

Chapter 19

Introduction to Nonlinear Control

This chapter gives a brief introduction to nonlinear control. We will build on the linear methods so as to benefit maximally from linear insights. We also give a *taste* of more rigorous nonlinear theory.

Linear Control of a Nonlinear Plant

An initial question that the reader might reasonably ask is what happens if a linear controller is applied to a nonlinear plant. We know from experience that this must be a reasonable strategy in many cases because one knows that all real plants exhibit some (*presumably*) mild form of nonlinearity and yet almost all real-world controllers are based on linear design.

We will thus pause to analyze the effect of using a linear controller (C) on a nonlinear plant (*which we think of as being composed of a linear nominal part G_0 together with nonlinear additive model error G_ϵ*).

Analysis

In the sequel we will need to mix linear and nonlinear descriptions of systems.

A simple interpretation can be given by the analysis by simply thinking of an operator (such as G_ϵ) as some kind of nonlinear function. Of course, G_ϵ will in general be a dynamical system.

A more rigorous statement of what we mean by an operator G_ϵ is given on the next slide. (*However, this can be skipped on a first reading*).

Formal Definitions

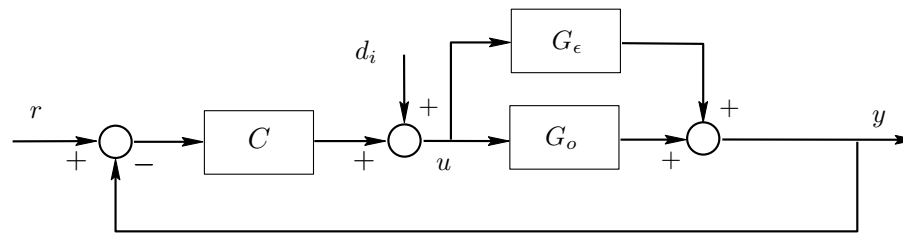
We define a nonlinear dynamic operator f as a mapping from one function space to another. Thus, f maps functions of time into other functions of time. Say, for example, that the time functions have finite energy; then we say that they belong to the space L_2 , and we then say that f maps functions from L_2 into L_2 . We will use the symbol y (*without brackets*) to denote an element of a function space; i.e. $y \in L_2$ is of the form $\{y(t) : \mathbb{R} \rightarrow \mathbb{R}\}$. We also use the notation $y = f(u)$ to represent the mapping (transformation) from u to y via f .

To keep track of the different components of the loop, the controller is represented by a linear operator C and the nominal plant by the linear operator G_0 ; the modeling error will be characterized by an additive nonlinear operator G_ϵ .

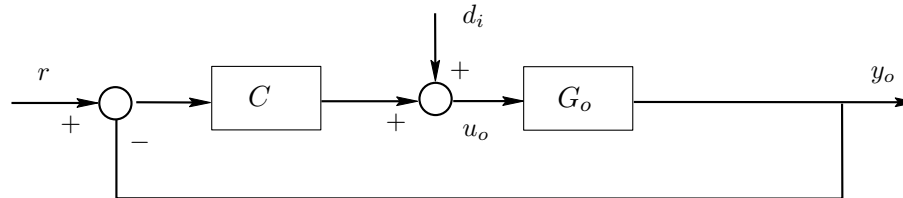
We consider a reference input and input disturbance, because both can be treated in a consistent fashion.

The nominal and the true one-degree-of-freedom loops are shown in Figure 19.1.

Figure 19.1: *True and nominal control loops*



a) True loop



b) Nominal loop

The Effect of Modeling Errors

To simplify the notation, we begin by grouping the external signals at the input by defining

$$d_i' = C\langle r \rangle + d_i$$

Note that this is possible because superposition holds for the linear part of the loop.

From Figure 19.1 (a), we see that

$$\begin{aligned}y &= (G_o + G_\epsilon)\langle u \rangle \\u &= -C\langle y \rangle + d_i'\end{aligned}$$

Similarly, for the nominal loop in Figure 19.1 (b) we have

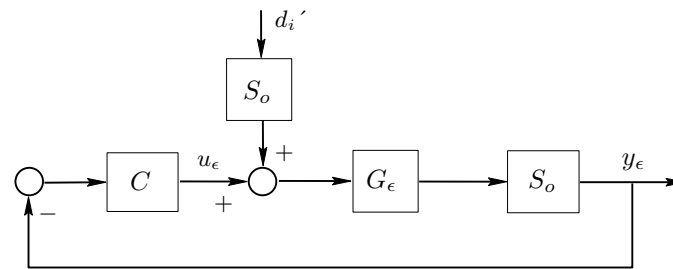
$$\begin{aligned}y_o &= -G_o\langle C\langle y_o \rangle \rangle + G_o\langle d_i' \rangle \\u_o &= -C\langle y_o \rangle + d_i'\end{aligned}$$

As usual, we define linear nominal sensitivities via

$$S_o\langle\circ\rangle = \frac{1}{1 + G_o\langle C\langle\circ\rangle\rangle}$$
$$S_{uo}\langle\circ\rangle = \frac{C\langle\circ\rangle}{1 + G_o\langle C\langle\circ\rangle\rangle}$$

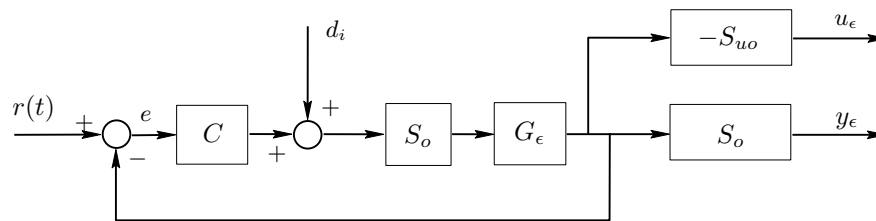
Some simple analysis then shows that the system can be represented as in Figure 19.2, where y_ϵ is the difference between the true loop output, y , and the nominal loop output, y_0 .

Figure 19.2: *Error feedback loop*



Reintroducing r and d_i leads to the final representation, as in Figure 19.3.

Figure 19.3: *Equivalent signal loop describing the effects of modeling errors*



In this figure, G_ϵ is the nonlinear model error. Of course, when G_ϵ happens to be linear, then the above error model reduces to the error model used earlier to describe the impact of linear model errors.

Figure 19.3 is a compact way of depicting the effect of unmodeled plant nonlinearities on the robustness and performance of the feedback loop. Stability, robustness, and performance robustness can be studied by using this representation.

We will next use the *error loop* described above to study the effect of nonlinearities on a linearized control system design.

Example 19.1

Consider the nonlinear plant having its state space model given by

$$\begin{aligned}\frac{dx_1(t)}{dt} &= x_2(t) + (x_2(t))^3 \\ \frac{dx_2(t)}{dt} &= -2x_1(t) - 3x_2(t) + u(t) + 0.1(x_1(t))^2 u(t) \\ y(t) &= x_1(t)\end{aligned}$$

Design Based on Linearized Model

Assume that a linear controller is designed for this plant, and also assume that the design has been based upon a small-signal linearized model. This linearized model has been obtained for the operating point determined by a constant input $u(t) = u_Q = 2$, and the (*linear*) design achieves nominal sensitivities given by

$$T_o(s) = \frac{9}{s^2 + 4s + 9}$$

$$S_o(s) = 1 - T_o(s) = \frac{s(s + 4)}{s^2 + 4s + 9}$$

The operating point is given by

$$x_{1Q} = y_Q \triangleq \lim_{t \rightarrow \infty} x_1(t) = 1.13 \quad \text{and} \quad x_{2Q} \triangleq \lim_{t \rightarrow \infty} x_2(t) = 0$$

The values for u_Q , x_{1Q} , and x_{2Q} are then used to obtain the linearized model $G_o(s)$, which is given by

$$G_o(s) = \frac{1.13}{s^2 + 3.0s + 1.55}$$

To achieve the desired $T_0(s)$, we have that the controller transfer function must be given by

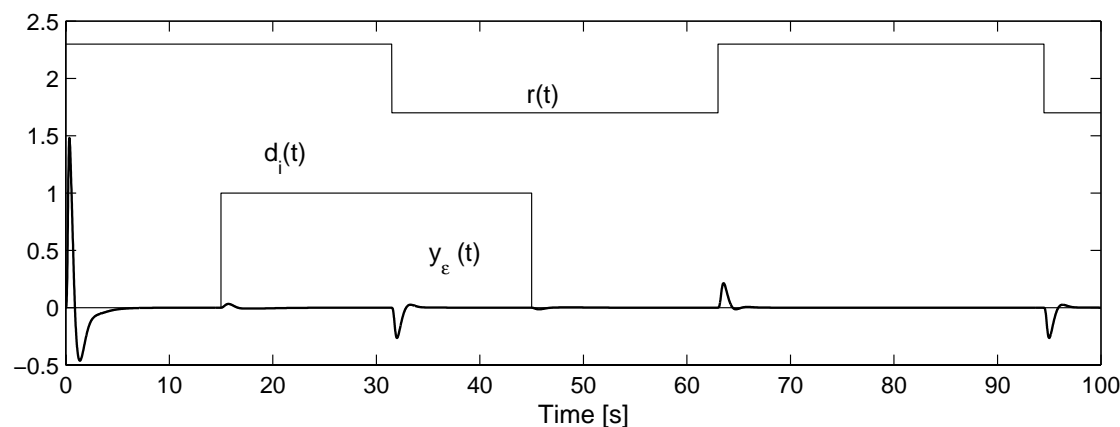
$$C(s) = [G_o(s)]^{-1} \frac{T_o(s)}{S_o(s)} = 7.96 \frac{s^2 + 3.0s + 1.55}{s(s + 4)}$$

We use SIMULINK to implement the *error loop* block diagram shown in Figure 19.3, where $S_{u0}(s)$ is given by

$$S_{u0}(s) = [G_o(s)]^{-1}T_o(s) = 7.96 \frac{s^2 + 3.0s + 1.55}{s^2 + 4s + 9}$$

A simulation is run with a reference, $r(t)$, which has a mean value equal to 2 and a superimposed square-wave of amplitude 0.3. A pulse is added as an input disturbance. Figure 19.4 on the next slide shows the plant output error, $y_{\epsilon}(t)$.

Figure 19.4: *Effects of linear modeling of a nonlinear plant on the performance of linear-based control design.*



linear model of a nonlinear plant] effects of linear modeling of a nonlinear plant on the performance of linear based control design.

We see from the above plot that, for this example, and for these particular reference signals and disturbances, the output error $y_{\epsilon}(t)$ is quite small. This indicates that the design based on the linearized model is probably adequate here.

We see from the above example that linear designs can perform well, at least, on some nonlinear plants, provided that the nonlinearity is sufficiently small. However, as performance demands grow, one is inevitably forced to carry out an inherently nonlinear design. A first step in this direction is described in the next section, where we still use a linear controller, but a different linear controller is chosen at different operating points.

Switched Linear Controllers

One useful (*engineering*) strategy for dealing with nonlinear systems is to split the state space up into small regions inside which a localized linear model gives a reasonable approximation to the response. One can then design a set of fixed linear controllers - one for each region. Two issues remain to be solved:

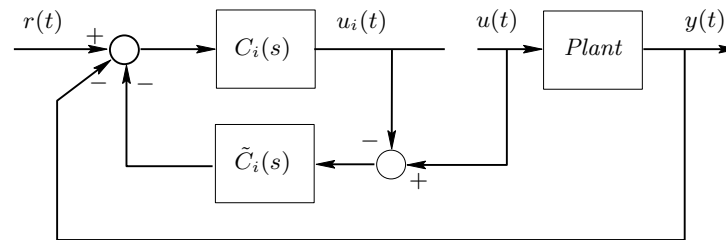
- (i) *how to know which region one is in, and*
- (ii) *how to transfer between controllers.*

The first problem above is often resolved if there exists some measured variable that is a key indicator of the dynamics of the system. These variables can be used to *schedule* the controllers. For example, in high-performance aircraft control, Mach number and altitude are frequently used as scheduling variables.

The second problem requires that each controller run in a stable fashion regardless of whether it is *in charge* of the plant. This can be achieved by the anti-wind-up strategies described in Chapter 11.

An alternative architecture for an anti-wind-up scheme is shown in Figure 19.5. Here, a controller ($\tilde{C}_i(s)$) is used to cause the i^{th} controller output to track the true plant input. To describe how this might be used, let us say that we have k_c controllers and that switching between controllers is facilitated by a device that we call a *supervisor*. Thus, depending on the state of the system at any given time instant, the *supervisor* may select one of the k_c controllers to be the *active controller*. The other $k_c - 1$ controllers will be in *standby mode*.

Figure 19.5: *Bumpless controller transfer*



We wish to arrange *bumpless transfer* between the controllers as directed by supervisor.

To cope with general situations, we place each of the standby controllers under separate feedback control, so that their outputs track the output of the active controller. Here we use the scheme previously illustrated in Figure 19.5.

In Figure 19.5, $u(t)$ denotes the active controller output (*this controller is not shown in the diagram*), and $C_i(s)$ denotes a standby controller having reference signal $r(t)$ for the particular plant output $y(t)$. In the figure, $\tilde{C}_i(s)$ denotes the controller for the i^{th} controller, $C_i(s)$. Thus, in a sense, $\tilde{C}_i(s)$ is a *controller-controller*. This controller-controller is relatively easy to design: the plant in this case is well known, because it is actually the i^{th} controller. For this controller-controller loop, the signals $r(t)$ and $y(t)$ act as disturbances.

