

Chapter 11

Dealing with Constraints

Topics to be covered

An ubiquitous problem in control is that all real actuators have limited authority. This implies that they are constrained in amplitude and/or rate of change. If one ignores this possibility then serious degradation in performance can result in the event that the input reaches a constraint limit. This is clearly a very important problem. There are two ways of dealing with it:

(i) reduce the performance demands so that a linear controller never violates the limits, or

(ii) modify the design to account for the limit.

Here we give a first treatment of option (ii) based on modifying a given linear design. This will usually work satisfactorily for modest violations of the constraint (up to say 100%). If more serious violations of the constraints occur then we would argue that the actuator has been undersized for the given application.

We will also show how the same ideas can be used to avoid simple kinds of state constraints.

Wind-Up

One very common consequence of an input hitting a saturation limit is that the integrator in the controller (assuming it has one) will continue to integrate whilst the input is constrained.

Example 11.1: *Consider the following nominal plant model:*

$$G_o(s) = \frac{2}{(s+1)(s+2)}$$

Say that the target complementary sensitivity is

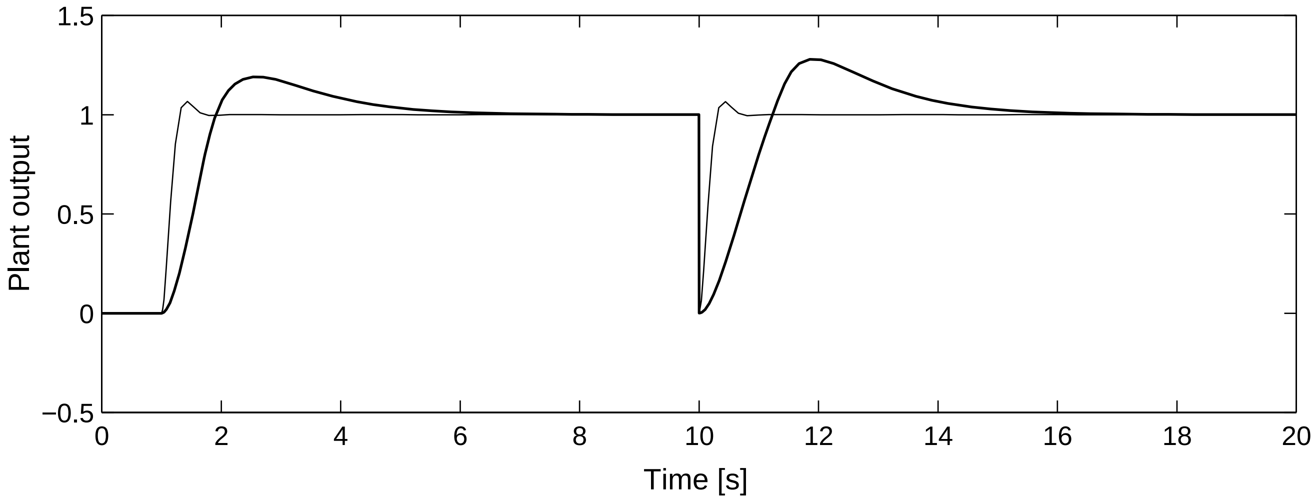
$$T_o(s) = \frac{100}{s^2 + 13s + 100}$$

It is readily seen that this is achieved with the following controller.

$$C(s) = \frac{50(s+1)(s+2)}{s(s+13)}$$

A unit step reference is applied at $t=1$ and a negative unit step output disturbance occurs at $t=10$. The plant input saturates when it is outside the range $[-3, 3]$. The plant output $y(t)$ is shown in Figure 11.1.

Figure 11.1: *Loop performance with (think line) and without (thin line) saturation at the plant input.*



We observe from Figure 11.1 that the plant output exhibits undesirable transient behavior which is inconsistent with the linear nominal bandwidth of approximately 10[rad/s]. This deficiency originates from the saturation, since a unit step in the reference produces an instantaneous demanded change of 50 in the controller output and hence saturation occurs, which a linear design procedure for C(s) does not take into account.

Anti-Windup Scheme

There are many alternative ways of achieving protection against wind-up. All of these methods rely on making sure that the states of the controller have two key properties; namely

(i) the state of the controller should be driven by the actual (i.e. constrained) plant input;

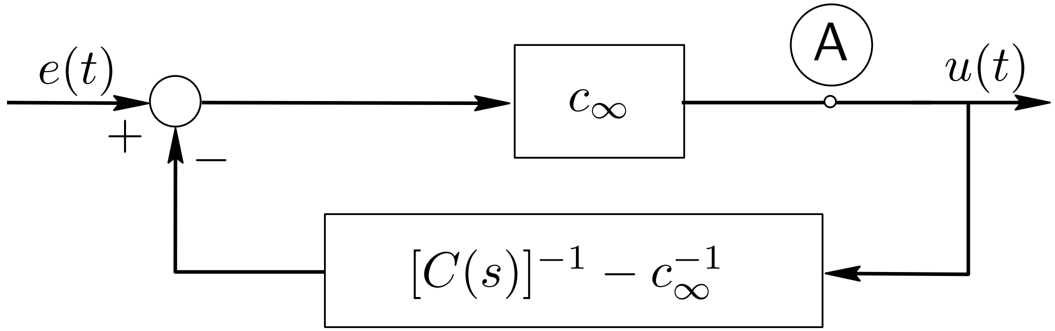
(ii) the states of the controller should have a stable realization when driven by the actual plant input.

