Abstract

In this work a comprehensive review of anti-windup, bumpless and conditioned transfer techniques is given in the framework of PID types of controllers, controllers with general rational transfer function and for state-space controllers. The so-called realisable reference [Hanus, 1989] is used as a powerful tool, when comparing different anti-windup, bumpless and conditioned transfer methods. The usefulness of the realisable reference lays in the fact that it transforms the system limitations (and other kind of non-linearity) into changed reference signal - realisable reference. System becomes linear and thus much easier to deal with. The notion of the realisable reference is given for all mentioned types of controllers. With the help of the realisable reference, the conditioning technique turned out as the most suitable anti-windup method for usual applications. The exception is the case in which the input limitations are too restrictive. In this case we proposed the method with variable feedback anti-windup compensator. The new notion of conditioned transfer is also introduced and it will be shown to be a more suitable solution than bumpless transfer.

Some additional practical recommendations, concerning anti-windup design, are also given.

All the derivations, definitions and discussions are supported by simulations.

Keywords: PID control, generalised PID controller, controller limitations, anti-windup, bumpless transfer, conditioned transfer, realisable reference

1. Introduction

All industrial processes are submitted to *constraints*. For instance, a controller works in a limited range 0-10V or 0-20mA, a valve can not be opened more than 100% and less than 0%, a motor driven actuator has a limited speed. Such constraints are usually referred to as *plant input limitations*.

In addition, a commonly encountered control scheme is switching from manual to automatic mode or between different controllers. Such mode switches are usually referred to as *plant input substitutions*.

As a result of limitations or substitutions, the actual plant input is temporarily different from the controller output. When this happens, if the controller is initially designed to operate in a linear range, the closed-loop performance will significantly deteriorate with respect to the predicted linear performance. This performance deterioration is referred to as *windup*. Besides windup, in the case of substitution, the difference between the outputs of different controllers results in a *bump* in the plant input.

A rational way to handle the problem of windup is to take into account the input limitations already in the stage of control design. However, this approach is very involved and the resulting control law is very complicated. A more common approach in practice is to add a simple heuristic compensation at the stage of control implementation. As this compensation aims to diminish the performance deterioration caused by windup, it is referred to as *anti-windup*.

In the case of mode switching, the method which aims to minimise the bump is referred to as the *bumpless transfer*. Yet, to minimise the bump is not always justified, since this may cause a poor tracking performance. Moreover, reducing bump to *zero* usually has no rational explanation except our fear against bump. Thus we refer to the method which will not only reduce the bump but also keep a good tracking performance as the conditioned transfer.

In the case of mode switching, an anti-windup strategy is usually implemented as a bumpless transfer technique. Indeed, an anti-windup method will usually diminish the bump, too.

The topic of anti-windup and bumpless transfer has been studied over a long period of time by many authors, and the most popular techniques are described in [Fertik and Ross, 1967] [Åström and Wittenmark, 1984] [Hanus et al., 1987] [Morari et al, 1993]. However, although the concept of anti-windup and bumpless transfer is introduced in almost every basic control textbook, it is not always clearly illustrated and is sometimes misinterpreted. For instance, many authors thought that anti-windup was aimed at reducing the output overshoot in its step response, or that anti-windup was the synonym of bumpless transfer, or that the best transfer transition is to have no bump. These thoughts need to be corrected.

Therefore, the objectives of this work are as follows. First, we would like to illustrate the phenomena of windup and bump, and the improved results obtained using the techniques of anti-windup, bumpless transfer and conditioned transfer. By the aid of simulations we will review the majority of existing methods and compare them by using the notion of realisable reference. Finally, we will investigate the case of mode switching and introduce the new notion of *conditioned transfer*. It will be shown that conditioned transfer is a more suitable solution than bumpless transfer.

The remainder of this work is organised as follows. In *section 2*, the background material concerning system limitations, the structure of used controllers, windup, anti-windup, bump, bumpless and conditioned transfer is provided.

In *section 3*, some most frequently used anti-windup algorithms for PID types of controllers, controllers with general rational transfer function and state-space controllers are reviewed.

In *section 4*, the realisable reference is introduced to compare different anti-windup algorithms. Due to realisable reference, some general solutions of anti-windup were also introduced.

Section 5 gives us some practical considerations when using anti-windup methods in practice and some special precautions when system is highly limited. An information is also given how non-linearity should be handled.

In *section 6*, some more examples on process models and on the laboratory plants are given to illustrate the effectiveness of the proposed anti-windup design.

In *section 7*, a short and comprehensive review of some mentioned terms and solutions is given and finally, conclusions are carried out in *section 8*.

2. Background Materials

2.1. System Limitations

Most controllers, practically running in the industry, are tuned to operate in a linear range. They assume no controller and process limitations. Such assumption is valid if disturbances are small or there are only slight set point changes. Fig. 2.1 represents the unlimited closed-loop linear system.

