Part I. Anti-windup and bumpless-transfer protection

1. *Introduction to anti-windup*

1.1 System limitations

The process input is often limited in practice due to actuator constraints. The most common types of limitations are magnitude and rate limitations. A magnitude limitation and a rate limitation can respectively be described by the following two equations:

$$u^r = \begin{cases} U_{\mathbf{max}} & ;\ u > U_{\mathbf{max}} \\ u & ;\ U_{\mathbf{min}} \le u \le U_{\mathbf{max}} \\ U_{\mathbf{min}} & ;\ u < U_{\mathbf{min}} \end{cases} , \tag{1}$$

$$\frac{\partial u^r}{\partial t} = \begin{cases} v_{\max} & ;\ \dfrac{\partial u}{\partial t} > v_{\max} \\ \dfrac{\partial u}{\partial t} & ;\ v_{\min} \le \dfrac{\partial u}{\partial t} \le v_{\max} \\ v_{\min} & ;\ \dfrac{\partial u}{\partial t} < v_{\min} \end{cases} , \tag{2}$$

where u and u^r are also referred to as the controller output and the real process input respectively.

In the following text, the block diagram LIM represents the magnitude and/or rate limitations (see Fig. 1). Note that LIM can represent real limitations or an estimation of real limitations. In the last case, the actual measurement of the signal u^r is not required because the estimated limitation (LIM) becomes a part of a controller. More information about this subject can be found in Vrančić (1995a), and Vrančić and Peng (1995b, 1996).

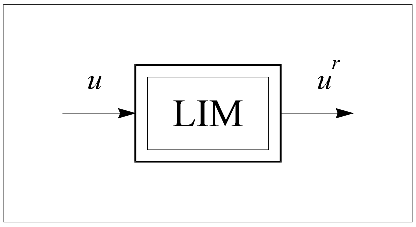

Fig. 1. The magnitude and/or rate limitations

1.2 Controllers

A *generalised PID controller* can be described by Fig. 2 and expression (3)

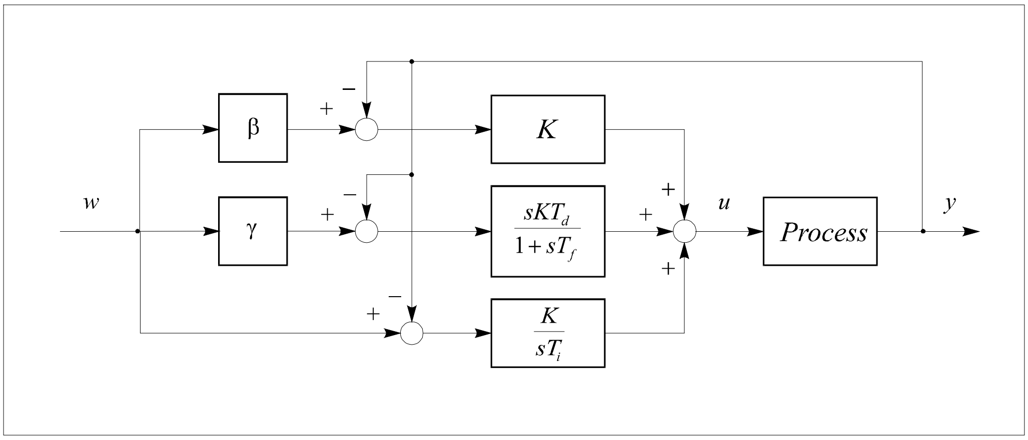

Fig. 2. The generalised PID controller

$$U(s) = K\left[\beta + \frac{1}{sT_i} + \gamma\frac{sT_d}{1+sT_f}\right]W(s) - K\left[1 + \frac{1}{sT_i} + \frac{sT_d}{1+sT_f}\right]Y(s) \quad , \tag{3}$$

where u is the controller output, w is the reference signal, y is the process output, and the capital letters U, W, and Y denote the Laplace transforms of u, w, and y, respectively.

The controller parameters are the proportional gain K, the integral time constant T_i, and the derivative time constant T_d with derivative filter time constant T_f.

Factors β and γ define the strength of the proportional (P) and the derivative (D) parts of the PID controller connected to the reference (set-point) w, respectively. Usually, values β and γ lie between 0 and 1. Note that the PID controllers most used in industry are a special case of the generalised PID controller with β=1 and γ=0. Such a type of generalised PID controller will be referred to as the *PID controller*. More details about factors β and γ and different structures of PID controllers can be found in Åström and Wittenmark (1984), Åström and Hägglund (1995a), Hang et al. (1991), Hang and Cao (1993), and Rundqwist (1990).

A more general controller in the *polynomial form* is given by Fig. 3 and expression (4).

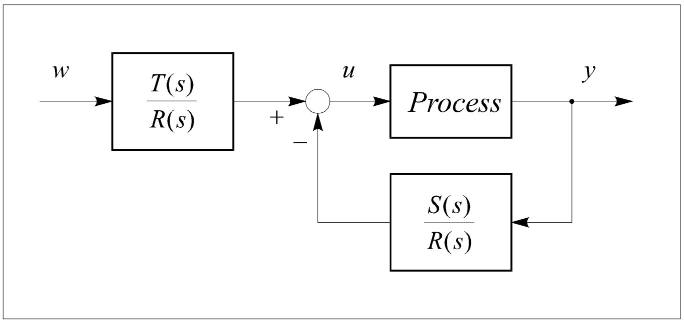

Fig. 3. A controller given in the polynomial form

$$U(s) = \frac{T(s)}{R(s)}W(s) - \frac{S(s)}{R(s)}Y(s) \quad . \tag{4}$$

The single-input single-output (SISO) *state-space controller* is described by Fig. 4 and expression (5)

$$\begin{aligned} \dot{x} &= Ax + bw - ey \\ u &= cx + dw - fy \end{aligned} \quad , \tag{5}$$

where A is the controller's *system* matrix, b and e are the controller's *input* vectors, c is the controller's *output* vector and d and f are scalar parameters. Generally, x is a vector and u, w, and y are scalars.

The *multi-input multi-output* (MIMO) state-space controllers are described in more detail in Chapter 8.

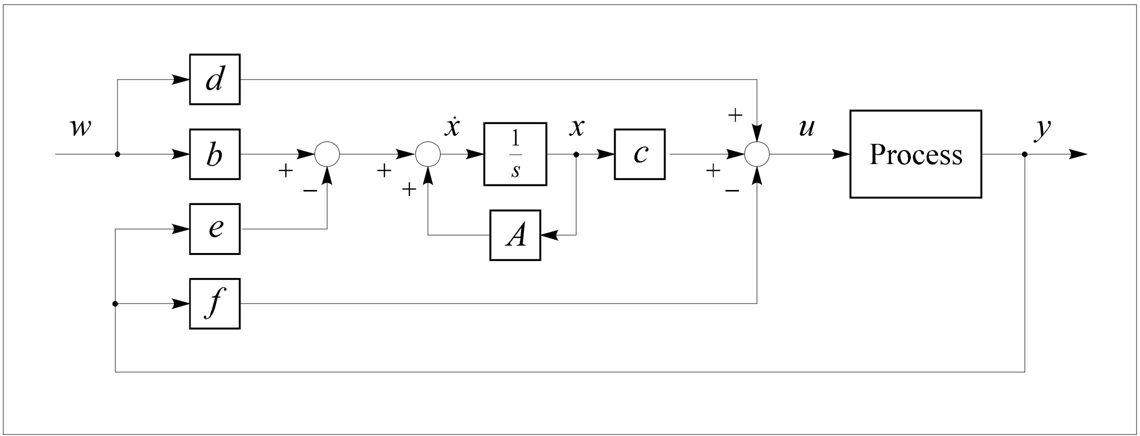

Fig. 4. State-space controller

1.3 Windup and anti-windup

Consider a closed-loop system containing a PID controller (β=1, γ=0) and a magnitude limitation LIM (Fig. 5). Suppose both the controller and process are in steady state. Assume a large positive step change in w that causes a jump in u, so that the actuator saturates at high limit if $K > 0$. Thus, u^r becomes smaller than u, and y is slower than in the unlimited case. Due to the slower y, e decreases slowly. The integral term increases much more than the one in the unlimited case, and it becomes large. When y approaches w, u still remains saturated or close to saturation due to the large integral term; u decreases after the error has been negative for a sufficiently long time. This leads to a large overshoot and a large settling time of the process output.

In order to illustrate the above phenomenon, we have made a simulation with process:

$$G(s) = \frac{1}{(1+8s)(1+4s)} \quad , \tag{6}$$

and controller:

$$K = 20,\ \ T_i = 30s,\ \ T_d = 0.95s,\ \ T_f = 0.095s\ . \tag{7}$$

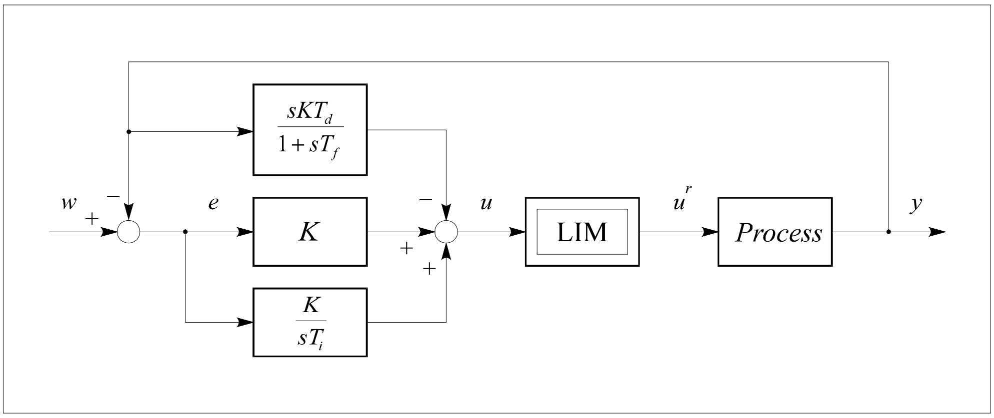

Fig. 5. Limited system in closed-loop with PID controller

The input limitations were U_{max}=2, U_{min}=0, v_{max}=2s^{-1} and v_{min}=-2s^{-1}. The closed-loop step responses for both limited (dashed lines) and unlimited cases (dotted lines) are shown in Figs. 6 and 7.

In Fig. 6, a large overshoot and a long process settling time can be seen in the limited case (see dashed line) as compared to the unlimited case. This closed-loop performance deterioration with respect to the unlimited case is called *windup*.

In fact, windup appears due to the fact that the integral term increases too greatly during saturation. Thus, during saturation the increase should be slowed down. This can be realised by an extra compensation that feeds back u-u^r to the integral term through a compensator with a transfer function $F(s)$ (see Fig. 8). Since this compensation aims at reducing the effect of windup, it is called *anti-windup* (AW).

From Fig. 6, it can be seen that by using an anti-windup compensator $F(s)$=1/K, the overshoot is *smaller* and the settling time is *much shorter* than in the absence of an anti-windup compensator (solid line).

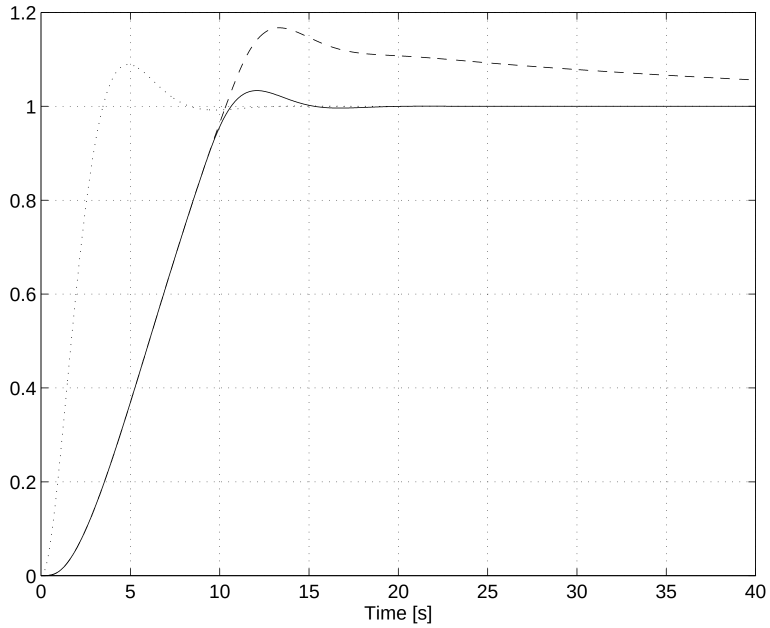

Fig. 6. Process output (y); __ Process input limited with AW, ... Process input unlimited, -- Process input limited without AW

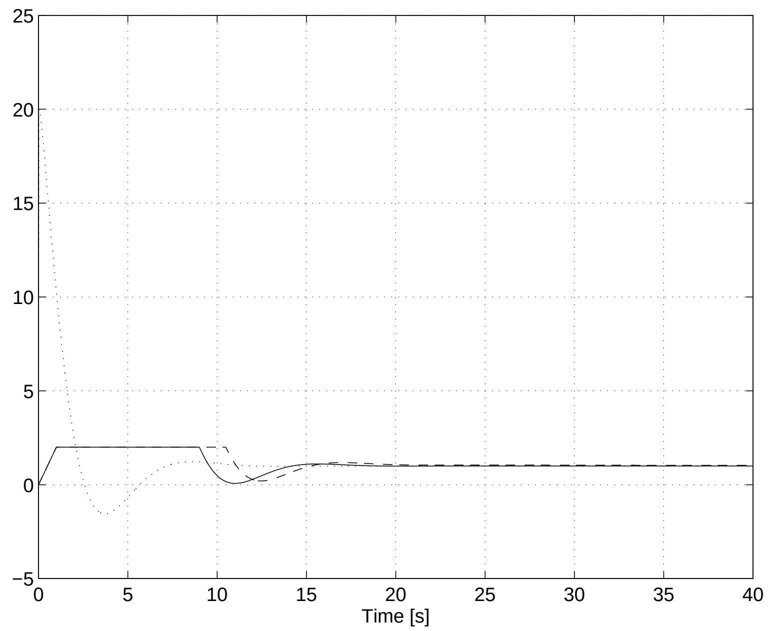

Fig. 7. Process input (u^r); __ Process input limited with AW, ... Process input unlimited, -- Process input limited without AW

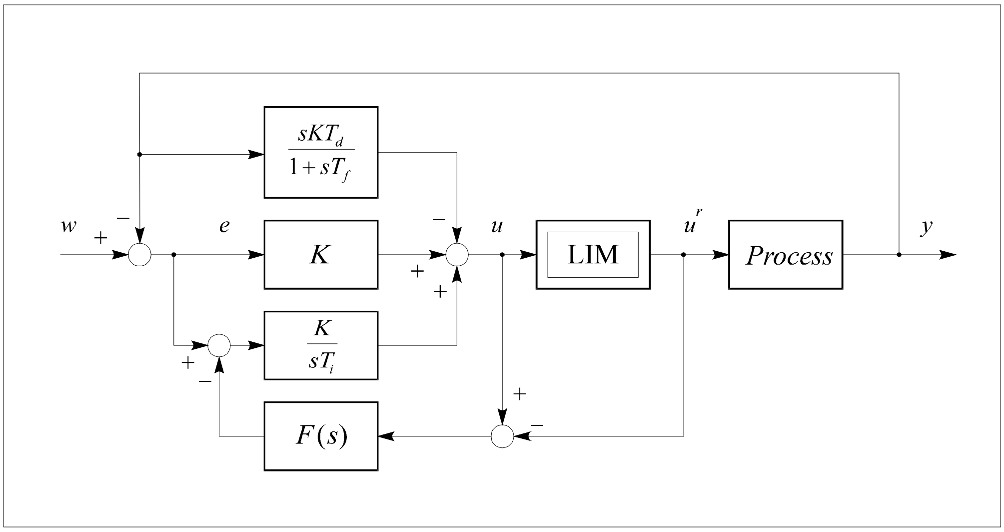

Fig. 8. The limited closed-loop system with AW

1.4 Bump transfer, bumpless transfer and conditioned transfer

Let us consider a control scheme with the capability of switching between manual and PID control mode (Fig. 9). Assume that the switch goes from automatic to manual control. If u^m is such that for some time $e>0$, then the integral term increases in an uncontrolled way to very high values and u becomes high and much greater than u^m. Now, assume that the switch goes back from manual to automatic control. At that moment, even if $e=0$, a large jump occurs at u^r due to the high values of the integral term. Moreover, u decreases only if $e<0$ for a sufficiently long time. This leads to a long settling time of the process output.

To illustrate the above phenomenon, a simulation using the same process (6) and controller (7) as in the previous example was performed. The reference signal was taken as 0. The process was manually controlled in the period from 0 to 40s. Then, its input was switched to the PID controller. The results of the simulation are shown in Figs. 10 and 11.

From Figs. 10 and 11 (dashed lines), it can be seen that, at the instant of switching, a large change occured at the process input, and this also caused a long settling time of the process. This mode switching with a change at the plant input is called *bump transfer*.

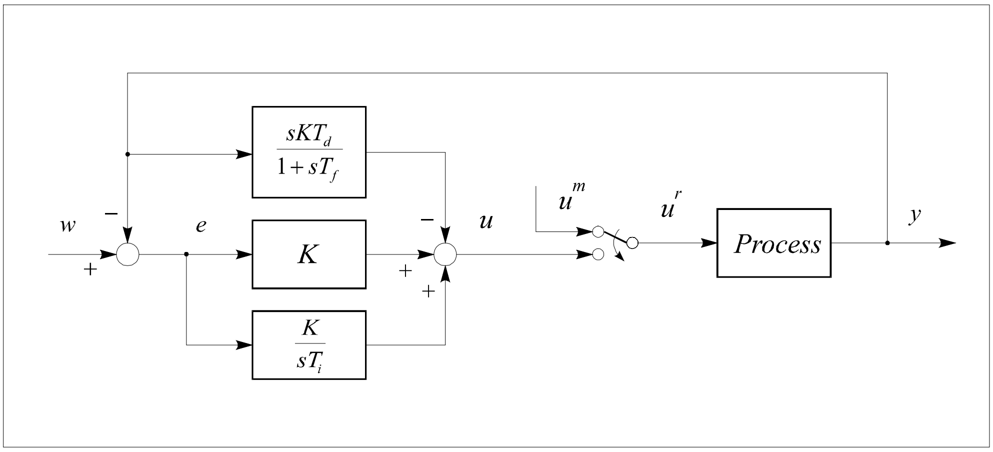

Fig. 9. Bump transfer from manual to automatic mode

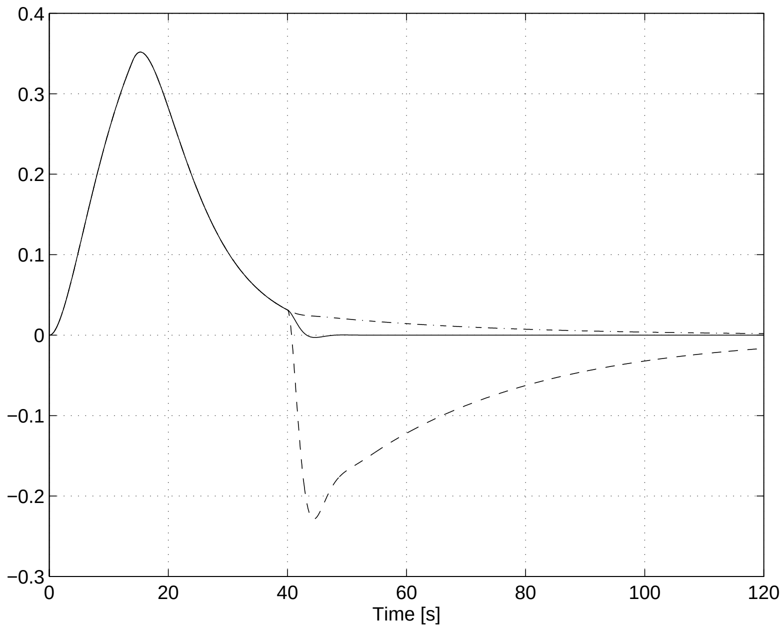

Fig. 10. Process output (y);
__ conditioned transfer, -- bump transfer, -.- bumpless transfer

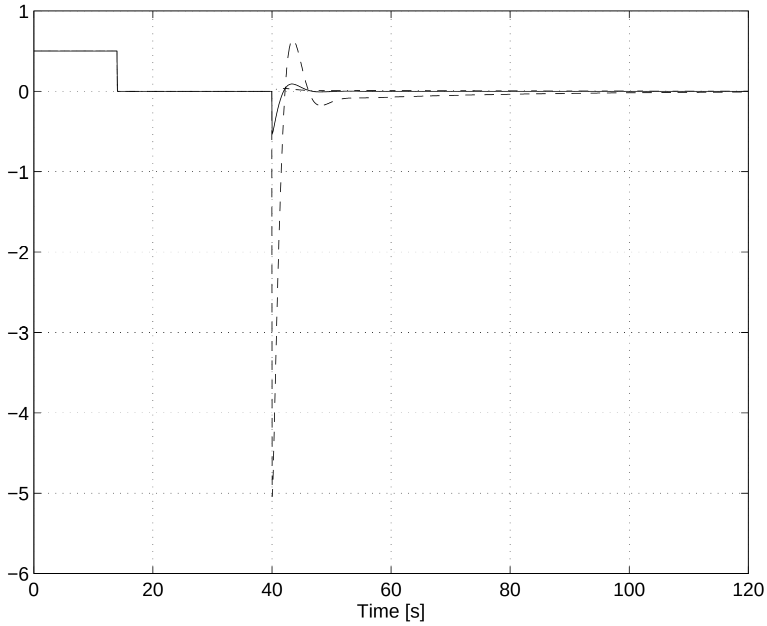

Fig. 11. Process input (u^r);__ conditioned transfer,
-- bump transfer, -.- bumpless transfer

The transfer which minimises the bump at the instant of switching is called *bumpless transfer* (BT). In the manual mode ($u^r=u^m$), the integral term should be kept under control so that u is as close as possible to u^m.

If the controller output u is adjusted so that after switching from manual to automatic control the plant input y tracks the reference w with the same dynamics as the closed-loop step response, then this mode switching is called *conditioned transfer* (Peng et al., 1996a,b; Vrančić and Peng, 1994a, 1995b; Vrančić, 1997). In other words, after switching good tracking performance will be assured when using a conditioned transfer. Note that the jump is usually small but not minimised in this case (Henrotte, 1989).

As with the anti-windup methods, both bumpless and conditioned transfers can also be realised by a compensation which feeds back u-u^r to the integral term, as shown in Fig. 12.

The results of using the BT and CT methods are shown in Figures 10 and 11 (dash-dotted and solid lines).

For bumpless and conditioned transfer, we choose $F(s)$=100 and $F(s)$=1/K respectively, the reason for this choice will be explained in the next section.

It can be seen that bumpless transfer produces no change at the process input (u^r) at the instant of switching from manual to automatic mode (Fig. 11), but the settling time of the closed-loop response is relatively long (Fig. 10).

On the other hand, the conditioned transfer yields a short settling time and produces some change in u^r when switching from manual to automatic mode.

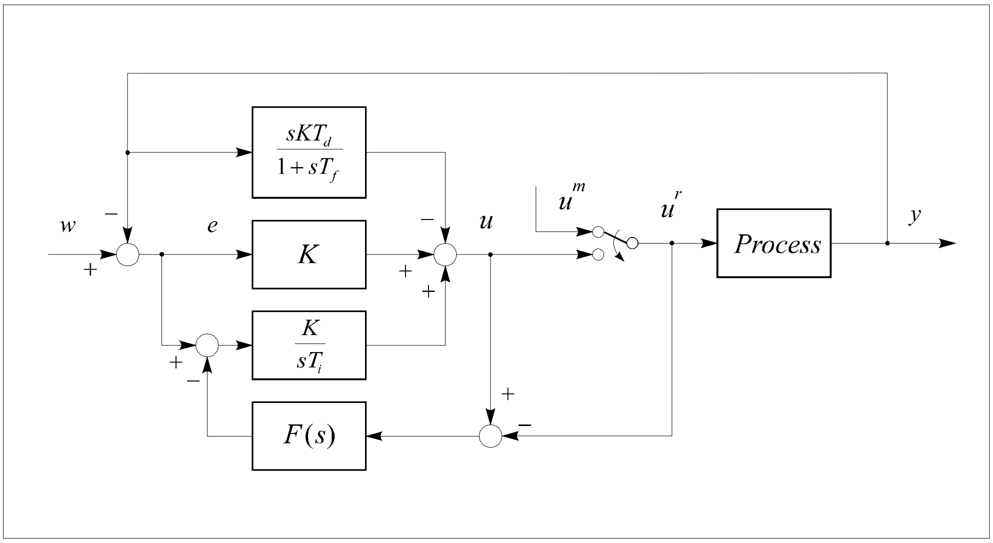

Fig. 12. Bumpless and conditioned transfer from manual to automatic mode

1.5 A short history

The windup phenomenon and bump transfer were first experienced in analog controllers. A common example is a batch process where, during the discharge, charging, and heat-up parts of the cycle, the controller output is not permitted to drive the control valve. Due to control error, the controller continues to integrate its output signal up and down. The same phenomenon was also observed in continuous processes when the pump goes down and the block valve is shut in the line, etc. The increased use of "override" or "selective" control systems (Khandheria and Luyben, 1976) increased the occurrence of integral windup and bump transfer problems since the manipulative variable could be controlled by one of the several different controllers at different times. The controllers that are not in use will windup if they have an integral action.

The problem could be prevented by switching the controller to manual mode when a windup condition occurrs and then switching it back to automatic mode at the right time. This requirs constant operator attention and might increase safety risks (Khandheria and Luyben, 1976).

The first cure proposed was the "batch controller", a pneumatic controller that had a switch which vented the reset bellows of the controller when the error signal exceeded set limits. This prevented saturation of the controller, but the transition from automatic to manual and back again was not bumpless.

A more satisfactory concept for analog controllers was the so-called "external reset feedback" (Shinskey, 1967). Such anti-windup protection is shown in Fig. 13.

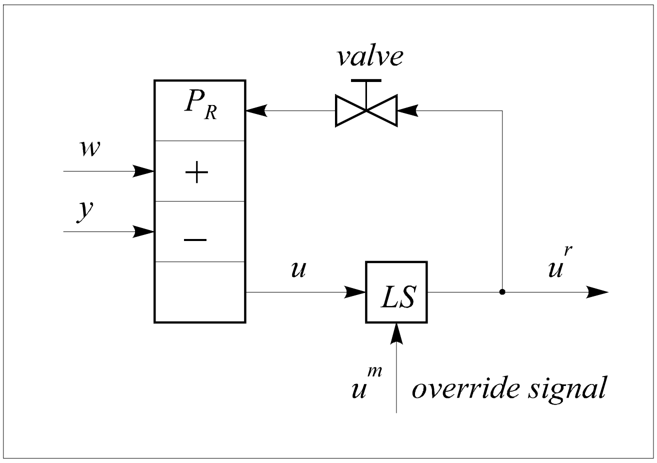

Fig. 13. Anti-windup protection for an analog controller, using the "external reset feedback"

If the signal to the actuator (u^r) is fed back to the reset chamber, the controller output can be expressed in the following way:

$$u = K(w - y) + \frac{1}{1 + sT_i} u \quad , \tag{8}$$

where K is the controller proportional gain, and T_i is the integral time constant, which can be adjusted by the valve (see Fig. 13).