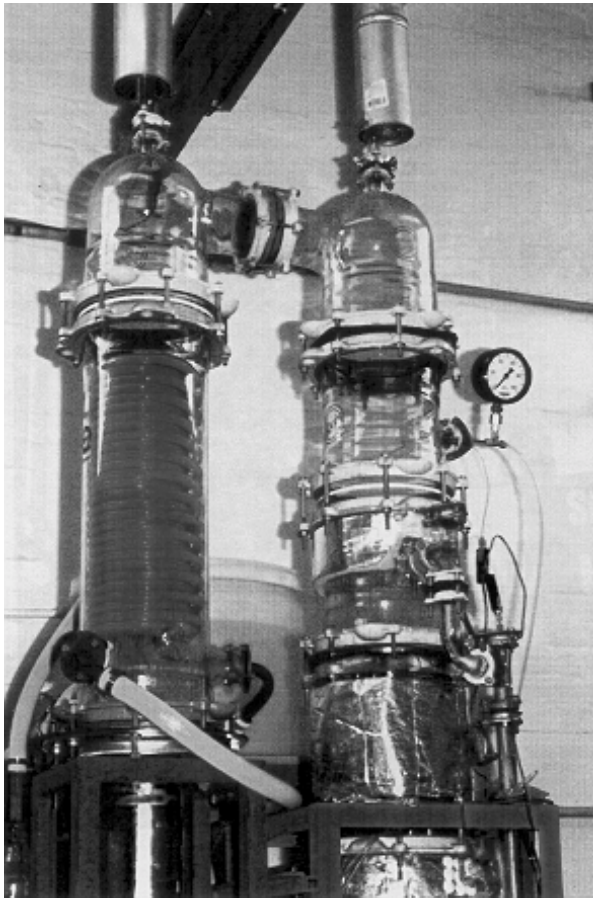
Pilot Scale Distillation Column Example

To illustrate the use of switched linear control of a nonlinear plant, we will take the example of a pilot scale distillation column. A photo is shown, on the next slide, of the plant which is in the Department of Chemical Engineering at the University of Sydney.

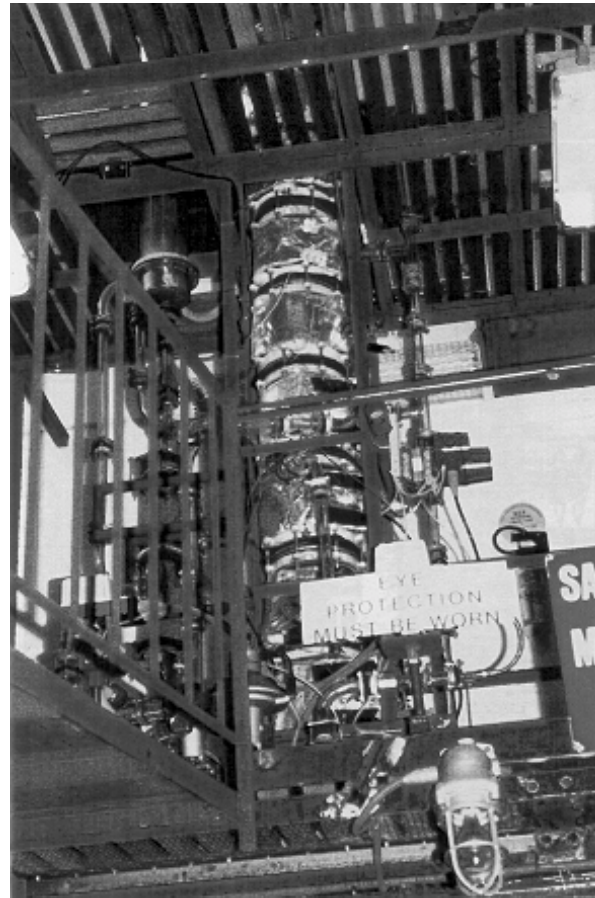
The work on multi-model control of this plant was carried out by Julio Rodriguez as part of his Ph.D. studies.

Pilot Binary Distillation Column

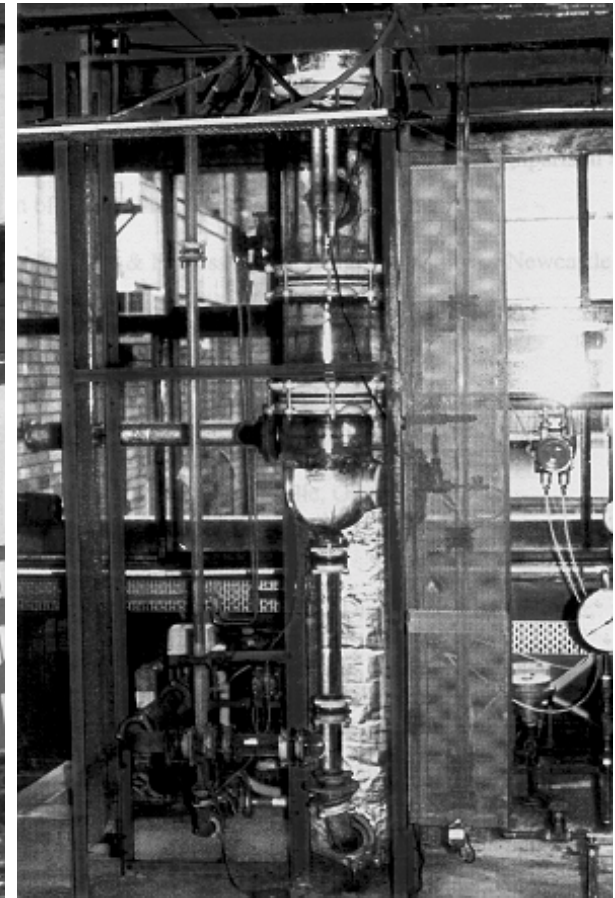
Condenser



Feed-point



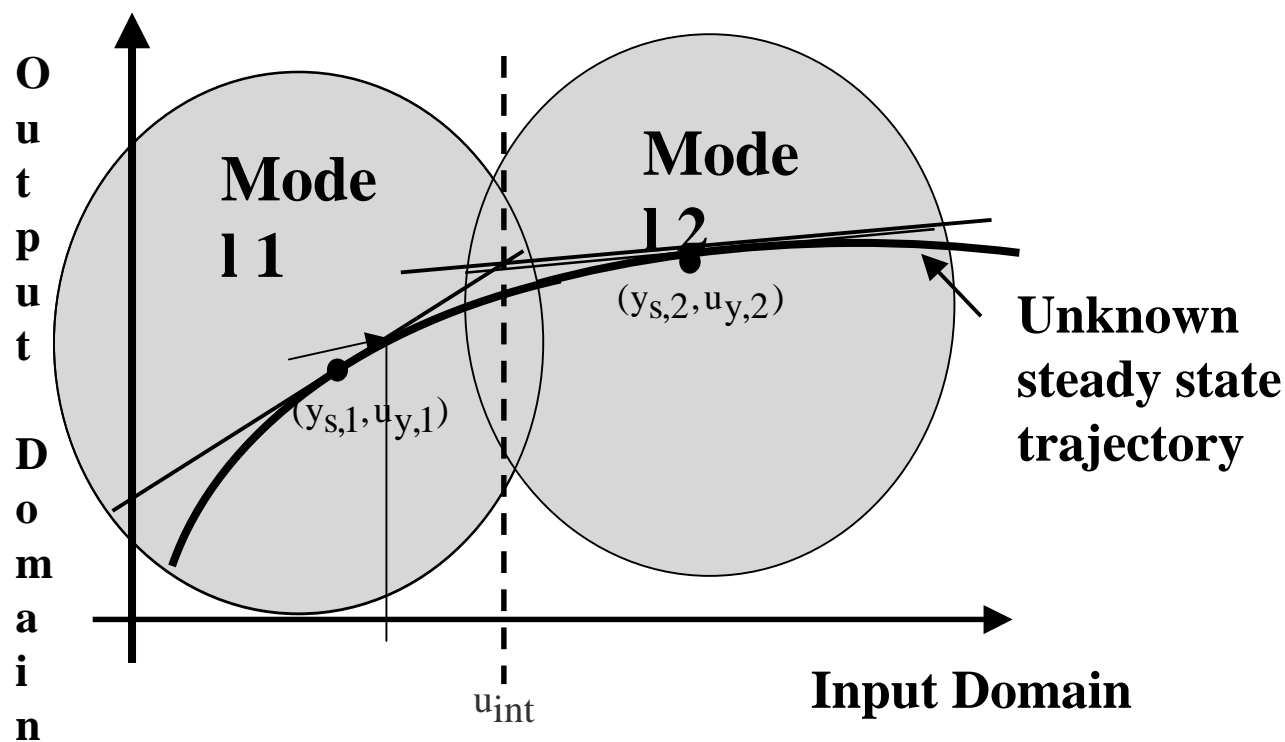
Reboiler



For this system, 2 linear controllers were fitted to 2 different operating points - see the next slide.

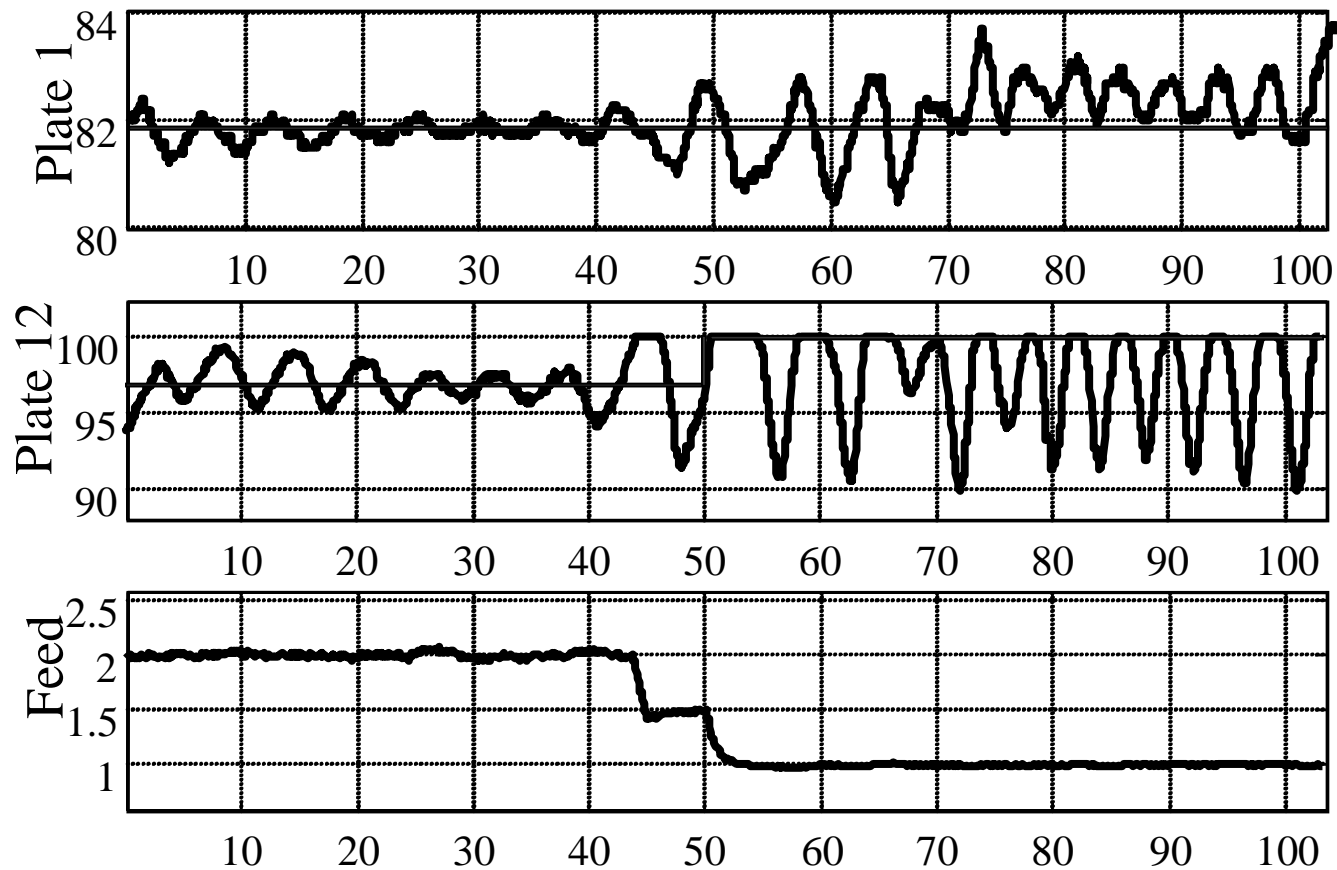
Multiple Model Modelling

The global model can be represented as an LPV system, given by

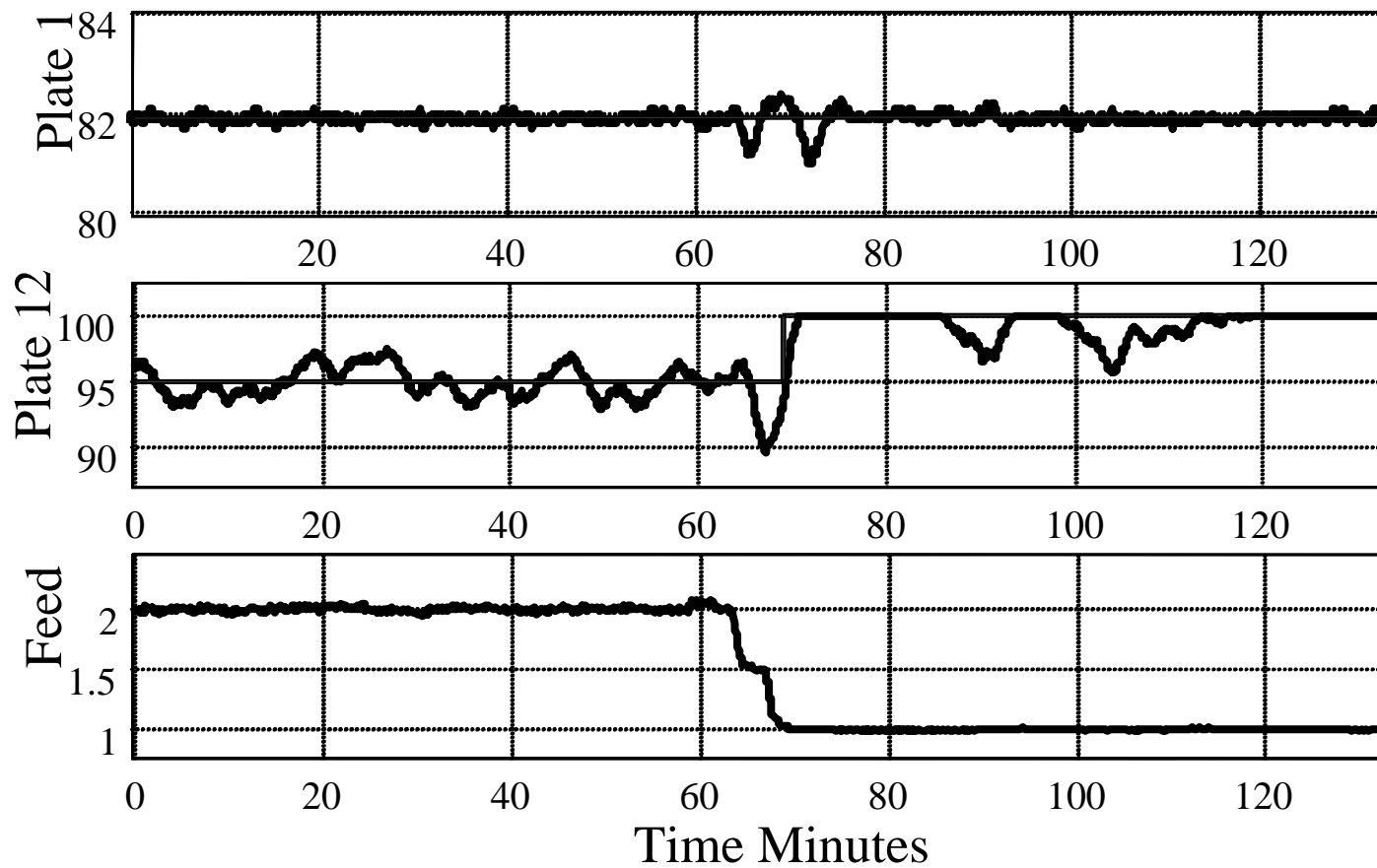
$$\begin{aligned}\dot{x} &= A(\lambda)x + B(\lambda)u \\ y &= C(\lambda)x + D(\lambda)u\end{aligned}$$


The performance achieved with a single fixed linear controller was then compared with the performance achieved with a switched linear controller comprising 2 linear controllers designed at 2 different operating points.

