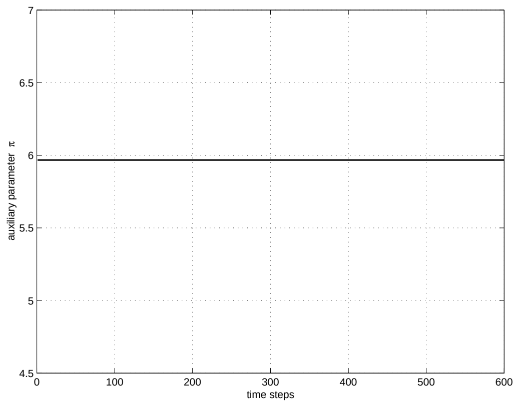


Fig. 7 Control signals (simulation2)



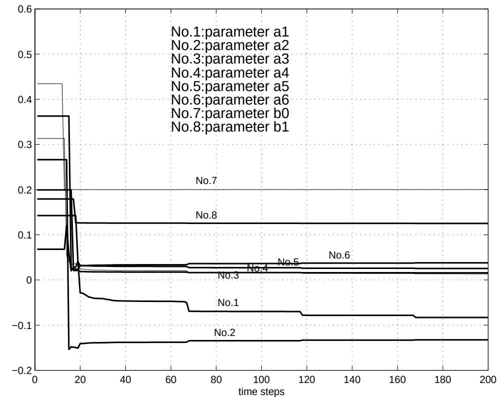


Fig.8 Auxiliary parameter π (simulation 2)

Fig.9 Identified parameter curves (simulation 2)

6. The Pressure Tank System

The real time system considered here is a Pressure Tank (Fig.10) through which air flows from a regulated source. Control valves are installed on both inlet and exit flow. The pressure in the vessel and the flow rate in exit flow are measured and transmitted to a computer. Data collection and system control are achieved by use of a microcomputer with an input-output (I/O) interface board. The controlled variables are pressure and flow

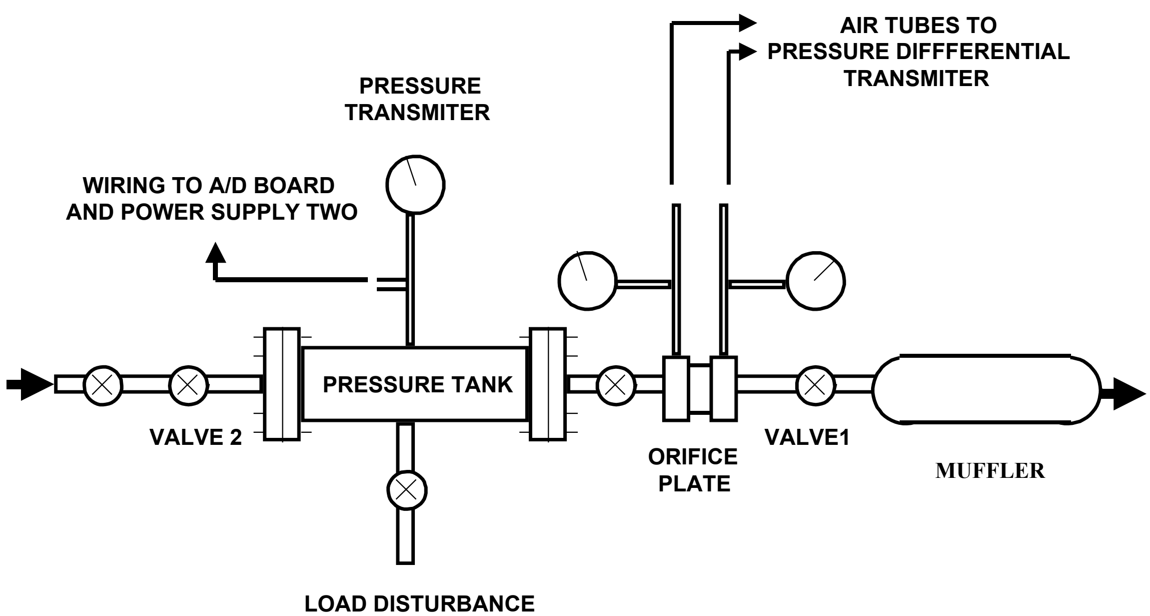


Fig. 10 Pressure Tank System

The real time control is also implemented by Matlab/Simulink. When implementing real time control, there are certain practical considerations, for example the controller output calculation is converted via a D/A board to produce an output signal in the range of 0-10 volts. If the control calculation results in a conversion outside this range, the output is clipped. Similarly for signal inputs, the range is converted by a A/D board which limits the range to 4-20 mA. Some `ad hoc' engineering methods such as deviation control are needed to take into account the clipped values as opposed to the calculated values, otherwise, instability may occur. In our algorithm, the saturation nonlinearity has been specifically considered. Therefore the signal in and out of the process should automatically be within the linear zone and clipping is longer required.