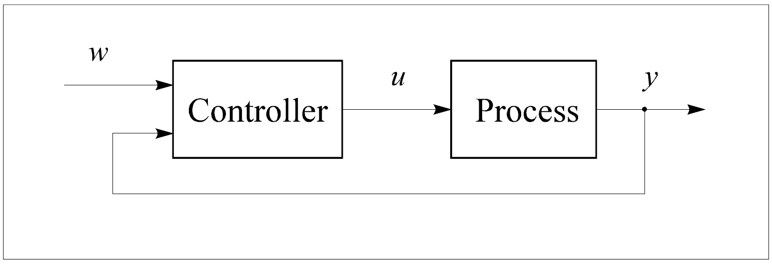

Fig. 2.1. The unlimited closed-loop system

However, real systems are always imposed to limitations. In such real systems, in the case of large set point change, control signal is limited what breaks a closed-loop path between controller and process. Broken path can produce undesired controller's behaviour (**windup**) if controller has the integral term (e.g. PI or PID controllers) or some other types of "memory" as in controllers with general rational transfer function or state-space controllers.

The next problem often related with windup is a transfer from manual to the automatic mode and transfer between different controllers.

If no special protection is used when controller works in an open-loop, a big bump (**bump transfer**) can appear at the process input when switching to closed-loop configuration (from manual to automatic mode or switching between different controllers).

However, in practice it is always the case that controller and process (specially actuators) are limited. The most common types of limitations are **magnitude** and **rate** limitations. Figure 2.2 and equation 2.1 describe **magnitude limitation**.

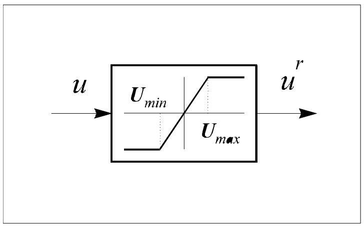

Fig. 2.2. Magnitude limitation.

$$u^r = \begin{cases} U_{\max} & ;\ u > U_{\max} \\ u & ;\ U_{\min} < u < U_{\max} \\ U_{\min} & ;\ u < U_{\min} \end{cases} \tag{2.1}$$

The output of the limitation is the same as the input if input signal lays between U_{min} and U_{max}. Output can not be greater than U_{max} and less than U_{min}. The representatives of the magnitude limitations are controllers (they can give the output values in limited range: 0-10V, 0/4-20mA, etc.), valves (valve can not be opened more than 100% and less than 0%), pumps, compressors, etc. All of them have limited working ranges.

Another type of limitation is **a rate limitation**. Figure 2.3 and equation 2.2 describe the rate limitation.

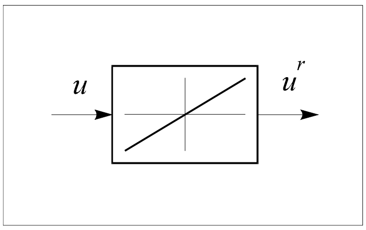

Fig. 2.3. Rate limitation.

$$\frac{\partial u^r}{\partial t} = \begin{cases} v_{\max} & ;\ \dfrac{\partial u}{\partial t} > v_{\max} \\ \dfrac{\partial u}{\partial t} & ;\ v_{\min} < \dfrac{\partial u}{\partial t} < v_{\max} \\ v_{\min} & ;\ \dfrac{\partial u}{\partial t} < v_{\min} \end{cases} \tag{2.2}$$

The output of the limiter can not change faster or slower than prescribed by the velocity limit (v_{max} and v_{min}). Representatives of the rate limitations are motor driven actuators, which have limited range of speed.

In the following text the magnitude and the rate limitations will be drawn in the same way (figure 2.4).

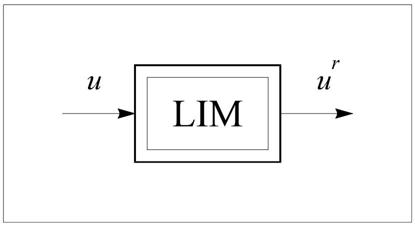

Fig. 2.4. The magnitude and/or rate limitations

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

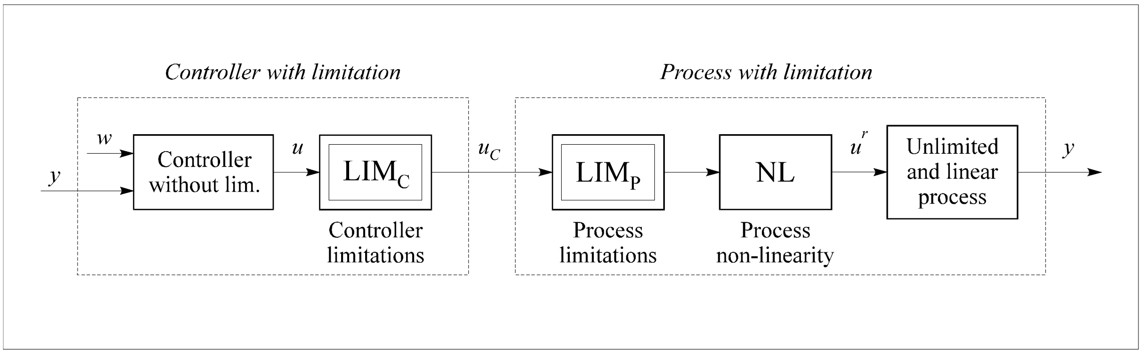

Fig 2.5. Controller limitations and process limitations with non-linearity

Actuators are not only imposed to pure limitations. Usually they can also have a non-linear behaviour (e.g. actuator of the valve has non-linear characteristics between output flow and input opening angle). In general, the controller limitations and process limitations with non-linearity can be described by Fig 2.5.