When the controller output is overridden by another signal (u^m), there is no positive feedback and no integration. The transfer from "override" to the closed-loop control was claimed to be bumpless (Khandheria and Luyben, 1976).

The method should work quite well for "override" or "selective" control, but probably had several drawbacks in the classical anti-windup protection, because it was not so easy to build up the pneumatic limiter, which would be placed instead of the "override" selector LS in Fig. 13. As will be outlined later, the concept given by Fig. 13 and expression (8) is the same as obtained by using the conditioning technique on the PI controller (Hanus, 1980).

The so-called "tracking algorithm" is in fact a digital realisation of the analog "batch controller".

Shinskey (1967) proposed the modification of the "external reset feedback" algorithm, known as the "batch unit". In fact, the solution he proposed is similar to the concept which was later obtained by using the incremental algorithm in digital controllers (real bumpless transfer from manual to automatic mode).

However, the first controllers which included bumpless transfer were made by the Bailey Meter Co. in 1959 for solid-state controllers (using transistors and magnetic amplifiers) (Morris, 1990) and by Moore Products in 1965 for pneumatic controllers (Babb, 1990).

The windup problem became more severe when direct digital control (DDC) algorithms started to replace analog ones. The integral term in such algorithms can increase to much higher values than in analog controllers. Moreover, the designers of DDC algorithms often had limited experience in designing analog controllers and therefore they were not aware of windup problems.

They often simply limited the controller output signal, but this did not eliminate windup (Khandheria and Luyben, 1976).

The earliest attempt to overcome the windup problem in discrete time PID controllers appears to be the work done by Fertik and Ross (1967). The windup phenomenon was explained as the “improper storage or loss of control information due to constraints on the controller output”. They divided the controller windup phenomenon into three forms of windup: integral windup, proportional windup, and derivative windup. It was claimed that the design of a “no-windup” controller is not a straightforward task since the solution depends upon the controller configuration and application, because “compensating for one windup problem may lead to another forms of windup”.

They concluded that the use of velocity (incremental) algorithm in DDC algorithms avoids the integral windup problem, but this form of PID controller implementation suffers from proportional windup. In order to overcome this problem, they introduced the so-called “back calculation method”. The main principle was to calculate an “effective error” e_n that the actuator can follow, whenever the change of the PI controller action is higher than permitted by the actuator constraints:

$$e_n = \frac{\Delta U_{\max} + K e_{n-1}}{K\left(1 + \frac{T_S}{T_i}\right)} \ , \tag{9}$$

where ΔU_{max} is the maximum change of controller action (u) between two sampling intervals, K is the proportional gain, T_i is the integral time constant, T_S is the sampling time, and e_{n-1} is the previous “effective error” value. The “effective error” was then used in the control algorithm instead of the real one.

It was also stated that the test results for various other no-windup (anti-windup) schemes could be misleading since the results depended on the degree of velocity limitation, the dynamics of the process and the tuning criteria for the PI controller. However, the authors claimed that the back calculation technique gave a superior performance under a wide variety of test conditions for PI controllers. As will be shown later, the method was used as the basis for a more general anti-windup method called the conditioning technique (Hanus, 1980).

When using the PID controller, the authors observed a drawback of the back-calculation method (additionally modified for the D-part of the controller), because the back calculation resulted in a delayed derivative action which sometimes caused an overshoot of the controlled variable. The integral windup for a PID controller was prevented by ignoring the integral action when it was the same sign as the desired output.

After Fertik and Ross' paper, the next contribution was a paper written by Kramer and Jenkins (1971). They derived the algorithm which minimised the disruptive effect of certain types on nonlinearities so as to eliminate windup. Their approach depended on the type of saturation (hard or rate limit). In short, their approach was such that if the actuator is saturated, the controller operation is suspended, keeping the actuator in saturation until the error reaches a value that would give desaturated control. When the actuator desaturates, the linear controller is *optimally* reinitialised. In the case when a rate limit occurs, the algorithm accepts the process state at the time of the desaturation, and if hard limit is hit, the states are constrained so that the actuator desaturates smoothly ($u=U_{max}$ or U_{min}). In a few experiments they showed that their approach gave better results than when using the back-calculation method proposed by Fertik and Ross (1967).

The next contribution to the field of anti-windup was the paper of Khandheria and Luyben (1976), where they firstly described the four existing anti-windup methods (limiting, freeze and tracking algorithms). They were particularly interested in the anti-windup protection for the cascade system. Their proposition for cascade systems was a modified freeze algorithm which was basically a combination of the "batch controller" and the "external reset feedback". All the algorithms were tested on a methanol-water distillation column.

The next significant contribution to the field was made in 1980, when Hanus (1980) proposed an anti-windup technique for SISO controllers, namely the "conditioning technique". The technique is based on the idea of Fertik and Ross (1967), but is extended to the state-space and controllers given in the polynomial form. The technique is based on using the realisable reference (following Fertik and Ross, it would be called "effective reference") instead of the actual one for changing the *controller states*. The conditioning technique gives the same anti-windup protection for a PI controller that was obtained by using the "external reset feedback" in pneumatic controllers (Shinskey, 1967) or using the "back calculation method" proposed by Fertik and Ross (1967). It should be pointed out that the conditioning technique neither depends on the process model nor the type of the nonlinearity (constraints).

A general approach, whereby an observer is introduced into the controller structure in order to prevent windup, was proposed by Åström and Wittenmark (1984). It offers some tuneable parameters which can be used in order to achieve desired controller anti-windup performance.

In the same decade, a considerable number of papers were published, concerning anti-windup protection. Krikelis (1980) introduced the "intelligent integrator", as an anti-windup protection (see also Zupančič, 1996). It can be viewed as the modification of the "freeze" algorithms tested by Khandheria and Luyben (1976), where in the event the control error exceeds certain limits, the integration is merely braked and not completely stopped (frozen). Davidson (1989) proposed freezing the integral term in the controller when the set-point is changed, until the new set-point is neared in order to prevent reset windup. Some stability issues of constrained systems were carried out by Glattfelder and Schaufelberger (1983). Hanus et al. (1987) also extended the conditioning technique to multivariable systems and complex control structures.

Over the last decade, even more attention has been paid to the windup problem. Campo and Morari (1990) suggested the so-called "direction-preserving" method for MIMO systems in order to prevent the "deadlock situation" which can occur in non-diagonally dominant systems when one or more controller outputs are saturated. Rundqwist (1991) studied the performance of several SISO anti-windup methods for disturbances and showed that the optimal tuning of the anti-windup compensator for disturbances differs from one obtained by the reference change. Hanus and Peng (1991, 1992) extended the use of the conditioning technique also for controllers with nonminimum phase zeros and/or singular feed-through controller matrix *D*. Walgama et al. (1992) generalised the conditioning technique by proposing additional filtering of the set-point signal. Hansson et al. (1994) suggested a fuzzy approach in designing an anti-windup compensator, which showed improved results in tested nonlinear plants. Park and Choi (1995) proposed a dynamic anti-windup feedback instead of a static one that minimises a reasonable performance index. Rönnbäck (1996) proposed a change of controller parameters when the system is saturated, so as to achieve improved performance when the control signal is close to saturation in the steady-state. Wurmthaler and Hippe (1993, 1996) introduced the "phase criterion" which, by using the appropriate observer poles, guarantees the stable behaviour of the limited closed-loop system. They also distinguish between the "controller windup" and "plant windup".

Some good survey papers concerning anti-windup techniques, amongst others, have been given by Hanus (1989), Walgama and Sternby (1990), Morari (1993), Kothare et al. (1994), and Edwards and Postlethwaite (1996).

Several practical aspects of using anti-windup protection are given by Henrotte (1989), and Vrančić and Peng (1995b, 1996).

2. *Review and comparison of some existing anti-windup algorithms*

2.1 Anti-windup schemes

As in the previous chapter, an anti-windup compensator can be similarly added to several other types of controllers. Fig. 14 and expression (10) provide the anti-windup compensation for the *generalised PID controller*. It is clear that the protection is similar to that given by Fig. 8 (the PID controller). Note that the term “*PID controller*” will be used for a special case of the generalised PID controller when using parameters $\beta=1$, and $\gamma=0$.

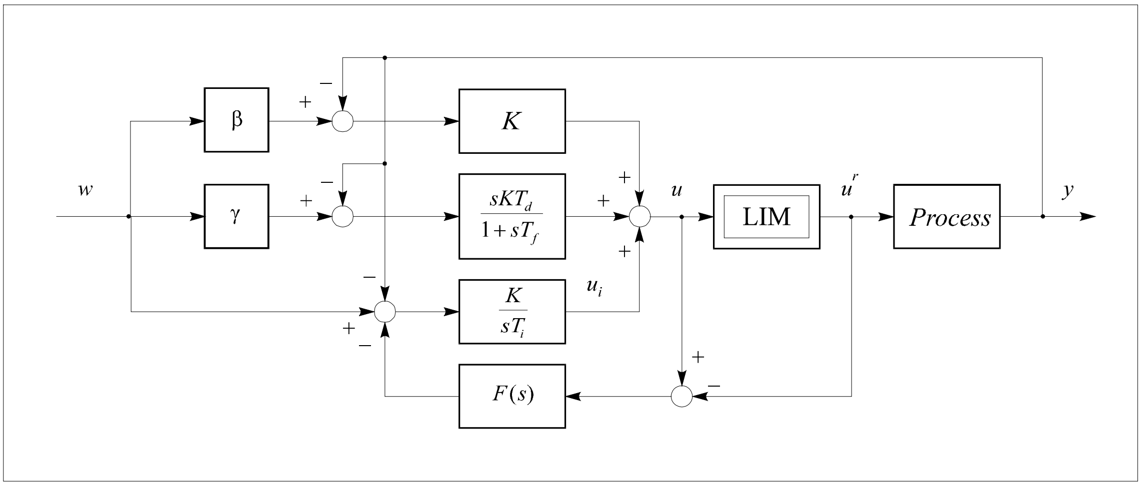

Fig. 14. Limited closed-loop system with AW for the generalised PID controller

$$U(s) = K\left[\beta + \frac{1}{sT_i} + \gamma\frac{sT_d}{1+sT_f}\right]W - K\left[1 + \frac{1}{sT_i} + \frac{sT_d}{1+sT_f}\right]Y - \frac{K}{sT_i}F(s)\left(U - U^r\right). \quad (10)$$

Fig. 15 and expression (11) give the anti-windup realisation for a controller given in the polynomial form, and Fig. 16 and expression (12) give the anti-windup realisation for controllers given in the state-space form.

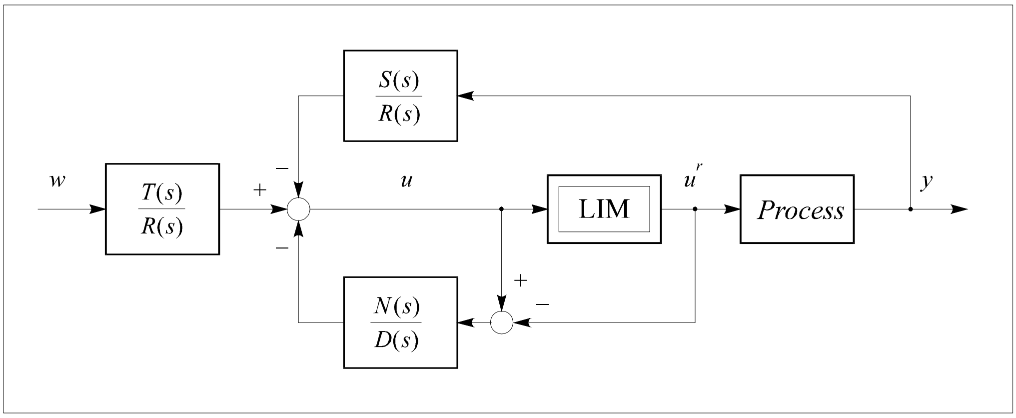

Fig. 15. Limited closed-loop system with AW for the controller given in the polynomial form

$$U(s) = \frac{T(s)}{R(s)}W - \frac{S(s)}{R(s)}Y - \frac{N(s)}{D(s)}\left[U - U^r\right] \tag{11}$$

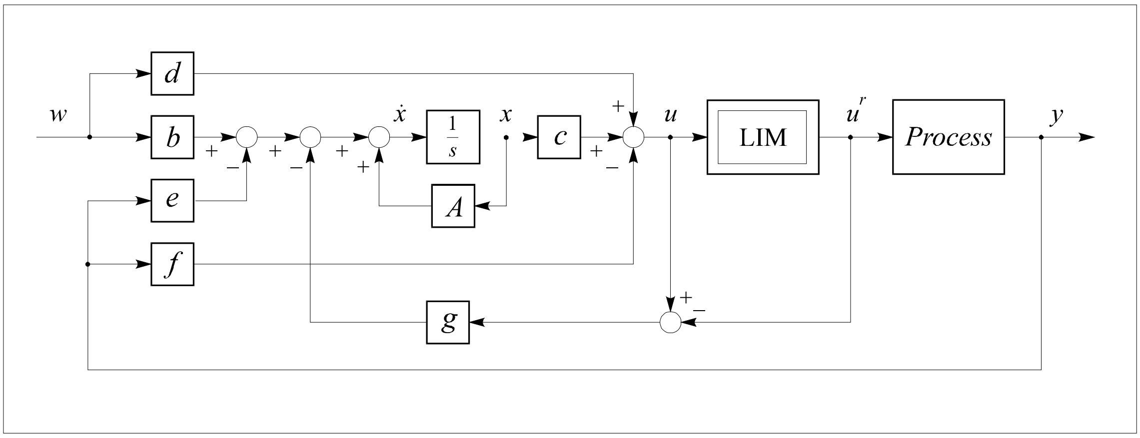

Fig. 16. Limited closed-loop system with AW for the controller given in state-space form

$$\begin{aligned} \dot{x} &= Ax + bw - ey - g\left(u - u^r\right) \\ u &= cx + dw - fy \end{aligned} \tag{12}$$

2.1.1 Observer approach

This AW approach was first presented in Åström and Wittenmark, (1984).

The interpretation of the windup phenomenon is that the states of the controller do not correspond to the control signal being fed to the process (Åström and Rundqwist, 1989; Åström and Wittenmark, 1984; Rundqwist, 1990). In order to correctly estimate the states when $u^r \neq u$, an observer is introduced.

For the PID controller (β=1, γ=0), the correction of the state is proportional to the difference between u and u^r through a static gain L (Peng at al., 1996a,b). The solution corresponds to the anti-windup protection given in Fig. 8 by substituting

$$F(s) = L\ . \tag{13}$$

The anti-windup compensation for the controller given in the polynomial form (Fig. 15) can be represented by the modified scheme given by Fig. 17, where the transfer function A_0(s) denotes the observer polynomial.

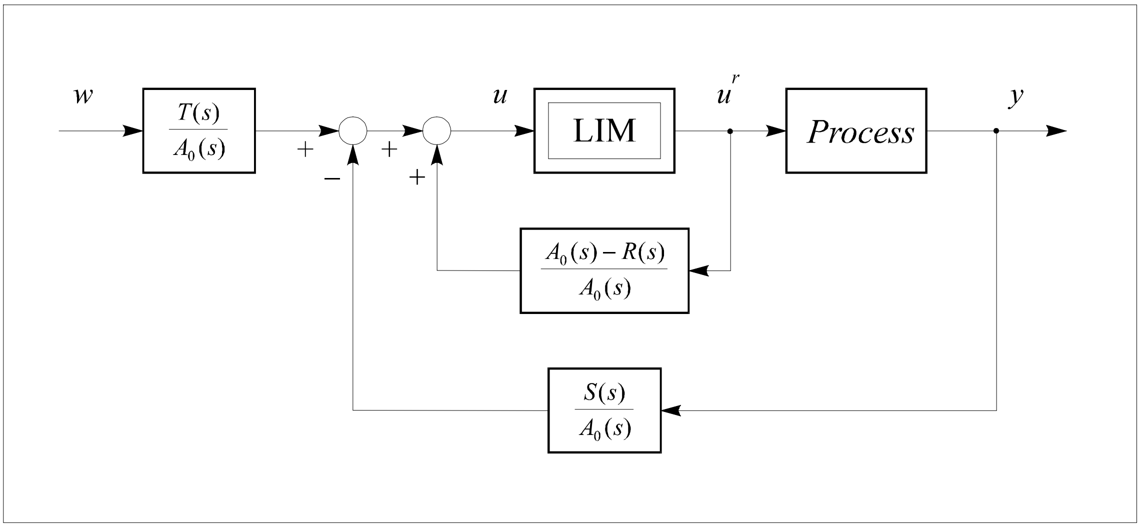

Fig. 17. Anti-windup protection for the controller given in the polynomial form using the observer approach

Some other forms of observer-based anti-windup schemes for the controller given in the polynomial form are given in Hanus (1989), and Wurmthaler and Hippe (1996). The state-space realisation of the observer approach can be found, amongst others, in Hanus (1989), Åström and Rundqwist (1989), and Walgama and Sternby (1990).

2.1.2 The conditioning technique

This AW approach was first presented in Hanus, (1980) as an extension of the back calculation method for digital controllers proposed by Fertik and Ross, (1967), and the external reset feedback method for analog controllers given in Shinskey, (1967).

The anti-windup protection for the PID controller is realised by the same scheme as given in Fig. 8 by substituting

$$F(s) = \frac{1}{K\left(\beta + \gamma \dfrac{T_d}{T_f}\right)} \frac{1 + s\left(T_f - \gamma \dfrac{T_i T_d}{T_f}\right)}{\left(1 + sT_f\right)} \ . \qquad (14)$$

Note that $F(s)$ is reduced to a static gain if the derivative term of the controller is not connected to the reference (γ=0):

$$F(s) = \frac{1}{K\beta} \tag{15}$$

Similarly, when applying the conditioning technique to the PI controller (by applying β=1, and T_d=0 to Fig. 8 and expression (15)), the controller output becomes:

$$U = K(W - Y) + \frac{U^r}{1 + sT_i} \ . \tag{16}$$

Note that expression (16) is the same as that obtained using the "external reset feedback" anti-windup protection for an analog PI controller given by Shinskey (1967) (see Fig. 13 and expression (8)).

The anti-windup protection for controllers given in the polynomial form (11) (see Fig. 15), using the conditioning technique, is the following (see Peng (1996), and Vrančić et al. (1995b):

$$\frac{N(s)}{D(s)} = \frac{T(s)}{R(s)} \lim_{s \to \infty} \frac{R(s)}{T(s)} - 1 \tag{17}$$

Similarly, the anti-windup protection for controllers given in the "observer" form (see Fig. 17) can be achieved by substituting

$$A_0(s) = T(s) \lim_{s \to \infty} \frac{R(s)}{T(s)} \ . \tag{18}$$

The anti-windup protection for the state-space controller (12) (see Fig. 16), when using the conditioning technique, is the following:

$$g = bd^{-1} \tag{19}$$

Some extensions and testings of the conditioning technique are given in Hanus and Peng (1991, 1992), Peng (1996), Peng et al. (1996a,b), Pušnik et al. (1995), Vrančić et al. (1995b), and Walgama et al. (1992).

2.1.3 Incremental algorithm

The incremental algorithm is very often used to prevent windup phenomena in practice (Åström and Rundqwist, 1989; Hanus, 1989; Hyde and Glover, 1993; Vandenbussche, 1975). It is also a relatively simple method to be incorporated in a digital controller. Fig. 18 shows a typical discrete-time implementation of the PID controller. Some other incremental forms are also given by Walgama and Sternby (1990).

It can be shown (Peng et al., 1996a,b) that the continuous-time implementation of the incremental algorithm can be represented as a special case of the anti-windup protection given in Fig. 8 by applying strong anti-windup compensation:

$$F(s) \to \infty \ . \tag{20}$$

From expression (20) and Fig. 8 it is obvious that due to a high gain of $F(s)$, the controller output (u) tightly follows the process input (u^r) when a system is saturated.

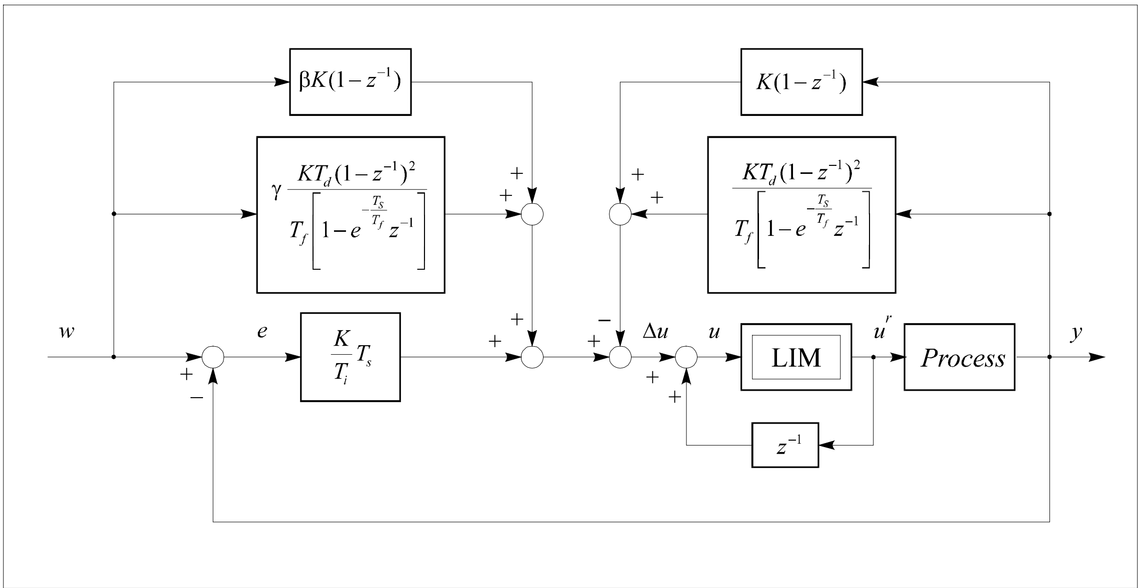

Fig. 18. Incremental algorithm (discrete-time implementation with sampling time T_S)

One of the general incremental forms for the controller given in the polynomial form is given by Walgama and Sternby (1990):

$$U(z) = z^{-1}U^{r}(z) + \left(1 - z^{-1}\right)\frac{T(z)}{R(z)}W(z) - \left(1 - z^{-1}\right)\frac{S(z)}{R(z)}Y(z) \; . \tag{21}$$

It can be seen from expression (21) that the controller output (u) tightly follows the process input (u^{r}) except at the instant when the reference changes.

Besides the anti-windup protection methods presented, other methods also exist like conditional integration methods (Rundqwist, 1991; Peng et al., 1996a,b) or model-based methods (Hanus, 1989; Walgama and Sternby, 1990; Edwards and Postlethwaite, 1996). However, these methods will not be considered in this thesis.

2.2 Comparative study

The anti-windup schemes presented can be compared using different criteria. However, it will be shown that most of them are not adequate for comparing system responses under limitations.

2.2.1 Classical comparison criteria

The most popular comparisons are based on the following criteria:

- process overshoot,
- process settling time,
- integral absolute error (IAE), integral squared error (ISE), etc.

Process overshoot is the most popular criterion when comparing the process responses of different anti-windup schemes. One of the most undesired effects caused by windup is excessive overshoot. When applying the appropriate anti-windup protection to a limited system, the overshoot is usually reduced. However, this is only one of the results of using the appropriate anti-windup protection and is not its primary goal. The goal is to reduce controller windup (in order to properly update the controller states). The question is how high an overshoot, when using the proper anti-windup system, should be obtained? We believe that the smallest possible overshoot is usually not the optimal solution (see also Huba et al., 1997).

Let us explain the above statements in one example by using the process:

$$G_P = \frac{1}{(1+s)^3} \tag{22}$$

and the following PID controller:

$$K = 5,\ \ T_i = 4s,\ \ T_d = 0.3s,\ \ T_f = 0.03s \tag{23}$$

The process limitations were U_{max}=1.5 and U_{min}=-1.5. Fig. 19 shows process responses when using different anti-windup compensators *F(s)*. It is obvious that the anti-windup compensator *F(s)*=1.92/*K* results in no overshoot, but it is also obvious that the solution with *F(s)*=1/*K* results in better tracking performance.

There is one more question to be answered. If the unlimited response gives about 40% process overshoot, why should the anti-windup protection reduce it to e.g. 0%? We believe that anti-windup protection is aimed at reducing windup and not "shaping" the limited response. If a smaller overshoot of the limited system is desired, why isn't the unlimited response made with a smaller overshoot?

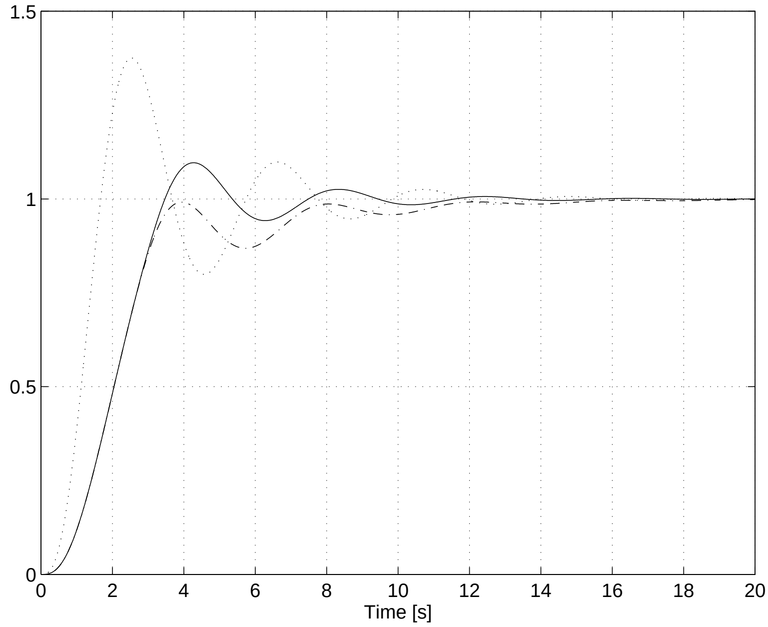

Fig. 19. Process responses under limitations U_{max}=1.5, U_{min}=-1.5; ... Unlimited system, __ Limited system with F(s)=1/K, -.- Limited system with F(s)=1.92/K.

Process settling time is also quite a popular criterion for comparing different anti-windup schemes. Like the overshoot criterion, this one has several drawbacks. The problem starts with defining the settling time. It can be 5%, 2%, etc. but which one should be chosen?

Let us show another example using the same process (6) and controller (23) as in the previous case, but with the following system limitations: U_{max}=2.5, U_{min}=-2.5. The process responses are shown in Fig. 20.