Results for single linear controller are shown below.



Results for switched linear controller are shown below.



Comparing the above two results we see the superior performance achieved by the switching control law.

Control of Systems with Smooth Nonlinearities

When the nonlinear features are significant, the above switching could lead to a conservative design. If this is the case, inherently nonlinear strategies should be considered in the design. To highlight some interesting features of nonlinear problems, we will begin by considering a simple scenario in which the model is both stable and stably invertible. We will later examine more general problems.

Static Input Nonlinearities

We will first examine the nonlinear control problem as an extension of the ideas in Chapter 15.

Basically, we mirror the idea of the “ Q ” form of all stabilizing controllers for a stable system. This leads to the nonlinear version of this scheme shown on the next slide.

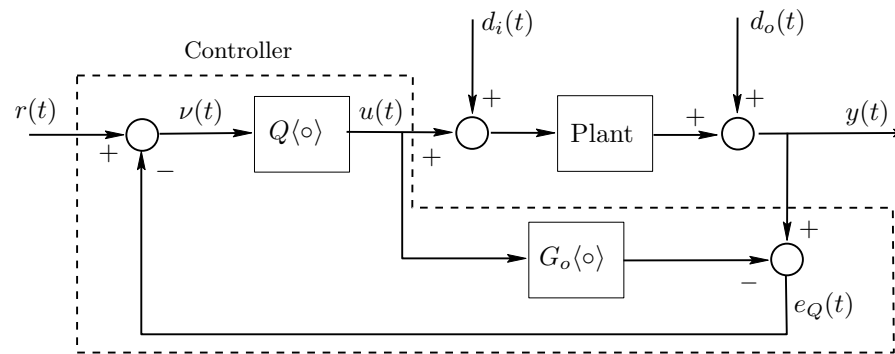


Figure 19.6: *IMC architecture for control of (smooth) nonlinear systems*

Note that this looks like the “ Q ” controller described in the linear case - here however, the various operators may be nonlinear.

We also recall, from Chapter 15, that, in the linear case of stable and minimum phase plants, $Q(s)$ was chosen as

$$Q(s) = F_Q(s)[G_o(s)]^{-1}$$

In the linear case, this leads to the result

$$y(t) = F_Q\langle\nu(t)\rangle$$

The same basic principle can be applied to nonlinear systems. In the sequel we describe several methods for obtaining approximate inverses for nonlinear dynamic systems.

Static Input Nonlinearities

The simplest situation occurs when the plant has a static input nonlinearity, and there is only an output disturbance; then the nominal model output is given by

$$y(t) = G_o \langle u(t) \rangle + d_o(t) = \bar{G}_o \langle \phi(u) \rangle + d_o(t)$$

Note that static input nonlinearities are quite common in practice since they correspond to known nonlinear behaviour of input actuators (*e.g. valves, etc.*)

Introducing \bar{F}_Q as a linear stable operator of appropriate relative degree, the approximate nonlinear inverse is simply obtained in this case by inverting the static nonlinearity. This leads to:

$$Q\langle\circ\rangle = \phi^{-1} (\bar{G}_o^{-1} \langle \bar{F}_Q \langle \circ \rangle \rangle)$$

Hence,

$$u(t) = \phi^{-1} (\bar{G}_o^{-1} \langle \bar{F}_Q \langle \nu(t) \rangle \rangle)$$

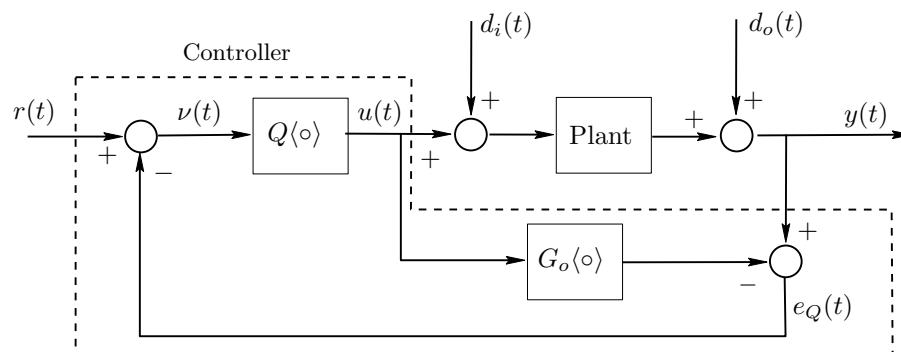
We see that the static nonlinear function, ϕ , is cancelled between the control law and the plant. Hence, the final control loop performance as a linear loop.

We next consider more complex situations where the dynamics of the system also contain nonlinear elements.

Smooth Dynamic Nonlinearities for Stable and Stably Invertible Models

For this class of systems, we again use the IMC architecture on the next slide. The parallel model is easily constructed. However, some attention needs to be focussed on the approximate inverse required in the function $Q(\circ)$.

Figure 19.6: *IMC architecture for control of (smooth) nonlinear systems*



Approximate Inversion for Models Having Stable Inverses

We consider a plant having a nominal model of the form

$$\begin{aligned}\rho x(t) &\triangleq \frac{dx(t)}{dt} = f(x) + g(x)u(t) \\ y(t) &= h(x)\end{aligned}$$

We recall that in the linear case that to develop an approximate inverse, it was necessary to know the relative degree. The nonlinear version of this idea is given in the next slide.

Nonlinear Relative Degree

Definition 19.1: The relative degree of the system is the minimum value $\eta \in \mathbb{N}$ such that if we differentiate the system output η times then the input appears explicitly, i.e. we have

$$\rho^\eta y(t) = \beta_\eta(x) + \alpha_\eta(x)u(t)$$

where $\alpha_\eta(x)$ is not identically zero.

Example 19.2

Consider a system having the model

$$\rho x_1(t) = -x_1(t) - 0.1(x_2(t))^2$$

$$\rho x_2(t) = -2x_1(t) - 3x_2(t) - 0.125(x_2(t))^3 + \left[1 + 0.1(x_1(t))^2\right] u(t)$$

$$y(t) = x_1(t)$$

Then, if we compute the first- and second-time derivatives of $y(t)$, we obtain

$$\begin{aligned}\rho y(t) &= \rho x_1(t) = -x_1(t) - 0.1(x_2(t))^2 \\ \rho^2 y(t) &= 1.4x_1(t) + 0.6x_2(t) + 0.1x_2(t)^2 + 0.25x_2(t)^3 \\ &\quad - \left[0.2 + 0.02x_1(t)^2\right] u(t)\end{aligned}$$

from which we see that the relative degree of the system is equal to 2.

Note that, for the system

$$\rho x(t) \triangleq \frac{dx(t)}{dt} = f(x) + g(x)u(t)$$
$$y(t) = h(x)$$

having relative degree η , we have that, if the i^{th} derivative of $y(t)$, $i=0, 1, 2, \dots, \eta$, takes the form

$$\rho^i y(t) = \beta_i(x) + \alpha_i(x)u(t),$$

then $\alpha_i(x) \equiv 0$ for $i = 0, 1, 2, \dots, \eta - 1$.

Consider now the operator polynomial $p(\rho) = \sum_{i=0}^{\eta} p_i \rho^i$; then we see that $p(\rho)y(t)$ can be expressed as

$$p(\rho)y(t) = b(x) + a(x)u(t)$$

where

$$b(x) = \sum_{i=0}^{\eta} p_i \beta_i(x) \quad \text{and} \quad a(x) = p_{\eta} \alpha_{\eta}(x)$$

Final Construction of the Approximate Inverse

The approximate inverse for the plant is finally obtained simply by setting $p(\rho)y(t)$ equal to $v(t)$. This leads (*after changing the argument of the equation*) to the following result for $u(t)$:

$$u(t) = (a(x))^{-1} (\nu(t) - b(x)) = Q\langle\nu\rangle$$

Substituting this into the original model shows that the end result of using this control law is that the output satisfies:

$$p(\rho)y(t) = \nu(t) \iff y(t) = [p(\rho)]^{-1}\langle\nu\rangle$$

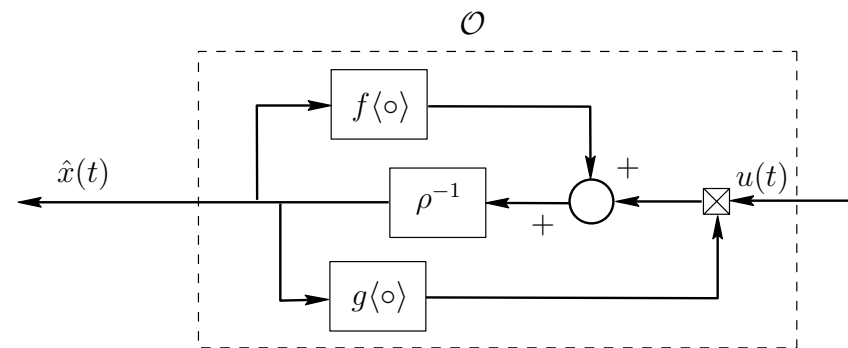
Note that, to obtain a perfect inverse at d.c., we set $p_0 = 1$.

The control strategy described above is commonly known as *input-output feedback linearization*, because it leads to a linear closed-loop system from the input $v(t)$ to the output $y(t)$.

Actually, we see that the final closed loop has only *denominator* dynamics (*i.e.* $y(t) = \frac{v(t)}{p(\rho)}$). Thus, the control law has cancelled the *numerator* dynamics of the plant. Hence a major restriction on the use of this method is that the system must be stably invertible (*i.e. have stable numerator dynamics*).

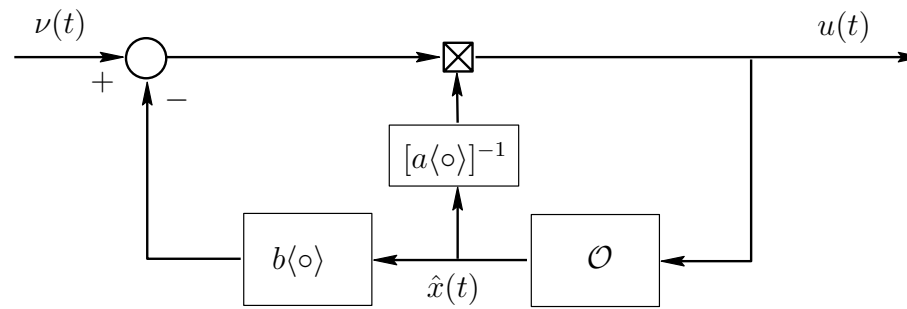
A remaining issue is how to implement $Q\langle\circ\rangle$ because it depends on the plant state, which is normally unavailable. Following the same philosophy as in the linear case, we can estimate the state by means of a nonlinear state observer. For the moment, we are, assuming that the plant is open-loop stable, so we can use an open-loop observer. An open-loop observer is driven only by the plant input $u(t)$. It is basically an open loop model of the system and can be represented as shown on the next slide, where ρ^1 denotes the integral operator.

Figure 19.7: *Nonlinear (open loop) observer*



With this observer, we can build the Q block, using the feedback linearization scheme. This leads to the final implementation of Q as shown on the next slide. Here, the observer is represented by the block labelled O .

Figure 19.8: *Implementation of the Q block*



We next illustrate how we can put all of these ideas together to design a feedback control law for a system that is *both* stable and stably invertible.

Example 19.3

Consider the nonlinear plant having a state space model given by

$$\begin{aligned}\frac{dx_1(t)}{dt} &= x_2(t) + (x_2(t))^3 \\ \frac{dx_2(t)}{dt} &= -2x_1(t) - 3x_2(t) + u(t) + 0.1(x_1(t))^2 u(t) \\ y(t) &= x_1(t)\end{aligned}$$

Differentiating the plant output we see that

$$\begin{aligned}\rho y(t) &= x_2(t) + (x_2(t))^3 \\ \rho^2 y(t) &= (1 + 3(x_2(t))^2) \frac{dx_2(t)}{dt} \\ &= -(1 + 3x_2^2(t))(2x_1(t) + 3x_2(t)) + (1 + 3(x_2(t))^2)(1 + 0.1(x_1(t))^2)u(t)\end{aligned}$$

Thus the system has relative degree 2.