Through all experiments in the following text we will suppose there are no process non-linearities, except in chapter 5.2. where is described how to handle process non-linearity through the concept of anti-windup design.

2.2. Controllers

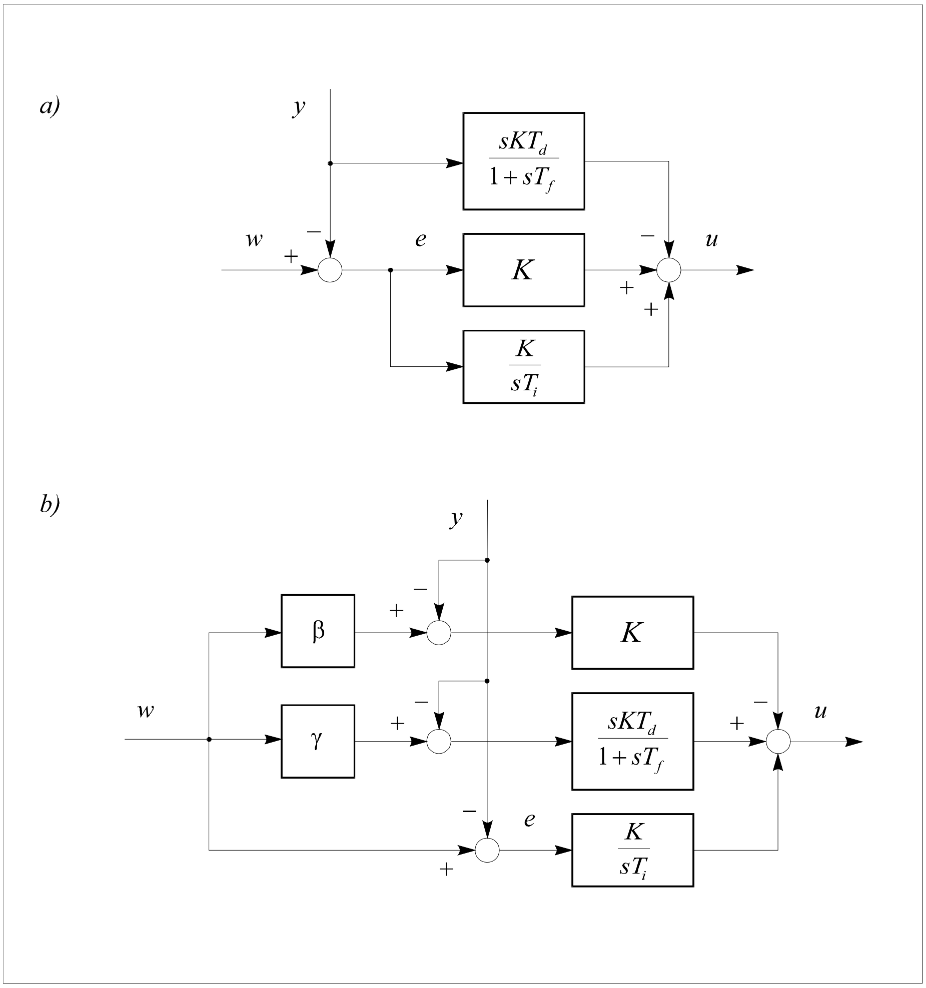

Fig 2.6. a) PID controller b) Generalised PID controller

As a controller, we have used PID controllers, controllers with general rational transfer function and state-space controllers. Fig 2.6. represents two chosen types of PID controllers. First type (case a) is widely used (specially in industry) and is referred as ***PID controller***.

The second one is generalised structure of PID controller. With changing factors β and γ, we can make different PID structures as textbook PID controller (β=1, γ=1), already mentioned industrial PID controller (β=1, γ=0), set-point-on-I-only controller (β=0, γ=0) [Åström and Wittenmark, 1984], etc. Note that factors β and γ can have any real value (usually between 0 and 1). The second type (case b) of PID controller is therefore in the following text referred to as ***generalised PID controller***.

The *PID controller* can be described by equation

$$U(s) = K\left[E(s) + \frac{1}{sT_i}E(s) + \frac{sT_d}{1+sT_f}Y(s)\right] \tag{2.3}$$

where U, Y and E denote the Laplace transfer function of controller output, process output and process tracking error *(e=w-y)*, respectively.

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

A ***generalised PID controller*** is described by expression (see Fig 2.6):

$$U(s) = K\left[\beta + \frac{1}{sT_i} + \gamma\frac{sT_d}{1+sT_f}\right]W(s) - K\left[1 + \frac{1}{sT_i} + \frac{sT_d}{1+sT_f}\right]Y(s) \tag{2.4}$$

where w denote a Laplace transform of the reference (set-point). Factors β and γ define the strength of proportional (P) and derivative (D) part of PID controller connected to the reference (set-point) w, respectively. More details about factor β, can be found in [Åström and Wittenmark, 1984].