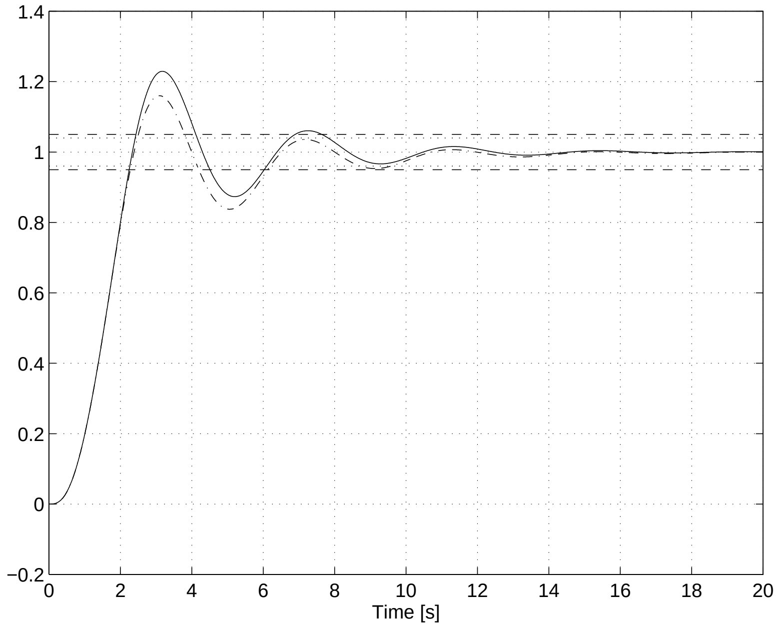

Fig. 20. Process responses under limitations U_{max}=2.5, U_{min}=-2.5; __ Limited system with F(s)=1/K, -.- Limited system with F(s)=1.92/K, -- 5% bandwidth, ... 4% bandwidth

The anti-windup compensator protection where *F(s)*=1.92/*K* gives faster 5% settling time, but faster 4% settling time is obtained where *F(s)*=1/*K*. However, both settling times are shorter where *F(s)*=1/*K* with the process limitations U_{max}=1.5 and U_{min}=-1.5 (see Fig. 19).

Therefore, as with the overshoot criterion, the settling time depends on the system limitations.

Moreover, if the unlimited system has relatively long settling times, is it really desirable to have short settling times under saturation? If so, then the unlimited system should also be tuned so as to have a short settling time.

Integral of absolute error and ***integral of squared error*** are also frequently used criteria for comparing different anti-windup methods, where the error can be defined as the difference between the limited and unlimited response and the difference between the limited response and the reference. We shall now introduce four criteria:

$$IAE_1 = \int_0^{\infty} |y_0(t) - y_1(t)| dt \tag{24}$$

$$IAE_2 = \int_0^{\infty} |w(t) - y_1(t)| dt \tag{25}$$

$$ISE_1 = \int_0^{\infty} \left(y_0(t) - y_1(t)\right)^2 dt \tag{26}$$

$$ISE_2 = \int_0^{\infty} \left(w(t) - y_1(t)\right)^2 dt \;, \tag{27}$$

where y_0, y_1 and w denote the unlimited process response, limited process response and the reference, respectively.

Let us make another experiment using the same process (6) and controller (23) as in the previous case, but with the following system limitations: U_{max}=1.1, U_{min}=-1.1. The process responses are shown in Fig. 21.

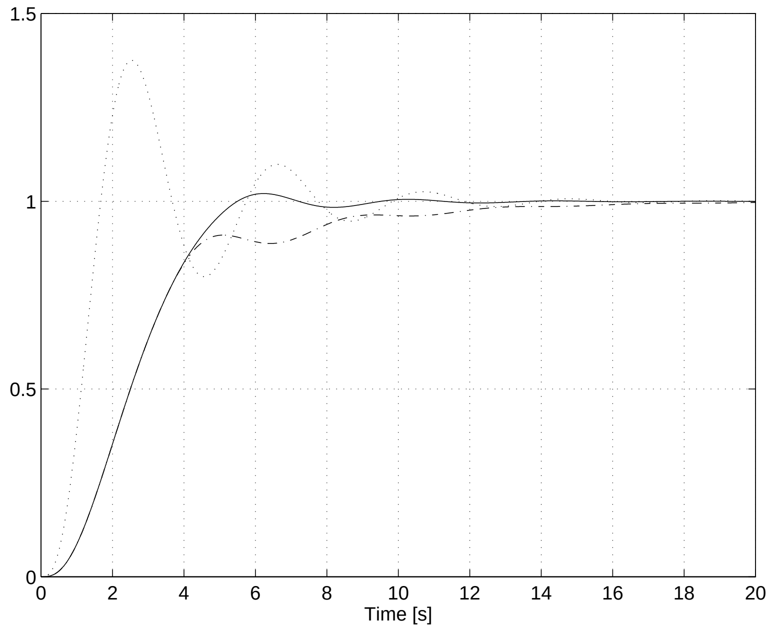

Fig. 21. Process responses under limitations U_{max}=1.1, U_{min}=-1.1; ... Unlimited system, __ Limited system with F(s)=1/K, -.- Limited system with F(s)=1.92/K.

The values of the criteria for all three examples (see Figs. 19 to 21) are given in Table 1.1.

It is obvious that for the process limitations U_{max}=1.5, and U_{min} =-1.5, the values of IAE_1 and ISE_1 are smaller for $F(s)$=1.92/K, whilst IAE_2 and ISE_2 are smaller when applying $F(s)$=1/K. For the process limitations U_{max}=2.5, and U_{min} =-2.5, the values of all the criteria are smaller when using $F(s)$=1.92/K, and just the opposite happens for the limitations U_{max}=1.1, and U_{min} =-1.1.

It is therefore obvious that all of the criteria mentioned strongly depend on system limitations and cannot be successfully used for comparing different anti-windup schemes.

Table 1.1. The criterion values at different process limitations

$F(s)$	U_{max}	U_{min}	IAE_1	IAE_2	ISE_1	ISE_2
1/K	1.5	-1.5	2.149	2.28	0.914	1.575
1.92/K	1.5	-1.5	2.058	2.49	0.881	1.595
1/K	2.5	-2.5	1.201	2.089	0.277	1.26
1.92/K	2.5	-2.5	1.141	2.050	0.271	1.246
1/K	1.1	-1.1	2.231	2.68	1.302	1.911
1.92/K	1.1	-1.1	2.515	3.154	1.350	1.950

Let us consider another example using the process

$$G_{PR} = \frac{1}{(1+4s)(1+s)^2}, \tag{28}$$

and the PID controller

$$K = 10,\ \ T_i = 30s,\ \ T_d = 0.5s,\ \ T_f = 0.05s,\ \ \beta = 1,\ \ \gamma = 0, \tag{29}$$

with the following limitations

$$U_{\max} = 2,\ \ U_{\min} = 0. \tag{30}$$

The response of such system, when the reference changes from 0 to 1 at time origin, is shown in Figures 22 and 23.

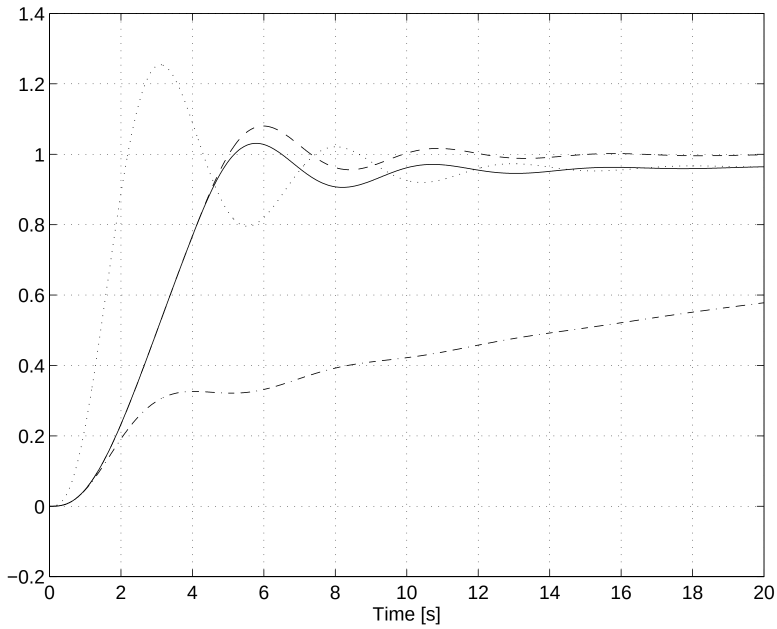

Fig. 22. Process output (y); __ Limited with F(s)=1/K, -.- Limited with F(s)=100, -- Limited without AW protection (F(s)=0), ... Unlimited response

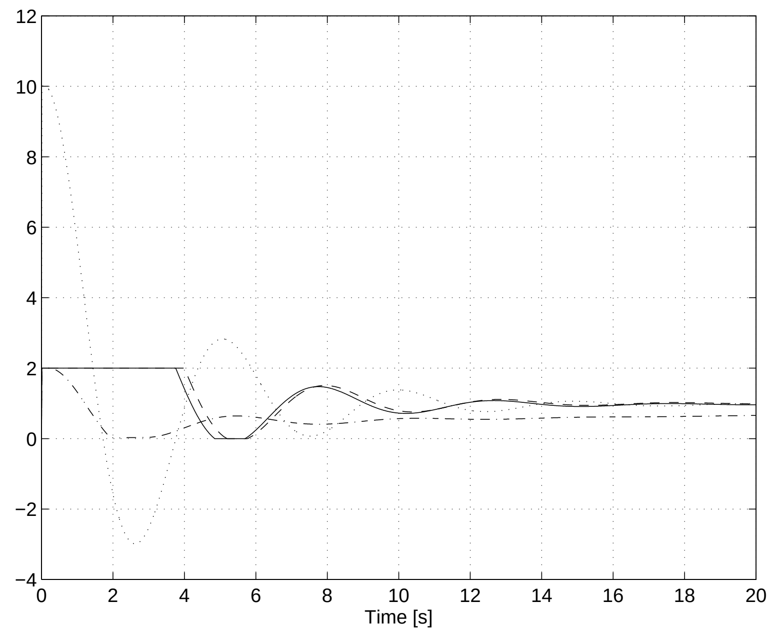

Fig. 23. Process input (u^r); __ Limited with F(s)=1/K, -.- Limited with F(s)=100, -- Limited without AW protection (F(s)=0), ... Unlimited response

Fig. 22 shows the process output (y). The dotted line denotes the unlimited process response.

It can be seen that the PID controller is not tuned well. The unlimited process response has a long settling time (in Fig. 22 it appears as a steady-state error), because the integral time constant (T_i) is too high. The limited process time response when no AW protection is used (denoted by a dashed line) seems better than when using the anti-windup compensator $F(s)=1/K$ (a solid line).

Of course, this does not mean that the windup phenomenon produces useful and desired system behaviour in general. It only means that the controller parameters should be properly tuned for the unlimited process. How this is done for the PID controller can be found in Part II of this thesis.

In the field of anti-windup design, most researchers seek the "optimal" anti-windup compensator. However, in the text above it was shown that the "optimality" (overshoot, settling time, IAE, ISE,...) depends on system constraints, on the process set-point and dynamics, and on the reference (disturbance) changes. It is, therefore, practically impossible to find one "optimal" anti-windup compensation for all possible cases. Nearly each form of anti-windup protection has its drawbacks and advantages under different conditions.

However, it *is* possible to obtain an "optimal" process response when using certain approaches known from the field of optimal control (Bisták et al., 1996; Kulha and Huba, 1996; Huba et al., 1996; Rawlings and Muske, 1993; Scokaert, 1997; Scokaert and Rawlings, 1996; Šulc, 1993). Such solutions are usually quite involved, and require more extensive computations. However, in this thesis we are dealing with classical anti-windup schemes, which mainly aim to prevent controller *windup* and not reduce the overshoot, settling time, etc.

The remaining question then is how to compare different anti-windup schemes. It is shown that all of the mentioned criteria depend on factors, such as controller tuning, reference changes and process limitations. We are therefore seeking a criterion for making comparisons of different anti-windup compensators which is independent of those factors. It will be shown that a *realisable reference* could prove to be such a criterion.

2.2.2 Realisable reference

The notion of the realisable reference is closely related to the work of Fertik and Ross, (1967), where the authors used the term "effective error", which is in fact the difference between the realisable reference and the process output. However, the term "realisable reference" was first mentioned in Hanus et al., (1987) when describing the conditioning technique.

The realisable reference w^r is such that if it had been applied to the controller instead of the reference w, the control output u would have been equal to the real plant input u^r

obtained with the reference w. In this case, the limitation is not activated (Peng et al, 1996a,b).

If w^r is used in the control scheme described by Fig. 8, by definition, the limiter is not activated (u^r is always the same as u), so it can be put away as shown in Fig. 24. Moreover, u^r and y described in Fig. 24 are equivalent to u^r and y described in Fig. 8, respectively.

We can see that the control scheme described by Fig. 24 does not include any implicit nonlinearity. The nonlinearity is hidden in the realisable reference w^r. From Fig. 24, it can clearly be seen that y tracks w^r instead of w, with the expected linear performance.

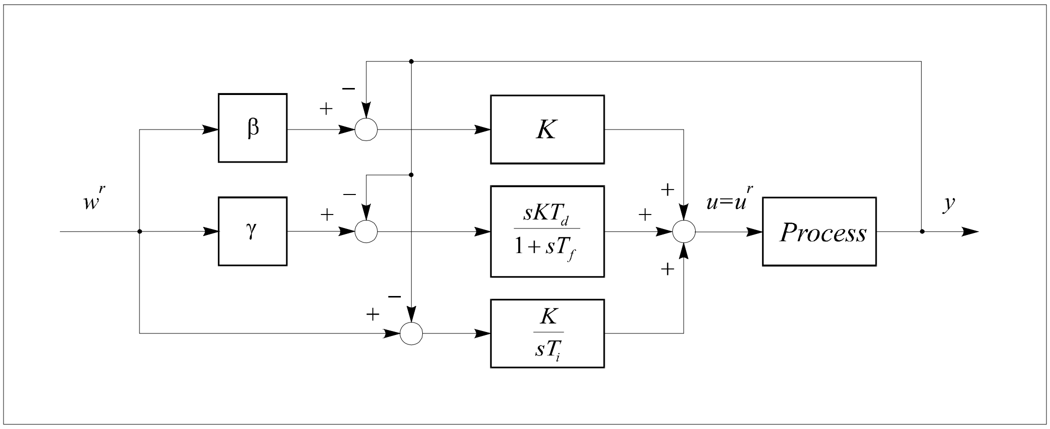

Fig. 24. Equivalent scheme of Fig. 8

From the definition of the realisable reference, we have

$$U^r = K\left(\beta + \frac{1}{sT_i} + \gamma\frac{sT_d}{1+sT_f}\right)W^r - K\left(1 + \frac{1}{sT_i} + \frac{sT_d}{1+sT_f}\right)Y \quad , \tag{31}$$

which yields

$$W^r = \frac{sT_i\left(1+sT_f\right)U^r + K\left[s^2T_i\left(T_d+T_f\right) + s\left(T_i+T_f\right) + 1\right]Y}{K\left[s^2T_i\left(\gamma T_d + \beta T_f\right) + s\left(\beta T_i + T_f\right) + 1\right]} \quad . \tag{32}$$

During the limitation, U^r is the same for whatever $F(s)$ is used, and so is Y, if the initial conditions are the same. Hence, from (32), it can be seen that during limitation, W^r is the same for whatever $F(s)$ is used. For the linear feedback AW algorithm, we have:

$$U = K\left(\beta + \frac{1}{sT_i} + \gamma\frac{sT_d}{1+sT_f}\right)W - K\left(1 + \frac{1}{sT_i} + \frac{sT_d}{1+sT_f}\right)Y + \frac{K}{sT_i}F(s)\left(U^r - U\right) \quad (33)$$

Subtracting (31) from (33), we can calculate w^r as

$$\begin{aligned} W^r &= W + \frac{\left(KF(s) + sT_i\right)\left(1+sT_f\right)}{K\left[s^2T_i\left(\gamma T_d + \beta T_i\right) + s\left(\beta T_i + T_f\right) + 1\right]}\left(U^r - U\right) = \\ &= W + G_w(s)\left(U^r - U\right) \end{aligned} \quad . \quad (34)$$

As $G_w(s)$ is a dynamic transfer function, w^r will not become the same as w at the instant when the controller leaves the limitation ($u^r{=}u$), unless $G_w(s)$ is reduced to a static gain. Indeed, when applying the conditioning technique (14), (34) yields:

$$G_w(s) = \frac{1}{K\left(\beta + \gamma\dfrac{T_d}{T_f}\right)} \quad , \quad (35)$$

$$w^r = w + \frac{u^r - u}{K\left(\beta + \gamma\dfrac{T_d}{T_f}\right)} \quad . \quad (36)$$

At the instant the controller leaves the limitation ($u^r{=}u$), w^r becomes w. This choice thus gives the best tracking performance.

In order to compare the above-mentioned AW algorithms, we have made a simulation with process:

$$G(s) = \frac{1}{\left(1+8s\right)\left(1+4s\right)} \quad , \quad (37)$$

and the PID controller:

$$K = 20,\ \ T_i = 30s,\ \ T_d = 0.95s,\ \ T_f = 0.095s,\ \ \beta = 1,\ \ \gamma = 0\,. \qquad (38)$$

The incremental algorithm is achieved by using the scheme given in Fig. *18*. The sampling time was T_S=0.01s.

The process limitations were U_{max}=2, U_{min}=0, v_{max}=2s^{-1}, and v_{min}=-2s^{-1}. The closed-loop step responses for both limited and unlimited cases are shown in Figs. 25 and 26.

It is clearly seen from Fig. 25 that the conditioning technique gives a w^r which is the closest to w. Fig. 26 demonstrates that y tracks w^r instead of w.

It is now clear why Fertik and Ross (1967) made the following observations: "Evaluation of tests results for various no-windup schemes can be misleading since the results depend on the degree of velocity limiting, the dynamics of the process and the tuning criteria of the controller. The back calculation method (equivalent to conditioning technique for PI controller), however, *has shown superior performance under a wide variety of test conditions*".

Similar observations are found in Vrančić et al. (1993a), Vrančić and Petrovčič (1993), and Vrančić and Petrovčič (1996).

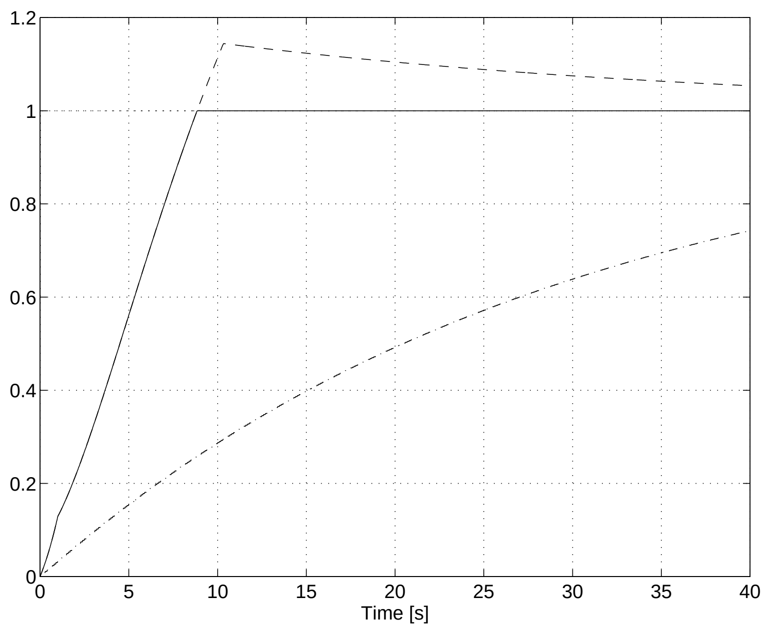

Fig. 25. Realisable reference (w^r); — Conditioning technique, -- Without AW, -.- Incremental algorithm

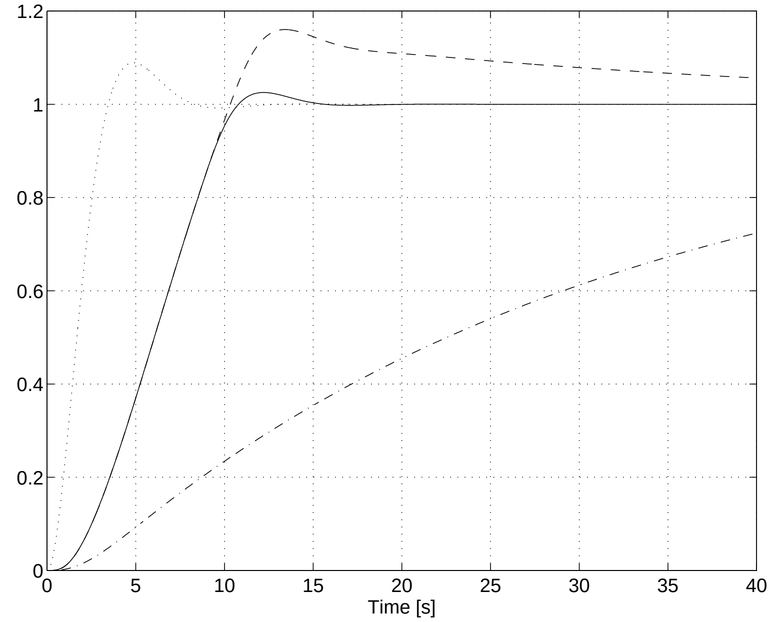

Fig. 26. Process output (y); — Conditioning technique, -- Without AW,
-.- Incremental algorithm, ··· Unlimited response

Similar derivations, as for the PID controller, can be made for controllers given in a rational form (see Fig. 17). The realisable reference can be expressed as

$$w^r = w + G_w\left(u^r - u\right), \tag{39}$$

where

$$G_w(s) = \frac{A_0(s)}{T(s)} \tag{40}$$

The rational function $G_w(s)$ is reduced to a static gain if

$$A_0(s) = K_S T(s) \ , \tag{41}$$

where K_S is a static gain chosen so as to make the anti-windup algorithm implementable. As given by Fig. 17, the algebraic loop can be formed from u^r to u if $A_0(\infty)$-$R(\infty)\neq 0$ (for more details see Peng (1996)). In order to avoid the algebraic loop, the following relation is to be satisfied:

$$A_0(\infty) = R(\infty)\,. \tag{42}$$

Applying (42) into (41) results in

$$A_0(s) = T(s)\lim_{s\to\infty}\frac{R(s)}{T(s)}\ . \tag{43}$$

This is the same as the previously given solution for the conditioning technique (18). As with the PID controller, at the instant the controller leaves the limitation (u^r=u), w^r becomes w. This choice (43) thus gives the best tracking performance.

In order to compare the AW algorithms for controllers given in a polynomial form, a simulation was made using the same process (6) as before and with the following controller:

$$\begin{aligned} &T = 40.70s^2 + 34.32s + 8.83 \\ &S = T \\ &R = 0.276s^2 + s \end{aligned} \tag{44}$$

The observer polynomial, when using the conditioning technique, was derived from (43):

$$A_0(s) = T(s)\lim_{s\to\infty}\frac{R(s)}{T(s)} = 0.276s^2 + 0.233s + 0.06 \tag{45}$$

The incremental algorithm was applied using the expression (21). The sampling time was T_S=0.01s and the transformation from continuous time to discrete-time controller was made by using the step-invariance method.

The input limitations were U_{max}=10, and U_{min}=-10. The closed-loop step responses for both limited and unlimited cases are shown in Figs. 27 and 28.

It can be seen that the conditioning technique gives a very good process tracking performance.

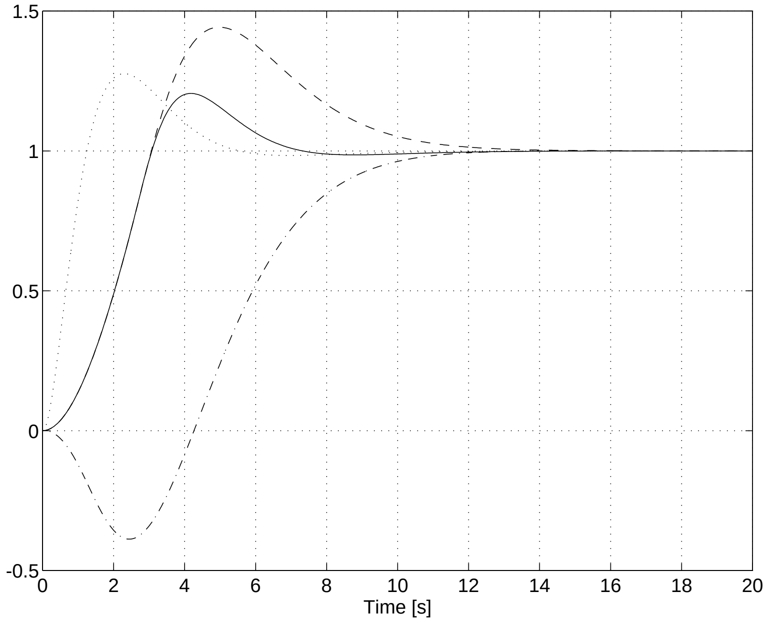

Fig. 27. Process output (y); __ conditioning technique, -- without anti-windup protection, -.- incremental algorithm, ... unlimited response

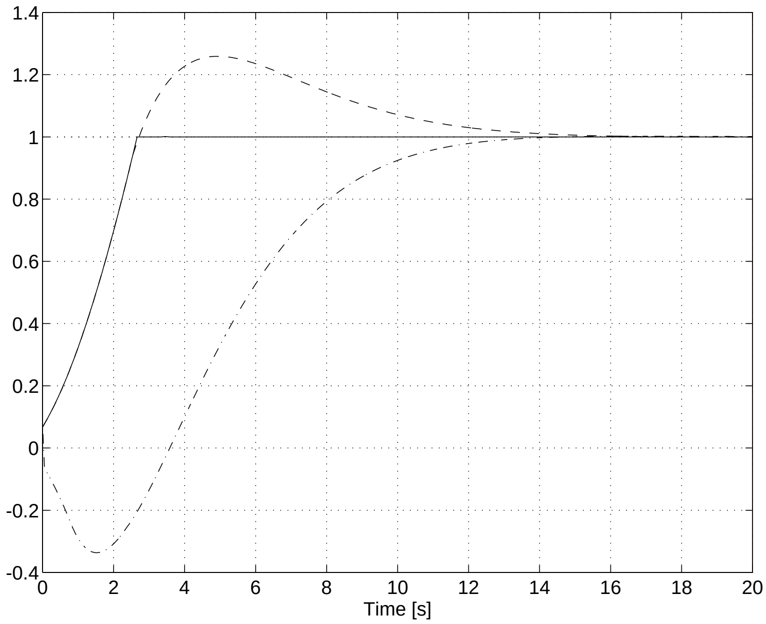

Fig. 28. Realisable reference (w^r); __ conditioning technique, -- without anti-windup protection, -.- incremental algorithm

Similarly, the realisable reference can also be derived for the controllers given in the state-space form.