We now choose

$$p(\rho) = (\rho^2 + 4\rho + 9)/9$$

we then obtain $a(x)$ and $b(x)$ as

$$a(x) = \frac{(1 + 3(x_2(t))^2)(1 + 0.1(x_1(t))^2)}{9}$$

$$b(x) = \frac{7x_1(t) + x_2(t) - 5(x_2(t))^3 - 6x_1(t)(x_2(t))^2}{9}$$

The control loop can be implemented as in Figure 19.6 with the nonlinear operator $Q\langle\circ\rangle$ implemented as below:

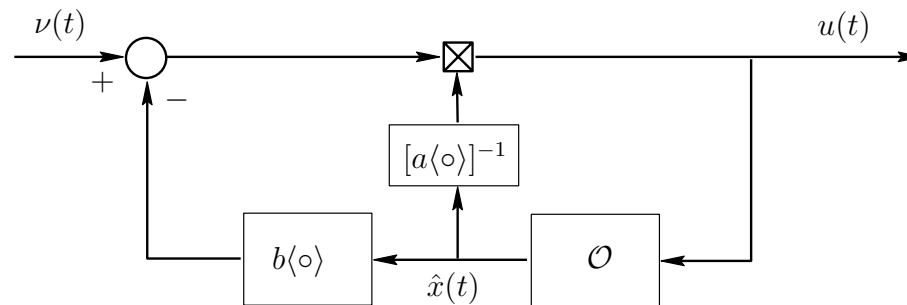


Figure 19.8

To assess the performance of this design, we compare it with the linear design based upon a linearized plant model.

$$G_o(s) = \frac{1.13}{s^2 + 3.0s + 1.55}$$

We choose an appropriate complementary sensitivity, say

$$T_o(s) = \frac{9}{s^2 + 4s + 9}$$

Which leads to the following linearized control law:

$$C(s) = [G_o(s)]^{-1} \frac{T_o(s)}{S_o(s)} = 7.96 \frac{s^2 + 3.0s + 1.55}{s(s + 4)}$$

The results of the simulation are shown below. This figure shows the superior tracking performance of the nonlinear design.

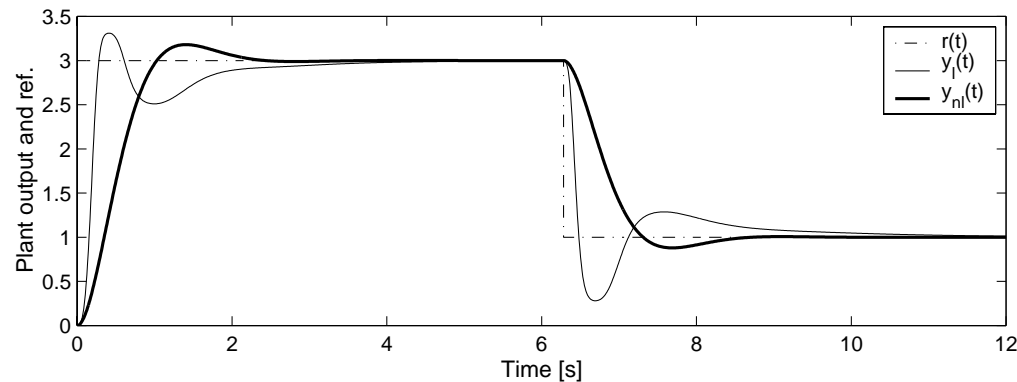


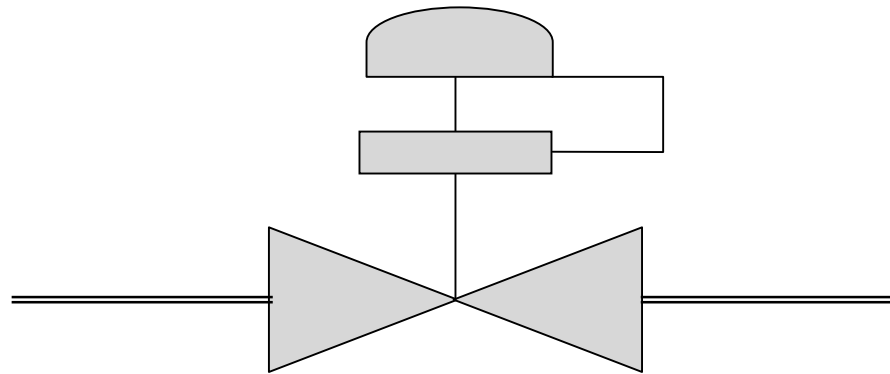
Figure 19.9: *Tracking performance of linear (y_l) and nonlinear (y_{nl}) control designs for a nonlinear plant*

Example 19.4 (pH neutralization)

pH control is an extremely difficult problem in practical situations, because of the large dynamic range needed in the controller.

To deal with this issue, it is often necessary to make structural changes to the physical setup - e.g., by providing additional mixing tanks and multiple reagent valves. The interested reader is referred, for instance, to the extensive discussion of this problem on the web page for the book.

To briefly summarize, we note that all reagent values are nonlinear (*i.e. they contain deadzones, hysteresis, etc.*)



Reagent Control Value

(Non-ideal !)

These nonlinear effects in the control values lead to control difficulties. We briefly discuss this below.

Problem (*Actuator Ranging*)

Control valve large enough to go from pH 11 \rightarrow 7

10^4 change in ion concentration

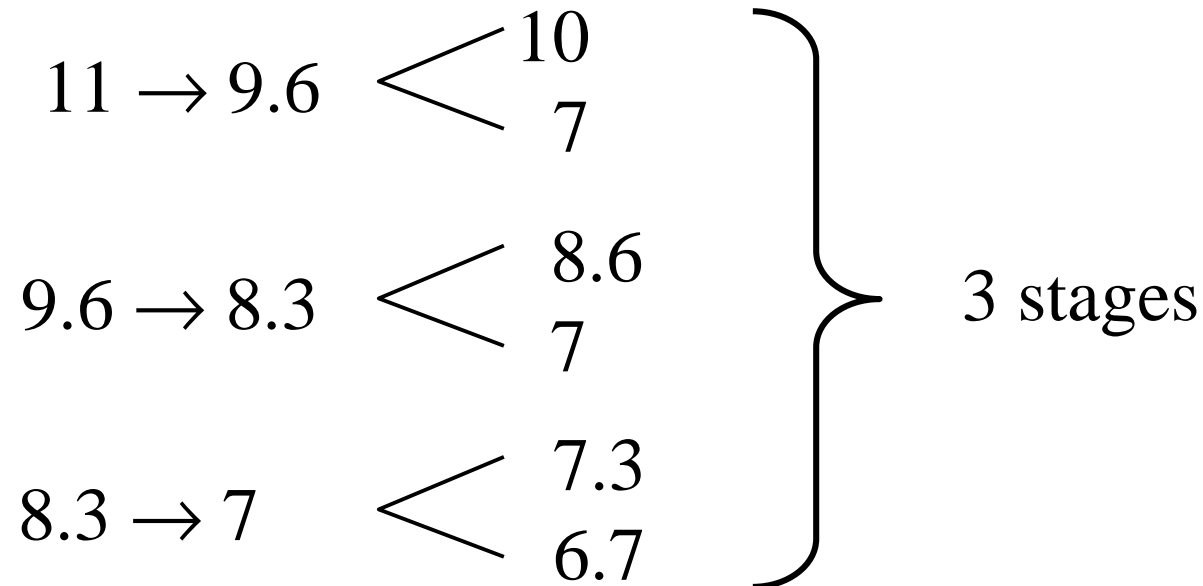
1% error in valve $\equiv 10^2$ change in ion concentration

Final pH error 7 ± 2 !

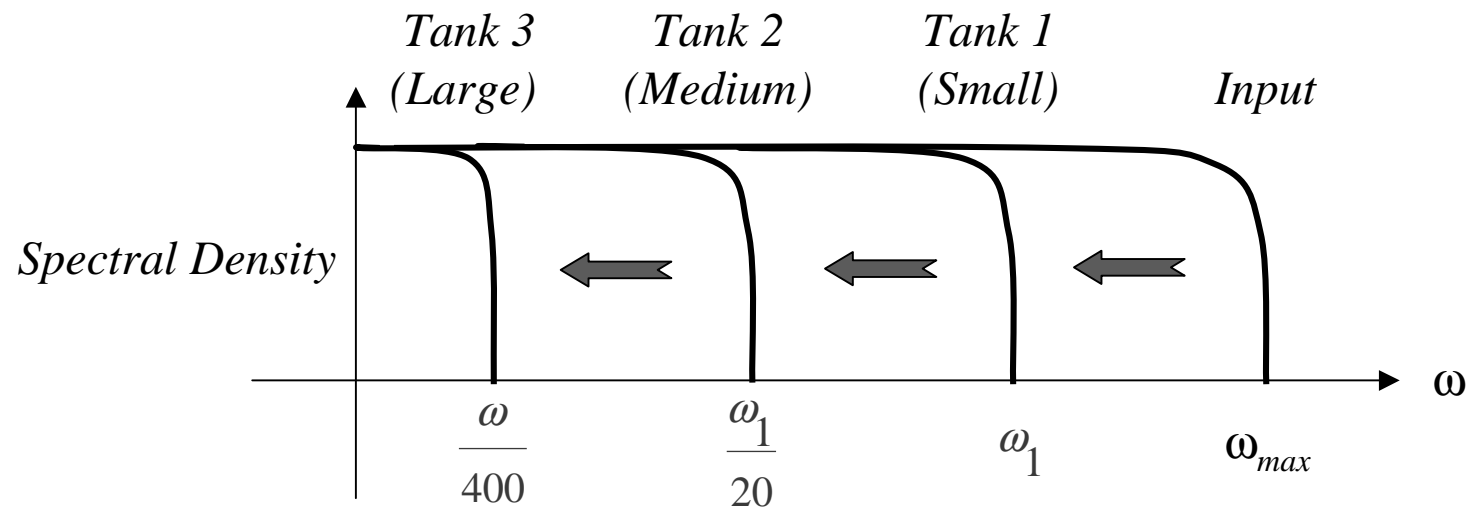
Control System Design (*Issues:*)

Thus, multiple valves are needed to achieve desired accuracy.

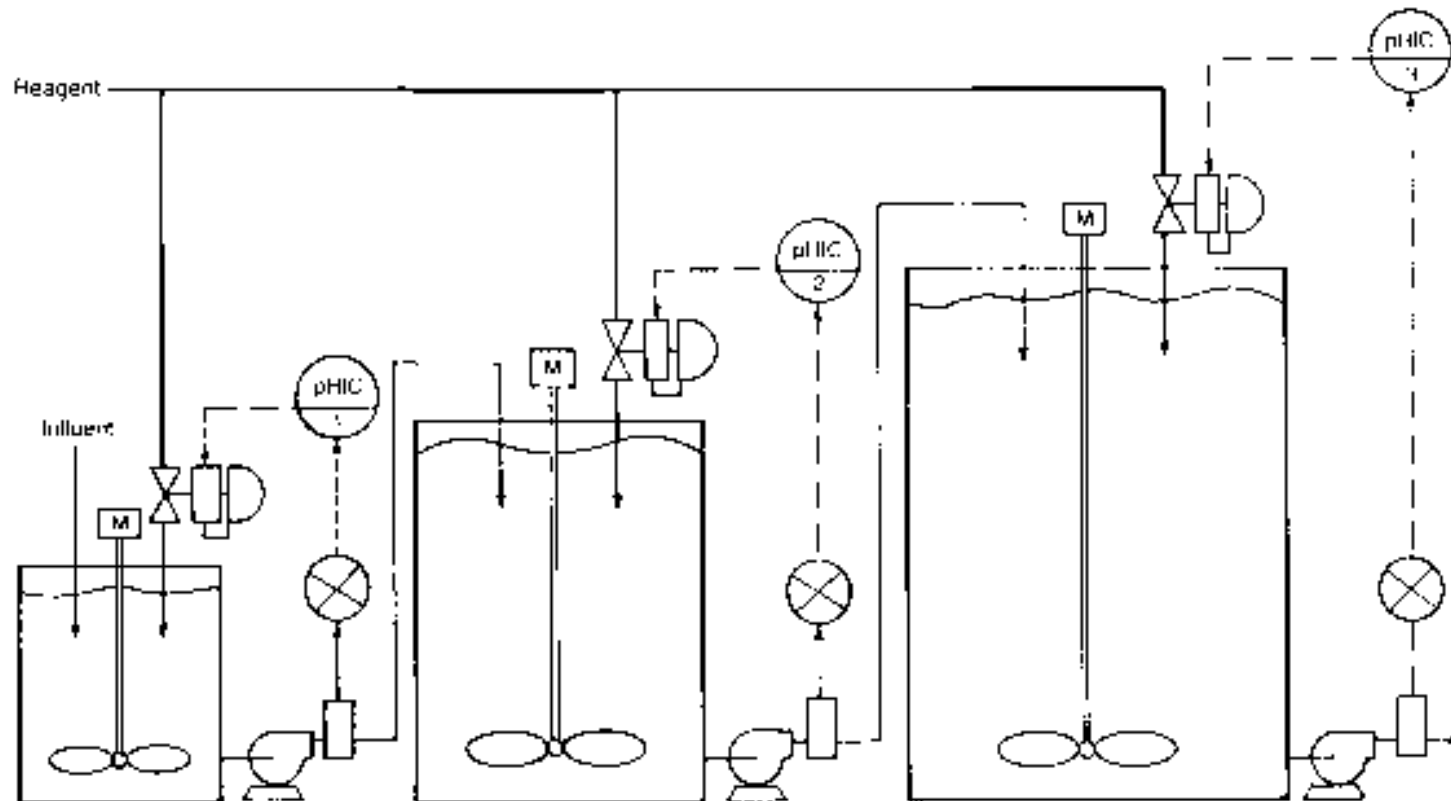
Assume valves 5% accuracy



Also, we can match the errors arising from valve nonlinearities to errors arising from input variations by mixing in tanks of different sizes.