This is particularly easy to achieve when the controller is biproper and minimum phase. Say that the controller has transfer function $C(s)$, then we split this into the direct feedthrough term C_∞ and a strictly proper transfer function $C(s)$; i.e.

$$C(s) = c_\infty + \overline{C}(s)$$

Then consider the feedback loop shown in Figure 11.2.

Figure 11.2: *Feedback form of biproper controller*



The transfer function from *e*(*t*) to *u*(t) in Figure 11.2 is readily seen to be

$$\frac{U(s)}{E(s)} = \frac{c_\infty}{1 + ([C(s)]^{-1} - c_\infty^{-1})c_\infty}$$
$$= \frac{c_\infty}{[C(s)]^{-1}c_\infty}$$
$$= C(s)$$

We next redraw Figure 11.2 as in Figure 11.3.

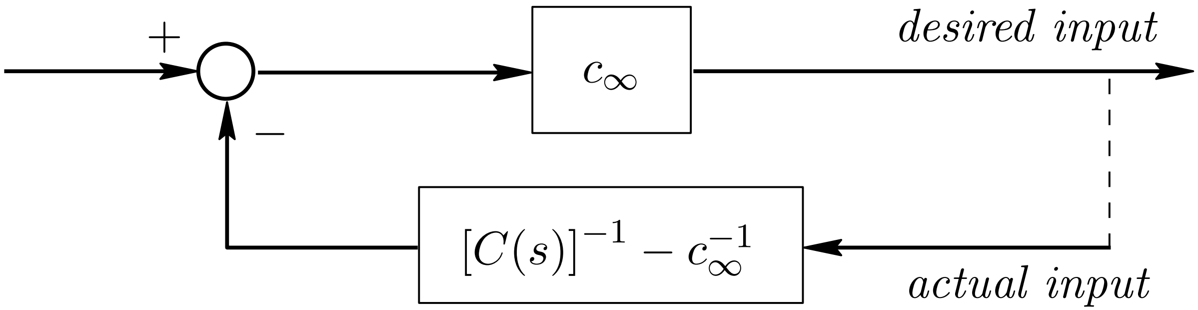


Figure 11.3: Desired and actual plant input

In the case of a limited input, all we now need to do is to ensure that the correct relationship is achieved between the desired and actual input.

Saturation

The appropriate function to describe input saturation is:

$$u(t) = Sat\langle \hat{u}(t) \rangle \triangleq \begin{cases} u_{max} & \text{if } \hat{u}(t) > u_{max}, \\ \hat{u}(t) & \text{if } u_{min} \leq \hat{u}(t) \leq u_{max}, \\ u_{min} & \text{if } \hat{u}(t) < u_{min}. \end{cases}$$

Where $\hat{u}(t)$ is the unconstrained controller output and $u(t)$ is the effective plant input.

Slew Rate Limit

Similarly, we can describe a limit on the rate of change of the input (called *slew rate*) as follows:

$$\dot{u}(t) = Sat\langle \dot{\hat{u}}(t) \rangle \stackrel{\triangle}{=} \begin{cases} \sigma_{max} & \text{if } \dot{\hat{u}}(t) > \sigma_{max}, \\ \dot{\hat{u}}(t) & \text{if } \sigma_{min} \leq \dot{\hat{u}}(t) \leq \sigma_{max}, \\ \sigma_{min} & \text{if } \dot{\hat{u}}(t) < \sigma_{min}. \end{cases}$$

A block diagram realization of a slew rate limiter is shown on the next slide.

Figure 11.4: *Slew rate limit model*

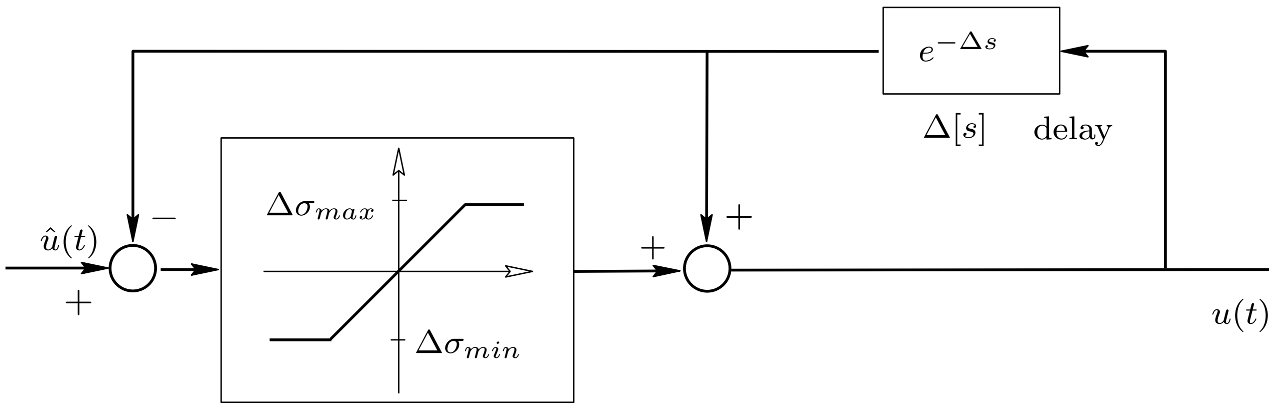


Figure 11.5: *Combined saturation and slew rate limit model*

A slew rate limiter can be combined with a saturation constraint as follows:

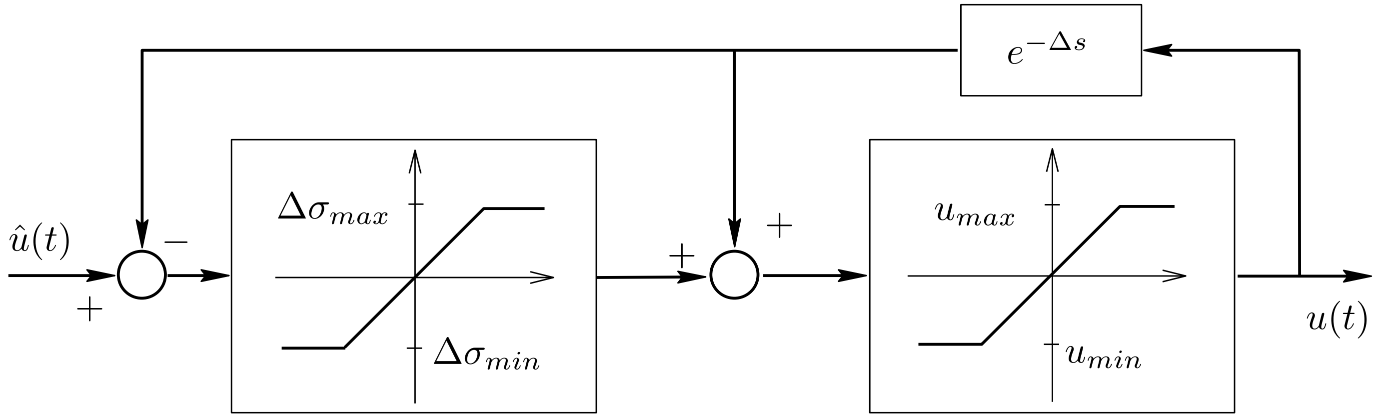
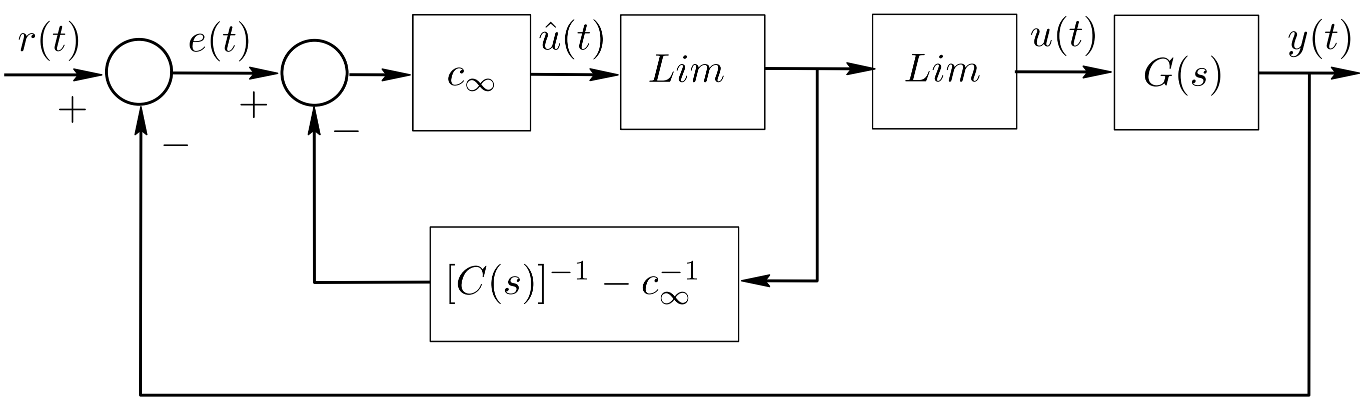


Figure 11.6: *Simplified anti wind-up control loop (C form)*

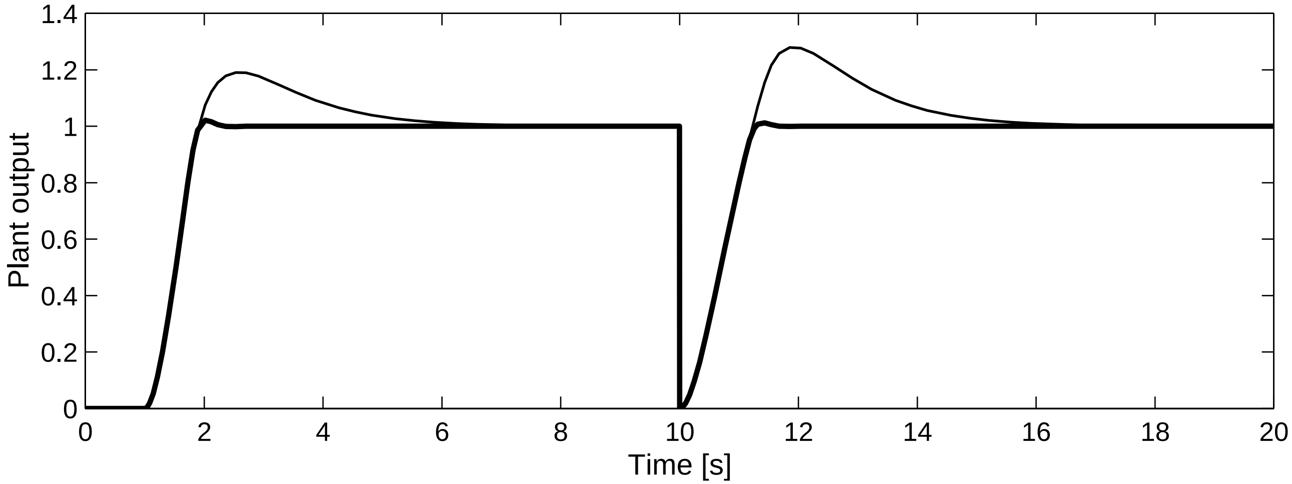
Referring back to Figure 11.3, we can realize an anti-windup compensated controller by placing the appropriate limiter into the block diagram connecting the desired input to the actual (or allowed) input. This leads to the feedback loop shown below:



Example 11.2: *Consider the same plant as in Example 11.1 with identical reference and disturbance conditions. However, this time we implement the control loop using anti-windup protection.*

On running a simulation, the results are shown in Figure 11.7, where the plant output has been plotted.

Figure 11.7: *Loop performance with anti wind-up controller (thick line) compared to performance achieved with no anti wind-up feature (thin line). The latter corresponds to the thick line in Figure 11.1*



A second example having slew rate limits is described below.

Consider a plant having a linear model given by

$$Y(s) = e^{-s}\left(\frac{1}{(s+1)^2}U(s) + D_g(s)\right)$$

Assume that a PI controller with K_P=0.5 and T_r=1.5[s], has been tuned for the linear operating range of this model, i.e., ignoring any nonlinear actuator dynamics.

If the input u(t) cannot change at a rate faster than 0.2[s^{-1}], verify that implementation of the controller as in Figure 11.6 provides better performance than ignoring the slew rate limitation.

Solution

We build a control loop with the controller structure shown in Figure 11.6 (see the next slide) with Lim replaced by the slew rate limiter in Figure 11.4.

$$c_\infty = K_p = 0.5; \qquad [C(s)]^{-1} - c_\infty^{-1} = -\frac{1}{K_p(T_r s + 1)} = -\frac{2}{(1.5s + 1)}$$

Figure 11.6: *Simplified anti wind-up control loop (C form)*

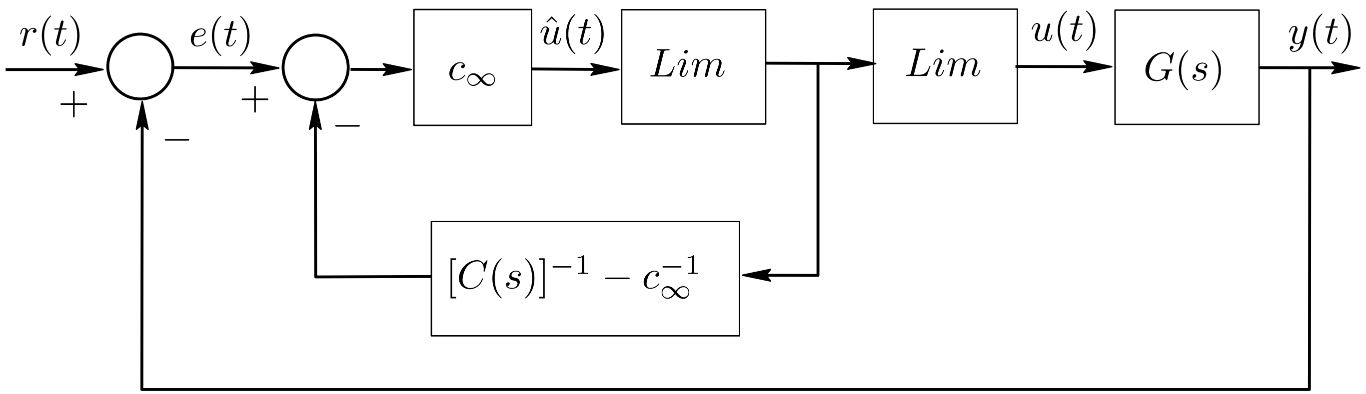
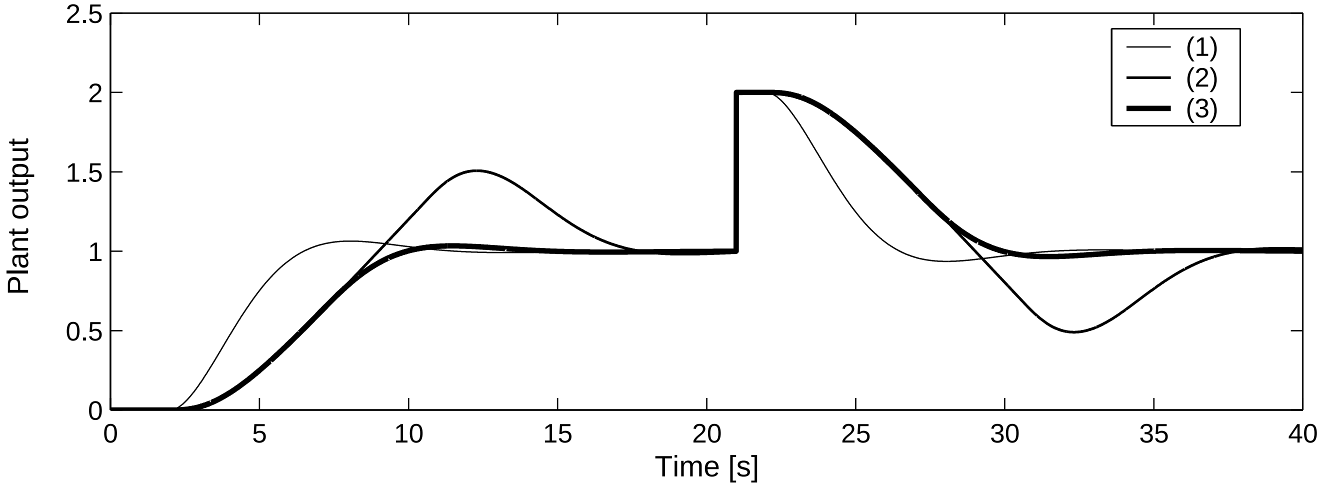