Consider the controller implementation scheme described by Fig. 16 and expression (12). Expressions (12) can be described by the input-output form through

$$X = (sI - A)^{-1}\left[bW - eY - g\left(U - U^r\right)\right] \tag{46a}$$

$$U = \left[c(sI - A)^{-1}\left[b \quad -e \quad -g\right] + \left[d \quad -f \quad 0\right]\right]\begin{bmatrix} W \\ Y \\ U - U^r \end{bmatrix}. \tag{46b}$$

The realisable reference for the SISO state-space controller is defined similarly as for other types of controllers. It can be asserted that the real process input (u^r) and the process output (y) in Fig. 16 are the same as in Fig. 29. Moreover, Fig. 29 does not include any limitation. The limitation is “hidden” in the realisable reference w^r. Because the second scheme is linear, it can be stated that y tracks w^r instead of w.

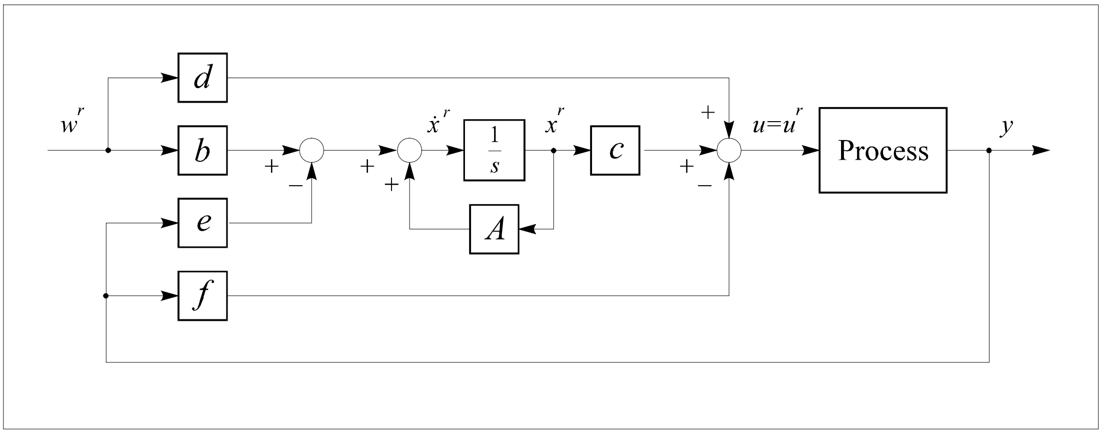

Fig. 29. The equivalent scheme of Fig. 16 from the process viewpoint

From Fig. 29 the u^r can be expressed as:

$$U^r = \left[c(sI - A)^{-1}\left[b \quad -e\right] + \left[d \quad -f\right]\right]\begin{bmatrix} W^r \\ Y \end{bmatrix}. \tag{47}$$

The realisable reference can be calculated by subtracting signal u (46b) from u^r (47):

$$W^r = W + \frac{c(sI - A)^{-1} g + 1}{c(sI - A)^{-1} b + d} \left(U^r - U\right) \ . \tag{48}$$

When using the conditioning technique (19), expression (48) becomes

$$W^r = W + \frac{U^r - U}{d} \ . \tag{49}$$

In order to illustrate the above derivations, a simulation was performed using a process

$$G_{PR} = \frac{1}{(1+8s)(1+4s)} \ ,$$

and controller

$$\begin{aligned} A &= \begin{bmatrix} 0 & 0 \\ 0 & -10 \end{bmatrix} \\ b &= \begin{bmatrix} 20 \\ 0 \end{bmatrix} \\ c &= \begin{bmatrix} 0.0333 & -100 \end{bmatrix} \ . \\ d &= 20 \\ e &= \begin{bmatrix} 20 \\ 20 \end{bmatrix} \\ f &= 220 \end{aligned}$$

The process input is subject to hard limits U_{max}=2 and U_{min}=0.

Figs. 30 and 31 show the results obtained when using different AW algorithms. The reference w goes from 0 to 1 at time t=0. The realisable reference (w^r) for the unlimited response is the same as w.

The effect of the windup is clearly seen (dashed line). The conditioning technique (19) gives quite a good AW response (solid line). The dash-dotted line represents the result obtained by using a 5-times stronger AW compensator than that used in the conditioning technique.

From Figs. 30 and 31 it can be seen that the process output follows the realisable reference (w^r) instead of the actual reference signal (w).

Now, let us investigate the case of mode switching. During manual mode ($u^r = u^m$), as the controller is not connected to u^m, its output is usually quite different from u^r. In this case, after switching, a jump will be produced at the plant input (bump transfer). In order to remove the jump, the controller output u should be made as close as possible to u^m during manual mode. As a result, the jump at the instant of switching will be minimised. This mode switching is called bumpless transfer (BT). Yet, there is no guarantee that the tracking performance will be good after mode switching.

If the controller output u is adjusted so that after switching from manual to automatic control mode, the plant output y tracks the reference w with the same dynamics as the closed-loop step response, then this mode switching is called conditioned transfer (CT). In other words, after switching, a good tracking performance will be assured when using conditioned transfer. Note that the jump is usually small but not minimised in this case.

An anti-windup strategy is usually implemented as a bumpless transfer technique as shown in Fig. 32. However, it will be shown that only the incremental algorithm is a solution for BT, whilst the conditioning technique is a solution for CT.

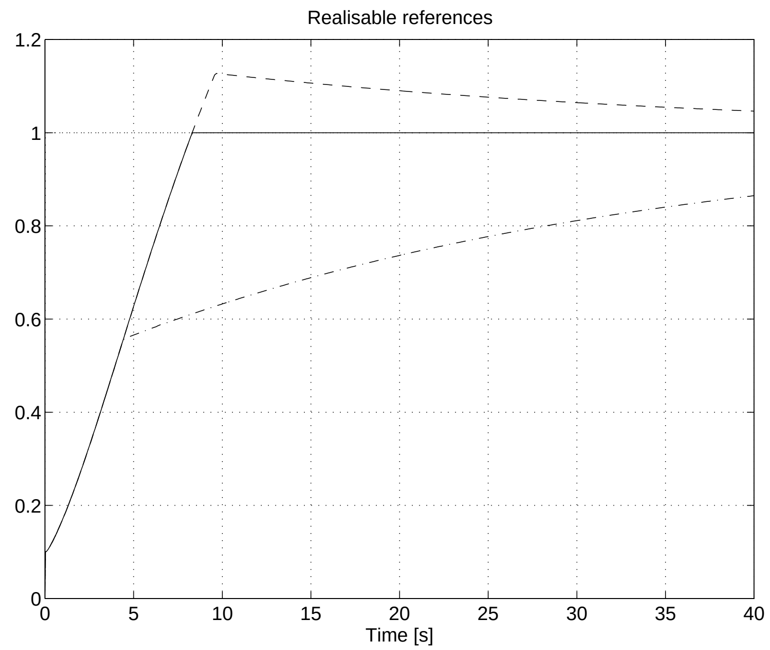

Fig. 30. Realisable references (w^r); __ Conditioning technique ($g=bd^{-1}$), -.- $g=5*bd^{-1}$, -- without AW ($g=0$)

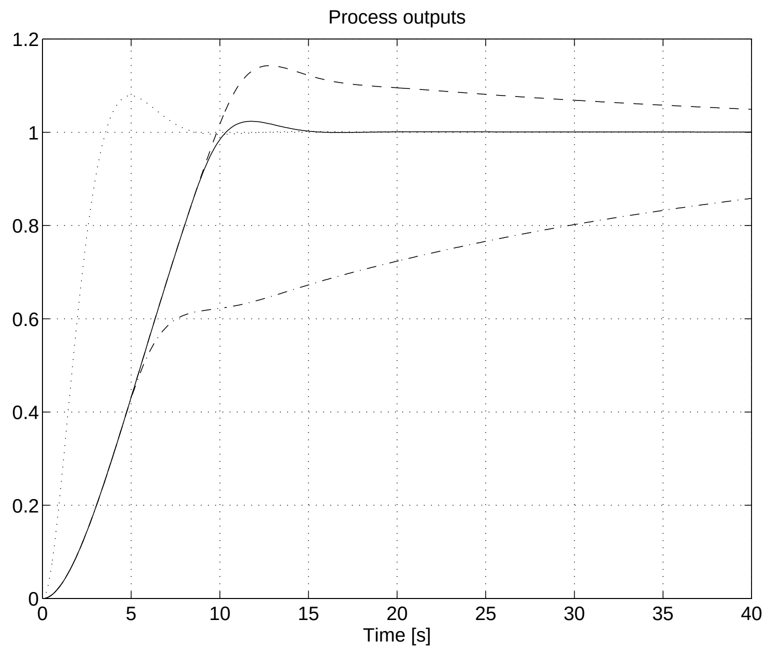

*Fig. 31. Process outputs (y); __ Conditioning technique (g=bd^{-1}), -.- g=5*bd^{-1}, -- without AW (g=0), ... Unlimited response*

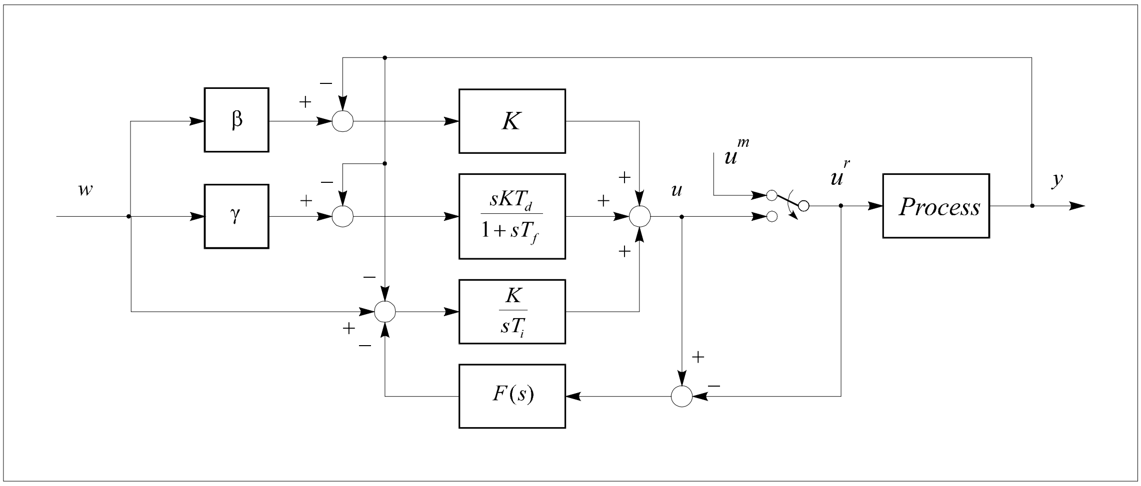

Fig. 32. Bumpless and conditioned transfer from manual to automatic modes for the generalised PID controller

Indeed, using the incremental algorithm ($F(s)\rightarrow\infty$), during manual mode we can see that u is made nearly equal to u^r. Thus, the incremental algorithm will not produce a jump at u^r at the time of switching, and can be used as a BT method.

The realisable reference w^r for BT and CT methods can be defined in the same way as for the AW scheme. Figs. 32 and 24 then represent the equivalent schemes from the process viewpoint. Note that the realisable reference is defined for all time. During manual mode ($u^r\neq u$), w^r is different from w, and y tracks w^r. After switching to automatic mode ($u^r=u$), $w^r=w$ is desired so that y will track w with the same dynamics as the closed-loop step response (CT). The only way to do this at the instant of switching is to use the conditioning technique. Usually, the conditioning technique will produce a jump at the input of the process, because w^r will jump to w when the switching occurs. This is normal, as a jump always occurs when the reference has a step change. Yet if a jump is not tolerable, we can either switch from manual to automatic mode when u is close to u^m (by driving y close to w before switching) or add a rate limitation at the process input (Vrančić et al., 1995b).

In order to support the above arguments, a simulation using the same process (6) and controller (23) as in the previous example was made. The reference signal was taken as 0. The process was manually controlled in the period from 0 to 20s. Then, its input was switched to the PID controller. The results are shown in Figs. 33 to 35.

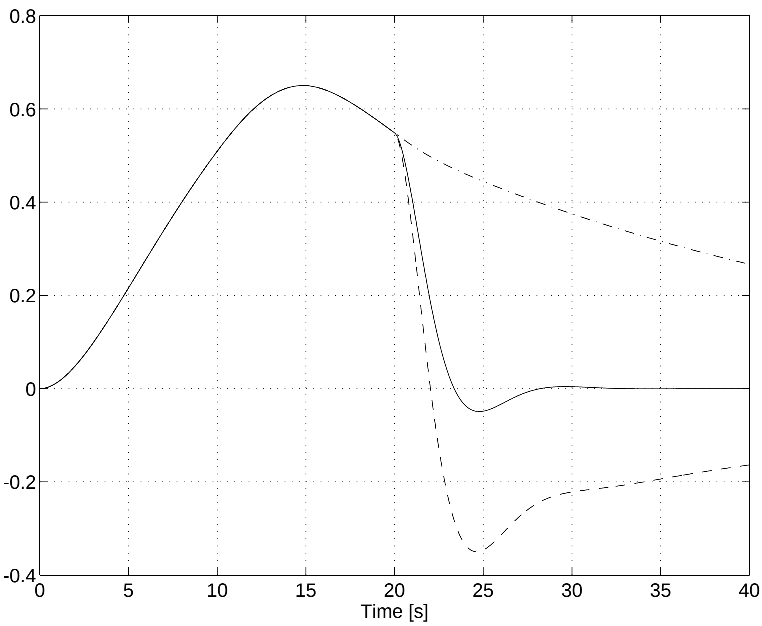

Fig. 33. Process output (y); —— Conditioned transfer, -- Bump transfer, -.- Bumpless transfer

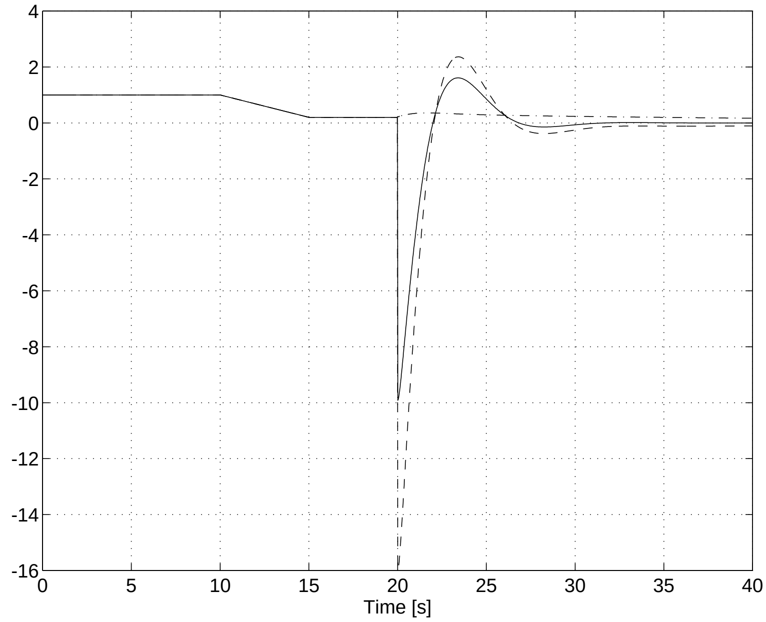

Fig. 34. Process input (u^r); — Conditioned transfer,
-- Bump transfer, -.- Bumpless transfer

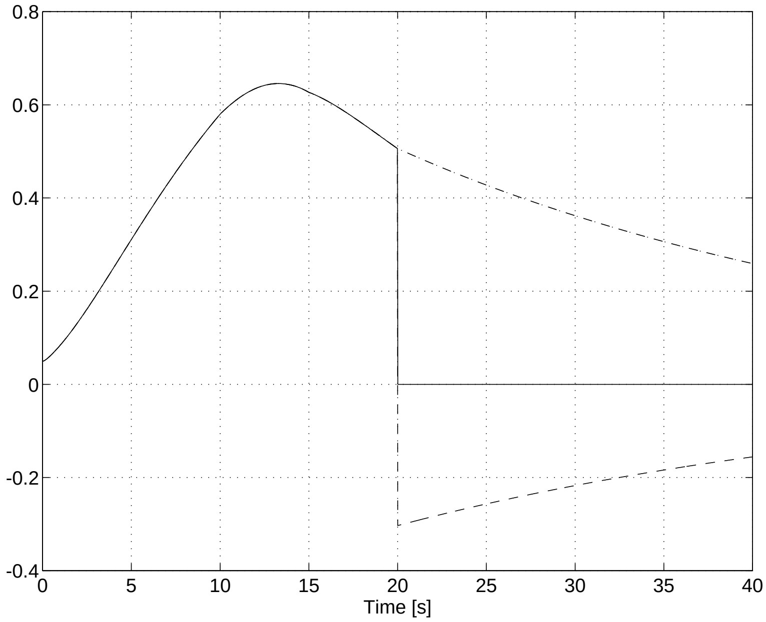

Fig. 35. Realisable reference (w^r); — Conditioned transfer,
-- Bump transfer, -.- Bumpless transfer

It can be seen that the incremental algorithm (BT) (sampling time T_S=0.01s) produces no jump at the process input (u^r) at the instant of switching from manual to automatic mode (Fig. 34), but the settling time of the closed-loop response is relatively long (Fig. 33). On the other hand, the conditioning technique (CT) yields a short settling time at the cost of producing a small jump at the process input. It can also be seen that only the conditioning technique can make w^r=w at the instant of switching from manual to automatic mode (Fig. 35).

Similar derivations also hold for controllers in a polynomial form. Only conditioned transfer, using the conditioning technique, assures a good tracking performance.

A simulation was made using the same process (6) and controller (44) as in the previous example. The reference signal was taken as 0. The process was manually controlled in the period from 0 to 10s. Then, its input was switched to the controller output. The results are shown in Figs. 36 to 38.

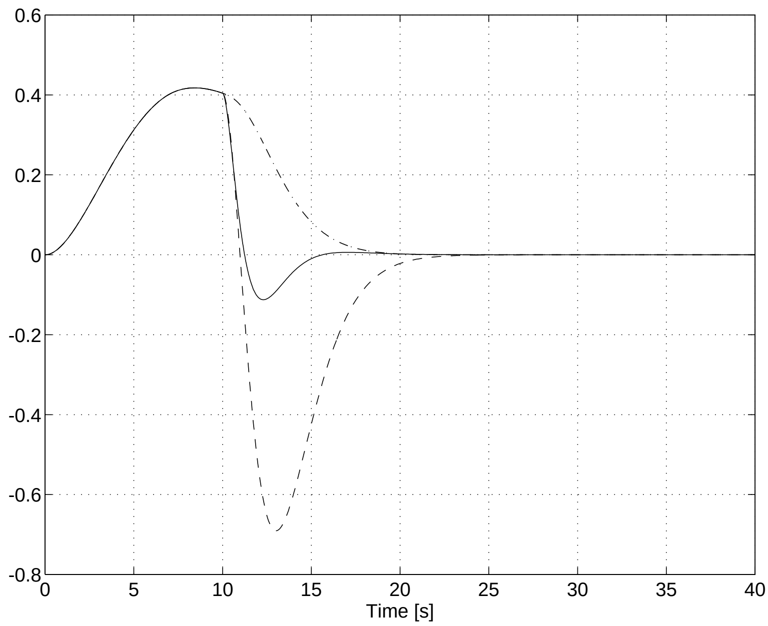

Fig. 36. Process output (y); __ conditioned transfer, -- bump transfer, -.- bumpless transfer

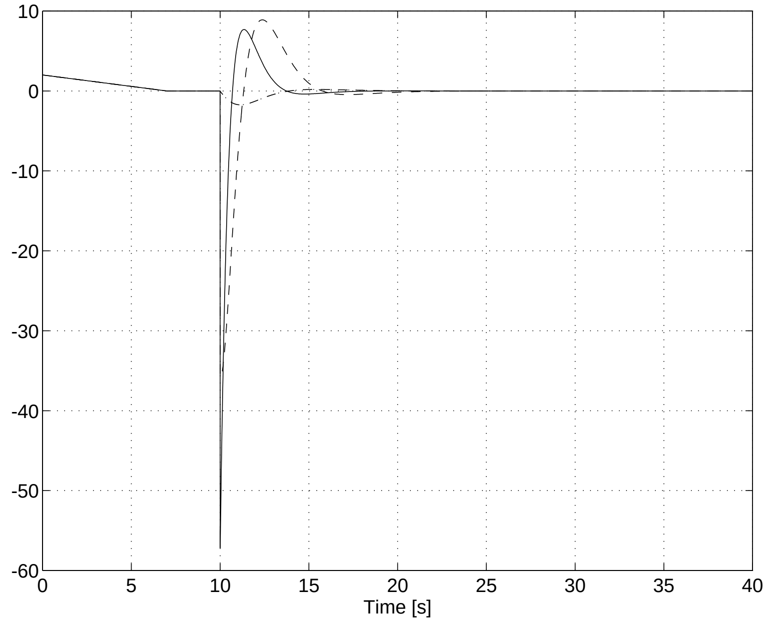

Fig. 37. Process input (u^r); __ conditioned transfer, -- bump transfer,
-.- bumpless transfer

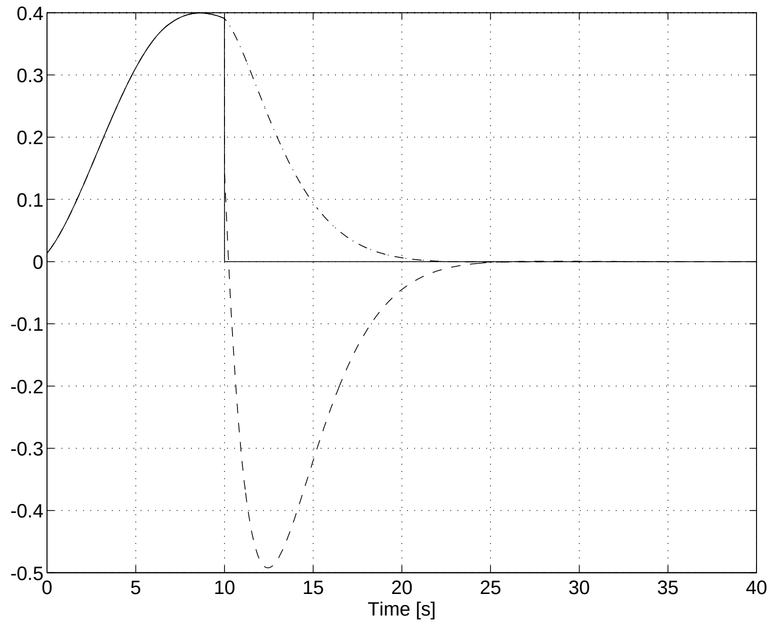

Fig. 38. Realisable reference (w^r); __ conditioned transfer, -- bump transfer,
-.- bumpless transfer

3. *Anti-windup for univariable controllers*

3.1 The shape of the realisable reference

In this chapter, the time-response of the realisable reference for the generalised PID controller will be derived. It will be observed that system behaviour, after it leaves the saturation, depends strongly on certain controller parameters, especially on the integral term time constant. For the PID controller, a smaller T_i generally gives a faster controller recovering after the system is no longer saturated (Vrančić et al., 1996e).

3.1.1 Evaluation of the time response

The anti-windup protection for the generalised PID controller (see Fig. 14) can be transformed into the scheme shown in Fig. 39. Note that K_{ai} and K_{ad} in Fig. 39 are *static* parameters, so an additional AW compensator is connected to the derivative term in order to compensate for the incorrectly updated state of the derivative term's filter during saturation. In Fig. 14 the incorrectly updated state of the derivative term is compensated (if $\gamma \neq 0$) by using a *dynamic* compensator $F(s)$ connected to the integral term. By comparing Figs. 14 and 39, the AW compensator's transfer function can be expressed in the following way:

$$F(s) = \frac{1 + s\left(T_f + \gamma T_i T_d \frac{K_{ai}}{K_{ad}} \right)}{K_{ai}\left(1 + sT_f\right)} \tag{50}$$

In general, anti-windup compensator parameters K_{ai} and K_{ad} can be freely tuned. The exception is the *conditioning technique* which fixes the compensator values at

$$K_{ai} = K\left(\beta + \gamma \frac{T_d}{T_f} \right),\ K_{ad} = -KT_f \left(\beta + \gamma \frac{T_d}{T_f} \right)\ . \tag{51}$$

The realisable reference for the PID controller can be calculated from the signals u and u^r in Figures 39 and 40, respectively.

$$\begin{aligned} U = {} & K\left(\beta + \frac{1}{sT_i} + \gamma \frac{sT_d}{1 + sT_f} \right) W - K\left(1 + \frac{1}{sT_i} + \frac{sT_d}{1 + sT_f} \right) Y + \\ & + \left[\frac{K}{sT_i K_{ai}} + \frac{\gamma K T_d}{K_{ad}\left(1 + sT_f\right)} \right] \left(U^r - U\right) \end{aligned} \tag{52}$$

$$U^r = K\left(\beta + \frac{1}{sT_i} + \gamma \frac{sT_d}{1 + sT_f} \right) W^r - K\left(1 + \frac{1}{sT_i} + \frac{sT_d}{1 + sT_f} \right) Y \tag{53}$$

When subtracting (33) from (53), we can get another form which is useful when comparing w^r and w:

$$\begin{aligned} W^r = {} & W + \\ & + \frac{1}{K} \frac{s^2 T_i T_f + s\left[T_i + \frac{K}{K_{ai}} T_f + \frac{K}{K_{ad}} \gamma T_i T_d \right] + \frac{K}{K_{ai}}}{s^2\left(\beta T_i T_f + \gamma T_i T_d\right) + s\left(\beta T_i + T_f\right) + 1} \left(U^r - U\right) \end{aligned}\ . \tag{54}$$

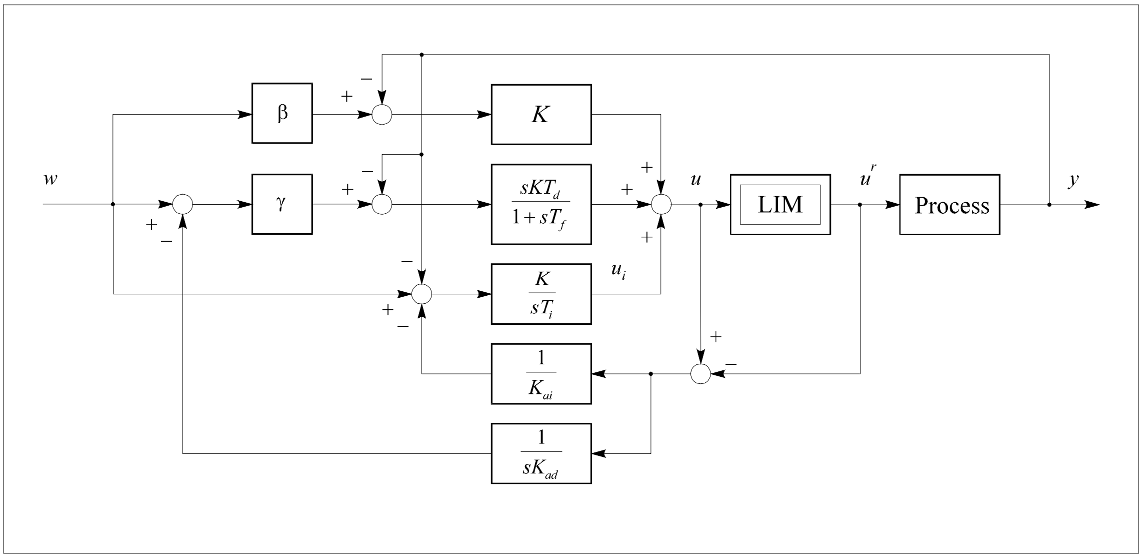

Fig. 39. AW method for the generalised PID controller

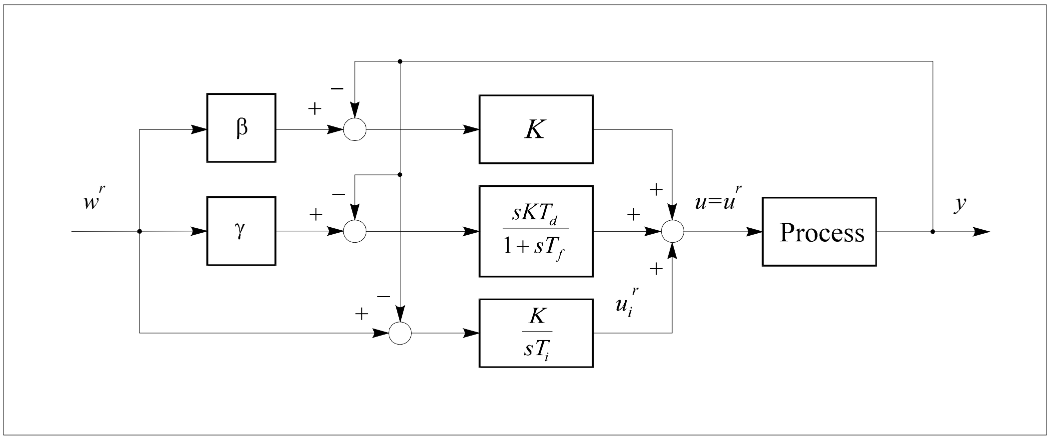

Fig. 40. The equivalent scheme of Fig. 39 from the viewpoint of the process

The realisable reference for the generalised PID controller is defined by expression (54) as

$$W(s) - W^r(s) = G_a(s)\left[U(s) - U^r(s)\right] , \quad (55)$$

where G_a is

$$G_a = \frac{1}{K}\frac{s^2 T_i T_f + s\left(T_i + \frac{K}{K_{ai}}T_f + \frac{K}{K_{ad}}\gamma T_i T_d\right) + \frac{K}{K_{ai}}}{\left(1+s\tau_1\right)\left(1+s\tau_2\right)} \quad , \tag{56}$$

with τ_1 and τ_2

$$\tau_1 = \frac{\beta T_i + T_f + \sqrt{\left(\beta T_i - T_f\right)^2 - 4\gamma T_i T_d}}{2} \tag{57}$$

$$\tau_2 = \frac{\beta T_i + T_f - \sqrt{\left(\beta T_i - T_f\right)^2 - 4\gamma T_i T_d}}{2} \quad . \tag{58}$$

If T_i, T_d, T_f, β and γ are greater than zero, the value of the square root in (57) and (58) is real and smaller than ($\beta T_i + T_f$) or becomes imaginary. This means that time constants τ_1 and τ_2 have *positive* real parts with or without imaginary components.