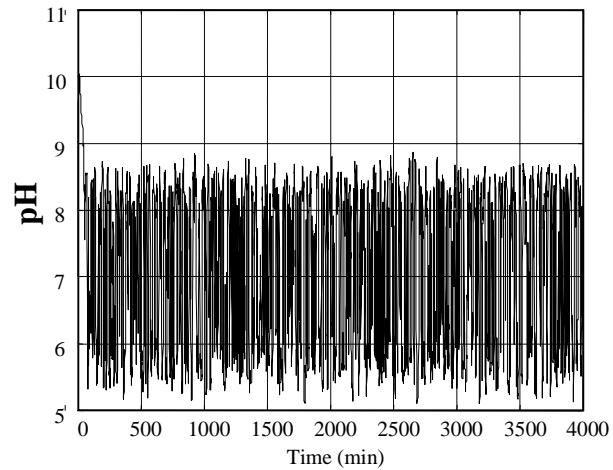
Final Design



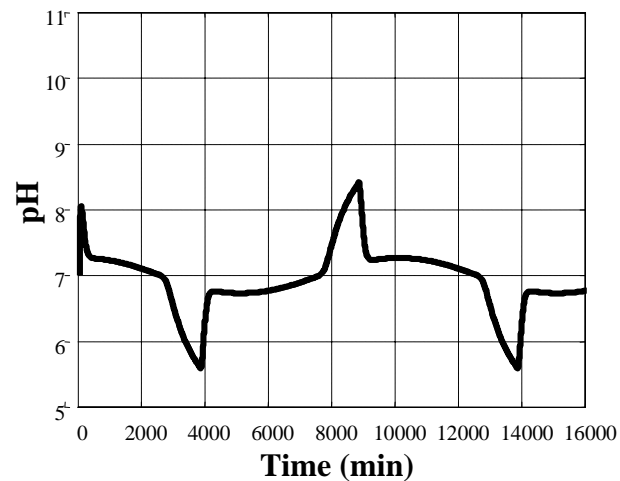
Reference: Gregory McMillan, *Pub. ISA*

Simulated Results

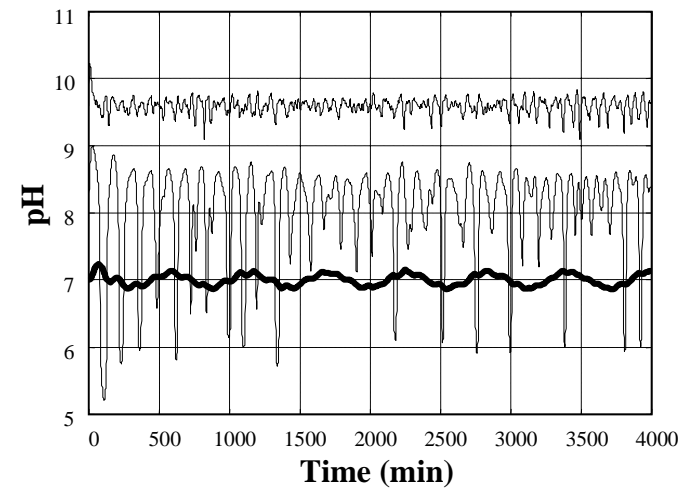
Noisy Flow



Valve Imperfections



Result with three tanks



In the sequel we take the above practical issues concerning valve imperfections for granted. Instead, we will focus on the design of the control law for one of the mixing tanks focusing on the inherent nonlinearity in the dynamics.

The key issues are:

- (i) a bilinear dependence of flow and concentration;
- (ii) pH is measured on a highly nonlinear (*logarithmic*) scale.

Simplified Model for pH Neutralization

From elementary mass-balance considerations, an appropriate state space model for pH neutralization of the strong-acid-strong-base type is given by (*We also include a lag on the measurement*):

$$\begin{aligned}\frac{dc_o(t)}{dt} &= \frac{u(t)}{V} (c_u - c_o(t)) + \frac{q}{V} (c_i + d(t) - c_o(t)) \\ \frac{dp_m(t)}{dt} &= \frac{1}{\alpha} (p_o(t) - p_m(t)) \\ p_o(t) &= -\log \left[\sqrt{0.25(c_o(t))^2 + 10^{-14}} + 0.5c_o(t) \right]\end{aligned}$$

Notice the two nonlinear effects mentioned above are captured in this model.

The relative degree of this nonlinear system is 2 (*except when $c_0 = c_u$*). In practice, this point of singularity can be avoided by appropriate choice of c_u .

We choose

$$p(\rho) = \alpha\beta\rho^2 + (\alpha + \beta)\rho + 1$$

An appropriate inverse is then obtained, if we choose

$$u(t) = \frac{V \ln(10)}{\beta(c_o(t) - c_u)} \sqrt{(c_o(t))^2 + 4 \times 10^{-14}} (\nu(t) - p_o(t)) + \frac{q(c_i - c_o(t))}{c_o(t) - c_u}$$

The implementation of this inverse requires that c_0 and p_0 be replaced by their estimates, \hat{c}_0 and \hat{p}_0 which we obtain by using a nonlinear observer.

The final performance of the feedback linearizing control law is shown below:

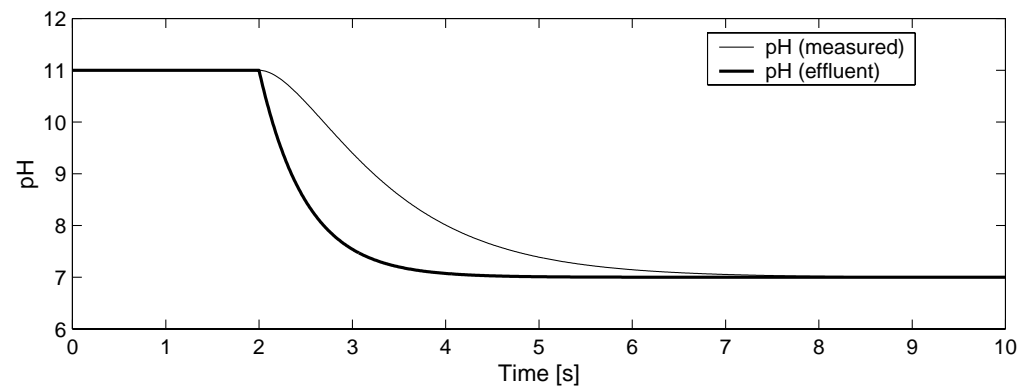


Figure 19.10: *pH control by using a nonlinear control-design strategy - the effluent pH (thick line) and measured pH (thin line) are shown.*

Notice that the final result exhibits linear performance due to the use of the feedback linearization design strategy.

Disturbances

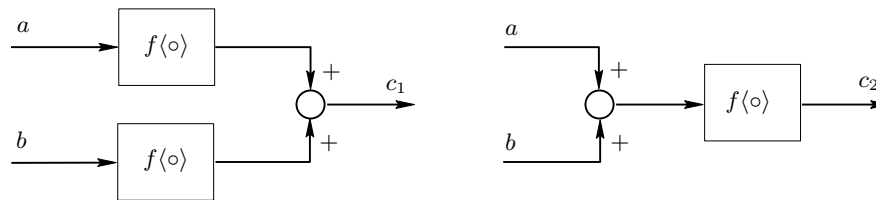
In the above designs we have focused primarily on the response to reference inputs. However, in practice, there will always be disturbances and, indeed, these are often the main reason for doing a control system design in the first place.

We thus briefly consider how we might (*slightly*) modify the above designs to account for input or output disturbances.

Disturbance Issues in Nonlinear Control

We have seen in Chapter 8, that disturbances need special attention in control-system design. In particular, we found that there were subtle issues regarding the differences between input and output disturbances. In the nonlinear case, these same issues arise, but there is an extra complication arising from nonlinear behavior. The essence of the difficulty is captured in Figure 19.11.

Figure 19.11: *Nonlinear operators*



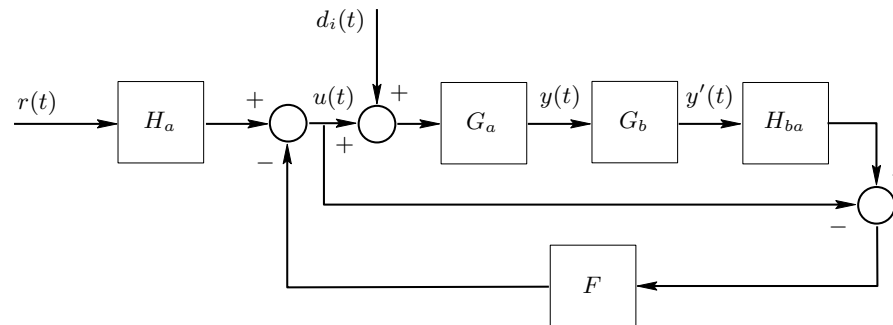
The output of these two systems are different because superposition does not hold for nonlinear operators. This implies, amongst other things, that different strategies may be needed to deal with input or output disturbances).

(i) Input disturbances

Here, we can operate on the measured plant output, $y'(t)$ and $u(t)$ to estimate $d_i(t)$, then cancel this at the input to the plant by action of feedback control. We will employ feedback linearization to implement the approximate inverses needed to implement this basic idea. This leads to the strategy illustrated on the next slide.

Figure 19.13: *Control strategy for input disturbances*

$$H_{ba} \approx [G_b \langle G_a \langle \bullet \rangle \rangle]^{-1}$$



Note H_{ba} can be evaluated by feedback linearization methods.

Note that the choice of the filter F is important. It serves two purposes:

- (i) to avoid an algebraic loop;
- (ii) to speed up the disturbance rejection for input disturbances (*as in the linear case*).

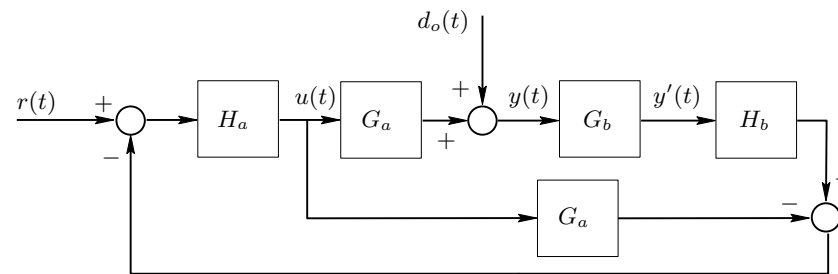
(ii) Output disturbance

Here we can operate on the measured plant output, $y'(t)$ and $u(t)$ to estimate $d_0(t)$, which is combined with the reference $r(t)$ and passed through an (*approximate*) inverse for $G_a(\cdot)$ so as to cancel the disturbance and cause $y(t)$ to approach $r(t)$. This leads to the strategy illustrated on the next slide.

Figure 19.14: Control strategy for output disturbances

$$H_b \approx G_b^{-1}$$

$$H_a \approx G_a^{-1}$$



The reader can verify that, in the linear case, one can commute the various inverses around the loops in Figures 19.13 and 19.14 to obtain (*essentially*) identical results. However, this relies on superposition which does not hold in the nonlinear case.

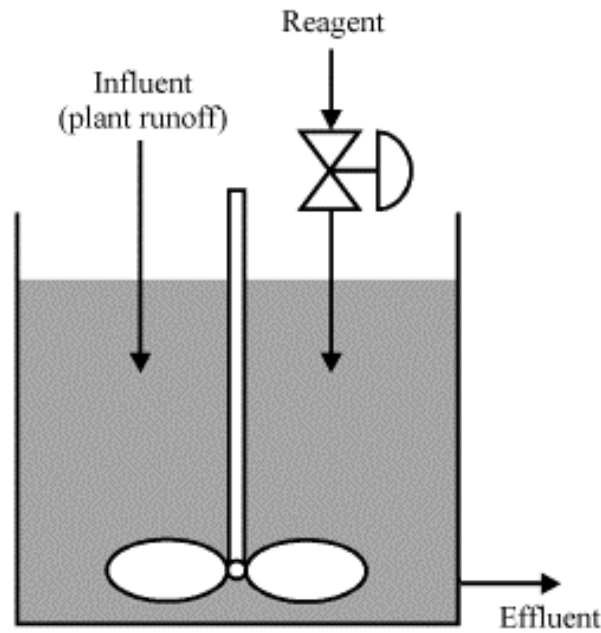
Example 19.5

We consider the pH problem again.

We assume one mixing tank and we ignore the lag on the output measurement.

A schematic of the system is shown on the next slide, together with the model.

Example: pH Neutralization



$$\dot{c}_o(t) = \frac{(u(t) + d_i(t))}{V} (c_u - c_o(t)) + \frac{q}{V} (c_i - c_o(t))$$

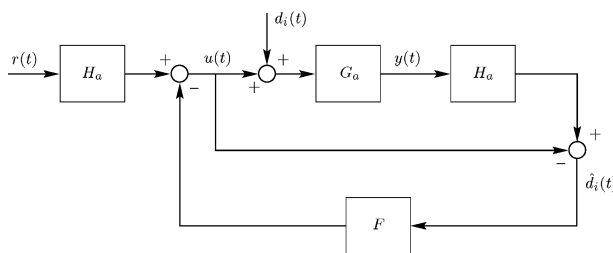
$$p_o(t) = -\log \left(\sqrt{0.25 c_o(t)^2 + 10^{-14}} + 0.5 c_o(t) \right) + d_o(t)$$

We assume an input disturbance and use the control strategy illustrated in Figure 19.3.