Figure 11.8: *Performance of PI control loop when no slew rate limitation exists (1), with slew rate limitation but no compensation (2) and with anti wind-up for slew rate limitation (3)*



Interpretation in terms of Conditioning

Here we ask the following question: What *conditioned* set-point $\bar{r}$ would have avoided producing an input $\hat{u}$ beyond the limits of saturation in the first place?

We assume that $C(s)$ is biproper and can hence be expanded in terms of its strictly proper and feed-through terms as

$$C(s) = \overline{C}(s) + c_\infty$$

Let us assume that we have avoided saturation up to this point in time by changing $e(t)$ to $\bar{e}(t)$. Then, at the current time, we want to choose $\bar{e}$ so that ...

$$C\langle \overline{e} \rangle = u_{sat} = Sat\langle \overline{C}\langle \overline{e} \rangle + c_\infty e \rangle = \overline{C}\langle \overline{e} \rangle + c_\infty \overline{e}$$

Clearly this requires that we choose $\overline{e}$ as

$$\overline{e} = c_\infty^{-1} \left[Sat\langle \overline{C}\langle \overline{e} \rangle + c_\infty e \rangle - \overline{C}\, \overline{e} \right]$$

The above formula can be represented as in Figure 11.9.

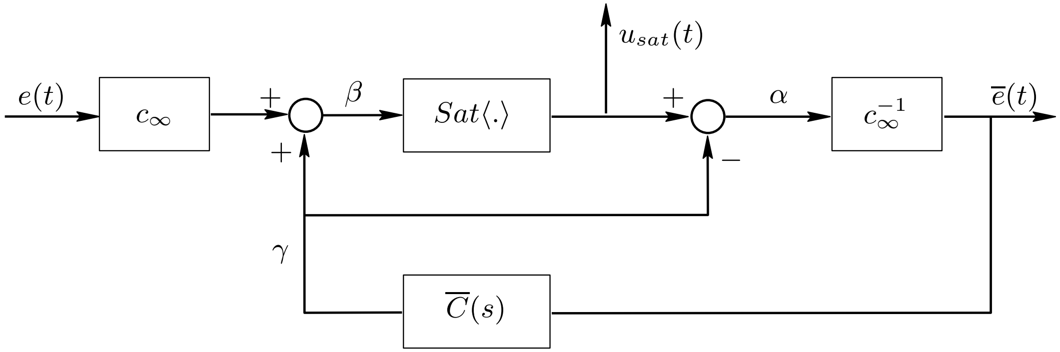


Figure 11.9: *Condition equivalent for the anti wind-up controller*

To show that this is equivalent to the previous design, we note that in Figure 11.9

$$\gamma(t) = c_\infty^{-1}\overline{C}\langle u_{sat}(t) - \gamma(t)\rangle \Leftrightarrow c_\infty\gamma(t) = \overline{C}\langle u_{sat}(t)\rangle - \overline{C}\langle\gamma(t)\rangle$$

From where

$$\gamma(t) = -c_\infty(C^{-1} - c_\infty^{-1})\langle u_{sat}(t)\rangle$$

Also

$$\beta(t) = c_\infty e(t) + \gamma(t) = c_\infty \left(e(t) - (C^{-1} - c_\infty^{-1})\langle u_{sat}(t)\rangle\right)$$

and

$$u_{sat}(t) = Sat\langle\beta(t)\rangle$$
$$u_{sat}(t) = Sat\left\langle c_\infty \left(e(t) - (C^{-1} - c_\infty^{-1})\langle u_{sat}(t)\rangle\right)\right\rangle$$

Hence the scheme in Figure 11.9 implements the same controller as that in Figure 11.6.

State Saturation

As a further illustration of the application of anti-windup procedures, we next show how they can be applied to maintain state limits.

We consider a plant with nominal model given by

$$Y(s) = G_o(s)U(s); \qquad Z(s) = G_{oz}(s)U(s)$$

We achieve state constraints based on switching between two controllers. One of these controllers (the prime controller) is the standard controller aimed at achieving the main control goal, i.e. that the plant output $y(t)$ tracks a given reference, say $r_y(t)$. The task for the secondary controller is to keep the variable $z(t)$ within prescribed bounds. This is achieved by use of a secondary closed loop aimed at the regulation of the estimated state, $\hat{z}(t)$ using a fixed set point.