To calculate $w(t)$-$w^r(t)$ from (55), we can use the convolution integral:

$$\begin{aligned} &w(t) - w^r(t) = g_a(t) * \left[u(t) - u^r(t)\right] = \\ &= \frac{T_i T_f}{K\tau_1\tau_2}\left[u(t) - u^r(t)\right] + c_1 e^{-\frac{t}{\tau_1}} \int_{-\infty}^{+\infty} e^{\frac{\tau}{\tau_1}}\left[u(\tau) - u^r(\tau)\right] d\tau - \quad , \\ &- c_2 e^{-\frac{t}{\tau_2}} \int_{-\infty}^{+\infty} e^{\frac{\tau}{\tau_2}}\left[u(\tau) - u^r(\tau)\right] d\tau \end{aligned} \tag{59}$$

where c_1 and c_2 are

$$\begin{aligned} c_1 &= \frac{T_i T_f}{K(\tau_1 - \tau_2)\tau_1^{\,2}} - \frac{T_i + \frac{K}{K_{ai}}T_f + \frac{K}{K_{ad}}\gamma T_i T_d}{K(\tau_1 - \tau_2)\tau_1} + \frac{1}{K_{ai}(\tau_1 - \tau_2)} \\ c_2 &= \frac{T_i T_f}{K(\tau_1 - \tau_2)\tau_2^{\,2}} - \frac{T_i + \frac{K}{K_{ai}}T_f + \frac{K}{K_{ad}}\gamma T_i T_d}{K(\tau_1 - \tau_2)\tau_2} + \frac{1}{K_{ai}(\tau_1 - \tau_2)} \end{aligned} \tag{60}$$

Consider a case in which the system leaves saturation at $t=t_1$. The controller output u becomes the same as the process input u^r and for $t \geq t_1$ the expression (59) changes:

$$w(t) - w^r(t) = C_1 e^{-\frac{t}{\tau_1}} - C_2 e^{-\frac{t}{\tau_2}} \;;\quad t \geq t_1 \;, \tag{61}$$

where C_1 and C_2 are constants:

$$\begin{aligned} C_1 &= c_1 \int_{-\infty}^{t_1} e^{\frac{\tau}{\tau_1}} \left[u(\tau) - u^r(\tau) \right] d\tau \\ C_2 &= c_2 \int_{-\infty}^{t_1} e^{\frac{\tau}{\tau_2}} \left[u(\tau) - u^r(\tau) \right] d\tau \end{aligned} \;. \tag{62}$$

After the system leaves saturation, w-w^r becomes the difference of two exponential functions, such as that given in (61). If τ_1 and τ_2 are complex (note that the real parts are positive), w-w^r becomes a damped sinusoidal function. For the PID controller ($\beta=1$, $\gamma=0$), the expression (61) simplifies to:

$$w(t) - w^r(t) = C e^{-\frac{t}{T_i}} \;;\quad t \geq t_1 \;, \tag{63}$$

where C is

$$C = \frac{1}{T_i} \left[\frac{1}{K_{ai}} - \frac{1}{K} \right] \int_{-\infty}^{t_1} e^{\frac{\tau}{T_i}} \left[u(\tau) - u^r(\tau) \right] d\tau \;. \tag{64}$$

This means that, when using the PID controller, after the process leaves the limitation, w-w^r becomes an exponential decreasing function with a time constant equal to the integral time constant T_i (Peng et al., 1996b).

To illustrate the above derivations, an example was made using the process

$$G_{PR} = \frac{1}{(1+s)(1+0.5s)} \,, \tag{65}$$

and controller

$$K = 10,\ \ T_i = 1.4s,\ \ T_d = 0.257s,\ \ T_f = 0.1s, \qquad (66)$$
$$\beta = 1,\ \ \gamma = 1 .$$

The limit was U_{max}=2.

Figures 41 and 42 show the realisable reference and process output when using (i) the conditioning technique ($K_{ai}=K\cdot(\beta+\gamma\cdot T_d/T_f)$, $K_{ad}=-T_f\cdot K_{ai}$), (ii) smaller values of K_{ai} and K_{ad} ($K_{ai}=0.7\cdot K\cdot(\beta+\gamma\cdot T_d/T_f)$, $K_{ad}=-T_f\cdot K_{ai}$) and (iii) when using no anti-windup protection ($K_{ai}=K_{ad}=\infty$). The realisable reference approaches w, such as that given by expression (61), after the process leaves the limitation. Because the response is not oscillatory, we can conclude that time constants τ_1 and τ_2 are real. In fact, from (57) and (58), they can be calculated as τ_1=1s and τ_2=0.5s.

Figures 25 and 26 show an example for the PID controller (β=1, γ=0). It can be seen that after the system leaves saturation, the realisable reference approaches the actual reference (w=1) exponentially with the time constant equivalent to T_i (30s). The greater T_i is, the longer is the settling time of the realisable reference from the instant when the system comes out of saturation.

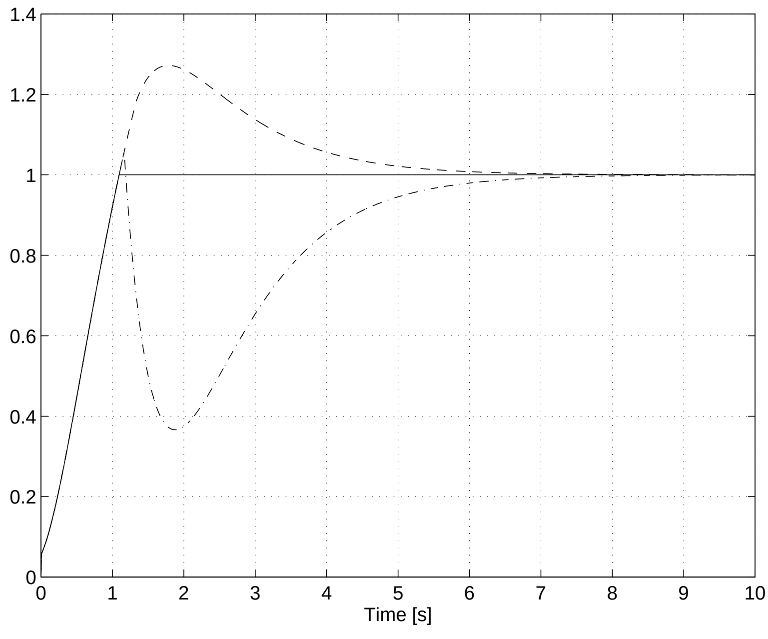

Fig. 41. Realisable reference (w^r);
__ Conditioning technique ($K_{ai}=K\cdot(\beta+\gamma\cdot T_d/T_f)$, $K_{ad}=-T_f\cdot K_{ai}$),
-.- ($K_{ai}= 0.7\cdot K\cdot(\beta+\gamma\cdot T_d/T_f)$, $K_{ad}=-T_f\cdot K_{ai}$), -- Without AW protection ($K_{ai}=K_{ad}=\infty$)

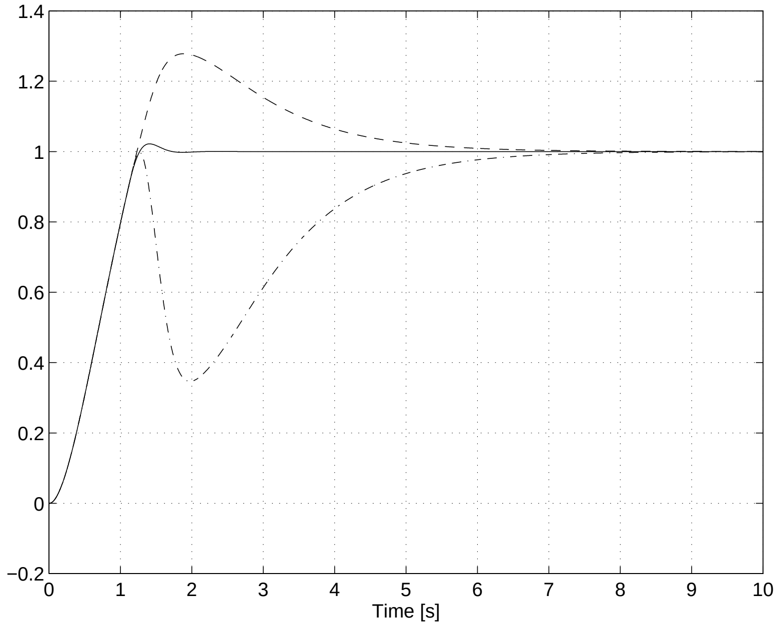

Fig. 42. Process output (y);
__ Conditioning technique ($K_{ai}=K\cdot(\beta+\gamma\cdot T_d/T_f)$, $K_{ad}=-T_f\cdot K_{ai}$),
-.- ($K_{ai}= 0.7\cdot K\cdot(\beta+\gamma\cdot T_d/T_f)$, $K_{ad}=-T_f\cdot K_{ai}$), -- Without AW protection ($K_{ai}=K_{ad}=\infty$)

From expressions (57), (58) and (61) it can also be seen that for the generalised PID controller, as for the PID controller, the main time constant is connected to controller time parameters, especially to T_i. Note that the controller time parameters are usually tuned so that their ratio is fixed (e.g. $T_d=T_i/4$, $T_f=T_d/10$, etc.). Therefore, a greater T_i gives a longer settling time of the realisable reference and consequently results in a longer process settling time. However, note that a smaller T_i can also produce larger overshoots (see Sub-chapter 3.3).

3.1.2 Calculation of K_{ai} and K_{ad}

One remaining question is how to find the appropriate parameters K_{ai} and K_{ad}. Originally, the controller goal was to make the process output (y) track the set-point (w), but y actually tracked the realisable reference w^r due to the system limitations. Thus, we would like to have that w^r which is as close as possible to w. We can tune w^r by changing the parameters K_{ai} and K_{ad}.

The realisable references, when using different values of K_{ai} and K_{ad}, separate from each other when the system is no longer saturated. The time responses of the realisable references when the system is no longer saturated are given by the expressions (61) and (63). According to the goal of having w^r as close as possible to w, constants C, C_1 and C_2 should be as small as possible. They become equal to zero when using the conditioning technique (see expressions (51), (60), (62) and (64)). This also means that the realisable reference w^r becomes and remains the same as the actual reference w at the instant the process leaves the saturation, so the system follows the reference w. Other anti-windup methods (values of K_{ai} and K_{ad}) produce a w^r that will not follow w after the system is no longer saturated, but will exponentially approach the reference w. Therefore, the windup effect will still be present in the system even when it is no longer saturated. Too small a value of K_{ai} can also produce problems, such as shown in Figures 25, 26, 41, and 42. Sometimes it can cause instability even when the process response - when not using any anti-windup protection - is stable (Vrančić et al., 1995b; Vrančić and Peng, 1996). On the other hand, by using the conditioning technique, the windup effect will vanish when the system is no longer saturated.

3.1.3 Discussion

Using the well-known definition of the realisable reference, it has been proved that the behaviour of the system, after it leaves saturation, depends strongly on the controller parameters, especially on the integral time constant (T_i). For the generalised PID controller, a smaller T_i gives a faster controller recovery after the system leaves saturation.

3.2 The extent of windup and the benefits of anti-windup

A variety of anti-windup (AW) methods is proposed in the literature (Hanus, 1989). Each method provides a somewhat different explanation of the cause of windup. Here we would like to show that the windup effect can be predicted and measured by calculating the integral of the difference between the limited and unlimited process time responses (Vrančić et al., 1996a; 1995b). This new observation can also be used to explain why the AW methods do work. Using this idea, we explain why the conditioning technique is usually the most suitable method from the large set of AW methods included in the observer approach.

3.2.1 Background material

The controller implementation in the state-space form is given Fig. 16 and expression (12).

The controller (12) can be represented by

$$\dot{x} = (A - gc)x + (b - gd)w - (e - gf)y + gu^r \ . \tag{67}$$

During the saturation, the controller is driven in an open-loop by u^r (either U_{max} or U_{min}). Thus, if the plant is open-loop stable, the sufficient condition for guaranteeing stability is that (*A-gc*) has no positive eigenvalue. Hence, the value of *g* can be freely tuned provided that this condition is met. Some guidelines for tuning are given in Åström and Rundqwist (1989), and Rundqwist (1990).

Expression (19) gives the value of the AW compensator when using the conditioning technique. In this case, ($A\text{-}gc\text{=}A\text{-}bcd^{-1}$) has the controller's zeros as its eigenvalues. Therefore, if the controller is inversely unstable, stability is not guaranteed by using the conditioning technique. In order to overcome this drawback, a modified conditioning technique was proposed in Hanus and Peng, (1991). It should also be pointed out that, in the SISO case, the controller is usually inversely stable, stability using the conditioning technique is thus guaranteed during saturation.

3.2.2 The extent of windup

During saturation, the process output changes more slowly than in the unlimited case. Due to this slower output, windup occurs and the process response has an enlarged overshoot and a long settling time. Therefore, ca a relationship between the limitation and the shape of the process output be found?

Fig. 43 shows typical control and process signals if the process is limited and the AW algorithm is running. Signal y_0 represents the process output of the unlimited system. Signals u_1, u_1^r and y_1 represent control output, process input and process output of the limited system, respectively. A_u denotes the area enclosed by u_1 and u_1^r, A_y denotes the area enclosed by y_0 and y_1, and A_w denotes the area enclosed by *w* and w^r.

In this case, we can compute the area A_y as

$$A_y = \int_0^\infty \left[y_0(t) - y_1(t) \right] dt \ . \tag{68}$$

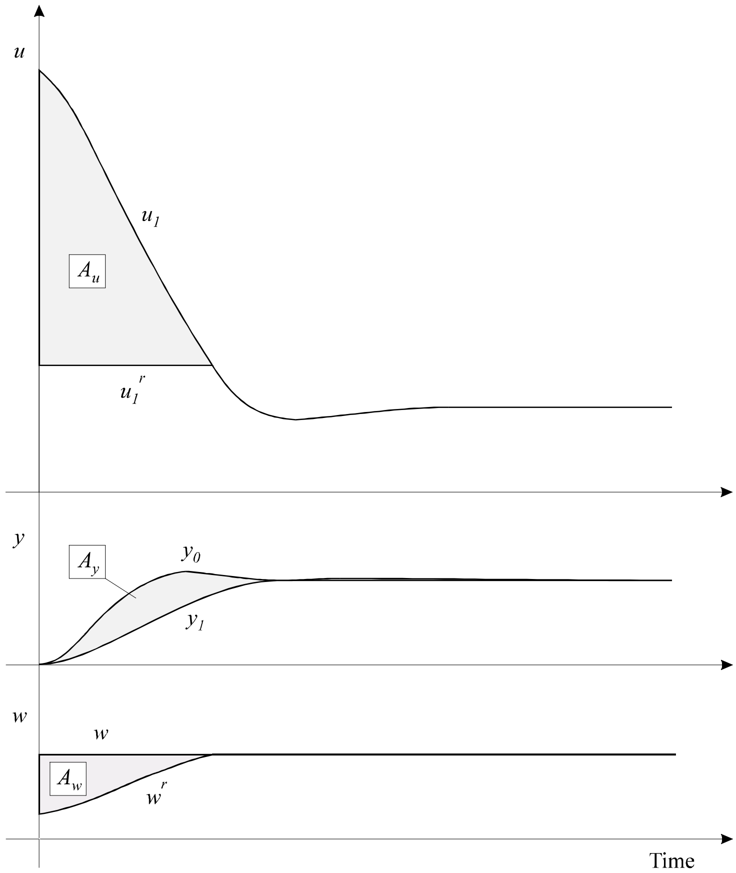

Fig. 43. Typical system time response;
u - control outputs, u^r - process inputs,
y - process outputs, w - references

As the integral (68) converges (the proof is given in Appendix A), it can be expressed in the Laplace form as

$$A_y = \lim_{s \to 0} \left(Y_0(s) - Y_1(s) \right) \; . \tag{69}$$

Similarly, we can deduce

$$A_u = \lim_{s \to 0} \left(U_1(s) - U_1{}^r(s) \right) \; , \tag{70}$$

and

$$A_w = \lim_{s\to 0}\left(W(s) - W^r(s)\right) \ . \tag{71}$$

Unlimited process output Y_0 can be expressed from Fig. 16 and (46b), when u^r=u, as

$$Y_0(s) = \frac{G_{PR}(s)G_{CW}(s)}{1+G_{PR}(s)G_{CY}(s)}W(s) = G_{CL}(s)W(s) \ , \tag{72}$$

where $G_{PR}(s)$ is the process transfer function, and $G_{CL}(s)$ is the closed-loop transfer function. Transfer functions $G_{CW}(s)$ and $G_{CY}(s)$ are the controller transfer functions from the reference (w) and from the process output (y) to the controller output (u). The limited system is not linear and cannot be expressed as (72), but it can be made linear by using the realisable reference w^r (see Fig. 29 and (48)).

The limited process output can therefore be expressed as:

$$Y_1(s) = G_{CL}(s)W^r(s) \ . \tag{73}$$

The property of the closed-loop transfer function is

$$\lim_{s\to 0} G_{CL}(s) = \xi \ . \tag{74}$$

Note that by using a controller with an integral action (no steady-state error), ξ=1.

Substituting (72), (73) and (74) into (69) gives

$$A_y = \xi \lim_{s\to 0}\left[W(s) - W^r(s)\right] = \xi A_w \ . \tag{75}$$

Substituting (48) into (75) gives

$$A_y = \xi A_w = \xi \lim_{s\to 0}\left[\frac{c(sI-A)^{-1}g+1}{c(sI-A)^{-1}b+d}\left(U_1(s) - U_1^{\ r}(s)\right)\right] = \xi\frac{c\cdot adj(A)g}{c\cdot adj(A)b}A_u \ . \tag{76}$$

If the AW compensator is not used (g=0), (76) simplifies to

$$A_y = A_w = 0 \ . \tag{77}$$

Fig. 44 shows the case where no AW protection is used. The area A_y (A_w) is divided into two parts, namely A_{y1} (A_{w1}) and A_{y2} (A_{w2}). The area A_{y1} denotes the integral between y_0 and y_1, when $y_0 > y_1$ and the area A_{y2} denotes the integral between y_0 and y_1, when $y_0 < y_1$. Note that A_{y2} has the opposite sign to A_{y1}. From (77) we see that A_{y1} and $-A_{y2}$ must be the same.

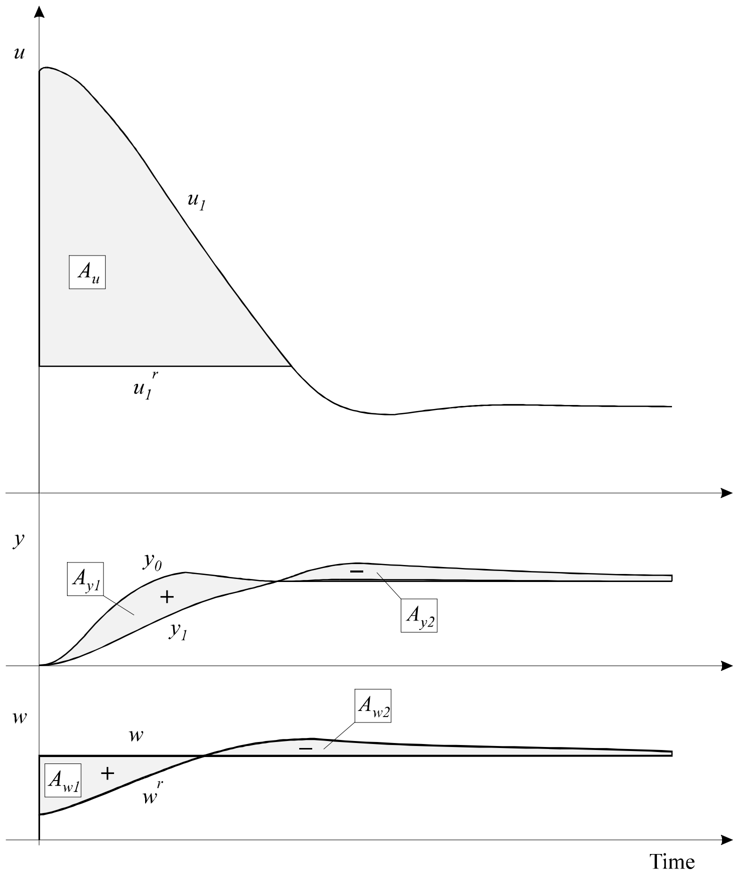

Fig. 44. System time response when no AW protection is used;
u - control outputs, u^r - process inputs,
y - process outputs, w - references

Assume that the system is limited from t=0s to t=t_1:

$$\begin{matrix} u^r < u \ ; & t < t_1 \\ u^r = u \ ; & t \geq t_1 \end{matrix} \ ; \ t_1 < \infty \ . \tag{78}$$

Note that the time t_1 depends on the value of the compensator g.

The realisable reference can be expressed from (47) as

$$W^r = \frac{U^r + \left[c(sI - A)^{-1} e + f\right] Y}{c(sI - A)^{-1} b + d} \quad . \tag{79}$$

During the time interval when the system is saturated ($u^r < u$), the time responses at the process input (u^r) are equal, no matter what AW compensator g is used. The same applies to the process output (y), if the initial conditions are the same. So, during $u^r < u$, the realisable reference, when not using any AW protection ($g=0$), equals that when using the conditioning technique ($g=bd^{-1}$).

According to expression (49), a greater difference between u and u^r (stronger limitation) results in a greater difference between w and w^r (greater area A_{w1}). This results in a higher $-A_{w2}$, if no AW protection is used.

The process output (y) follows the realisable reference (w^r), so a larger difference between w and w^r will, generally speaking, result in a larger difference between y_0 and y_1 and consequently in a larger A_{y1} (larger $-A_{y2}$). Hence, stronger process limitation results in a larger overshoot and/or a longer settling time of the process.

The process settling time also depends on the controller parameters. For the PID controller, settling time is strongly related to the integral time constant. The greater the integral time constant, the longer the process settling time (Peng et al., 1996b). This means that for the same area $-A_{y2}$ ($-A_{w2}$), the controller with the larger integral time constant will have a smaller overshoot and a longer settling time and vice versa, which explains why, even without any AW protection, some controllers do not produce a significant overshoot when the system saturates (which is often thought to be a synonym of windup).

From the above, it can be seen that the extent of the windup depends on the size of A_{y1}, the controller parameters, and the AW strategy used. Note that (77) does not necessarily imply windup, since it also holds in the unlimited case ($A_u=0$). In that case, A_{y1} is always zero, and vector g has no influence.

3.2.3 The benefits of anti-windup compensation

The aim of AW compensation is to reduce the area $-A_{w2}$ ($-A_{y2}$) without any significant increase of A_{w1} (A_{y1}). Actuator limitations will then be “transformed” into smaller process overshoots, and into shorter settling times. This can be done by changing vector g in order to sufficiently increase A_w (A_y) (see (76)). If A_w (A_y) are excessively increased, A_{w2} (A_{y2}) will equal zero and A_{w1} (A_{y1}) will become equal to A_w (A_y). This however

causes another drawback - a large undershoot and long settling time (see the dash-dotted lines in Figs. 30 and 31).
Applying (78) to (49), it can be stated that when using the conditioning technique,

$$\begin{aligned} w^r &< w\,; \quad t < t_1 \\ w^r &= w\,; \quad t \ge t_1 \end{aligned} . \tag{80}$$

As stated before, the realisable reference is the same as when no AW protection is used during $t<t_1$. So, the surfaces A_{w1} are therefore the same in both cases. It is obvious that the surface A_{w2} equals zero when using the conditioning technique (80). This is clarified by the solid line in Fig. 30. Therefore, the conditioning technique reduces the area $-A_{w2}$ to zero, whilst the area A_{w1} stays unchanged. The relationship between A_y, A_w and A_u for the conditioning technique (19) is:

$$A_y = \xi A_w = \xi \frac{A_u}{d} \ . \tag{81}$$

However, when the controller gain is too high and the system limitations are too restrictive, multiple opposite limitations may appear in the system (Morari, 1993). In such cases, the controller parameters should be tuned so as to decrease the oscillations, if disturbance rejection is not of particular importance. Another approach would be changing the controller or AW compensator structure or using the variable AW compensator (Peng et al., 1996a).