We use two different choices for the filter F ;

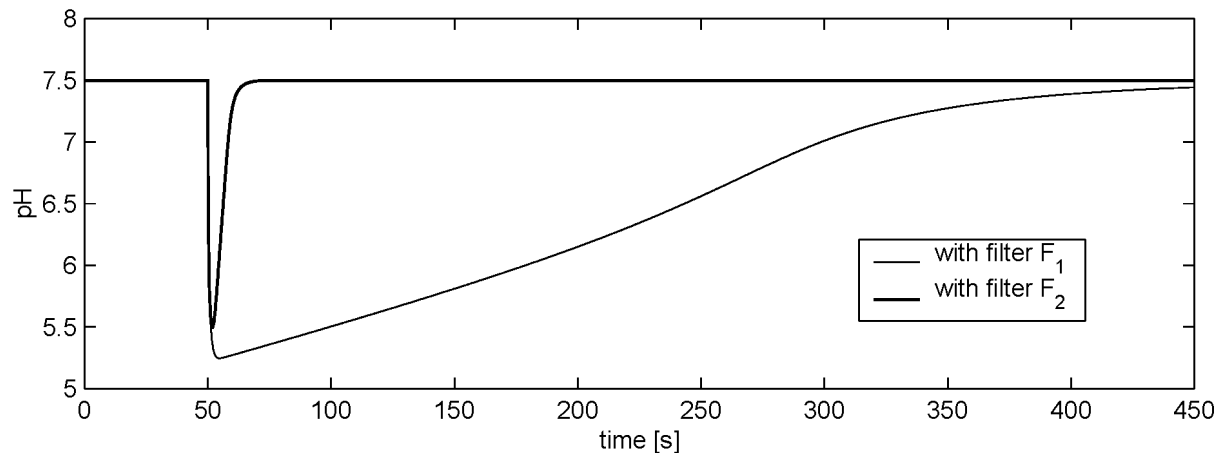
- (i) $F_1(s) = \frac{1}{\tau s + 1}$ [*This filter is a small time constant filter aimed simply at avoiding an algebraic loop.*]
- (ii) $F_2(s) = \frac{f_1 s + 1}{(\tau s + 1)^2}$ [*This filter contains numerator dynamics aimed at approximately cancelling the slow dynamics of the plant.*]

Input disturbance design: the effect of an appropriate choice of the F filter.



$$F_1(s) = \frac{1}{\tau s + 1}$$

$$F_2(s) = \frac{f_1 s + 1}{(\tau s + 1)^2}$$

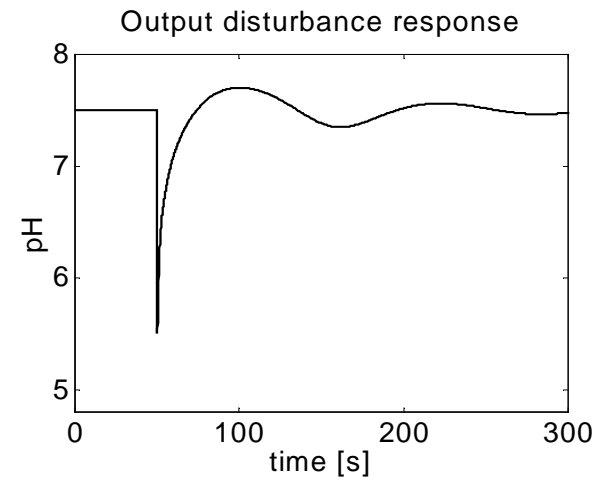
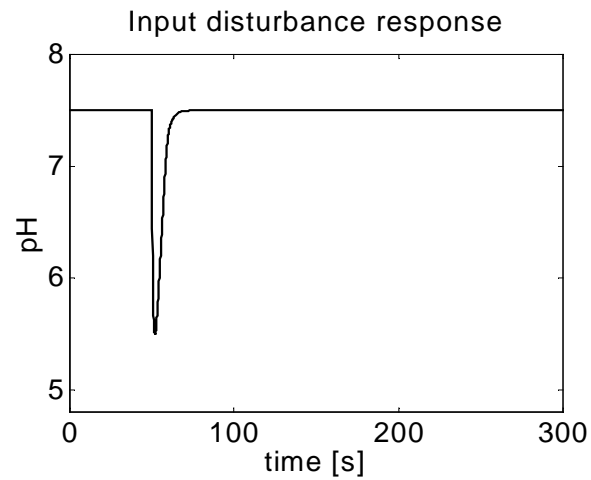


From the previous slide, we see that the choice of the filter $F(s)$ makes a significant difference to the transient associated with input disturbance rejection. This mirrors similar observations made in the linear case.

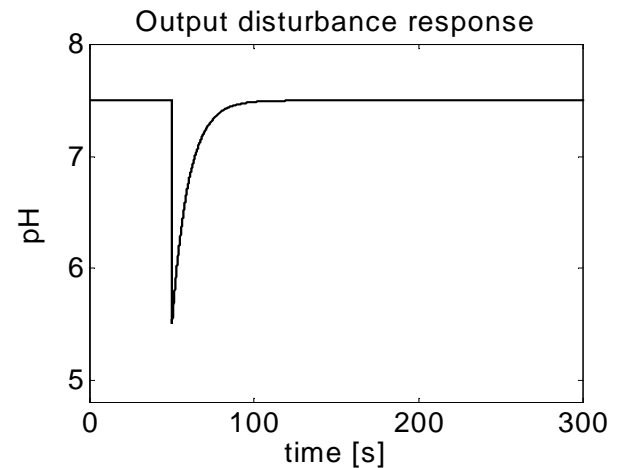
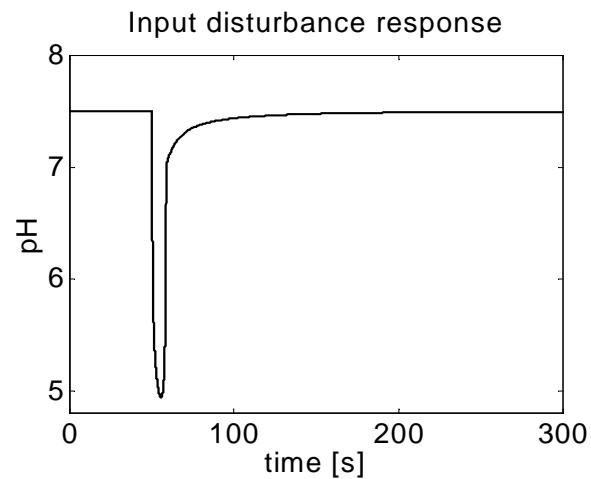
Next we compare the design for input disturbances (*Figure 19.3*) with the design for output disturbances (*Figure 19.4*). The next slide shows 4 plots

- 2 correspond to the case when the disturbance appears at the point it was designed for
- the other 2 plots correspond to the case when the disturbance is not injected at the design point

*Input
disturbance
design*



*Output
disturbance
design*



This example highlights the fact that, in nonlinear systems, certain linear insights no longer hold, because the principle of superposition is not valid. Thus, particular care needs to be exercised. We have seen that input and output disturbances need to be carefully considered and that design should (*in principle*) be targeted at the correct disturbance injection point. This is all part of the excitement associated with the control of nonlinear systems.

More General Plants with Smooth Nonlinearities

For simplicity, the discussion above has focused on plants that are both stable and stably invertible. We next examine briefly the more general situation. By analogy with the linear case, one might conceive of solving the general problem by a combination of a (*nonlinear*) observer and (*nonlinear*) state-estimate feedback. For example, we found in the linear case that the closed-loop poles are simply the union of the dynamics of the observer and state feedback, considered separately. Unfortunately, this does not hold in the nonlinear case where, *inter alia*, there will generally be ...

interaction between the observer and state feedback.
This makes the nonlinear problem much more difficult.

We proceed to describe some nonlinear designs which mirror linear ideas. However, these designs have deficiencies arising from the basic linear philosophy that underlies their design.

Nonlinear Observer

We first show how local linearization can be used to design a nonlinear observer.

Consider a plant of the form given in

$$\begin{aligned}\rho x(t) &\triangleq \frac{dx(t)}{dt} = f(x) + g(x)u(t) \\ y(t) &= h(x)\end{aligned}$$

Say that we are given an estimate $\hat{x}(t)$ of the state at time t . We will use linearization methods to see how we might propagate this estimate.

The linearized forms of the model about $\hat{x}(t)$, are respectively

$$\begin{aligned}\rho x(t) &\approx f(\hat{x}) + \left. \frac{\partial f}{\partial x} \right|_{\hat{x}} [x(t) - \hat{x}(t)] + g(\hat{x})u(t) + \left. \frac{\partial g}{\partial x} \right|_{\hat{x}} [x(t) - \hat{x}(t)]u(t) \\ y(t) &\approx h(\hat{x}) + \left. \frac{\partial h}{\partial x} \right|_{\hat{x}} [x(t) - \hat{x}(t)]\end{aligned}$$

For notational convenience, let

$$\mathbf{A} = \left. \frac{\partial f}{\partial x} \right|_{\hat{x}} + \left. \frac{\partial g}{\partial x} \right|_{\hat{x}} u(t)$$

$$\mathbf{B} = g(\hat{x}) - \left. \frac{\partial g}{\partial x} \right|_{\hat{x}} \hat{x}(t)$$

$$\mathbf{C} = \left. \frac{\partial h}{\partial x} \right|_{\hat{x}}$$

$$\mathbf{D} = h(\hat{x}) - \left. \frac{\partial h}{\partial x} \right|_{\hat{x}} \hat{x}(t)$$

$$\mathbf{E} = f(\hat{x}) - \left. \frac{\partial f}{\partial x} \right|_{\hat{x}} \hat{x}(t)$$

The linearized model is then

$$\begin{aligned}\rho x &= \mathbf{A}x + \mathbf{B}u + \mathbf{E} \\ y &= \mathbf{C}x + \mathbf{D}\end{aligned}$$

This suggests the following linearized observer:

$$\rho \hat{x} = \mathbf{A}\hat{x} + \mathbf{B}u + \mathbf{E} + \mathbf{J}(y - \mathbf{C}\hat{x} - \mathbf{D})$$

Of course, we must remember

- (i) that \mathbf{A} , \mathbf{B} , \mathbf{C} , \mathbf{D} , \mathbf{E} depend on the point about which the linearization is made, and
- (ii) the system is really nonlinear, so the above model is only an approximation.

Substituting for **A**, **B**, **C**, **D**, **E** leads to the following compact representation for the observer:

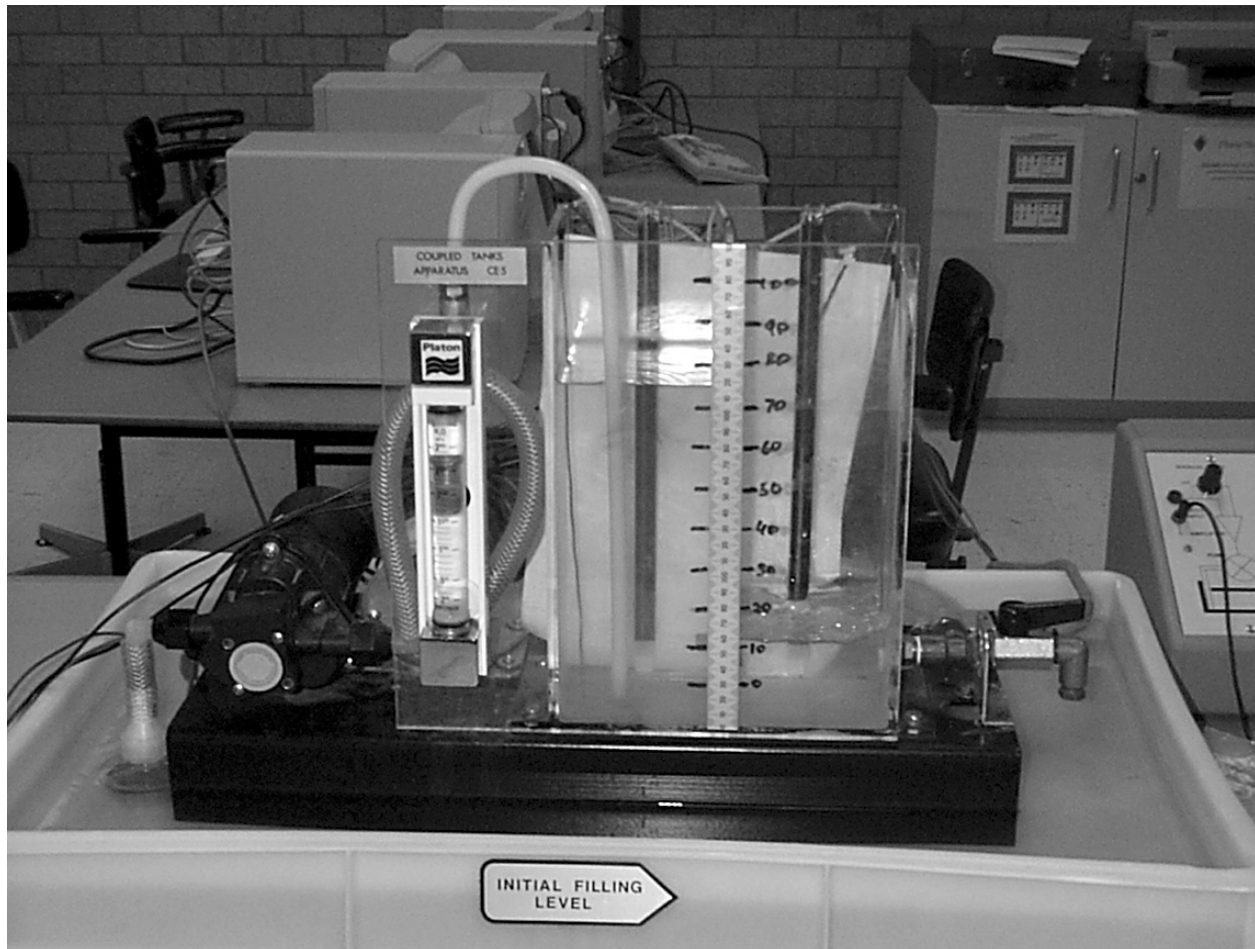
$$\rho \hat{x} = f(\hat{x}) + g(\hat{x})u + \mathbf{J}[y - h(\hat{x})]$$

This result is intuitively appealing, because the nonlinear observer so obtained constitutes an open-loop nonlinear model with (*linear*) feedback gain multiplying the difference between the actual observations, y , and those given by the nonlinear model $h(\hat{x})$. (This is generally known as linear output injection).