Our strategy will be to switch between the primary and secondary controller. However, it can be seen that there is a strong potential for wind-up since one of the two controllers, at any one time, will be running in open loop. We will thus implement both controllers in anti-windup form.

For simplicity of presentation, we assume that a bound is set upon $|z(t)|$, i.e. that $z(t)$ is symmetrically bounded.

Both controllers have been implemented as in Figure 11.6. Thus, the prime (linear) controller has a transfer function $C_y(s)$, given by

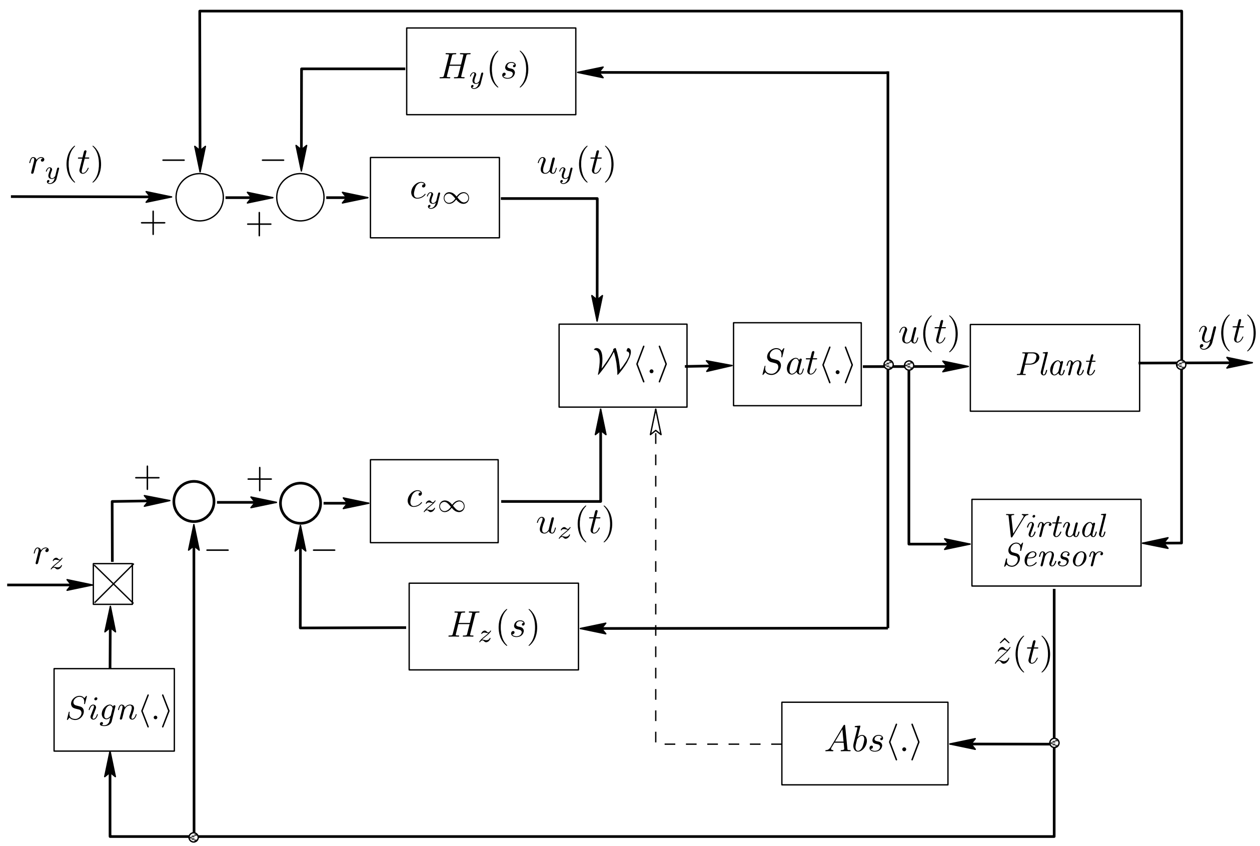
$$C_z(s) = \frac{c_{z\infty}}{1 + c_{z\infty} H_z(s)}; \qquad H_z(s) = [C_z(s)]^{-1} - c_{z\infty}^{-1}$$

Analogously, the secondary (linear) controller has a transfer function $C_z(s)$, given by

$$C_y(s) = \frac{c_{y\infty}}{1 + c_{y\infty} H_y(s)}; \qquad H_y(s) = [C_y(s)]^{-1} - c_{y\infty}^{-1}$$

The final composite controller is shown on the next slide.

Figure 11.10: *Switching strategy for state saturation*



$\mathcal{W}\langle \cdot \rangle$ is a switch which transfers between the two controllers.

Substitutive Switching with Hysteresis

A simple approach is to transfer the generation of the real plant input, $u(t)$ from one controller to the other, in such a way that, at any time, $u(t)$ is determined by either $u_y(t)$ or $u_z(t)$.

If we aim to keep $|z(t)|$ bounded by a known constant $z_{sat} > 0$, then this approach can be implemented using a switch with hysteresis, where the switching levels z_l and z_h, are chosen as $0 < z_l < z_h < z_{sat}$.