3.2.4 Discussion

It was outlined that the extent of the windup effect depends on the initial difference between the unlimited and limited process responses. The area enclosed by the unlimited and limited process responses equals zero when no protection against windup is used. The greater the initial difference between the two responses, the stronger the windup effect (a larger overshoot and/or longer process settling times). The results of various experiments performed on many process models and laboratory set-ups support these results.

The aim of the AW system is to sufficiently increase the area enclosed by the process responses. The conditioning technique gives the best compromise for normal applications by reducing the area $-A_{w2}$ to zero.

3.3 The overshoot of the realisable reference for PID controllers

Fig. 45 shows the reference and the realisable reference for the typical case where no AW protection is applied to the PID controller.

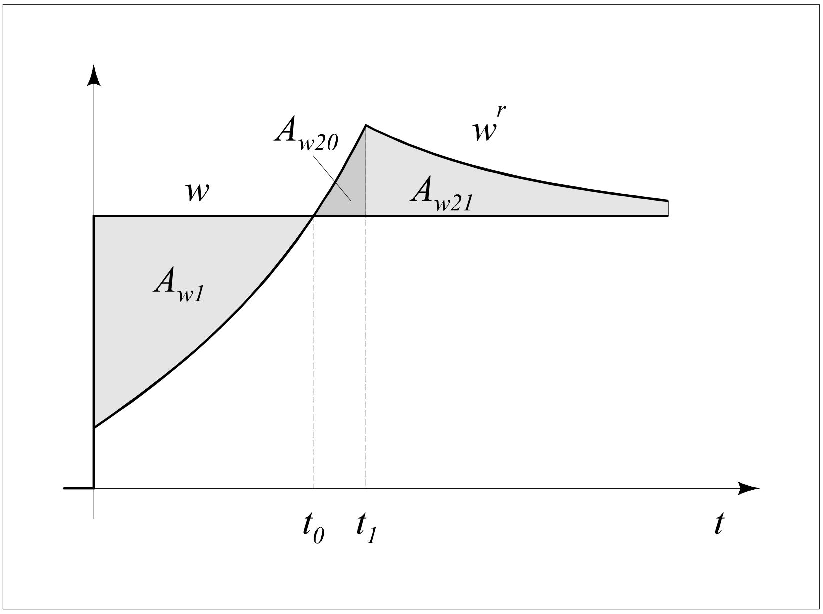

Fig. 45. The reference (w) and the realisable reference (w^r) when no AW protection is applied to the PID controller

According to expression (77), it holds:

$$\left|A_{w20} + A_{w21}\right| = \left|A_{w1}\right| \ , \tag{82}$$

where

$$\begin{aligned} A_{w20} &= \int_{t_0}^{t_1} \left(w(\tau) - w^r(\tau)\right) d\tau \ < \ 0 \\ A_{w21} &= \int_{t_1}^{\infty} \left(w(\tau) - w^r(\tau)\right) d\tau \ < \ 0 \end{aligned} \ . \tag{83}$$

After $t>t_1$, the realisable reference w^r approaches the real reference w exponentially with the integral time constant T_i, according to (63):

$$w(t) - w^r(t) = \left[w(t_1) - w^r(t_1)\right] e^{-\frac{t-t_1}{T_i}} \ ; \ t \geq t_1 \tag{84}$$

the surface A_{w21} can be calculated by inserting (84) into (83),

$$A_{w21} = \left[w(t_1) - w^r(t_1)\right] T_i \ . \tag{85}$$

Due to (82)

$$\left|A_{w21}\right| \leq \left|A_{w1}\right| \ , \tag{86}$$

the overshoot of the realisable reference can be expressed by the following inequality:

$$\left|w(t_1) - w^r(t_1)\right| \leq \left|\frac{A_{w1}}{T_i}\right| \tag{87}$$

As the surface A_{w1} is already known at $t=t_0$, the maximum possible overshoot $|w(t_1)-w^r(t_1)|$ can be predicted in advance. The expected overshoot of w^r will be higher for a smaller integral time parameter T_i, for the same A_{w1}, as follows from (87). However, the settling time of w^r will be shorter.

An experiment was conducted in order to depict the above results. Four different process models were chosen so that the time response of the realisable reference was the same, during the system saturation, for four different integral time constants of the PI controller. The chosen processes and integral time constants are given below.

Process	T_i
$G_{PR1}(s)=\dfrac{(1+1.6514s)(1-0.1514s)}{(1+2s)(1+s)(1+0.5s)}$	2s
$G_{PR2}(s)=\dfrac{1}{(1+s)(1+0.5s)}$	1s
$G_{PR3}(s)=\dfrac{(1+0.25s)}{(1+s)(1+0.5s)}$	0.5s
$G_{PR4}(s)=\dfrac{(1+(0.3+0.3317i)s)(1+(0.3-0.3317i)s)}{(1+s)(1+0.5s)(1+0.2s)}$	0.2s

The gain of the PI controller was the same for all the processes: K=2. The process limitation was U_{max}=1.2. The realisable references for all four cases are shown in Fig. 46. It can be seen that the settling time of the realisable reference is higher for higher values of the integral time constant, while just the opposite holds for the overshoot of the realisable reference, according to expression (87).

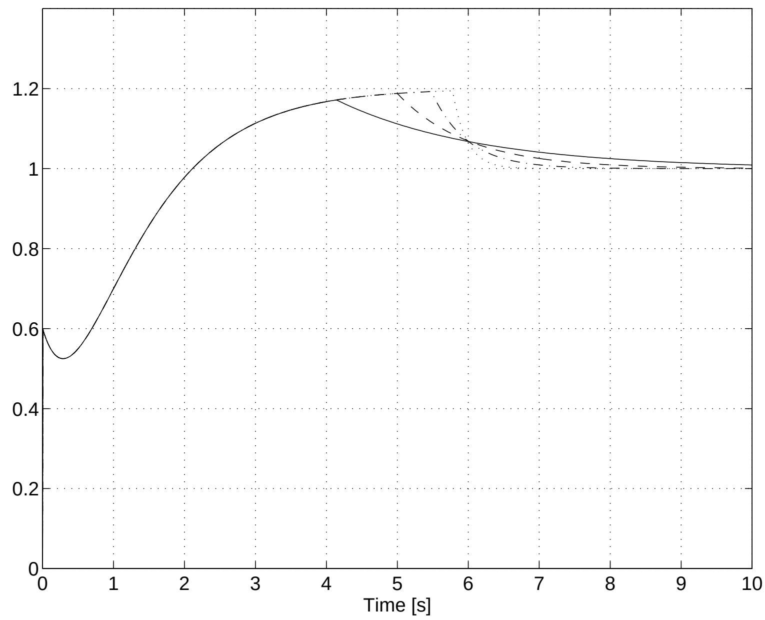

Fig. 46. The realisable references; __ $G_{PR1}(s)$ and T_i=2, -- $G_{PR2}(s)$ and T_i=1, -.- $G_{PR3}(s)$ and T_i=0.5, ... $G_{PR4}(s)$ and T_i=0.2,

3.4 Disturbance rejection and anti-windup

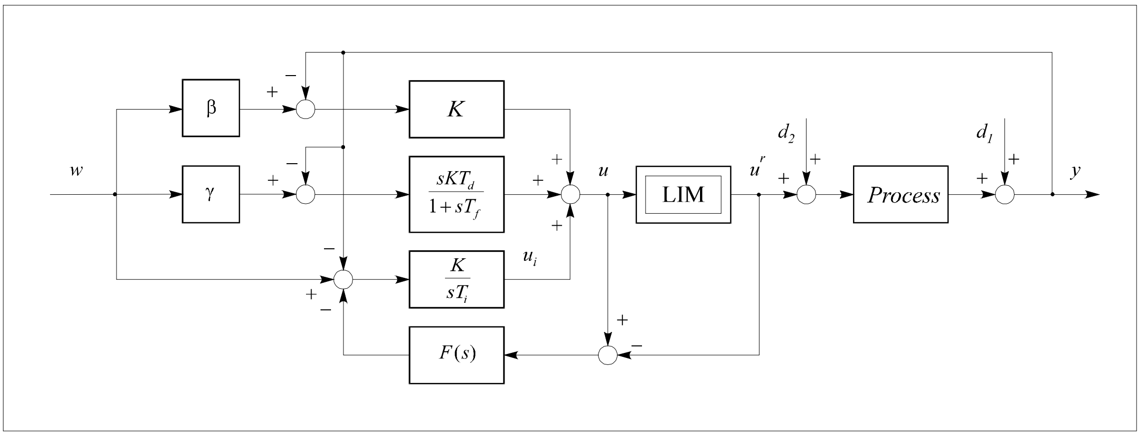

Fig. 47. Anti-windup compensation with disturbances

What happens if the limitation is caused by a disturbance instead of a reference change? Fig. 47 shows two typical disturbances: d_1, and d_2. Disturbance d_1 is called *process disturbance*. It acts directly on the measured process signal, whilst the influence of disturbance d_2 (*load disturbance*) depends on the process dynamics. Usually the process response to the disturbance d_2 is smooth and does not cause the saturation of u (Vrančić et al., 1995b). Therefore, we will focus on the process disturbance d_1.

Let us try an experiment with the following third-order process:

$$G_{PR}(s) = \frac{1}{(1+s)(1+0.2s)} \tag{88}$$

and the generalised PID controller:

$$K = 10,\ T_i = 0.5s,\ T_d = 0.05s,\ T_f = 0.005s,\ \beta = 0.7,\ \gamma = 0\ . \tag{89}$$

The process limitations are:

$$U_{\max} = 1.4,\ U_{\min} = 0\ . \tag{90}$$

Figs. 48 to 50 show the system responses when changing the reference from w=0 to w=1 at the time origin and then adding the step-change disturbance d_l, with amplitude d_l=-0.3 at t=5s. We chose two different anti-windup compensators. The first was the conditioning technique (14) and the second was

$$F(s) = \frac{T_i}{K\sqrt{T_i T_d}} \ , \tag{91}$$

as proposed by Åström and Rundqwist (1989), Rundqwist (1990), and Rundqwist (1991), as being the most appropriate anti-windup protection for process disturbances. In the following text, the compensator (91) will be denoted as "disturbance compensator".

From the results of experiments (see Figs. 48 and 49), it is clear that the conditioning technique gave considerable good system responses to the reference change as well as to disturbance rejection, whilst the disturbance compensator did not give as good responses even to disturbance rejection.

The question is what is the process response when disturbance occurs? Firstly, it is known that, whilst the process is not limited, the realisable reference (w^r) equals the actual one (w). Secondly, the limited system always acts as it follows the realisable reference instead of the actual one. By assuming that the superposition theorem holds, the limited process response to disturbance is equal to the sum of the unlimited process response to disturbance and the unlimited process response to the (changed) realisable reference (which differs from w when limitation occurs).

When using the conditioning technique, the realisable reference approaches the real one (w) proportionally to the difference between u and u^r. From the moment when the system is no longer saturated, the controller follows the real reference.

If the controller is well tuned for reference tracking and for disturbance attenuation, then the process response, when using the conditioning technique, should give quite a reasonable disturbance attenuation as shown in the last example.

The exception could prove to be when a very small factor β is used. In such a case, the AW compensator becomes relatively strong and desaturates the integral term too quickly during the time limitation occurs due to disturbance. This results in poorer disturbance rejection under saturation.

We made another experiment using the same process (88) and limitation (90), but with different controller parameters:

$$K = 15,\ T_i = 0.25s,\ T_d = 0.07s,\ T_f = 0.007s,\ \beta = 0.2,\ \gamma = 0 \tag{92}$$

Figs. 51 to 53 show the system responses. It is obvious that a disturbance compensator gives better disturbance rejection. Response to the reference change is, however, better when using the conditioning technique.

The situation changes when increasing factor γ. Figs. 54 to 56 show the results of experiments when using γ=0.5 in the same controller (92). It is again obvious that both responses are better when using the conditioning technique.

From the above experiments, we can conclude that the conditioning technique is usually quite an appropriate choice for anti-windup protection. However, for smaller values of factor β, disturbance response may not be so optimal. In such a case, as well as when disturbances are expected to cause frequent saturations, a disturbance compensator could be a more appropriate choice. However, a disturbance compensator (91) can also become too strong if T_d is considerably smaller than T_i. As a result, the general approach for such cases can only be to perform testings on the plant or to make simulations on a computer where the plant can be explicitly identified.

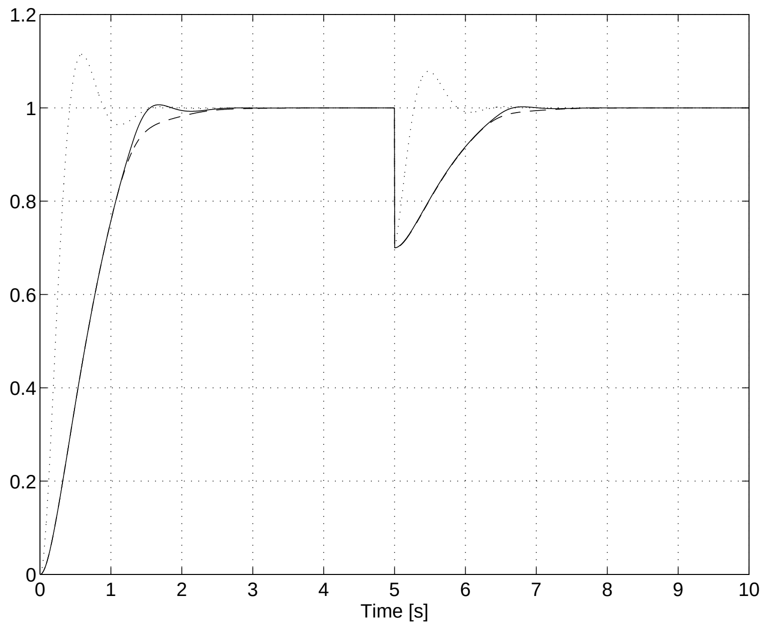

Fig. 48. Process output (y); __ limited response when using the conditioning technique, -- limited response when using disturbance compensator, ... unlimited response

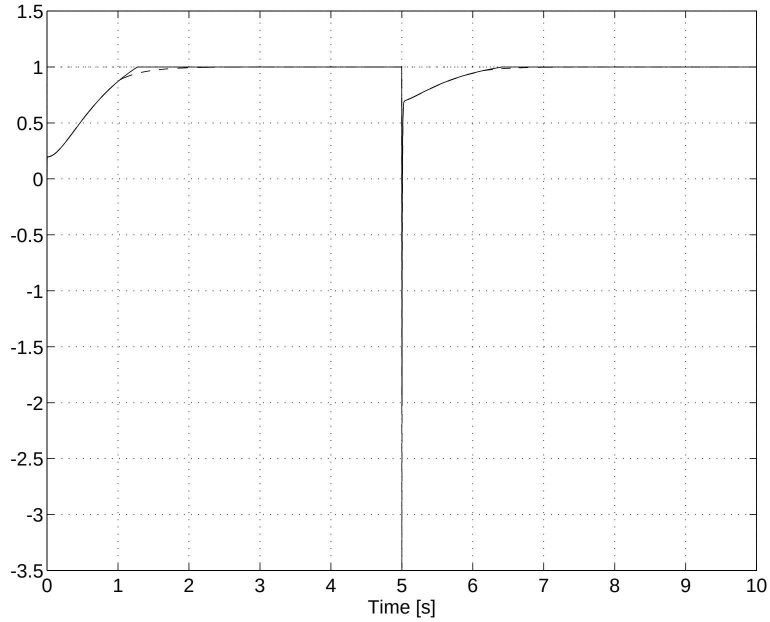

Fig. 49. Realisable reference (w^r); __ limited response when using the conditioning technique, -- limited response when using disturbance compensator, ... unlimited response

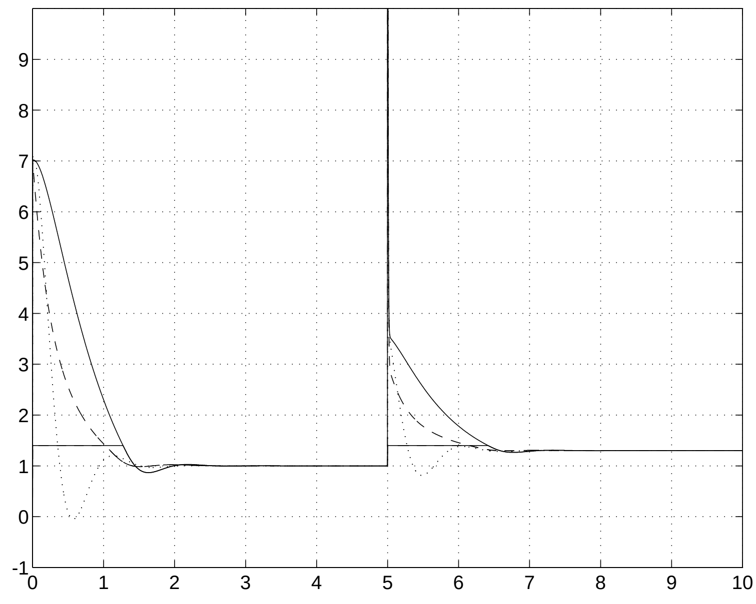

Fig. 50. Controller output (u) and process input (u^r); __ limited response when using the conditioning technique, -- limited response when using disturbance compensator, ... unlimited response

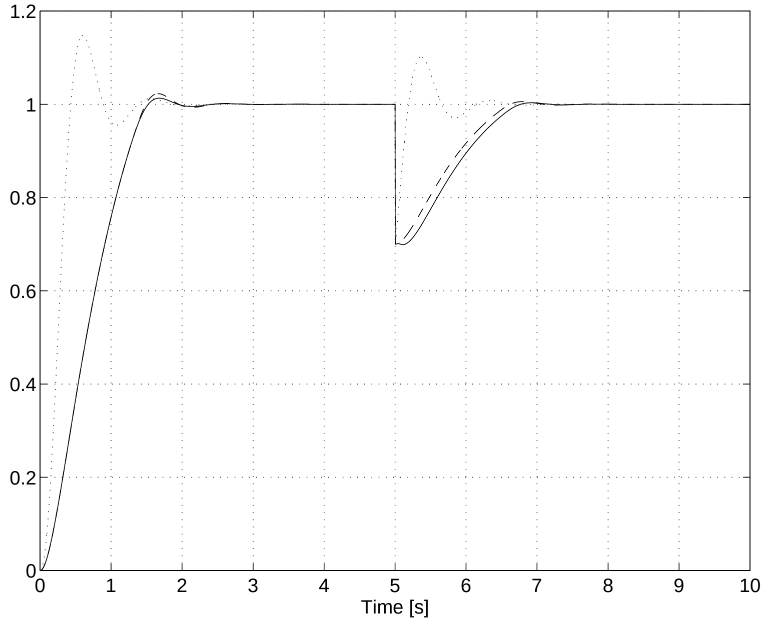

Fig. 51. Process output (y); __ limited response when using the conditioning technique, -- limited response when using disturbance compensator, ... unlimited response

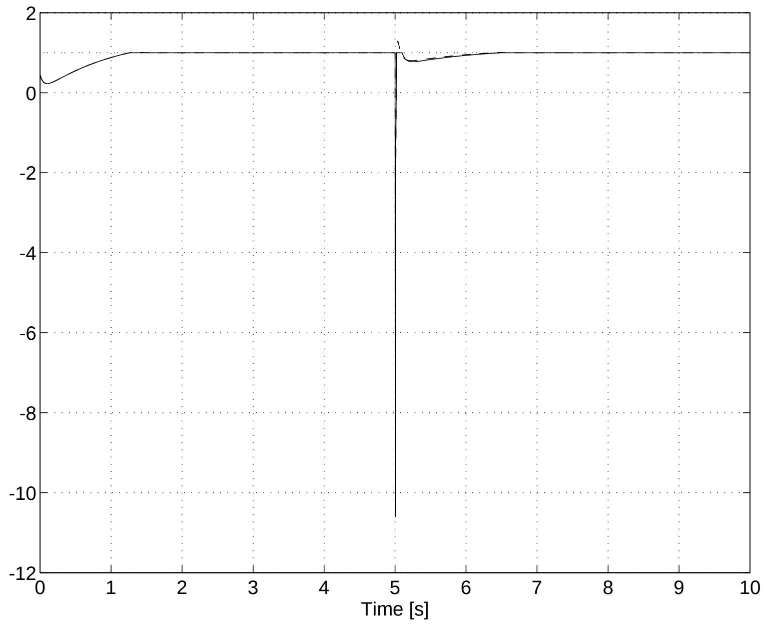

Fig. 52. Realisable reference (w^r); __ limited response when using the conditioning technique, -- limited response when using disturbance compensator, ... unlimited response

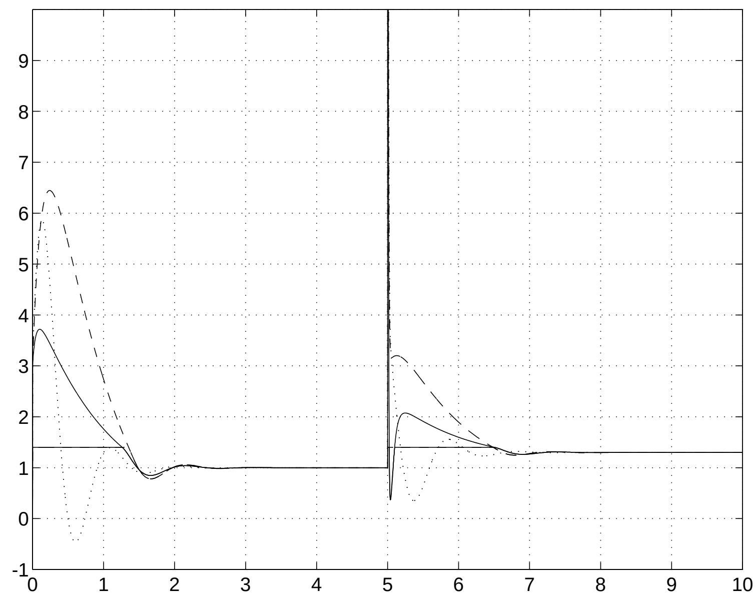

Fig. 53. Controller output (u) and process input (u^r);
__ limited response when using the conditioning technique,
-- limited response when using disturbance compensator, ... unlimited response

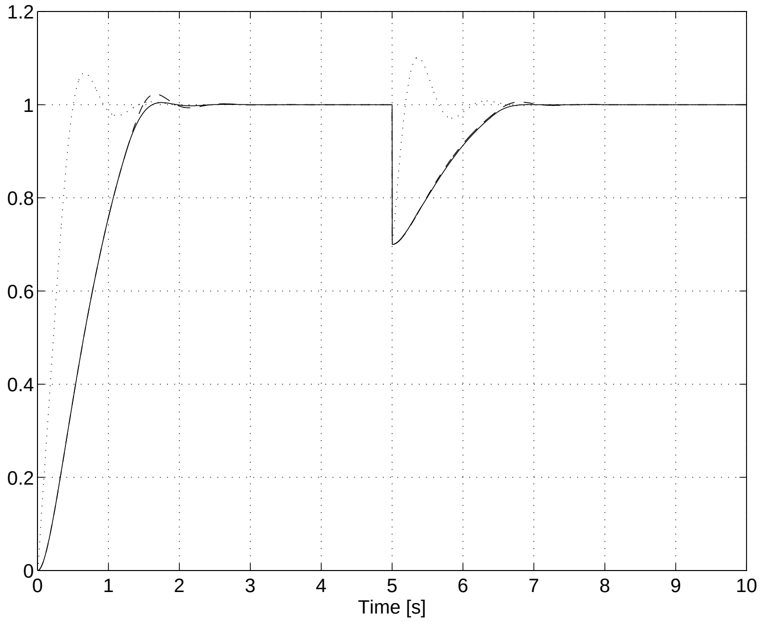

Fig. 54. Process output (y); __ limited response when using the conditioning technique,
-- limited response when using disturbance compensator, ... unlimited response

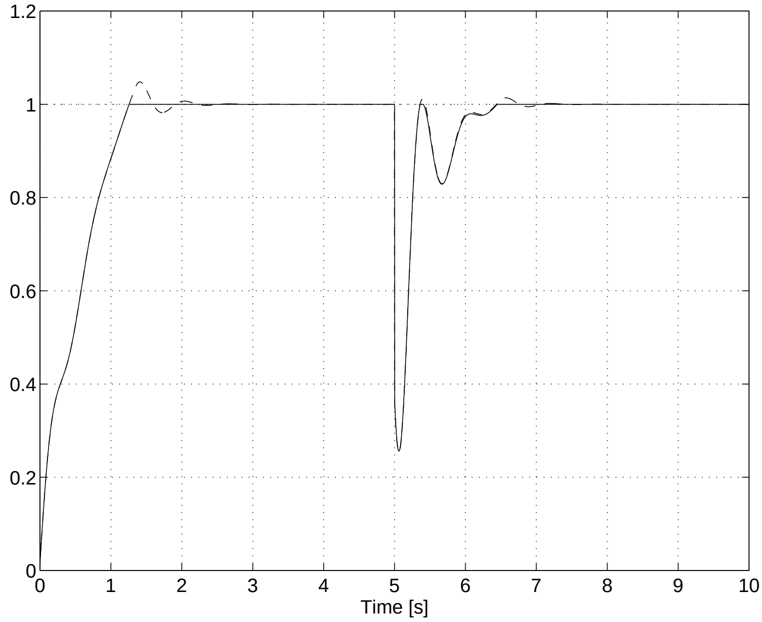

Fig. 55. Realisable reference (w^r); __ limited response when using the conditioning technique, -- limited response when using disturbance compensator, ... unlimited response

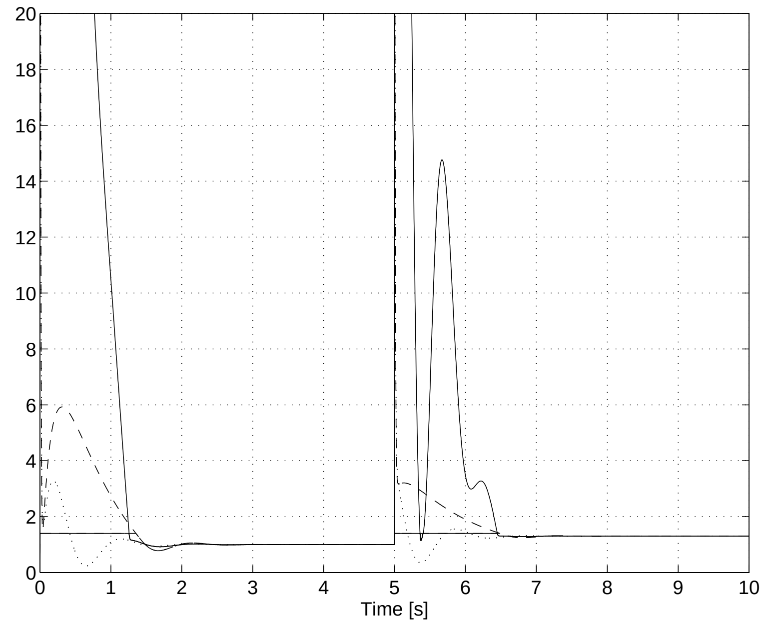

Fig. 56. Controller output (u) and process input (u^r); __ limited response when using the conditioning technique, -- limited response when using disturbance compensator, ... unlimited response

3.5 Noise and anti-windup

The problem of noise in saturated systems was noted by Khandheria and Luyben as early as 1976, but as far as we know, it has since not been closely elaborated by any other author. Here we would like to show which anti-windup techniques are more and which are less sensitive to process noise.

