

ELEC ENG 4CL4: Control System Design

Notes for Lecture #24

Wednesday, March 10, 2004

Dr. Ian C. Bruce

Room: CRL-229

Phone ext.: 26984

Email: ibruce@mail.ece.mcmaster.ca

Remedies

We next turn to the question of what remedial action one can take to overcome the kinds of limitations discussed above. Because these are fundamental limits, one really only has two options:

- (i) Live with the limitations but ensure that the design makes the best of the situation in terms of the desired performance goals; or
- (ii) Modify the very nature of the problem by changing the system either through
 - *new sensors*
 - *new actuators, or*
 - *alternative architectural arrangements.*

We will expand on point (ii) next.

Alternative Sensors

If the sensors are a key stumbling block then alternative sensors may be needed. One idea that has great potential in control engineering is to use other sensors to replace (or augment) a poor sensor. When other sensors are used together with a model to infer the value of a missing or poor sensor, we say we have used a *virtual* or *soft* sensor.

Thickness control in rolling mills revisited

We illustrate the use of virtual sensors by returning to Example 8.3 (*Rolling Mill Thickness Control*).

We recall, in that example, that the delay between the mill and thickness measuring device was the source of a fundamental limit in rolling mill thickness performance.

The solution to this problem is to replace the real measurement of exit thickness by a virtual sensor which does not suffer from the delay problem.

Development of Virtual Sensor

The force, F can be related to the thickness h and the roll gap σ via a simple spring equation of the form.

$$F(t) = M(h(t) - \sigma(t))$$

Then an essentially instantaneous estimate of $h(t)$ can be obtained by inverting to give:

$$\hat{h}(t) = \frac{F(t)}{M} + \sigma(t)$$

This estimator for existing thickness is called a *BISRA* gauge and is extremely commonly used in practice.

An alternative virtual sensor

Another possible virtual sensor is described below:

It turns out that the strip width is essentially constant in most mills. In this case, conservation of mass across the roll gap leads to the relationship

$$V(t)H(t) \simeq v(t)h(t)$$

where V , H , v , h denotes the input velocity, input thickness, exit velocity, and exit thickness respectively.

We can estimate the exit thickness from:

$$\hat{h}(t) = \frac{V(t)H(t)}{v(t)}$$

Actuator Remedies

Some potential strategies for mitigating the effect of a given poor actuator include:

- (i) One can sometimes model the saturation effect and apply an appropriate inverse to ensure appropriate control is executed with a poor actuator.
- (ii) One can sometimes put a high gain control loop locally around the offending actuator. This is commonly called *Cascade Control*. (This is discussed further in Chapter 10).
- (iii) One can sometimes arrange the hardware so that the actuator limitation is removed or, at least reduced.

We study below a special way of arranging the control law to mitigate the bad effects of controller saturation.

Anti-Windup Mechanisms

When an actuator is limited in amplitude or slew rate, then one can often avoid the problem by reducing the performance demands. However, in other applications it is desirable to push the actuator hard up against the limits so as to gain the maximum benefit from the available actuator *authority*. This makes the best of the given situation. However, there is a down-side associated with this strategy.

In particular, one of the costs of driving an actuator into a maximal limit is associated with the problem of integral wind-up.

In particular, when the input is saturated the control is constant and hence the error cannot be reduced. Under these conditions, the I term in the PID controller will grow leading to poor transient/response. This is called *wind-up*.

For the moment it suffices to remark that the core idea used to protect systems against the negative effects of *wind-up* is to turn the integrator off whenever the input reaches a limit. This can either be done by a switch or by implementing the integrator in such a way that it automatically turns off when the input reaches a limit.

As a simple illustration, consider the loop given in the figure below:

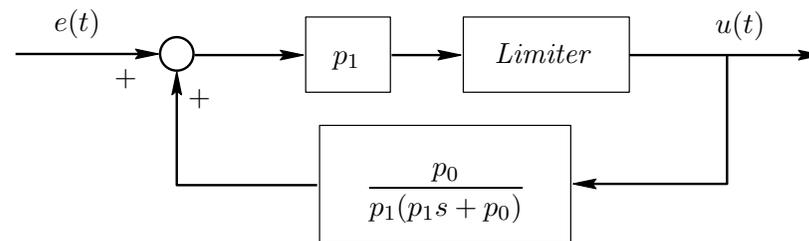


Figure 8.9: *Feedback loop with limiter*

We see that positive feedback using a stable transfer function $\left(\frac{p_0}{p_1(p_1s + p_0)} \right)$ has been used to obtain integral action.