7. Experimental Runs

A number of experimental runs were conducted on the pressure tank to test out the efficiency of the control algorithm under real time conditions.

Run I. The controlled variable is flow rate. Since the saturation value of the system is $u_s = \pm 1$, as determined by the real time software, the upper bound $u_m = 0.6$ is chosen. Based on our knowledge, this process is a high order system with unknown time delay so that a self-tuning control algorithm based on a fourth order model with delay *d=1* is used. The initial values of system parameters vector and data vector are given as a random vector and zeros vector respectively. The value of auxiliary variable $\pi_0 = 10$ is chosen for startup. A stable pole placement equation is chosen as

$$T(z^{-1}) = 1 - 0.15z^{-1}$$

In this experiment, the sample period is 1 second. The set-point of the flow rate in $ft^3/\min$ is varied as a square wave.

Fig.11-Fig.13 shows the closed-loop system response for the run. From Fig.11-Fig.13, a good control response with minor offset is obtained. This offset is due to the process nonlinearity which varies with setpoint level, but is minor in this range. Fig.12 shows the output of the controller, which for the most of part, maintains an absolute value of less than 0.6 (i.e. within linear control zone). Fig. 13 shows the trajectory of auxiliary parameter π_k with rapid convergence rate.

From Fig.12, we notice two points *S1* and *S2* where the control action overshoots it's boundary magnitude of 0.6. This overshoot rapidly re-enters the linear zone and the output trajectory remains unaffected by this brief excursion. One reason for this occasional overshoot can be seen by examining the Diophantine equation (7). Here we see that the controller parameters $G_t(z^{-1})$ depend on the adaptive model parameters $\hat{A}_t(z^{-1})$ and $P_t(z^{-1})$ as well as the estimated time delay *d*. Generally the model time delay is unknown and may even appear to vary due to process nonlinearities. Here a fixed, estimated time dalay is used. This appears to be sufficient for all but brief instances during which the process output remains unaffected.

A second reason is that the rate of change of the adaptive auxiliary parameter π is dependant on the length of time over which the constraint is violated. Thus π will not respond to brief departures and is better suited to more serious extended periods of saturation.

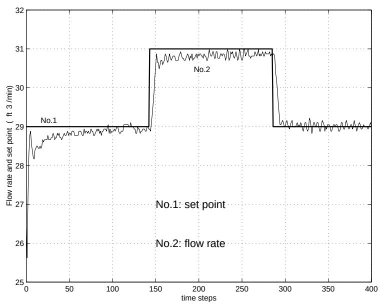


Fig.11 Flow rate tracking Trajectories (run 1)

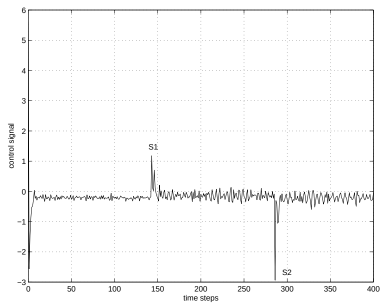


Fig. 12 Control signal (run 1)

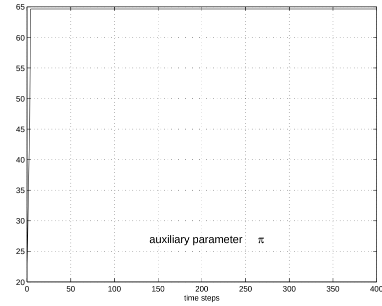


Fig.13 Auxiliary parameter curves (run 1)

Run 2. The controlled variable is now pressure, the linear model is based on a fourth order model with time delay *d=1*. The initial values of the system parameters vector and data vector are given as a random vector and zero vector respectively. An upper bound of $u_m = 0.6$ is also chosen. The initial value of auxiliary $\pi_0 = 10$ is chosen. The stable pole placement equation is chosen as

$$T(z^{-1}) = 1 - 0.1z^{-1}$$

In this experiment, the sample period is 0.2 second (smaller than the flow sample period since the pressure dynamics are faster). The set-point of the flow rate in ***psi*** is varied as a square wave.