Fig. 57 depicts the anti-windup system when process noise is present *during saturation*. Signal ε denotes the process noise signal (white noise, coloured noise, etc.). Note that u^r is the limited signal and therefore no additional noise is added to this signal. It is expected that the anti-windup algorithm will not be too sensitive to the process noise, meaning that the controller states should not be seriously affected by the presence of noise *during saturation* (Vrančić et al., 1995b). For the sake of simplicity, the PI controller will be investigated (Fig. 57) in our further elaboration, where the only state of the controller is the output of the integral term.

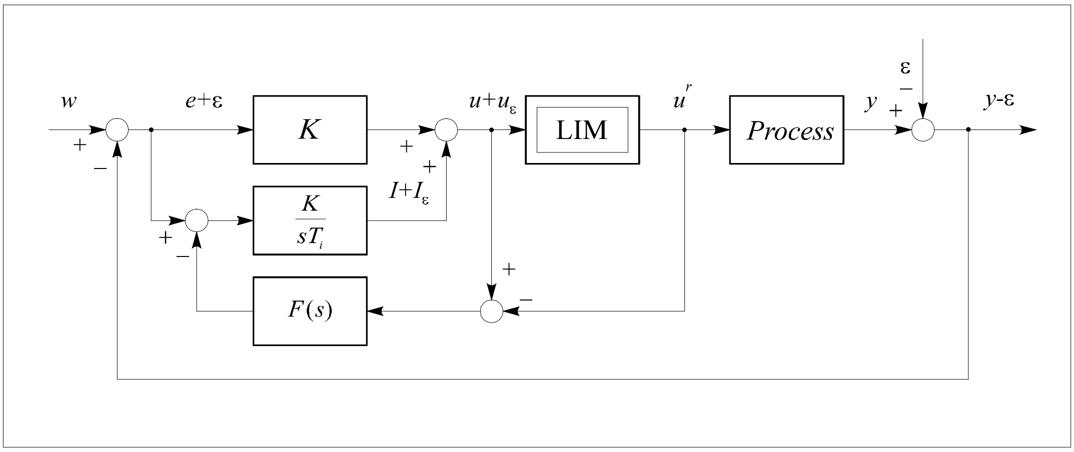

Fig. 57. The limited closed-loop system with noise (ε) during saturation

The noise (ε) at the process output produces a change at the output of the integral term (I_ε):

$$I_\varepsilon = \frac{K}{sT_i}\left[\varepsilon - u_\varepsilon F(s)\right] = \frac{1 - KF(s)}{F(s) + s\frac{T_i}{K}}\varepsilon = G_\varepsilon(s)\varepsilon \ , \tag{93}$$

where

$$G_\varepsilon(s) = \frac{1 - KF(s)}{F(s) + s\frac{T_i}{K}} \tag{94}$$

A dynamic transfer function $G_\varepsilon(s)$ depends on the chosen anti-windup compensator $F(s)$:

$$G_\varepsilon(s) = \begin{cases} \frac{K}{sT_i} & ;\ F(s) = 0 \\ 0 & ;\ F(s) = \frac{1}{K} \\ -K & ;\ F(s) \to \infty \end{cases} . \tag{95}$$

Let us now calculate the power spectrum of the noise at the output of the integral term (the controller state):

$$\Phi_{I_\varepsilon}(\omega) = \left|I_\varepsilon(j\omega)\right|^2 = \frac{\left|1 - KF(j\omega)\right|^2}{\left|F(j\omega) + j\omega\frac{T_i}{K}\right|^2}\Phi_\varepsilon(\omega)\ , \tag{96}$$

where $\Phi_\varepsilon(\omega)$ denotes the power spectrum of the noise signal ε.

Different power spectrums of the noise of the controller state depend on the chosen anti-windup compensator $F(s)$:

$$\Phi_{I_\varepsilon}(\omega) = \begin{cases} \frac{K^2}{\omega^2 T_i^2}\Phi_\varepsilon(\omega) & ;F(s) = 0 \\ 0 & ;F(s) = \frac{1}{K} \\ K^2\Phi_\varepsilon(\omega) & ;F(s) \to \infty \end{cases} \tag{97}$$

It is obvious that the controller state, when using the conditioning technique ($F(s)=1/K$), is the least sensitive to the process noise during system saturation. If no protection against windup is used ($F(s)=0$), the system is especially sensitive to low frequency noises (97) (windup effect) and when using a strong anti-windup compensator

($F(s)\to\infty$) (the analog counterpart of a discrete incremental algorithm), the noise power is transferred directly to the controller state through the gain K^2.

In order to depict the above sentences, an example is shown using process:

$$G(s)=\frac{1}{(1+s)^2}\ , \tag{98}$$

and the following PI controller:

$$K=2,\ \ T_i=1.2s\ . \tag{99}$$

The process limitations were U_{max}=1.2 and U_{min}=-1.2. The white noise with the unit standard deviation (generated in a programme package SIMULINK with the function *rand* and the sampling time T_S=0.01s.), filtered by the first-order low-pass filter

$$G_F(s)=\frac{1}{1+0.1s} \tag{100}$$

was added to the process output (see Fig. 57) as the signal ε. The sampling time used in the incremental algorithm was T_S=0.005s.

The results are shown in Figures 58 to 60. Fig. 58 shows the process output of the limited system when using different anti-windup algorithms. It can be seen that the incremental algorithm is very sensitive to system noise. The reference point cannot be reached due to periodic limitations of the controller output signal. The realisable reference is shown in Fig. 59. It is obvious that the conditioning technique gives quite a good response, whilst the incremental algorithm is considerably sensitive to periodic limitations. The controller output and the process input signals are given in Fig. 60.

Similar observations to those of the PI controller can also be made for the generalised PID controller. We prepared example with process:

$$G(s)=\frac{1}{(1+s)^3} \tag{101}$$

and the following generalised PID controller:

$$K = 4,\;\; T_i = 1.5s,\;\; T_d = 0.5s,\;\; T_f = 0.05s,\;\; \beta = 0.2,\;\; \gamma = 0 \tag{102}$$

The process limitations were U_{max}=1.5 and U_{min}=-1.5. The same white noise as in the previous example was filtered by the first-order low-pass filter

$$G_F(s) = \frac{0.2}{1 + 0.1s} \; , \tag{103}$$

and added to the process output. The sampling time for the incremental algorithm was the same as for the previous example (T_S=0.005s).

The experimental results are shown in Figures 61 to 63. Similar conclusions can be drawn as for the previous example. The incremental algorithm is again the most sensitive, whilst the conditioning technique was quite inert to the process noise.

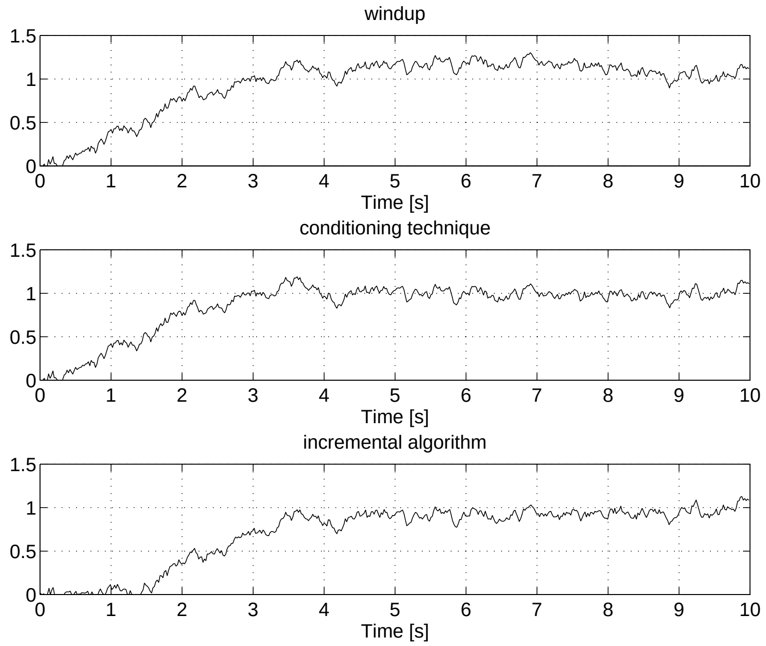

Fig. 58. The process output (y) when applying the PI controller

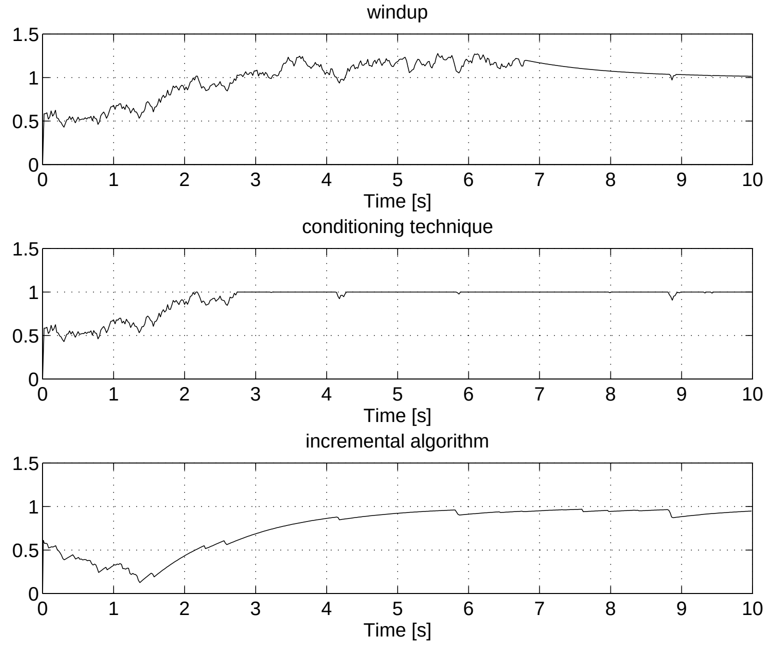

Fig. 59. Realisable reference (w^r) when applying the PI controller

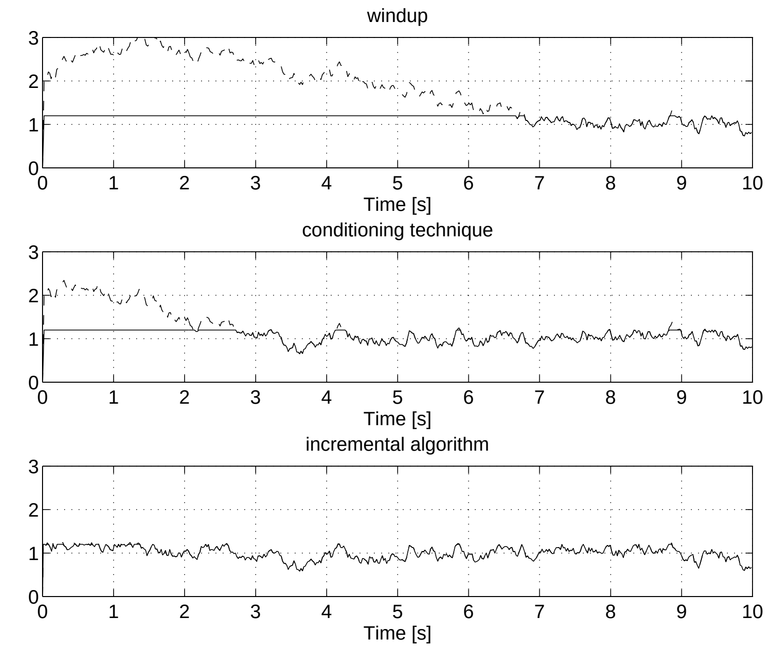

Fig. 60. The controller output and the process input when applying the PI controller; -- controller output (u), __ process input (u^r)

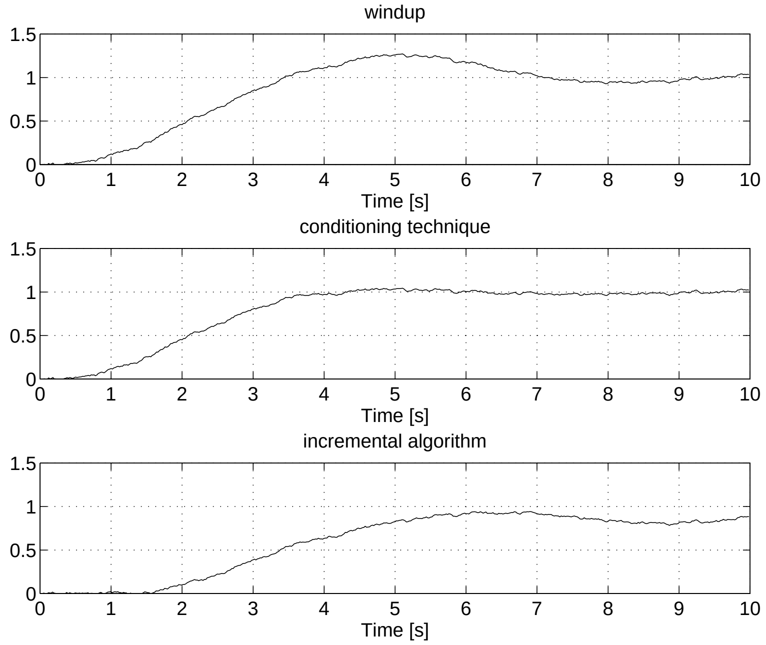

Fig. 61. The process output (y) when applying the generalised PID controller

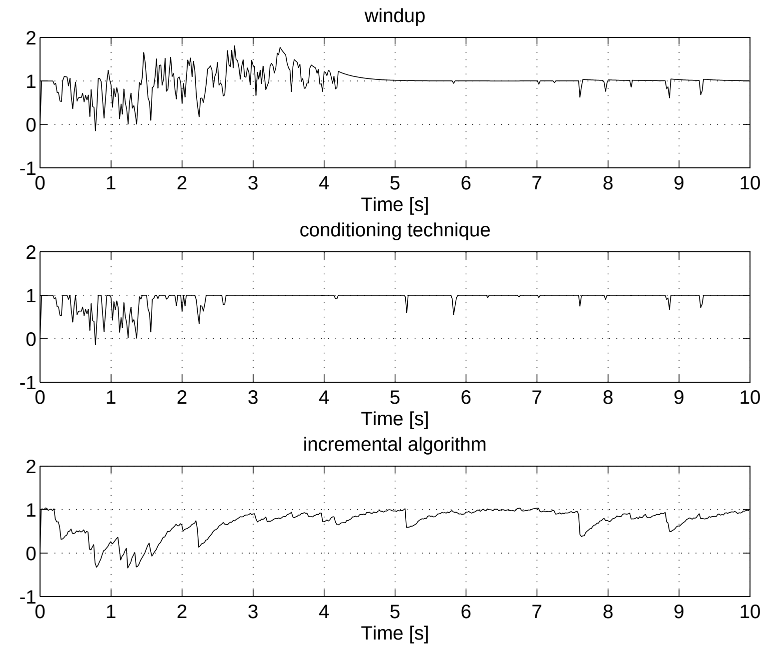

Fig. 62. The realisable reference (w^r) when applying the generalised PID controller

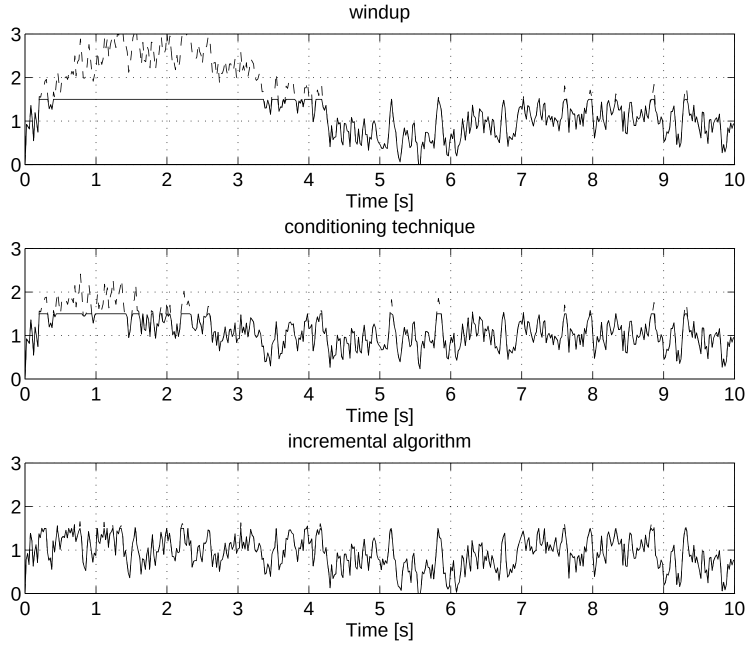

Fig. 63. The controller output and the process input when applying the generalised PID controller; -- controller output (u), __ process input (u^r)

4. *Anti-windup for multivariable controllers*

The topic of AW design has been studied for some four decades by many authors. However, anti-windup protection for multivariable systems has received more attention only in the last decade. The most popular AW techniques for such controllers are described in Hanus (1989) and Kothare et al. (1994), and the references contained therein. In Kothare et al. (1994), a general framework for AW design is presented in which all known AW compensation schemes arise as special cases.

Recently, some guidelines for designing AW, BLT and CT compensation for PID controllers have been provided (Peng et al., 1996a). Subsequent investigation has revealed that the conclusions can be extended to more general controllers. As a result, a key objective in this chapter is to provide a simple parameterisation for AW designs. Since the proposed parameterisation is based on the classical feedback control scheme, it has some advantages over the framework proposed by Kothare et al. (1994). First, the AW design can be explained in a natural and straightforward manner. Second, different existing AW designs can be easily compared, and their advantages and disadvantages readily understood.

One problem associated with saturation in multivariable (multi-input, multi-output, or MIMO) systems which has no scalar (single-input, single-output, or SISO) counterpart is that of control input vector directionality. This problem is addressed by means of an optimal design of an artificial nonlinearity (AN) (Peng et al., 1997).

4.1 Parameterisation of anti-windup designs

4.1.1 Windup problems and anti-windup compensation

Let us consider the unity feedback linear control system shown in Fig. 64, where the plant is described by a $m \times m$ linear transfer matrix $P(s)$, and a linear controller with $m \times m$ transfer matrix $K(s)$ was designed in order to meet certain performance specifications. Including the effect of input limitations leads to the system shown in Fig. 65, in which the block N is included to model the effects of input nonlinearities, where a distinction is made between the real plant input u^r and the expected plant input u, which is also the output of the linear controller $K(s)$.

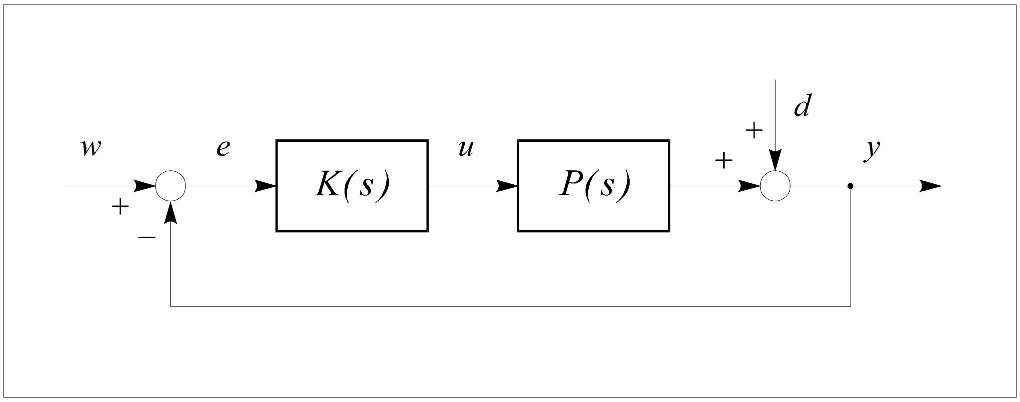

Fig. 64. Unity feedback control system

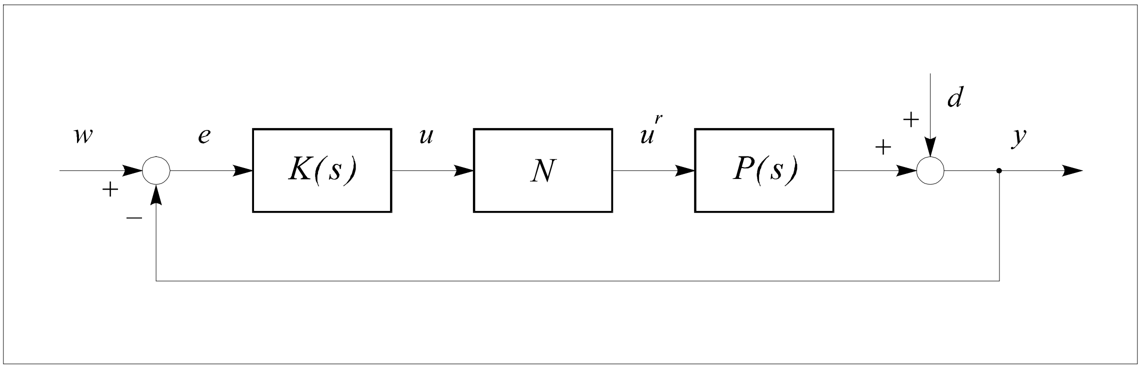

Fig. 65. Constrained unity feedback control system

If a dynamic controller is updated without using any information from the real plant input u^r as shown in Fig. 65, the controller states will be incorrectly updated when u^r differs from u, and this lack of consistency is referred to as "controller windup". Since the design of $K(s)$ ignores actuator nonlinearities, controller windup can result in significant performance degradation with respect to the expected linear performance.

In contrast, if the internal states of the dynamic controller $K(s)$ are updated using knowledge of u^r when $u^r \neq u$, the controller implementation is said to incorporate AW compensation. It should be pointed out, however, that the mere presence of AW compensation is not sufficient to eliminate the degradation of the closed-loop performance (Vrančić and Peng, 1995b, 1996). Our objective is, therefore, to investigate which AW compensation should be used in different cases. Appropriate AW design should lead to a graceful degradation with respect to the expected linear performance when the control input enters saturation.

A natural way of using knowledge of u^r is to feed (u- u^r) back to the controller through a linear filter with an appropriate transfer matrix $F(s)$ as shown in Fig. 66. As the AW compensation is achieved with a linear filter, AW designs of this type are referred to as linear AW compensators (Peng et al., 1996a,b). It is not a straightforward task to derive the conditions for controller implementability and closed-loop stability for the scheme in Fig. 66. Another way to parametrise any linear AW compensator for unity feedback controller $K(s)$ is shown in Fig. 67.

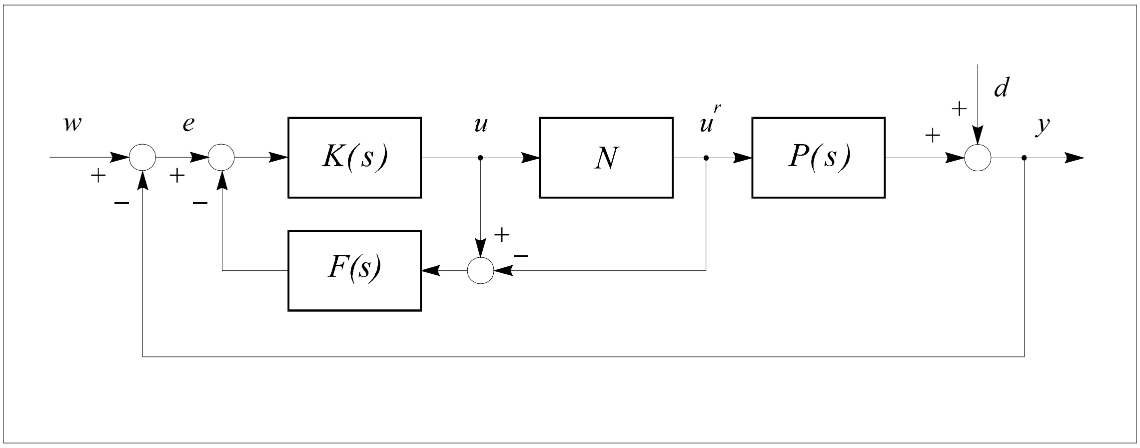

Fig. 66. Constrained unity feedback control system with linear AW compensator

Indeed, by block manipulation, it can be easily shown that the schemes represented in Fig. 66 and Fig. 67 are equivalent if and only if

$$K_1(s) = \left(I + K(s)F(s)\right)^{-1} K(s) \qquad (104)$$

$$K_2(s) = -\left(I + K(s)F(s)\right)^{-1} K(s)F(s) \tag{105}$$

Another possibility for modifying the AW scheme presented in Fig. 67 is shown in Fig. 68, where *AN* is an artificial nonlinearity chosen so that the output *v* never makes the nonlinearity *N* active, i.e. $u^r = v$ for all time. In practice, the block *AN* can only be added when block *N* is known a priori. The scheme in Fig. 4 can be therefore considered as a special case of the scheme shown in Fig. 68, corresponding to the case *AN* = *N*.

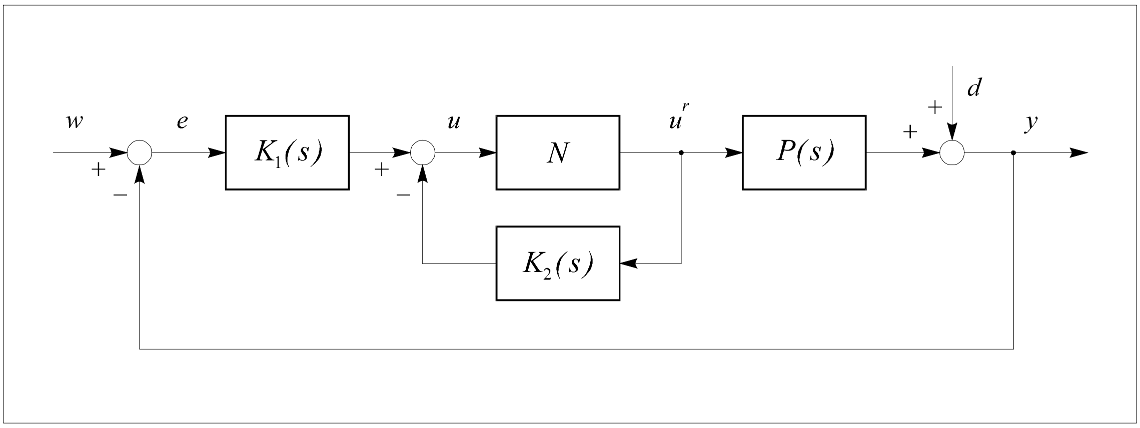

Fig. 67. Parameterisation of linear AW compensators

Kothare et al. (1994) presented a unified framework for the study of anti-windup designs in which all known anti-windup compensation schemes are shown to be special cases. It should be pointed out that our proposed parametrisation is equivalent to the framework of Kothare et al. (1994), but our interpretation is somewhat more intuitive, and provides the following additional insights. The anti-windup compensation framework of Kothare et al. can be abstracted as

$$u = V(s)e + \left(I - U(s)\right)v\,, \tag{106}$$

where

$$\begin{aligned} V(s) &= H_2 - H_2C\left(sI - A + H_1C\right)^{-1} H_1 \\ U(s) &= H_2D + H_2C\left(sI - A + H_1C\right)^{-1}\left(B - H_1D\right) \end{aligned} \,. \tag{107}$$

A, B, C, and D are state-space matrices of $K(s)$, and H_1 and H_2 are the design parameter matrices of an appropriate dimension. Indeed, expressions (106) and (107) are the state-space factorisation of $K(s)$. Noting $K_1(s)=V(s)$ and $K_2(s)=U(s)-I$, we can see that our parametrisation is the same as their framework.