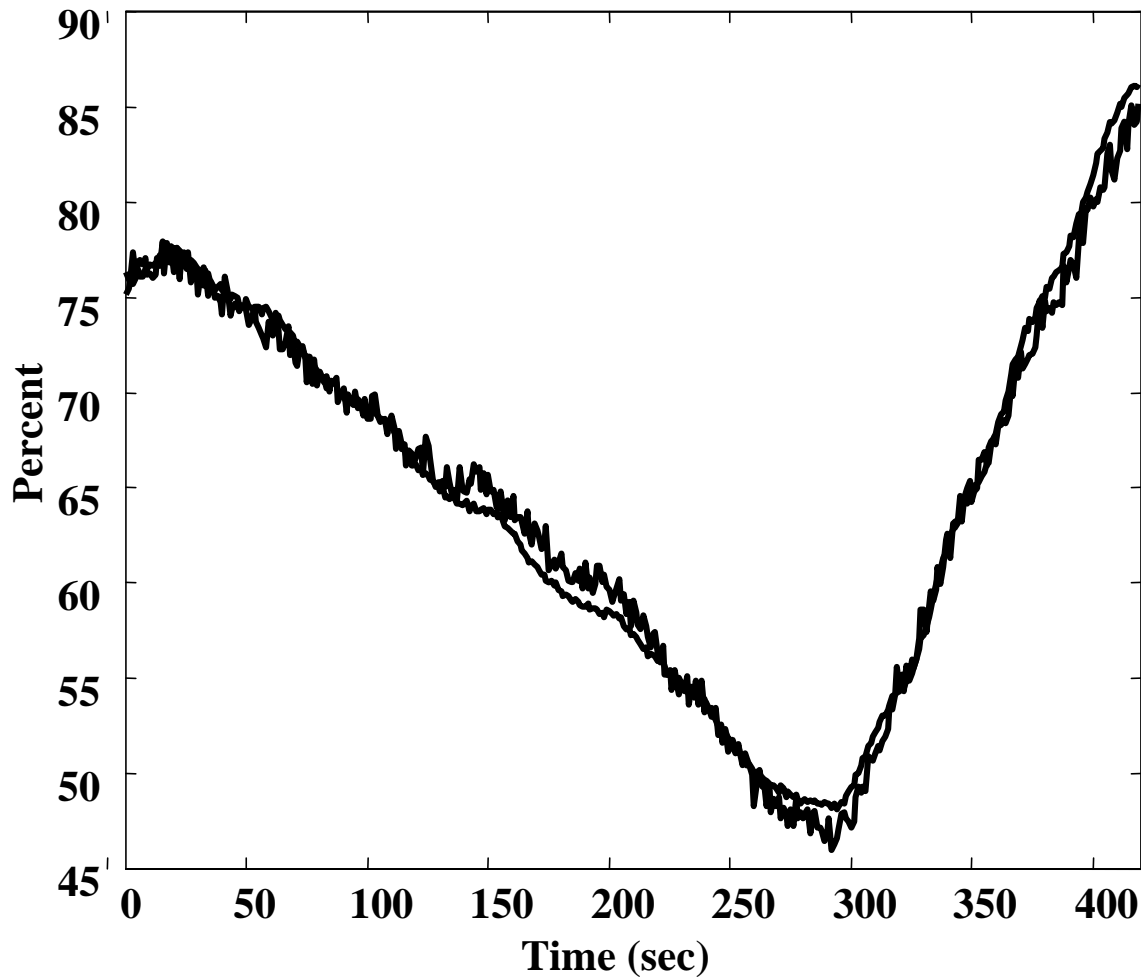
Notice that the open loop nonlinear observer used earlier in this chapter for stable nonlinear plants is an example of a linear output injection observer where the injection gain happens to be taken as zero. More generally, the injection gain will be designed using a linearized model as described above.

We illustrate by considering the estimation of liquid level in 2 coupled tanks as discussed earlier in Chapter 18. A photo of the system is shown on the next slide.

Coupled Tanks Apparatus



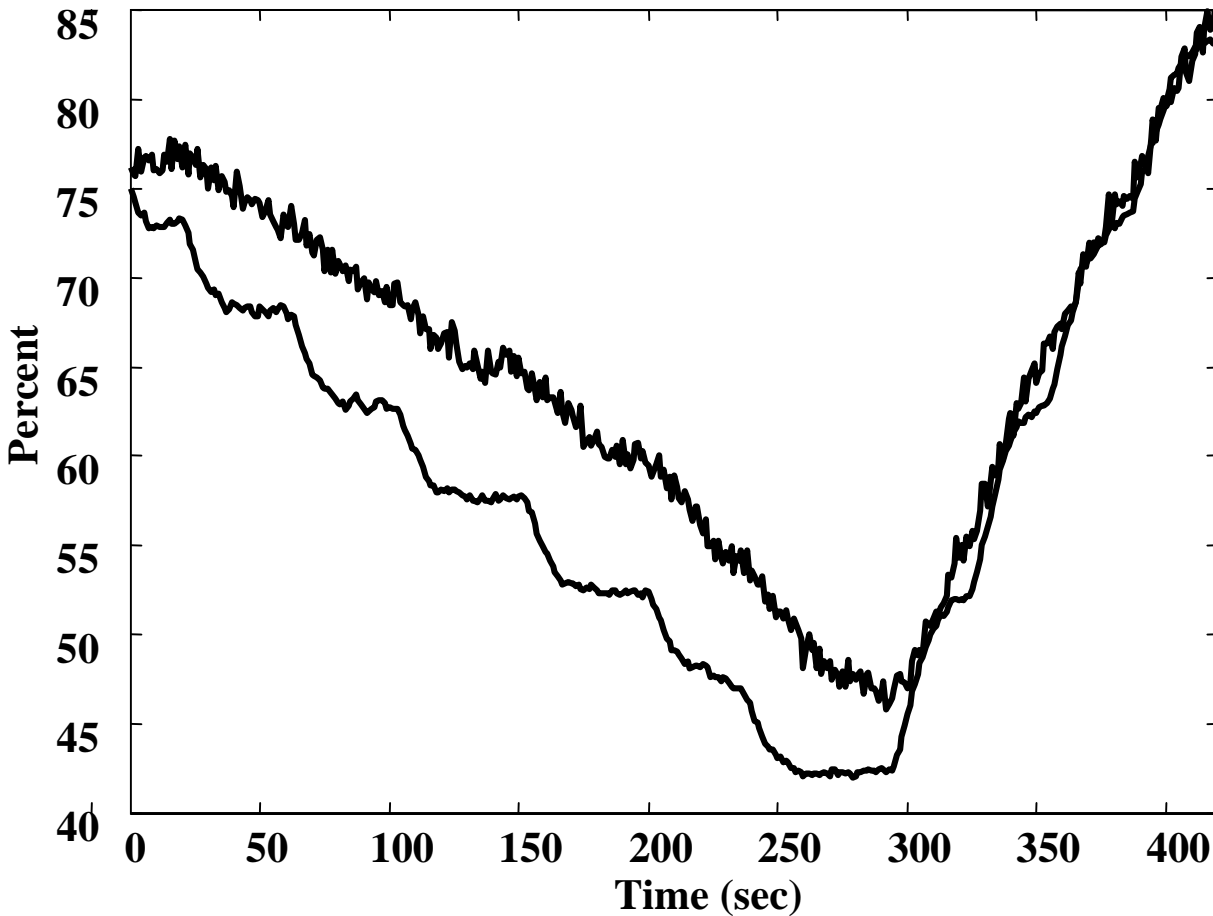
Recall that the height of water in one tank is measured and it is required to estimate the height of water in the other tank. In Chapter 18 a linear observer was designed. Here we design a *nonlinear* observer using linear injection as described above. The comparison of the true and estimated heights is given on the next slide.



*Actual height in tank 1 (blue),
Observed height in tank 1 (red)
Non-Linear*

The above results should be compared with the results obtained with the linear observer as found in Chapter 18 (*see the next slide*).

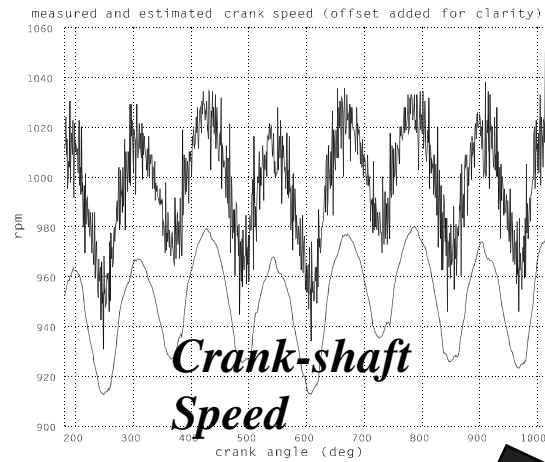
Comparison with linear observer in Chapter 18.



*Actual height in tank 1 (blue),
Observed height in tank 1 (red)*

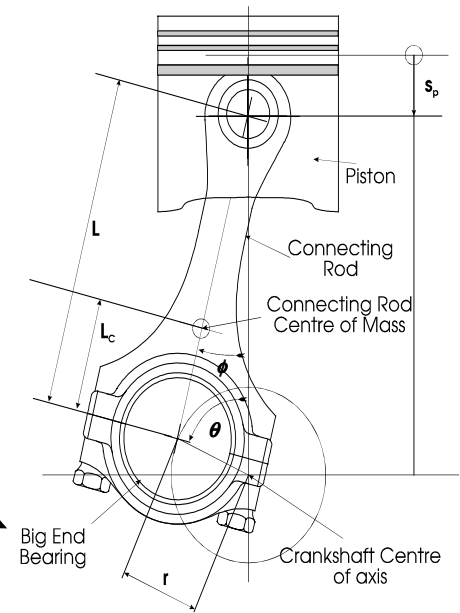
A practical application of the nonlinear observer using linear output injection is described on the next slide. Here the idea is to use accurate measurements of instantaneous rotational speed of an internal combustion engine to estimate internal cylinder pressure. Note that the latter variable is difficult to measure directly but plays an important role in engine control to minimize pollution.

In-Cylinder pressure estimation for Orbital Engine Company, Australia.

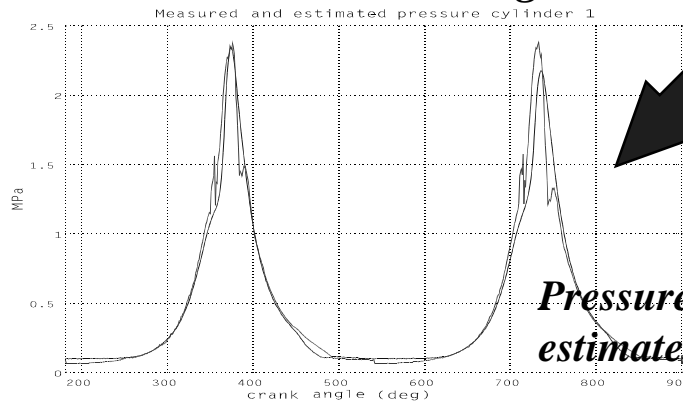


Crank-shaft Speed

Inertia and Gas Modelling



Engine Model



Pressure estimates

Nonlinear least squares

The model for the above system is actually very complex. The next slide shows details of the model. The reader should not try to follow the details of the model. It is simply intended to illustrate the flexibility of this type of observer to deal with rather complex problems.

$$\begin{aligned}
\tau_{\text{crit}} &= \frac{1}{\omega} M_c \{ (\ddot{x}_{\text{con}}(\theta) \ddot{x}_{\text{con}}(\theta) + \dot{y}_{\text{con}}(\theta) \dot{y}_{\text{con}}(\theta)) \\
&\quad + (\ddot{x}_{\text{con}}(\theta_2) \ddot{x}_{\text{con}}(\theta_2) + \dot{y}_{\text{con}}(\theta_2) \dot{y}_{\text{con}}(\theta_2)) \\
&\quad + (\ddot{x}_{\text{con}}(\theta_3) \ddot{x}_{\text{con}}(\theta_3) + \dot{y}_{\text{con}}(\theta_3) \dot{y}_{\text{con}}(\theta_3)) \} \\
&= -M_c \omega^2 r^2 \left[\left(\sin(\theta) + \frac{L_c r}{2LQ} \sin(2\theta) \right) \left(\cos(\theta) + \frac{L_c r}{LQ} \cos(2\theta) \right) + \right. \\
&\quad - \left(\sin(\theta_2) + \frac{L_c r}{2LQ} \sin(2\theta_2) \right) \left(\cos(\theta_2) + \frac{L_c r}{LQ} \cos(2\theta_2) \right) + \\
&\quad - \left(\sin(\theta_3) + \frac{L_c r}{2LQ} \sin(2\theta_3) \right) \left(\cos(\theta_3) + \frac{L_c r}{LQ} \cos(2\theta_3) \right) + \\
&\quad \left. \frac{1}{2} \left(1 - \frac{L_c}{L} \right)^2 \sin(2\theta) - \frac{1}{2} \left(1 - \frac{L_c}{L} \right)^2 \sin(2\theta_2) - \frac{1}{2} \left(1 - \frac{L_c}{L} \right)^2 \sin(2\theta_3) \right] \\
&= M_c \omega^2 r^2 \left[\left(\sin(\theta) + \frac{L_c r}{2LQ} \sin(2\theta) \right) \left(\cos(\theta) - \frac{L_c r}{LQ} \cos(2\theta) \right) + \right. \\
&\quad - \left(\sin(\theta_2) + \frac{L_c r}{2LQ} \sin(2\theta_2) \right) \left(\cos(\theta_2) + \frac{L_c r}{LQ} \cos(2\theta_2) \right) + \\
&\quad - \left(\sin(\theta_3) + \frac{L_c r}{2LQ} \sin(2\theta_3) \right) \left(\cos(\theta_3) + \frac{L_c r}{LQ} \cos(2\theta_3) \right) + \\
&\quad \left. - \frac{1}{2} \left(1 - \frac{L_c}{L} \right)^2 (\sin 2\theta + \sin 2\theta_2 + \sin 2\theta_3) \right] \\
&= M_c \omega^2 r^2 \left[0 + 0 \left(\sin \theta \right) \frac{L_c r}{LQ} \cos^2 \theta - \frac{3}{4} \left(\sin \theta \right) \frac{L_c r}{LQ} \right] \\
&= M_c \omega^2 \frac{9L_c r^3}{4LQ} \sin \theta (1 - \cos^2 \theta) \\
&= M_c \omega^2 \frac{9L_c r^3}{4LQ} (1 - 2\cos^2 \theta) \sin \theta \\
&= M_c \omega^2 \frac{9L_c r^3}{4LQ} \sin(3\theta)
\end{aligned}$$

Nonlinear Feedback Design

Next we discuss how one might use the state estimate \hat{x} in a nonlinear feedback control law.

There are myriad possibilities here. For example, if the system is unstable but has a stable inverse, then one could use feedback linearization. If the system does not have a stable inverse, then the basic feedback linearization scheme cannot be used. We thus describe a scheme that is close in spirit to feedback linearization but which can stabilize certain plants which are not stably invertible.