Equation (2.5) and Fig. 2.7 depict a ***controller with general rational transfer function***, where N_i and D_i represent numerators and denominators of the transfer function, respectively. In usual implementation, N_2(s) = N_1(s) and D_2(s)=D_1(s).

$$U(s) = \frac{N_1(s)}{D_1(s)}W(s) - \frac{N_2(s)}{D_2(s)}Y(s) \tag{2.5}$$

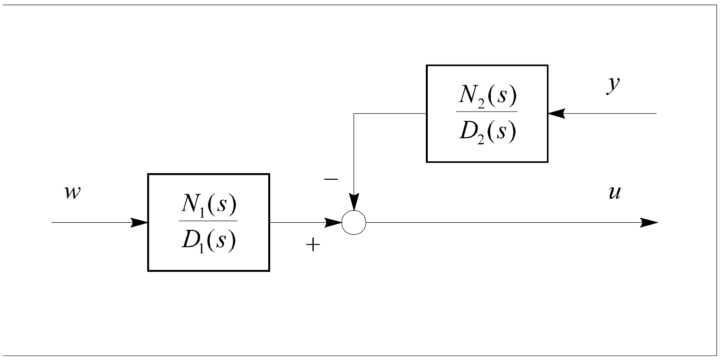

Fig. 2.7.: Controller with general rational transfer function

The ***controller in the state-space form*** is described by Fig 2.8 and equation 2.6.

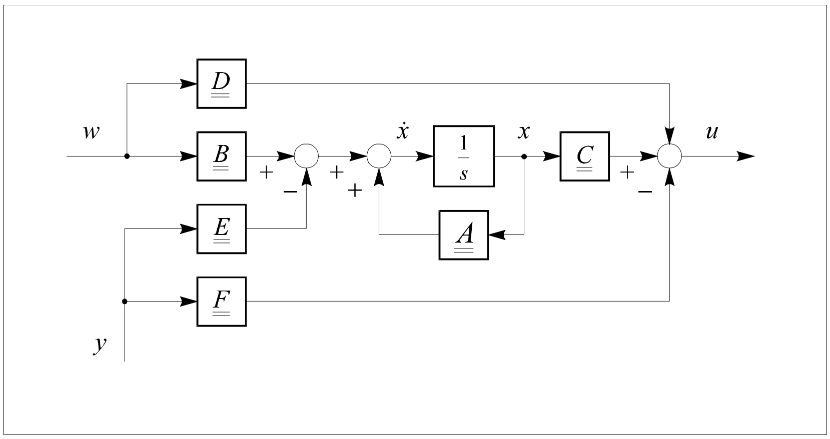

Fig. 2.8.: Controller in the state-space form

$$\underline{\dot{x}} = \underline{\underline{A}}\,\underline{x} + \underline{\underline{B}}\,\underline{w} - \underline{\underline{E}}\,\underline{y}$$
$$\underline{u} = \underline{\underline{C}}\,\underline{x} + \underline{\underline{D}}\,\underline{w} - \underline{\underline{F}}\,\underline{y} \qquad (2.6)$$

where *A* is controller’s *system* matrix, *B* and *E* are controller’s *input* matrices, *C* is controller’s *output* matrix and *D* and *F* are controller’s *input-output* matrices. Generally, *w*, *x*, *u* and *y* are vectors. When designing a controller, it is frequently chosen *B*=*E* and *D*=*F*.

2.3. Windup and Anti-Windup (AW)

Windup:

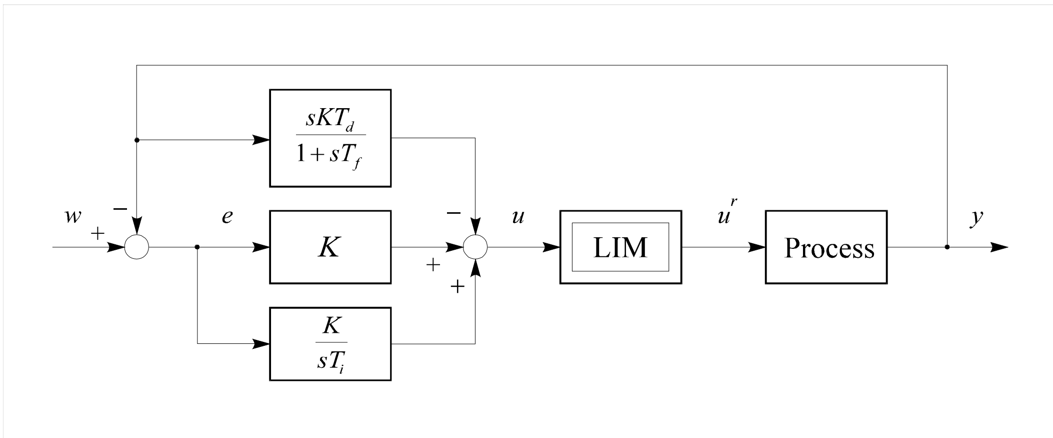

Fig. 2.9. PID controller: The limited closed-loop system without AW

Consider a closed-loop system containing a PID controller and a magnitude limitation LIM (Fig. 2.9). Suppose the controller and process are in steady state. A positive step change in w causes a jump in u which will leave the linear region of the actuator (the actuator saturates at higher limit if $K > 0$) and u^r becomes smaller than u. Thus y is slower than in 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. So 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.