However, as will be shown in the next sub-section, we found that H_2 should not be used as a free design parameter, but should be fixed to $H_2=I$. In such a case, the framework of Kothare et al. coincides with the observer approach proposed by Åström and Wittenmark (1984).

It should be pointed out that the scheme represented by Fig. 4 is one of the simplest ways to parametrise any linear AW compensator for a unity feedback controller $K(s)$. The reason is twofold. First, it is equivalent to the general framework given by Kothare et al. (1994) and therefore parametrises any linear AW compensator. Second, a controller with an anti-windup compensator should contain at least two blocks, and this scheme contains that many: $K_1(s)$ and $K_2(s)$.

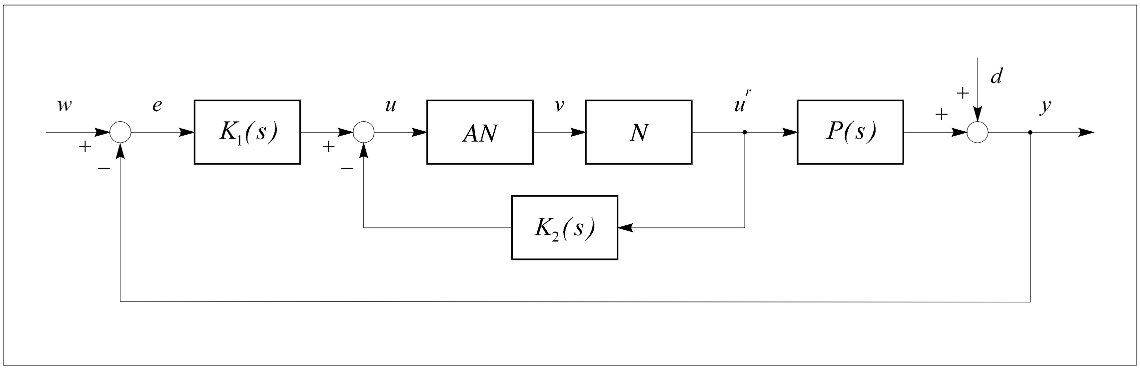

Fig. 68. AW schemes incorporating an artificial nonlinearity

4.1.2 Realisable reference

For a given controller $K(s)$, there are an infinite number of ways of assigning $K_1(s)$ and $K_2(s)$. Similarly, if N is known a priori, there are an infinite number of ways of designing AN. Our intention here is to investigate the pros and cons of several commonly-advocated AW schemes. To this end, the concept of the realisable reference is introduced as follows. Assume that AN is fixed so that N is never active, i.e. $u^r=v$. It is always possible to make process input and output in Fig. 68 equivalent to those in Fig. 69 by a suitable choice of reference w^r. Thus, different $K_1(s)$, $K_2(s)$ and AN will lead to different w^r. If such a reference w^r were applied to the controller instead of the reference w, the nonlinearity AN would not be active. For this reason, w^r is called the realisable reference (Hanus et al., 1987).

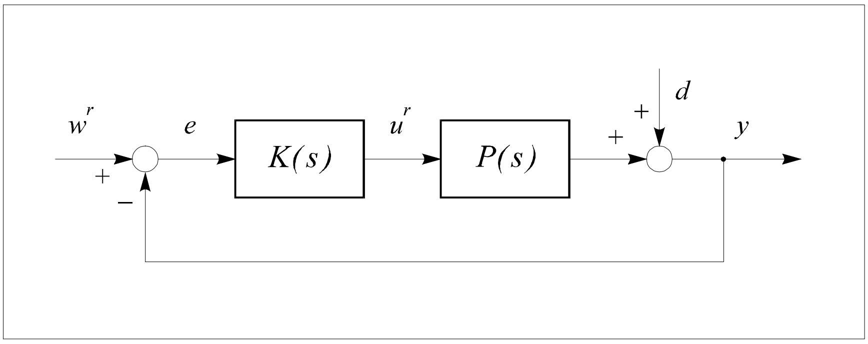

Fig. 69. Feedback control system with realisable reference w^r

From the viewpoint of tracking performance, the best AW strategy should make w^r as close as possible to w.

It can be easily deduced from Fig. 68 and Fig. 69 that

$$u = K_1(s)(w - y) - K_2(s)u^r , \tag{108}$$

$$v = AN(u) = K_1(s)(w^r - y) - K_2(s)u^r \ . \tag{109}$$

These two equations lead to

$$w^r = w + K_1^{-1}(s)(u^r - u) \ . \tag{110}$$

For SISO systems, in order to make w^r as close as possible to w, it is clear from (110) that AN should be the same as N, thereby ensuring $|u^r - u|$ is as small as possible. For this reason, there is no need to introduce AN for SISO systems. However, for MIMO systems, it often happens that one actuator is saturated whilst others are still working in linear regions. So it could prove to be useful to use AN to modify v in order to achieve the desired realisable reference. This aspect will be addressed in details in Sub-chapter 4.2.

4.1.3 Conditions on $K_1(s)$ and $K_2(s)$

The following condition on $K_1(s)$ and $K_2(s)$ must be met:

- C_1: $\left(I + K_2(s)\right)^{-1} K_1(s) = K(s)$;
- C_2: $K_2(s)$ is strictly proper;
- C_3: $K_1(s)$ and $K_2(s)$ are stable.

Condition C_1 , which can be derived from equations (104) and (105), implies that in the absence of N, the controller remains the same as the nominal linear one. Condition C_2 implies that the controller is free of algebraic loops, i.e. it can be implemented. Condition C_3 implies that whenever the plant input u^r is in saturation, the controller implementation is stable in the sense that the outputs of both $K_1(s)$ and $K_2(s)$ are bounded. Note that no stability requirement is imposed on $K(s)$.

One may argue that for a continuous time system, an algebraic loop can be tolerated and condition C_2 loses generality. For instance, in the general framework proposed by Kothare et al., (1994), a design parameter H_2 was proposed. It can be shown that condition C_2 will hold if and only if $H_2=I$ (I is an identity matrix). In other words, if $H_2 \neq I$, condition C_2 will be violated. However, by investigating the following two theorems, we can find that $H_2 \neq I$ is of no use.

Theorem 1: Consider any two AW schemes of the form shown in Fig. 68, and applied to the same constrained closed-loop system with identical initial conditions. Denote these systems as AWa with $K_1(s)=K_{1a}(s)$ and $K_2(s)=K_{2a}(s)$, and AWb with $K_1(s)=K_{1b}(s)$ and $K_2(s)=K_{2b}(s)$. If $K_{1b}(s)=\Gamma K_{1a}(s)$ where Γ is any non-singular matrix, then the realisable reference when using AWa is the same as that when using AWb.

Proof: See Appendix B.

Theorem 2: If $K_{2a}(s)$ is proper but not strictly proper, a non-singular matrix Γ exists such that $K_{2b}(s)$ is strictly proper when setting $K_{1b}(s)=\Gamma K_{1a}(s)$.

Proof: See Appendix C.

Indeed, $H_2 \neq I$ leads to K_{2a}(s) which is proper but not strictly proper. From Theorem 1, it follows that using such a K_{2a}(s) is equivalent to using any $K_{2b}(s)$ if it corresponds to $K_{1b}(s)=\Gamma \cdot K_{1a}(s)$, since the realisable references are the same for these two configurations. From Theorem 2, we can see that Γ exists such that $K_{2b}(s)$ is strictly proper. On the other hand, a strictly proper $K_{2b}(s)$ should correspond to $H_2=I$. Therefore, the imposition of $H_2=I$ in condition C_2 implies no loss of generality, and the introduction of an algebraic loop by imposing $H_2 \neq I$ is of no use.

To analyse the stability of the AW control system described by Fig. 68, only some sufficient conditions to guarantee closed-loop stability are given by techniques such as the Popov, Circle, and Off-axis Circle criteria, the passivity theorem and multiplier theory (Kothare and Morari, 1995). Note that the closed-loop AW control system contains a cascade link between nonlinearity N and $P(s)K_1(s)+K_2(s)$. We conject that a stable $K_1(s)$ and $K_2(s)$ lead to closed-loop systems having a greater chance of meeting the known sufficient conditions for stability, but a detailed investigation of this conjecture requires further research.

4.1.4 The choice of $K_1(s)$ and $K_2(s)$

Let us first give some explicit meanings of the conditions C_1, C_2, and C_3. From C_1, we can obtain

$$K_2(s) = K^{-1}(s)K_1(s) - I \tag{111}$$

Usually, a controller is biproper, and this implies that $K(\infty)$ exists. In such case, from (111) and noting that $K_2(s)$ should be strictly proper, we can deduce

- C_4: $K_1(\infty) = K(\infty)$ if $K(\infty)$ exists.

However, it may happen that a controller is not biproper. For instance, if one insists on diagonally decoupling a square MIMO plant whose interactor matrix is non-diagonal, any unity feedback controller must be strictly proper. In this case, C_4 is no longer valid. To deduce a condition for replacing C_4, we introduce an interactor matrix $\xi(s)$ for $K(s)$, such that

$$\xi(s)K(s) = \overline{K}(s) \tag{112}$$

$$\lim_{s\to\infty} \xi(s)K(s) = \lim_{s\to\infty} \overline{K}(s) = \overline{K}(\infty) \tag{113}$$

where $\overline{K}(\infty)$ is non-singular. Note that there are different ways to construct $\xi(s)$. From (112) and bearing in mind that $K_2(s)$ should be strictly proper, we can deduce

- C_5: $K_1(s) = \xi^{-1}(s)\overline{K}_1(s)$ and $\overline{K}_1(\infty) = \overline{K}(\infty)$.

Another potential problem arises when K(s) contains non-minimum phase zeros. For instance, if one insists on diagonally decoupling a square non-minimum phase MIMO plants whose generalised interactor matrix is non-diagonal, any unity feedback controller must be nonminimum phase (Goodwin et al., 1993; Weller, 1996). As $K_2(s)$ is stable, from (111) we must ensure that $K^{-1}(s)K_1(s)$ is also stable. The explicit condition for this can be derived using the concept of the generalised interactor matrix. By generalised matrix, we mean that not only (112) and (113) are to hold, but also ξ(s) contains all the non-minimum phase zeros. Due to this generalisation, C_5 is still valid since $K^{-1}(s)K_1(s)$ remains stable. The idea of using the controller interactor matrix in the AW context was first proposed in Hanus and Peng (1991, 1992).

It should be pointed out that in practice, K(s) is usually biproper and has no non-minimum phase zeros. In this case, we propose using the conditioning technique which sets

$$K_1(s) = K(\infty) \tag{114}$$

$$K_2(s) = K^{-1}(s)K(\infty) - I \tag{115}$$

There are two reasons for using the conditioning technique. First, as K_1(s) does not feedback any information of the real input u^r, it should avoid the inclusion of any dynamics which may suffer windup, as done in (114). The worst case is that K_1(s)=K(s), i.e. K_1(s) contains all the dynamics of K(s). This case corresponds to no AW compensation. Second, from (110), it is clear that w^r becomes w at the instant the controller leaves the limitation. Any dynamical K_1(s) will make w^r different from w after the controller leaves the limitation.

4.2 Optimal artificial nonlinearity design

For multivariable control systems, the potential exists for some, but not all, plant inputs to enter saturation. In such a situation, a commonly advocated strategy is to scale down all controller inputs so that the direction of $u(t)$ the plant input vector is maintained (Campo and Morari, 1990; Christen and Geering, 1996). This is accomplished using the following *AN* design strategy, known as “direction-preserving” AN design:

$$v(t) = AN(u(t)) = \begin{cases} u(t) & \textit{if } u(t) \textit{ is in a linear region} \\ \min\left\{\dfrac{\text{sat}(u_i(t))}{u_i(t)}\right\} u(t) & \textit{if } u(t) \textit{ enters the saturation} \end{cases} . \quad (116)$$

From equation (110), however, it is clear that such a strategy can easily make w^r far from w. An alternative strategy was proposed by Hanus and Kinnaert (1989), in which the artificial nonlinearity block *AN* is designed in such a way that w^r remains as close as possible to w, in some sense. Since by definition, *AN* is no less limited than *N*, the resulting w^r is bounded if it is bounded without using *AN*, and thus the optimal *AN* design will not endanger closed-loop stability. While optimal *AN* design was originally proposed in (Hanus and Kinnaert, 1989), the presentation there is insufficiently detailed to permit ready implementation. For this reason, we have decided to present a thorough treatment of optimal *AN* design in the present paper.

It should be emphasised that the optimal *AN* design is carried out after designing $K_1(s)$ and $K_2(s)$, so that design of *AN* is considered independent of any particular AW design. To simplify the presentation, the nonlinearity *N* is assumed to be restricted to the "*sat*" function, defined as

$$sat(u) = \left[sat(u_1) \quad \cdots \quad sat(u_m) \right]^T \quad (117)$$

where

$$sat(u_i) = \begin{cases} U_i^{\max} & u_i > U_i^{\max} \\ u_i & U_i^{\min} \le u_i \le U_i^{\max} \\ U_i^{\min} & u_i < U_i^{\min}. \end{cases} \quad (118)$$

Suppose that the conditioning technique (114)-(115) is used, and we denote by *D* the non-singular feedthrough matrix of the linear controller *K(s),* i.e. $D=K(\infty)$. Equation (110) can then be rewritten as

$$w^r = w + D^{-1}\left(u^r - u\right). \quad (119)$$

By introducing $H=[I_{m\times m}\ -I_{m\times m}]^T$ and $b=[-U_1^{\max},\ldots,-U_m^{\max},U_1^{\min},\ldots,U_m^{\min}]^T$, the inequality constraints in (118) can be concisely represented as

$$h_u(v) = Hv + b \leq 0\,. \tag{120}$$

Using (119), the inequality constraints can be expressed in terms of w^r and w:

$$h_w\left(w^r\right) = HD\left(w^r - w\right) + Hu + b \leq 0\,. \tag{121}$$

Optimal *AN* design is therefore a nonlinear programming problem, the solution of which instantaneously minimises the performance criterion $J\ (w^r - w)$ subject to the inequality constraints in equation (121). With the appropriate choice of $J(\bullet)$, both the criterion and constraint functions are convex. The optimal *AN* design problem therefore always has a solution which can be found via numerical techniques, e.g. the simplex method. However, since u is a time-varying signal, the optimal *AN* design should be carried out at every time instant, making such a solution computationally intensive. The closed-form solution first described by Hanus and Kinnaert (1989), and suitable for real-time applications is presented in Appendix D.

Block AN in Fig. 68 can then be simply represented by the following programme in the MATLAB programme package:

```
constr = H*u+b;
[cmax, imax] = max(constr);
if (cmax > 0)
  h0 = H(imax,:);
  b0 = b(imax);
  tmp = h0*D*inv(Lambda)*D'*h0';
  v = -D*Lambda*D'*h0'/tmp*(h0*u+b0)+u;
else
  v = u;
end
```

The meaning of the weighting matrix Λ (Lambda) is described in Sub-chapter 4.3 and in Appendix D.

4.3 Simulation results

In the previous sections, we have shown that the conditioning technique is usually a suitable AW technique and that optimal AN design can improve any AW technique. To support our theoretical development, we will compare the following four approaches, namely the original conditioning technique, the conditioning technique with direction-preserving AN, the conditioning technique with optimal AN, and the internal model control (IMC) with AW compensation. For the last approach, we chose the modified IMC approach (Zheng et al., 1994) since it was shown by Zheng et al. (1994) to be a superior solution to the original conditioning technique. Some simulations were performed to show the effectiveness of the proposed approach.

Consider the process described by

$$P(s) = \frac{10}{1+100s}\begin{bmatrix} 4 & -5 \\ -3 & 4 \end{bmatrix}. \tag{122}$$

This process is used in (Zheng et al.,1994). The classical IMC controller is given by $\widetilde{P}(s) = P(s)$ and

$$Q(s) = \frac{1+100s}{10(1+20s)}\begin{bmatrix} 4 & 5 \\ 3 & 4 \end{bmatrix}. \tag{123}$$

For the meaning of $\widetilde{P}(s)$ and $Q(s)$, refer to (Zheng et al.,1994). This controller is equivalent to a feedback controller with

$$K(s) = Q(I - \widetilde{P}Q)^{-1} = \frac{1+100s}{200s}\begin{bmatrix} 4 & 5 \\ 3 & 4 \end{bmatrix}. \tag{124}$$

In the modified IMC approach, Zheng et al. (1994) worked with a modified plant model in order to achieve a performance superior to the original conditioning technique. This implies that the controller too, was modified, with

$$\widetilde{P}(s)=\frac{10}{1+100s}\begin{bmatrix} 4 & -\frac{5}{0.1s+1} \\ -\frac{3}{0.1s+1} & 4 \end{bmatrix}, \tag{125}$$

and a filter *f=2.5(s+1)I* was introduced for the AW implementation. This leads to

$$\begin{aligned} K_1(s) &= fPQ \\ K_2(s) &= fP - I - fPQ\widetilde{P} \end{aligned}. \tag{126}$$

To apply the conditioning technique, we took the feedback controller (124) and decomposed it according to equations (114) and (115).

A set-point change of $[0.6\ 0.4]^T$ was applied. Both inputs were constrained between the saturation limits ±1. Fig. 70 shows the process outputs for unlimited system, and a limited system when using the original conditioning technique and the modified IMC AW compensator. It can be clearly seen that the classical conditioning technique resulted in quite a poor performance. The modified IMC approach resulted in improved performance. Fig. 71 shows the process outputs obtained using the conditioning technique with those achieved by direction-preserving AN, and the conditioning technique with optimal AN where the weighting factor was $\Lambda=I$. It is clear that both approaches resulted in improved responses compared with the original conditioning technique, even better than the modified IMC approach.

The following criteria can be used to compare the four AW design strategies:

$$J_1 = \sum_{i=1}^{2} \int_0^{\infty} \left| w_i^r(t) - w_i(t) \right| dt \tag{127}$$

$$J_2 = \sum_{i=1}^{2} \int_0^{\infty} \left(w_i^r(t) - w_i(t) \right)^2 dt \tag{128}$$

$$J_3 = \sum_{i=1}^{2} \int_0^{\infty} \left| y_{uli}(t) - y_{AWi}(t) \right| dt \tag{129}$$

$$J_4 = \sum_{i=1}^{2} \int_0^{\infty} \left(y_{uli}(t) - y_{AWi}(t) \right)^2 dt, \tag{130}$$

where y_{uli} and y_{AWi} are the *i*-th process output of the unlimited and limited process output with AW compensator, respectively.

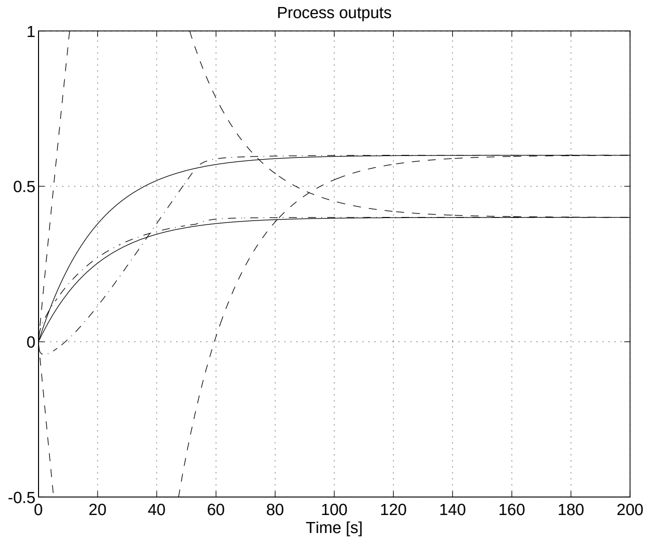

Fig. 70. The process outputs (y); __ Unlimited system, -- Original conditioning technique, -.- Modified IMC approach

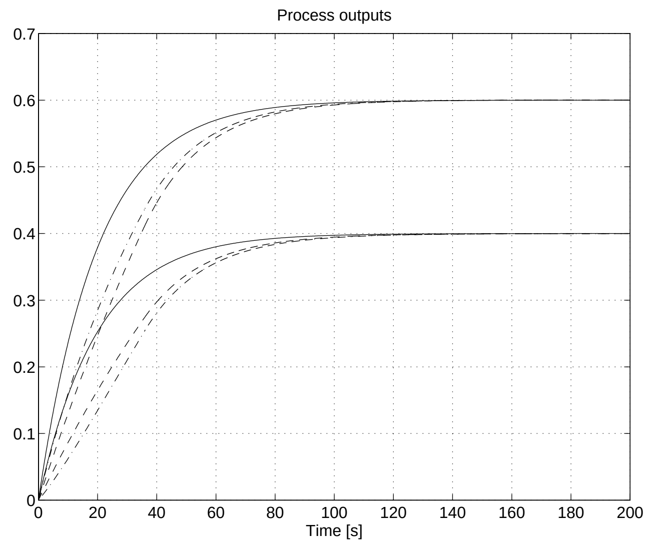

Fig. 71. The process outputs (y); __ Unlimited system, -- Conditioning technique with direction-preserving AN design, -.- Conditioning technique with optimal AN design

The values of criteria functions are given in Table 1.1. It is obvious that the conditioning technique with optimal AN gives the best result.

Table 1.1. The values of criteria functions achieved by several AW techniques

	Original conditioning technique	***Modified IMC with AW compensation***	***Conditioning technique with direction-preserving AN***	***Conditioning technique with optimal AN***
J_1	164.5	16.26	9.151	8.84
J_2	453.8	8.153	1.68	1.525
J_3	164.5	11.13	9.157	8.85
J_4	226.7	1.984	0.722	0.656

Figs. 72 and 73 show the effectiveness of the weighting factor Λ by taking

$$\Lambda = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix} \tag{131}$$

It is clear that, when using the conditioning technique with optimal AN, the unlimited and limited responses are quite similar for the first process output (at the cost of the greater differences between the second one). Therefore, if the control of certain process outputs must be tighter than that of the other process outputs, this can be achieved by changing the weighting factor Λ.

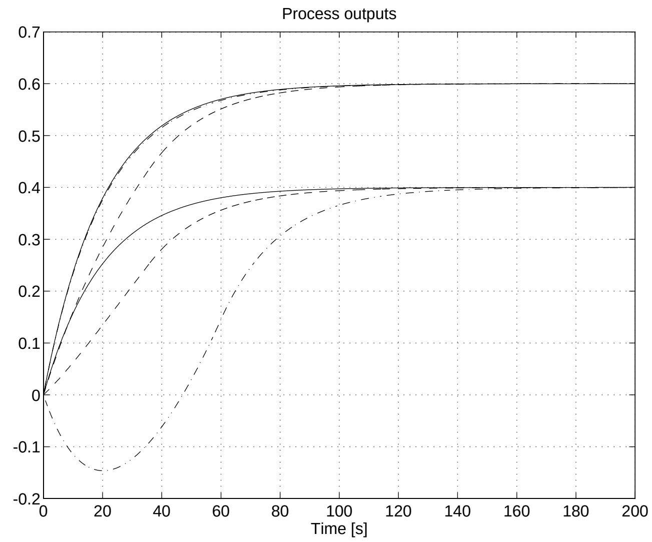

Fig. 72. Process outputs (y) when using conditioning technique with optimal AN design;
__ Unlimited system, -- Λ=I, -.- Λ =[10 0;0 1]

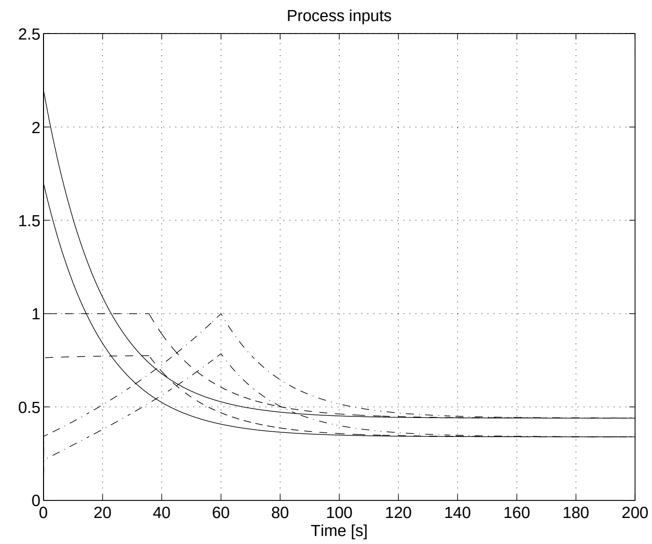

Fig. 73. Process inputs (u^r) when using conditioning technique with optimal AN design;
__ Unlimited system, -- Λ=I, -.- Λ =[10 0;0 1]

4.4 Discussion

Linear AW techniques for multivariable controllers can easily be parametrised using the classical feedback structure. Conditions for controller implementability and closed-loop stability can be stated in terms of this parametrisation.

The case where the controller is not biproper (has non-minimum phase zeros) is also studied. It is shown that in such a case the controller interactor (generalised interactor) matrix should be introduced in order to derive explicit conditions for controller implementability and closed-loop stability. This also helps in explaining why the conditioning technique has some limitations and how these limitations can be removed.

It has been shown that AW designs for MIMO systems can be enhanced with the inclusion of an artificial nonlinearity block, especially when mutual interactions between the process inputs and outputs are strong. As the conditioning technique is generally the most suitable AW technique for single-input single-output (SISO) systems (Peng et al. 1996a,b; Vrančić and Peng, 1996, Vrančić et al., 1996a), it seems that conditioning enhanced with an optimal AN design strategy is also a suitable AW technique for MIMO systems.

The practical implementation of the optimal AN has been presented, and the proposed algorithm, based on the Kuhn-Tucker theorem, appears to be simple and fast enough for real-time applications.

The advantage of using the conditioning technique as an AW technique is its simplicity: the resulting AW compensator depends only on the unconstrained controller transfer function, whereas most other approaches also require the process transfer function in order to calculate an appropriate AW compensator. Moreover, the robustness of AW compensators designed in this way when the identified process model is poor remains a topic for research.

It is sometimes claimed that a disadvantage of the conditioning technique is that it offers little design freedom in the form of additional tuning parameters. We would argue, however, that design freedom in the present context should be seen as a means of achieving satisfactory closed-loop performance, and that the freedom available in the design of the transfer matrix $K(s)$ and/or the additional artificial nonlinearity AN is almost always sufficient for this task.