Weighted Switching

A switching strategy which is an embellishment of the one described above is described next. It also relies on the use of the switching levels z_1 and z_h, but with the key difference that now the (unsaturated) plant input u(t) is a linear combination of $u_y(t)$ and $u_z(t)$, i.e.

$$u(t) = Sat\langle \lambda u_z(t) + (1-\lambda)u_y(t)\rangle$$

Where λ ∈ [0, 1] is a weighting factor. One way of determining λ would be:

$$\lambda = \begin{cases} 0 & \text{for } |z(t)| \leq z_l \\ \dfrac{|z(t)| - z_l}{z_h - z_l} & \text{for } z_h > |z(t)| \geq z_l \\ 1 & \text{for } |z(t)| > z_h \end{cases}$$

Example 11.4: *Consider a plant with a model*

$$G_o(s) = \frac{16}{(s+2)(s+4)(s+1)}; \qquad G_{oz}(s) = \frac{16}{(s+2)(s+4)}$$

where $Y = G_0 U$ *and* $Z = G_{0z} U$.

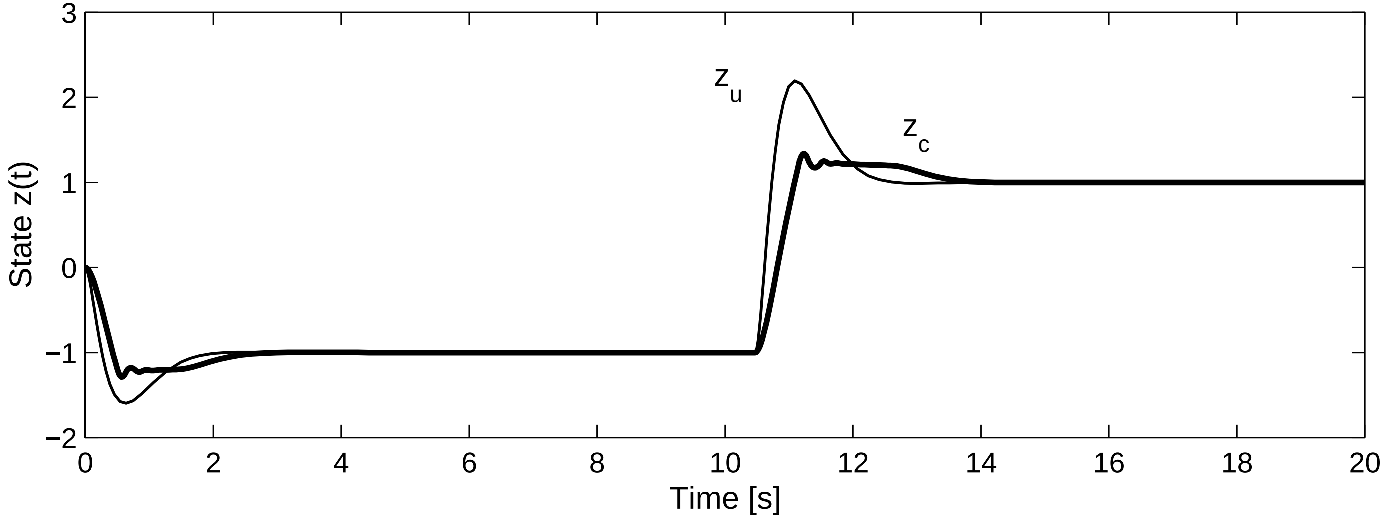
The reference is a square wave of unity amplitude and frequency 0.3[rad/s]. It is desired that the state z(t) does not go outside the range [-1.5; 1.5]. Furthermore, the plant input saturates outside the range [-2; 2].

For this plant, the primary and secondary controller are designed to be

$$C_y(s) = \frac{90(s+1)(s+2)(s+4)}{16s(s^2+15s+59)}; \qquad C_z(s) = \frac{16(3s+10)(s+4)}{s(s+14)}$$

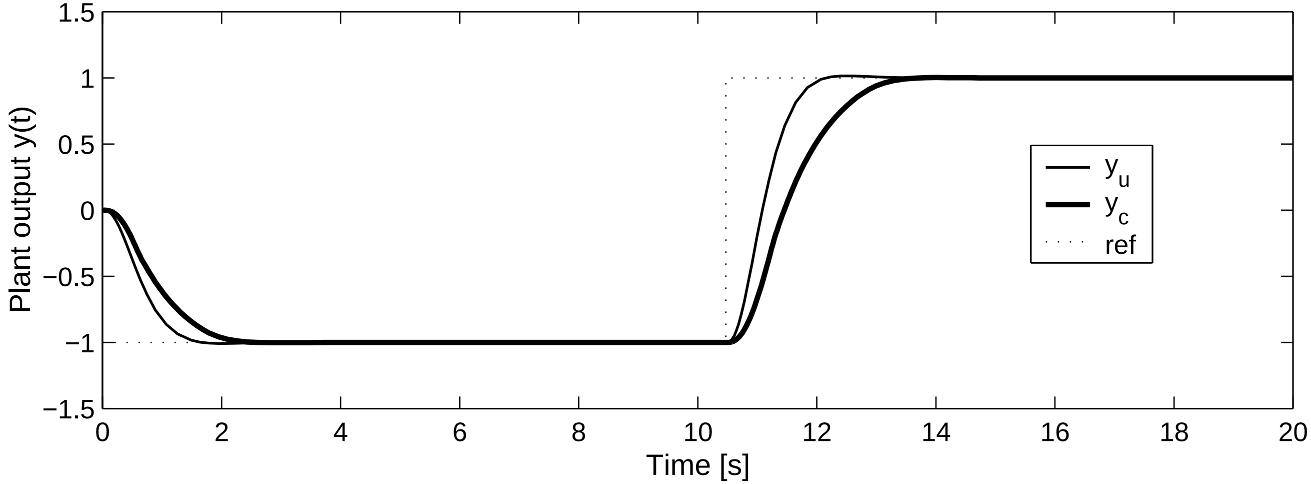
The basic guidelines used to develop the above designs is to have the secondary control loop faster than the primary control loop, so that the state z(t) can be quickly brought within the allowable bounds.