To illustrate the above phenomenon, we have made a simulation with process

$$G_{PR} = \frac{1}{(1+8s)(1+4s)} \tag{2.7}$$

and controller

$$K = 20,\ \ T_i = 30s,\ \ T_d = 1s,\ \ T_f = 0.1s \tag{2.8}$$

The process input limitations were:

$$v_{\max} = 1s^{-1}, v_{\min} = -1s^{-1} \tag{2.9}$$

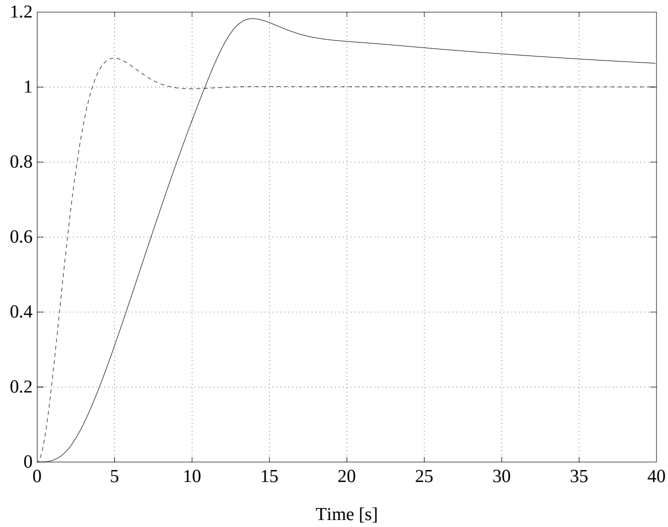

Fig. 2.10. Process output (y);
__ Process input limited, -- Process input unlimited

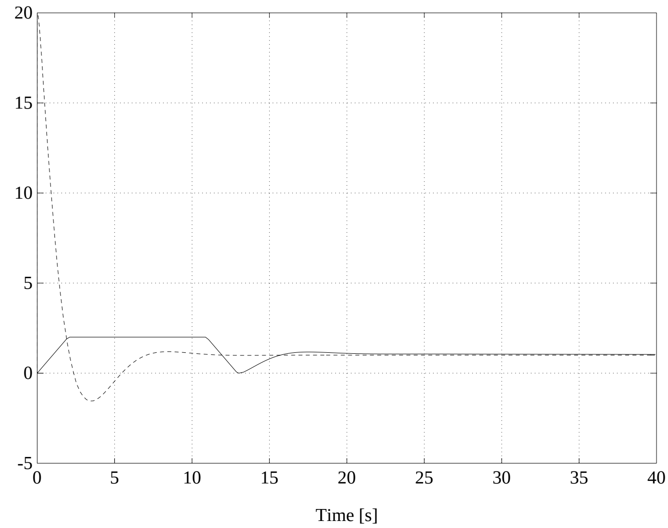

Fig. 2.11. Process input (u^r);
____ Process input limited, -- Process input unlimited

The closed-loop step responses for both, limited and unlimited cases, are shown in Figures 2.10 and 2.11.

In Figure 2.10, we can see a *large overshoot* and a *long process settling time* in the limited case with respect to the unlimited one.

This closed-loop performance deterioration due to the input limitations is called *windup*.

Anti-Windup:

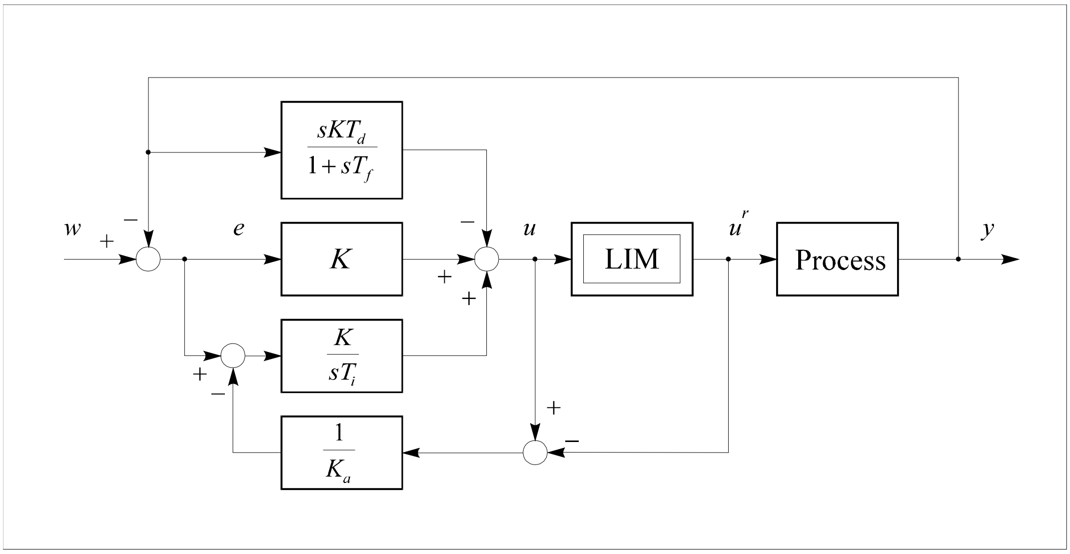

Fig. 2.12. The limited closed-loop system with AW

In fact, windup appears due to the fact that the integral term increases too much during saturation. Thus, during saturation, the increase should be slowed down. It can be realised by a compensation which feeds back u-u^r to the integral term, as shown in Figure 2.12. As this compensation aims at reducing the effect of windup, it is thus called anti-windup (AW).