Fig.14-Fig.16 shows the closed-loop system response for the run. From Fig.14-Fig.16, a good control response with minor offset is obtained. Fig.15 shows the most of output signal value of the controller is within linear control zone. The control signal overshoot appear in points *P1, P2, D1* and *D2*. The overshoot in point *P1* and *P2* are effected by the change of reference input signals, and the overshoot in point *D1* and *D2* are caused by unmeasurable disturbances. Since the choice of system time delay is a good enough approximation, the control signals return to the linear control zone quickly so that the system output trajectory is not affected. Fig. 16 shows the trajectory of the auxiliary parameter π_k.

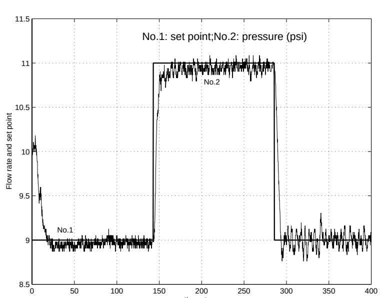


Fig.14 Flow rate tracking Trajectories (run 2)

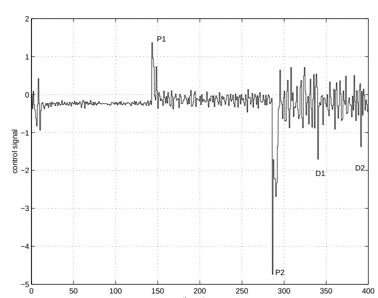


Fig. 15 Control signal (run 2)

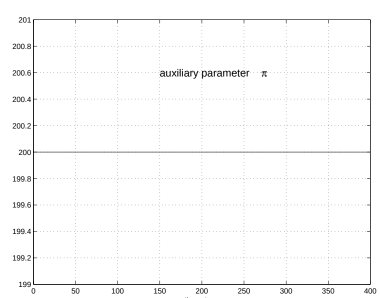


Fig.16 Auxiliary parameter curves (run 2)

8. Conclusion

This paper presents an algorithm for dealing with saturation signals in adaptive control systems. When the system signals are not saturated, the control system runs in normal adaptive mode. If a signal becomes saturated, a scaling parameter π is used to bring the signal back into the linear (unsaturated) zone. π is itself calculated adaptively, and tends to a constant when the process parameters converge to a static state. This auxiliary scaling parameter is chosen initially based on estimates of the plant parameters and the maximum value of the reference signal. An upper bound is also chosen for the control signal, which is less than the saturation limit. A suitable dead time must also be selected. The algorithm then operates to maintain the control signal within the linear (unsaturated) boundaries even in the initial stage when only rough process parameter values are available.

The new algorithm can be applied to unstable and/or nonminimum phase plants, including those of high order and large time delay. The system was tested via simulation and finally implemented on a real time pressure tank in SISO mode. The real time experiments demonstrate the success of the algorithm (except for a few transitory points) and is able to maintain the process operation within the linear control boundaries.

References

[1]. U. Borisson, Self-tuning regulator for a class of multivariable systems, *Automatica,* Vol. 15, pp. 209-215, 1979.

[2].G. C. Goodwin, P.J. Ramadge, and P.E. Caines, Discrete-time multivariable system, *IEEE Trans*. *Automat. Contr*. Vol. AC-25, pp. 449-456, 1980.

[3]. R.P. Singh and K. S. Narendra, Prior information in the design of multivariable adaptive controllers, *IEEE Trans. Automat. Contr*., Vol. AC-25, pp. 1241-1251, 1980.

[4].D. W. Clarke and P.J. Gawthrop, Self-tuning controller, *IEE Proc. Pt. D*., Vol. 122, pp. 929-935, 1975.

[5]Y. A. Daniel and G. F. Franklin, On the stability of adaptive pole-placement controllers with a saturating actuator, \QTR{it}{IEEE Trans. on Autom. Contr}., Vol. 35, No. 3, pp. 303-306, 1990.

[6]G. C. Goodwin and K. S. Sin, *Adaptive filtering prediction and control* (Information and System Science Series), Englewood Cliffs, NJ: Prentice-Hall, 1984.

[7] D. Y. Abramovitch, R. L. Kosut, and G.F. Franklin, Adaptive control with saturating inputs, *Proc. 25th Conf. Decision Contr*., 1986.