Figure 11.11: *Process variable z(t) (i) with state control saturation (z_c) and (ii) without state control saturation (z_u)*



Note that, whereas the unconstrained state exceeds 1.5 in magnitude, the solution with switched controller leads to *z* not exceeding the desired magnitude constraint.

Figure 11.12: *Plant output with (y_c) and without (y_u) state control saturation*

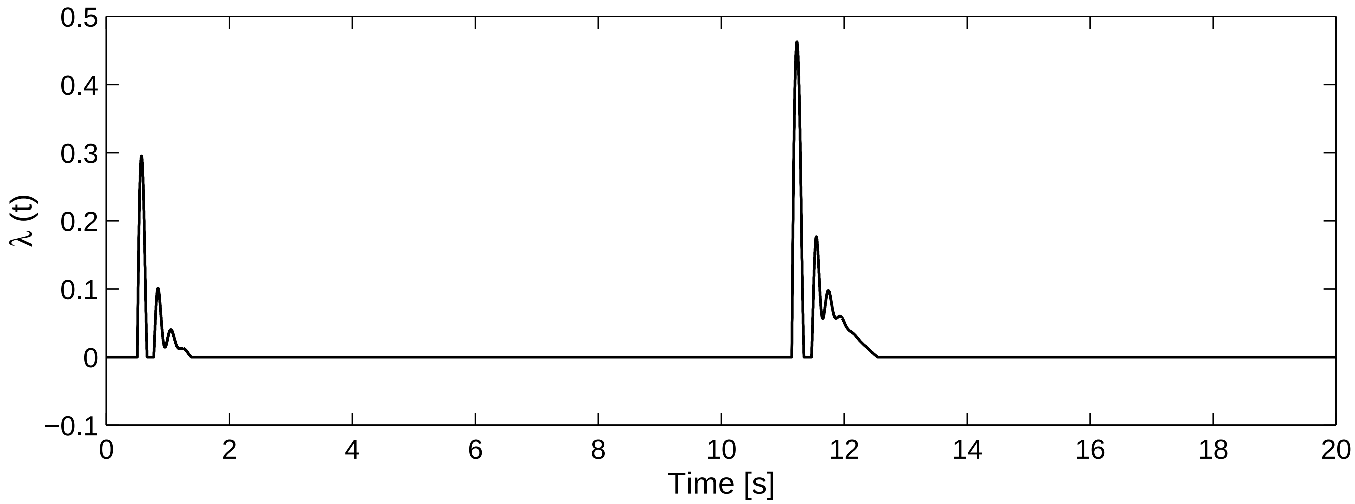


We observe that the effect of imposing the state constraint is to cause the output (y_c) to respond more slowly than when the state is unconstrained (y_u).

The evolution of the weighting factor $\lambda(t)$ is shown in Figure 11.13.

Figure 11.13 shows that the strategy uses a weight which does not reach its maximum value, i.e. the upper level z_h is never reached.

Figure 11.13: *Weighting factor behavior*



Summary

- Constraints are ubiquitous in real control systems
- There are two possible strategies for dealing with them
  - limit the performance so that the constraints are never violated
  - carry out a design with the constraints in mind
- Here we have given a brief introduction to the latter idea

- A very useful insight is provided by the arrangement shown below:

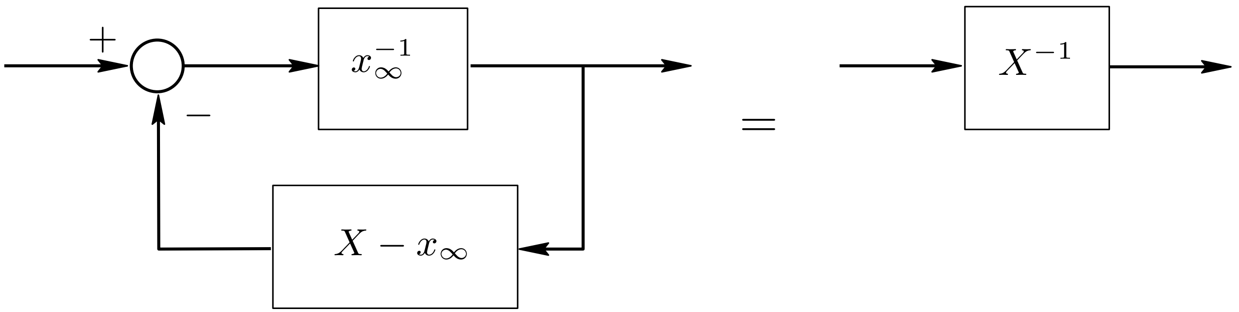


Figure 11.14: *Implicit inversion X^{-1}*

This idea has been used throughout this chapter to generate control strategies incorporating anti-windup protection.