To show improvements made by feedback compensator $1/K_a$, we used the same process (2.7), controller (2.8) and limitations (2.9) as in the previous case. Figures 2.13 and 2.14 show results obtained by choosing the value $K_a=K$.

It can be seen that from Fig. 2.13, using an anti-windup algorithm, the overshoot is *smaller* and the settling time is *much shorter* than in the absence of an anti-windup algorithm.

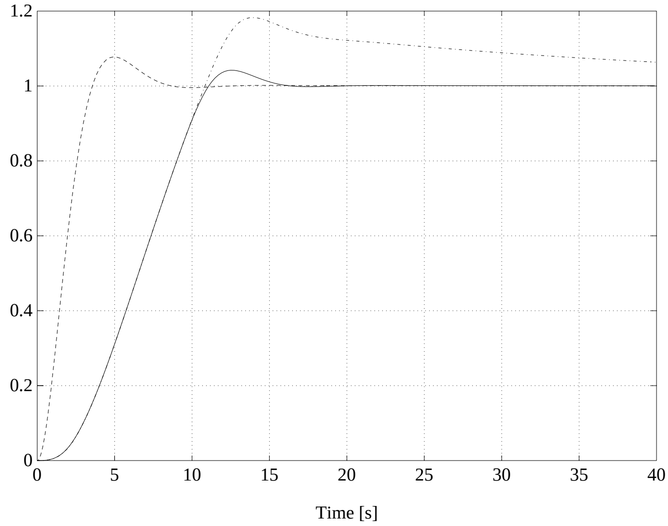

Fig. 2.13. Process output (y); __ Process input limited with AW, -- Process input unlimited, -.- Process input limited without AW

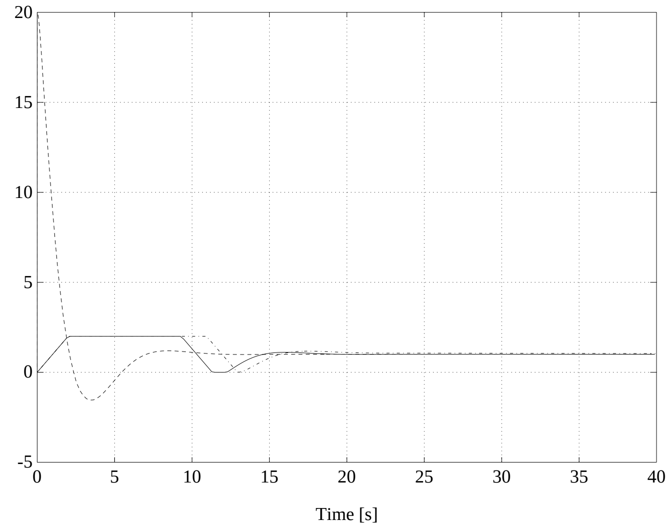

Fig. 2.14. Process input (u^r); __ Process input limited with AW, -- Process input unlimited, -.- Process input limited without AW

2.4. Bump, Bumpless and Conditioned Transfer (BT, CT)

Bump Transfer:

Let us consider the control scheme with the capability of switching between manual and automatic control mode (Fig. 2.15). Note that u^m *can also represent an output from another controller instead of manual signal*. Assume that the switch goes from automatic to manual control (u^r goes from u to u^m). If u^m is such that for some time e>0, then the integral term increases uncontrolled 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 (u^r goes from u^m to u). At that moment, even if e=0, a big 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 large settling time of the process output.

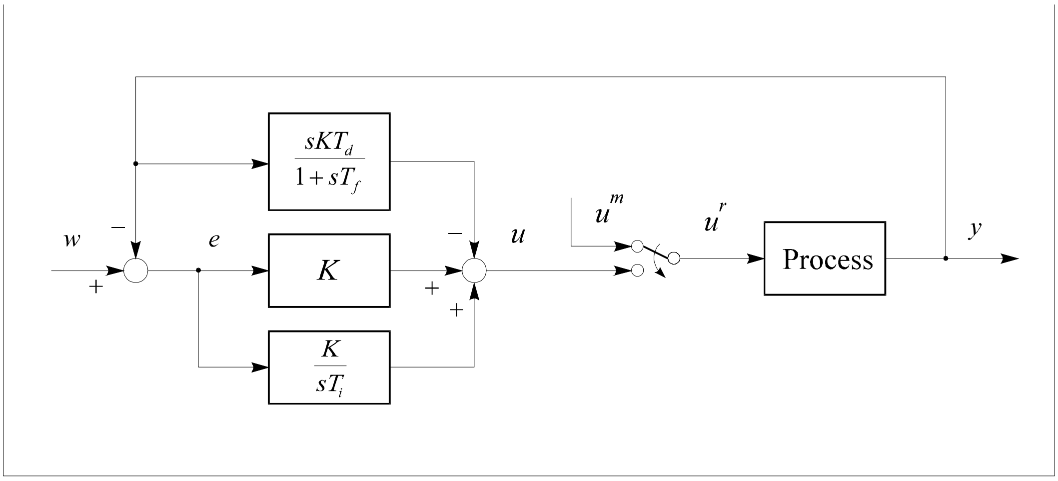

Fig. 2.15. Transfer from manual to automatic mode

To illustrate the above phenomenon, we have made a simulation with process

$$G_{PR} = \frac{1}{(1+10s)^2} \tag{2.10}$$

and controller

$$K = 20, \quad T_i = 40.8s, \quad T_d = 1.16s, \quad T_f = 0.116s \tag{2.11}$$

The reference signal is taken as 0. The controller is manually driven in the period from 0 to 80s. Then, it is switched to automatic mode. The result of the simulation is shown in Figures 2.16 and 2.17.