If we replace the limiter by a unity gain, then elementary block diagram analysis shows that the transfer function for e to u is

$$\frac{U(s)}{E(s)} = \frac{p_1 s + p_0}{s}$$

Thus we have a simple PI controller. However, the arrangement shown in Figure 8.9 is a useful way of implementing a PI controller. Specifically, when the controller output is limited, the integrator is removed since the positive feedback loop is open circuited.

As an illustration of what form the limiter in Figure 8.9 might take, we show a particular limiter in Figure 8.10 which when used in Figure 8.9 achieves anti-windup for an input amplitude limit.

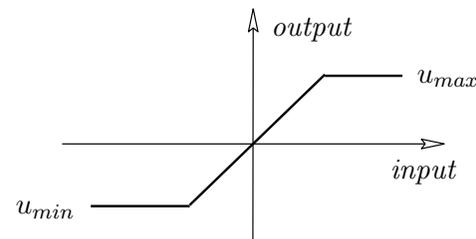


Figure 8.10: *Limiter to achieve saturation*

An alternative limiter that achieves both slew rate and amplitude limits is shown below.

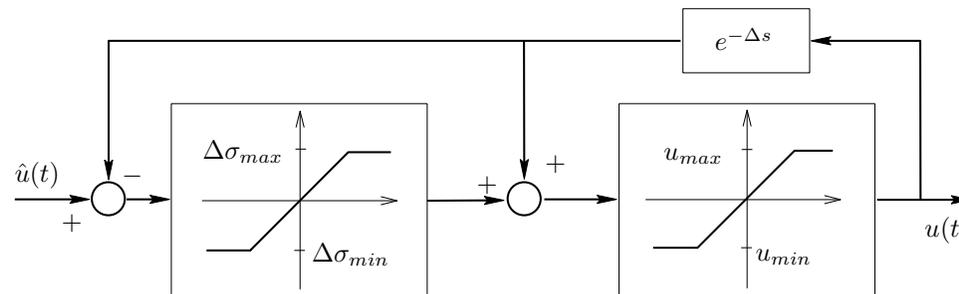


Figure 8.11: *Combined saturation and slew rate limit model*

We will discuss the above kind of anti-windup protection in much greater detail in:

Chapter 11 (*Dealing with Constraints*)

and

Chapter 23 (*Model Predictive Control*).

Remedies for Minimal Actuation Movement

Minimal actuator movements are difficult to remedy. In some applications, it is possible to use *dual-range* controllers wherein a large actuator is used to determine the majority of the control *force* but a smaller actuator is used to give a *fine-trim*.

An example of this is given on the book's web page in relation to pH Control.

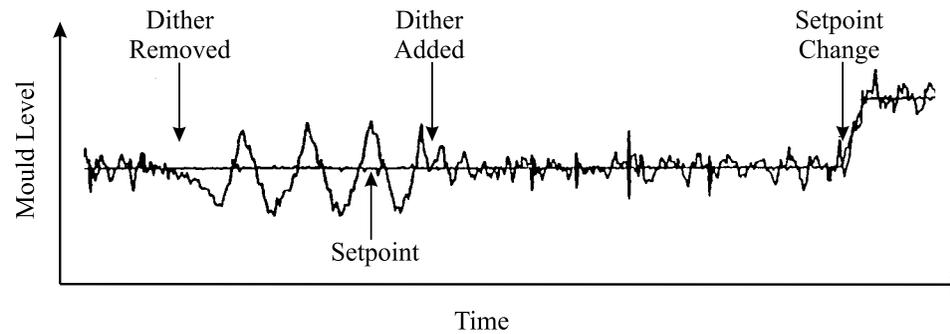
In other applications we must *live* with the existing actuator.

Continuous Caster Revisited

We recall the sustained oscillation problem due to actuator minimal movements described in Example 8.2.