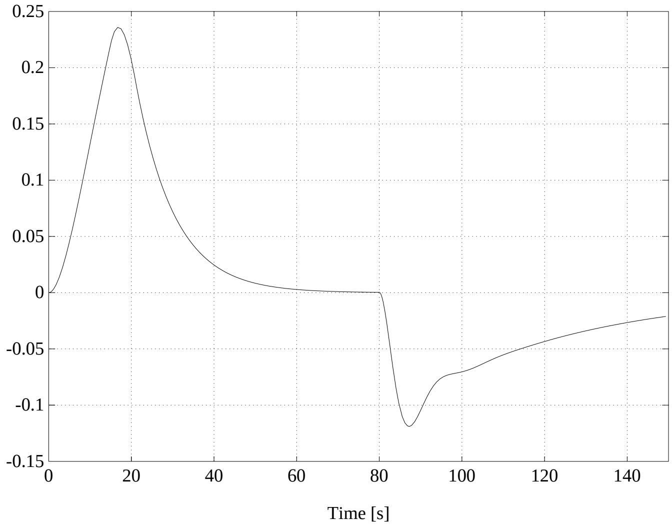

Fig. 2.16. Process output (y) - bump transfer

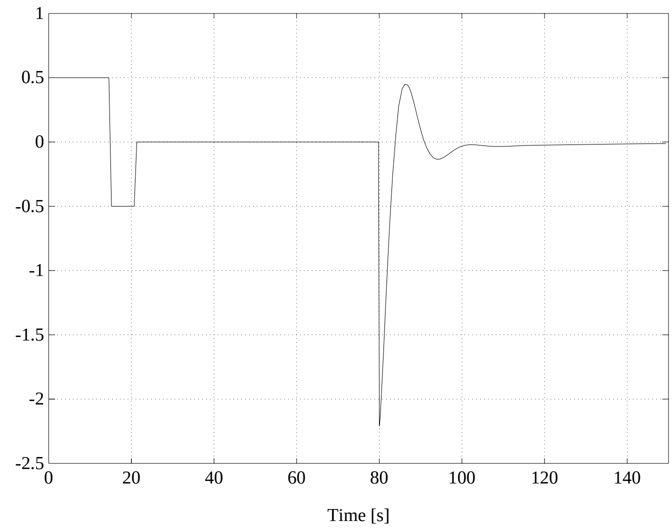

Fig. 2.17. Process input (u^r) - bump transfer

From Fig. 2.16, it can be seen that, at the instant of switching, a big jump occurs at the process input and this also causes a long settling time of the process.

This transfer without any protection is called *bump transfer*.

Bumpless Transfer:

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

Conditioned Transfer:

After switching from manual to automatic control, y is wished to become equal to w with the same dynamics as for the closed-loop step response. In other words, after switching, good tracking performance is expected. This transfer is called conditioned transfer (CT). Note that, using conditioned transfer, the bump is not minimised.

As done for 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. 2.18.

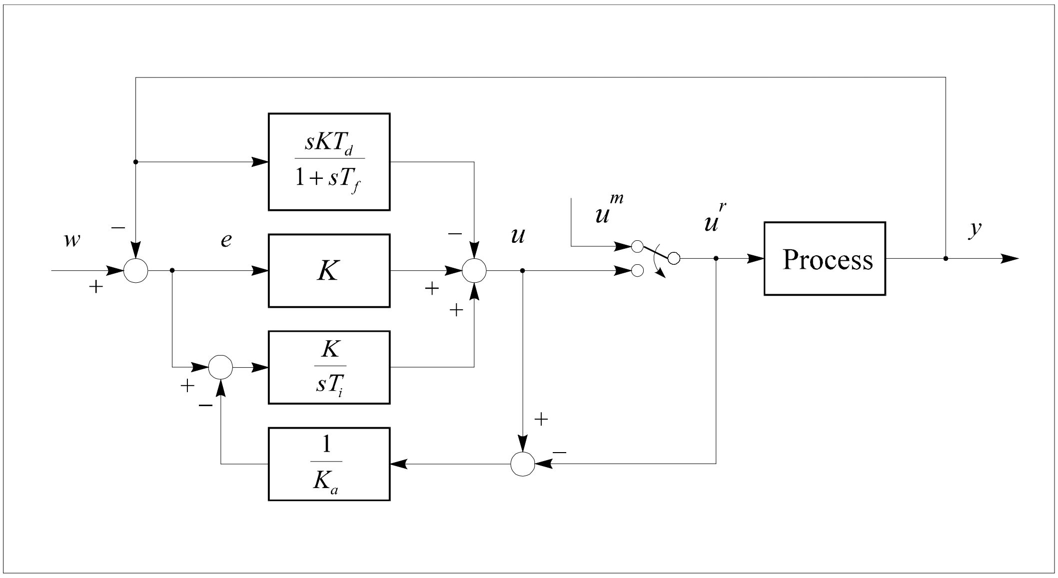

Fig. 2.18. Bumpless and conditioned transfer from manual to automatic mode

For the same situation as the simulation just described above, the result of using BT and CT methods can be seen in Figures 2.19 and 2.20.

For bumpless and conditioned transfer, we choose $K_a \to 0$ and $K_a = K$ respectively, the reason for this choice will be explained in next two sections.

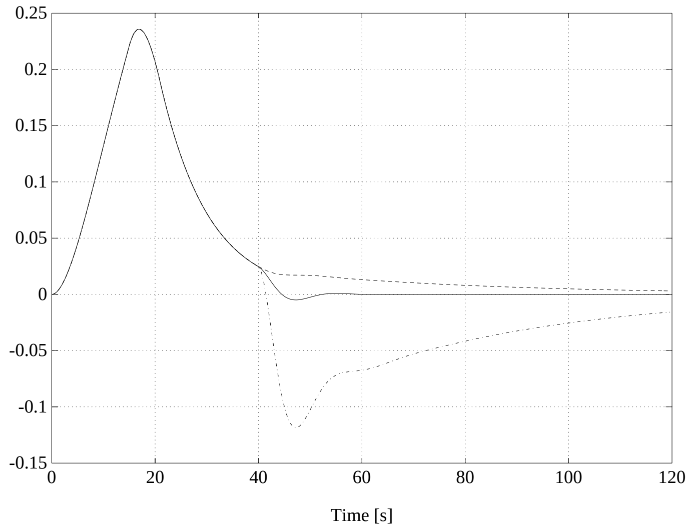

Fig. 2.19. Process output (y);
__ Conditioned transfer, -- Bumpless transfer, -.- Bump transfer

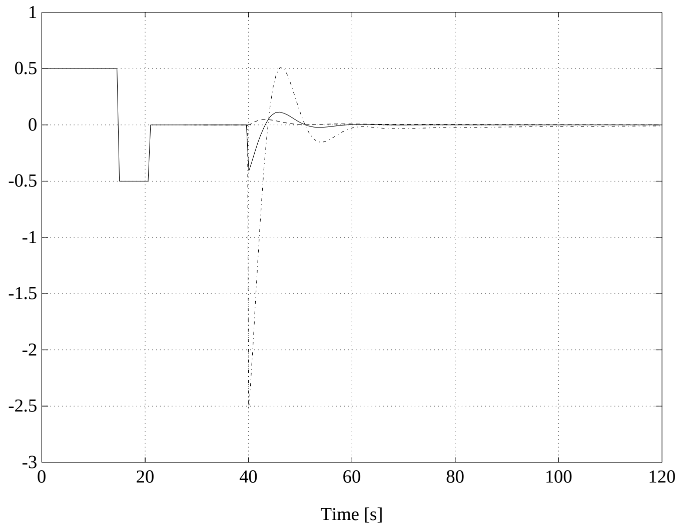

Fig. 2.20. Process input (u^r);
__ Conditioned transfer, -- Bumpless transfer, -.- Bump transfer

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. 2.20), but the settling time of the closed-loop response is quite long (Fig. 2.19).

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.

Both windup and bump are caused by the fact that the controller works in open-loop ($u^r \neq u$) for some time. As a consequence, the integral term is not properly updated. To solve this problem, we can update the integral term by feeding back the difference between u and u^r into controller. The same principle is applied for anti-windup and bumpless or conditioned transfer.

Figure 2.21 shows the way how to realise anti-windup and bumpless or conditioned transfer in the same time. In fact, the solution can be generalised for all other kind of controllers.

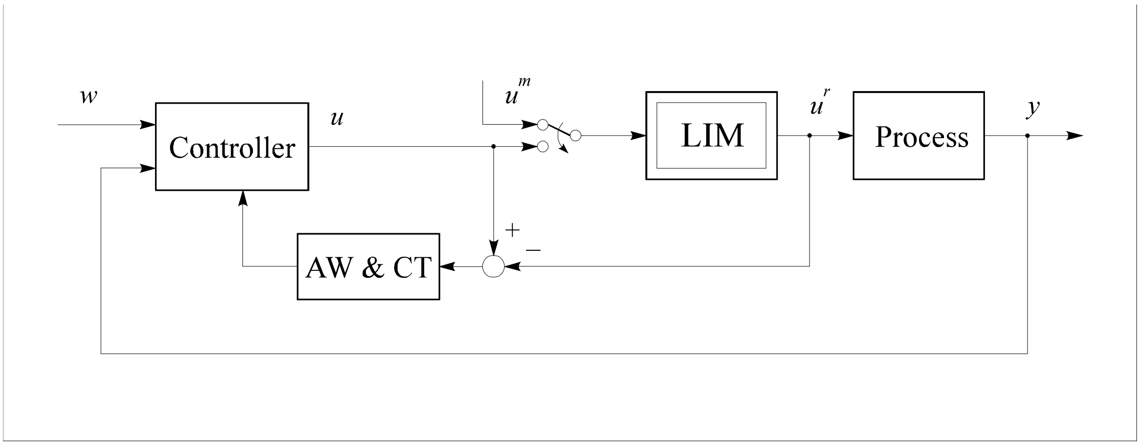

Fig. 2.21. Anti-windup and conditioned transfer