One cannot use dual-range control in this application because a small-high-precision valve would immediately clog with solidified steel. A solution we have used to considerable effect in this application is to add a small high frequency dither signal to the valve. This keeps the valve in motion and hence minimizes stiction effects. The high frequency input dither is filtered out by the dynamics of the process and thus does not have a significant impact on the final product quality. Of course, one does pay the price of having extra wear on the valve due to the presence of the dither signal. However, this cost is off-set by the very substantial improvements in product quality as seen at the output. Some real data is shown in Figure 8.12.

Figure 8.12: *Real data showing effect of adding dither*

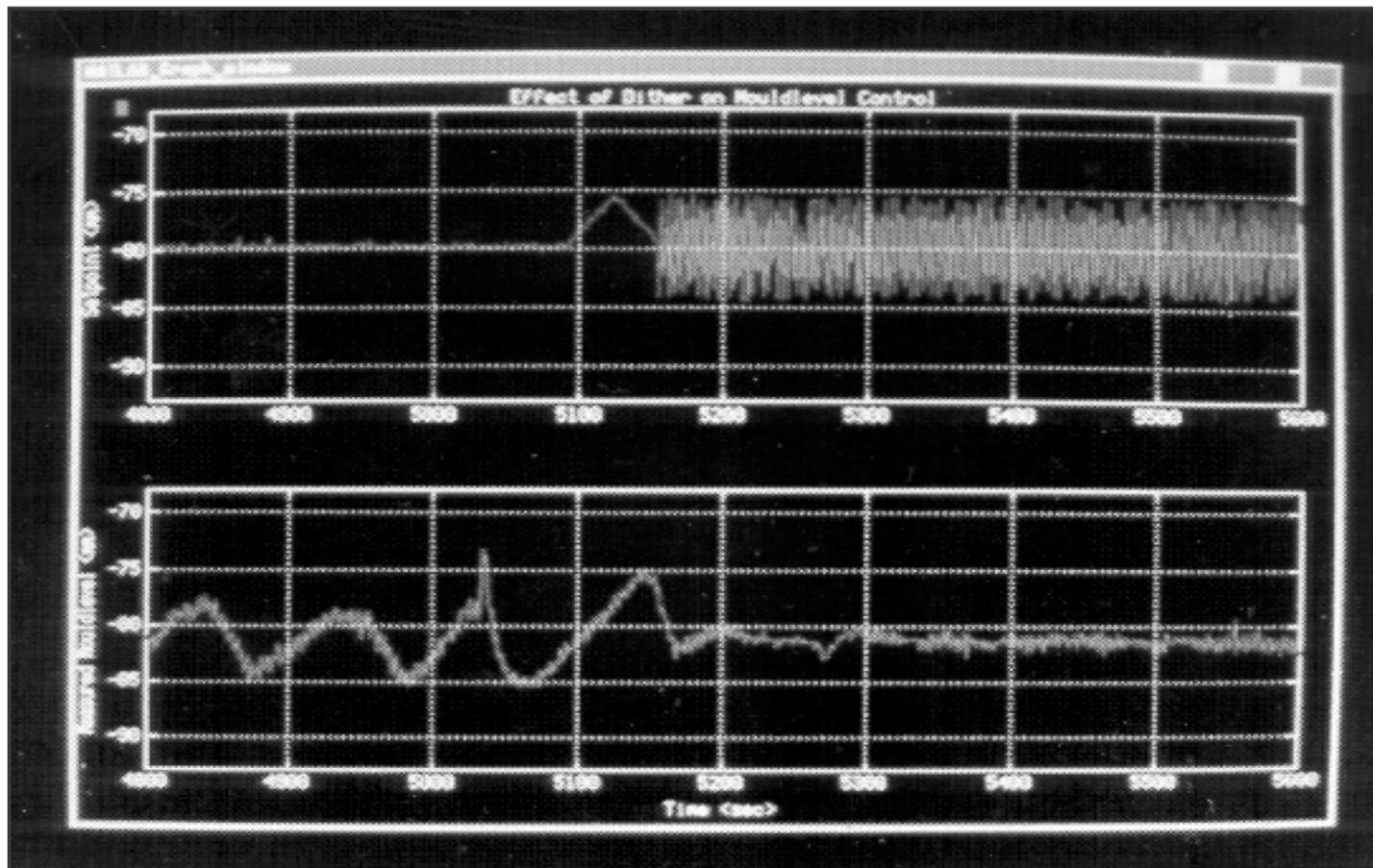


Further Real Data:

Top Trace - External Dither

Bottom Trace - Mould Level

(Note: oscillation disappears once dither applied)



Finally, we turn to the impact of the process itself. We have discussed these limitations under the headings of:

delays

open loop plant poles

open loop plant zeros

The limitations arising from these effects are fundamental **WITHIN THE GIVEN ARCHITECTURE !** This suggests that the one to overcome these limitations is to consider changing the basic architecture of the problem.

Architectural Changes

The fundamental limits we have described apply to the given set-up. Clearly, if one changes the physical system in some way then the situation changes. Indeed, these kinds of change are a very powerful tool in the hands of the control system designer.

The above idea will actually be a central theme as we move forward in these notes. Indeed, we will give many industrial examples of the power of architectural changes. For example, in Chapter 10 we will show how feedforward and cascade loops can dramatically improve performance. We will also see how a simple architectural change can resolve the fundamental problem of the hold-up effect in Rolling Mills (*see earlier in this chapter*).

Simple Illustration:

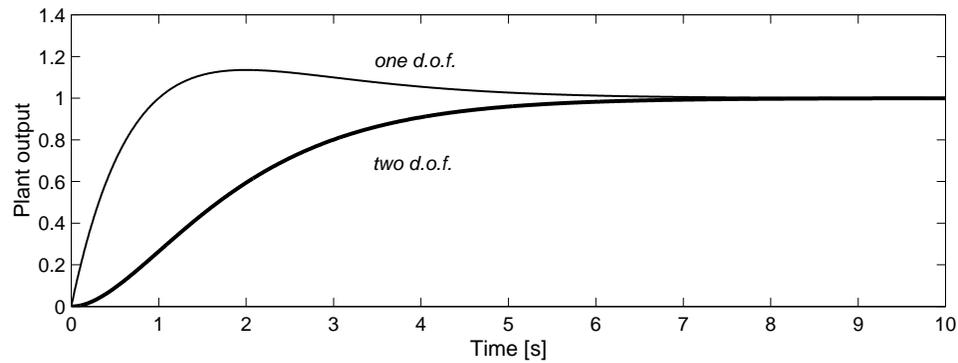
Effect of Two Degree of Freedom Architecture on Closed Loop Response with PI Control

Consider the feedback control of plant with nominal model $G_0(s)$ with a PI controller, $C(s)$, where

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

Then, the closed loop poles are at $(-1; -1)$ and the controller has a zero at $s = -0.5$. Equation (8.6.12) correctly predicts overshoot for the one d.o.f. design. However, if we first prefilter the reference by $H(s) = 1/2s+1$, then no overshoot occurs in response to a unit step change in the reference signal. Figure 8.13 shows the plant output for the one d.o.f. design; this is due to the fact that now the transfer function from $R(s)$ to $E(s) = R(s) - Y(s)$ has only one zero at the origin.

Figure 8.13: *Effects of two d.o.f. architecture*



Design Homogeneity Revisited

We have seen above that limitations arise from different effects. For example, the following factors typically place an upper limit on the usable bandwidth

- ◆ *Actuator slew rate and amplitude limits*
- ◆ *Model error*
- ◆ *Delays*
- ◆ *Right half plane or imaginary axis zeros*

This leads to the obvious question: *which of these limits, if any, do I need to consider?* The answer is that it is clearly best to focus on that particular issue which has the most impact.

This is because the greatest return comes from influencing the most significant factor.

Indeed, in an ideal situation, the final errors due to various sources should all be comparable (*otherwise the possibility exists that one has over expended effort in reducing one source of error when it wasn't dominant*). We call this design *homogeneity*.

Summary

- ❖ This chapter has addressed design issues for SISO feedback loops
- ❖ It has been shown that the following closed loop properties cannot be addressed independently by a (*linear time invariant*) controller:
 - ◆ *speed of disturbance rejection*
 - ◆ *sensitivity to measurement noise*
 - ◆ *accumulated control error*
 - ◆ *required control amplitude*
 - ◆ *required control rate changes*
 - ◆ *overshoot, if the system is open-loop unstable*
 - ◆ *undershoot, if the system is non-minimum phase*
 - ◆ *sensitivity to parametric modeling errors*
 - ◆ *sensitivity to structural modeling errors*

-
- ❖ Rather, tuning for one of these properties automatically impacts on the others.
 - ❖ For example, irrespectively of how a controller is synthesized and tuned, if the effect of the *measurement noise* on the output is $T_0(s)$, then the impact of an *output disturbance* is necessarily $1 - T_0(s)$. Thus, any particular frequency cannot be removed from both an output disturbance and the measurement noise as one would require $T_0(s)$ to be close to 0 at that frequency, whereas the other would require $T_0(s)$ to be close to 1. One can therefore only reject one at the expense of the other, or compromise.

-
- ❖ Thus, a faster rejection of disturbances, is generally associated with
 - ◆ *higher sensitivity to measurement noise*
 - ◆ *less control error*
 - ◆ *larger amplitude and slew rates in the control action*
 - ◆ *higher sensitivity to structural modeling errors*
 - ◆ *more undershoot, if the system is non-minimum phase*
 - ◆ *less overshoot if the system is unstable.*

❖ The trade-offs are made precise by the following fundamental laws of trade-off:

(1) $S_0(s) = 1 - T_0(s)$

that is, an output disturbance is rejected only at frequencies where $|T_0(j\omega)| \approx 1$;

(2) $Y(s) = -T_0(s)D_m(s)$

that is, measurement noise $d_m(t)$, is rejected only at frequencies where $|T_0(j\omega)| \approx 0$;

(3) $S_{u0}(s) = T_0(s)[G(s)]^{-1}$

that is, large control signals arise at frequencies where $|T_0(j\omega)| \approx 1$ but $|G_0(j\omega)| \ll 1$, which occurs when the closed loop is forced to be much more responsive than the open loop process.

-
- (4) $S_{u0}(s) = S_0(s)G_0(s)$
that is, open-loop poles of the process must necessarily either appear as zeros in $S_0(s)$ (*resulting in overshoot when rejecting output step disturbances and additional sensitivity*), or if they are stable, the designer can choose to accept them as poles in $S_{i0}(s)$ instead (*where they impact on input-disturbance rejection*).
- (5) $S(s) = S_0(s)S_\Delta(s)$ where $S_\Delta(s) = (1 + T_0(s)G_\Delta(s))^{-1}$
that is, being responsive to reference changes and against disturbances at frequencies with significant modeling errors, jeopardizes stability; note that the relative (multiplicative) modeling error G_Δ usually accumulates phase and magnitude towards higher frequencies.
- (6) Forcing the closed loop faster than unstable zeros, necessarily causes substantial undershoot.

-
- ❖ Observing the fundamental laws of trade-off ensures that inadvertently specified, but unachievable specifications can quickly be identified without wasted tuning effort.
 - ❖ They also suggest where additional effort is profitable or wasted:
 - ◆ if a design does not fully utilize the actuators and disturbance rejection is poor due to modeling errors (i.e., the loop is constrained by fundamental trade-off law (5), then additional modeling efforts are warranted.
 - ◆ If, on the other hand, loop performance is constrained by non-minimum phase zeros and a constraint on undershoot (i.e., the loop is constrained by fundamental trade-off law (6), then larger actuators or better models would be wasted.

-
- ❖ It is important to note that the design trade-offs
 - ◆ *are fundamental to linear time invariant control*
 - ◆ *are independent of any particular control synthesis methods used.*
 - ❖ However, different synthesis methods
 - ◆ *choose different closed loop properties as their constructed property,*
 - ◆ *therefore rendering different properties as consequential.*
 - ❖ Some design constraints, such as the inverse response due to NMP zeros, exist not only for linear control systems, but also for any other control approach and architecture.

-
- ❖ Remedies for the fundamental limits do exist but they inevitably require radical changes, e.g.
 - ◆ *seeking alternative senses*
 - ◆ *seeking alternative actuators*
 - ◆ *modifying the basic architecture of the plant or controller.*