Generalized Feedback linearization for Nonstability-Invertible Plants

We recall that the feedback-linearization scheme in essence brings $p(\rho)y(t)$ to the set point signal $v(t)$ where $p(\rho)$ is a differential operator of degree equal to the relative degree of the nonlinear system. A drawback of the scheme, however, was that it cancelled the zero dynamics and hence required that the system have a stable inverse.

We note that the basic feedback-linearization scheme achieves

$$p(\rho)y(t) = \nu(t)$$

However, a difficulty in the nonstably-invertible case is that the corresponding input will not be bounded. By focusing temporarily on the input, it seems desirable to match the above equation by some similar requirement on the input. Thus, we might ask that the input satisfy a linear dynamic model of the form

$$\ell(\rho)u(t) = u_s$$

Of course, the above requirements will, in general, not be simultaneously compatible. This suggests that we might determine the input by combining them in some way. For example, we could determine the input as that value of $u(t)$ that satisfies a linear combination of the form:

$$(1 - \lambda)(p(\rho)y(t) - \nu(t)) + (\lambda)(\ell(\rho)u(t) - u_s) = 0$$

We call the resultant control policy:

Generalized Feedback Linearization

To develop the control law defined above, we introduce a dummy input $\bar{u}(t)$ defined by

$$\ell(\rho)u(t) = \bar{u}(t)$$

Say that $\ell(\rho)$ has degree h , and that the nonlinear system has relative degree m . Then the nonlinear system between $\bar{u}(t)$ and $y(t)$ will have relative degree $m+h$. Hence, if we use an $(m+h)$ degree operator, $p(\rho)$, then $p(\rho)y(t)$ will depend explicitly on $\bar{u}(t)$. Hence, we can write

$$p(\rho)y = b(x) + a(x)\bar{u}$$

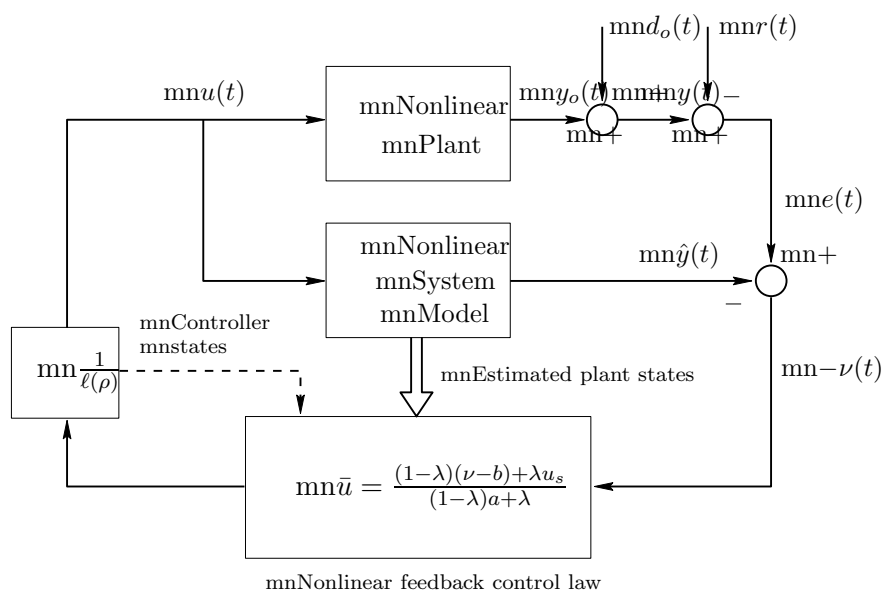
Substituting into $(1 - \lambda)(p(\rho)y(t) - \nu(t)) + (\lambda)(\ell(\rho)u(t) - u_s) = 0$ gives the following nonlinear feedback control law:

$$\bar{u} = \frac{(1 - \lambda)(\nu - b) + \lambda u_s}{(1 - \lambda)a + \lambda}$$

This control law does not cancel the zero dynamics unless $\lambda=0$. Clearly for $\lambda \rightarrow 0$, the scheme reduces to the feedback-linearizing control law and for $\lambda \rightarrow 1$, $u(t)$ becomes the open-loop control policy $\ell(\rho)u(t) = u_s$. Because when $\lambda \rightarrow 1$, the control law becomes open loop, then it follows that one class of systems that this scheme will handle is all open-loop stable nonlinear systems whether or not they are stably invertible.

For the case of stable plants, we can use an open loop observer. This leads, finally, to the feedback structure shown on the next slide.

Figure 19.23: *Generalized feedback linearization for open-loop stable plant*



We summarize the idea below - see next two slides

General Nonlinear Systems

- ❖ We consider general nonlinear systems which are not necessarily stable nor stably invertible.
- ❖ To handle this type of system we introduce the Generalized Feedback linearization (GFL) strategy:

$$(1 - \lambda)(p(\rho)y(t) - y^*) + \gamma(l(\rho)u(t) - u^*) = 0$$

$$0 \leq \lambda \leq 1$$

$p(\rho)$ and $l(\rho)$ are suitable differential operators.

With $\lambda = 0$ we revert to the usual feedback linearization strategy.

To develop the control law, define:

$$l(\rho)u(t) = \bar{u}(t)$$

Then:

$$p(\rho)y(t) = b(x') + a(x')\bar{u}(t).$$

Finally

$$\bar{u}(t) = \frac{(1-\lambda)(y^*(t) - b(x')) + \lambda u^*}{(1-\lambda)a(x') + \lambda}$$

Example 19.8

Consider the nonlinear system

$$\dot{x}_1 = 10x_1 - 10x_2$$

$$\dot{x}_2 = 16.925x_1 - 16x_2 - 0.1(u - \tan^{-1} x_2)$$

$$y = x_2 + d_o$$

where d_o represents a constant output disturbance.

Note that this system is open loop stable but is *not* stably invertible.

-
-
- (1) The zero dynamics can be evaluated by setting $y = 0$ with $d_0 = 0$. This leads to

$$\dot{x}_1 = 10x_1$$

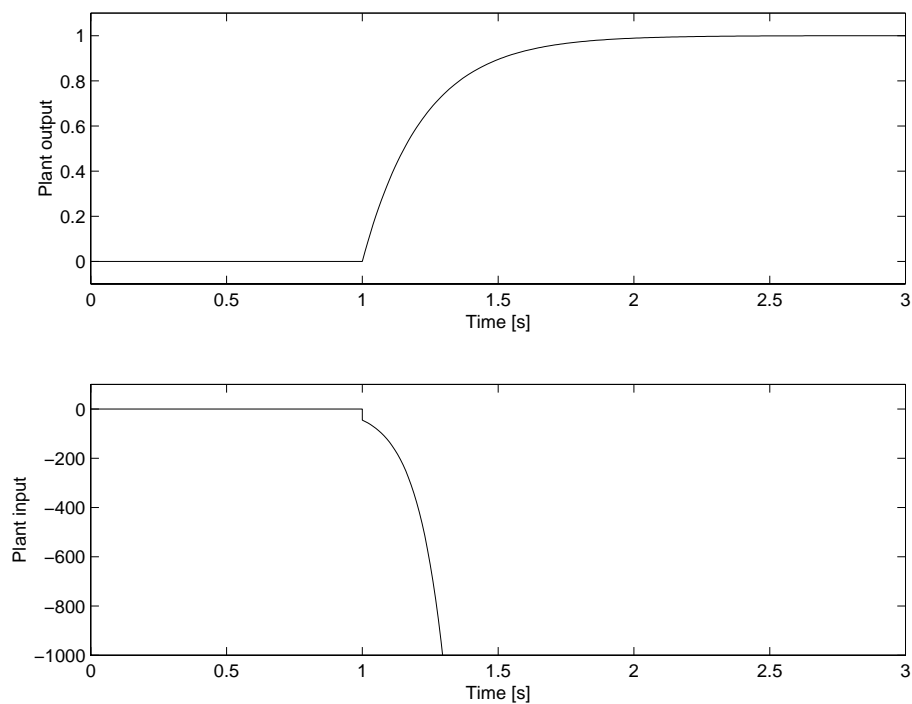
Clearly the above zero dynamics are unstable showing that the plant does not have a stable inverse.

- (2) The system is open loop stable.
- (3) We design a generalized feedback linearizing control law as described above.

Check on the failure of feedback linearization for this example.

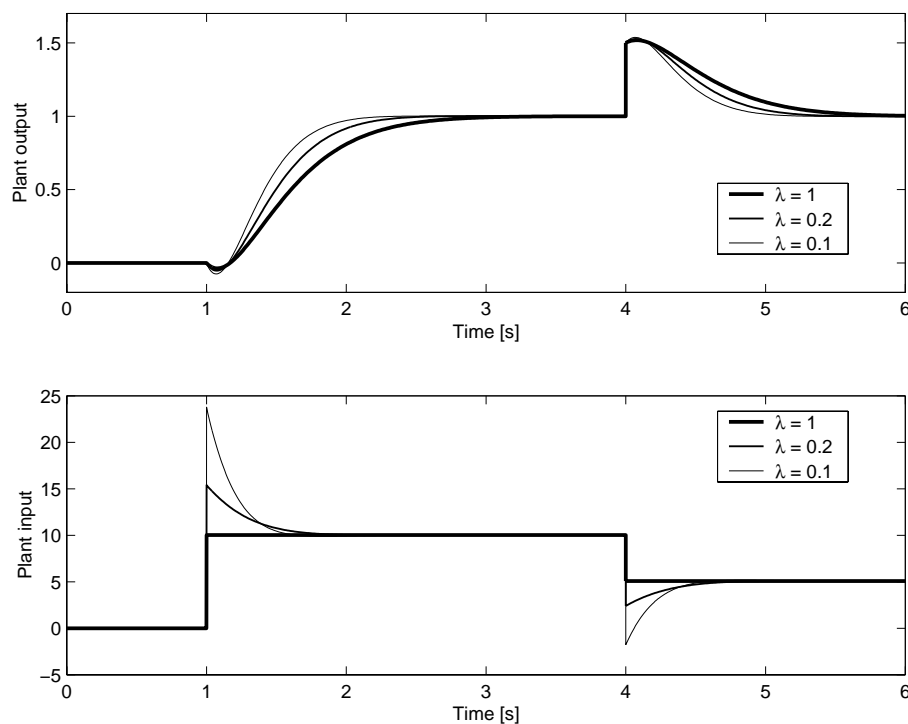
The system was simulated with $\lambda = 0$ and $\ell(\rho) = 1$. This corresponds to the (*basic*) feedback linearization control law. We do not expect that this will work here because the system does not have a stable inverse. Indeed, as shown on the next slide the input blows up when we test the scheme. Note that the output response follows the desired trajectory, however, the input grows without bound. The latter outcome is a result of the nonstable invertibility of the system.

Figure 19.24: *Simulation of basic feedback-linearization scheme*



Next we simulate the generalized feedback linearization scheme for different values of λ (*see next slide*)

Figure 19.25 *Simulation of the generalized feedback-linearization scheme*



Note that as λ decreases, so the response becomes faster and the undershoot increases. Of course, there is a lower limit of λ consistent with u remaining bounded.

We thus see that the Generalized feedback linearization scheme appears to have led to a satisfactory solution in this case.

Of course, this system was (*at least*) open loop stable. We next consider an example which is both open loop unstable and nonstably invertible.

We begin by assuming that the full state is measured.
The results are shown on the next slide.

General Nonlinear Systems

Example of the application of the GFL strategy assuming complete state knowledge:

$$\dot{x}_1 = 10x_1 - 10x_2$$

$$\dot{x}_2 = 9.9x_1 - 10x_2 + 0.1x_2^3 + 0.1u$$

$$y = x_2$$

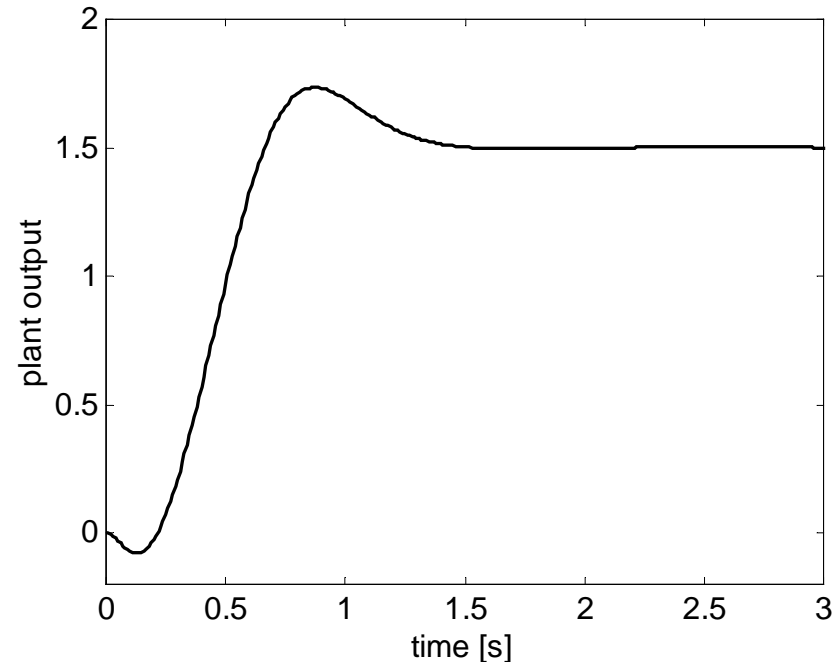
Using:

$$p(\rho) = 0.1\rho^2 + 0.4\rho + 1$$

$$l(\rho) = -4.1\rho + 1$$

$$\lambda = 3.4 \cdot 10^{-3}$$

Step response:



Finally, we use a nonlinear observer of the type described in Section 19.8.1. Note that we also use the observer to estimate the disturbance state which we assume here to be a constant input disturbance.

To achieve integral action with the GFL strategy, we estimate the disturbance using a nonlinear observer.

Model including input disturbance:

$$\dot{x}(t) = f(x) + g(x)(u(t) + d_i(t))$$

$$y(t) = h(x)$$

$$\dot{d}_i(t) = 0$$

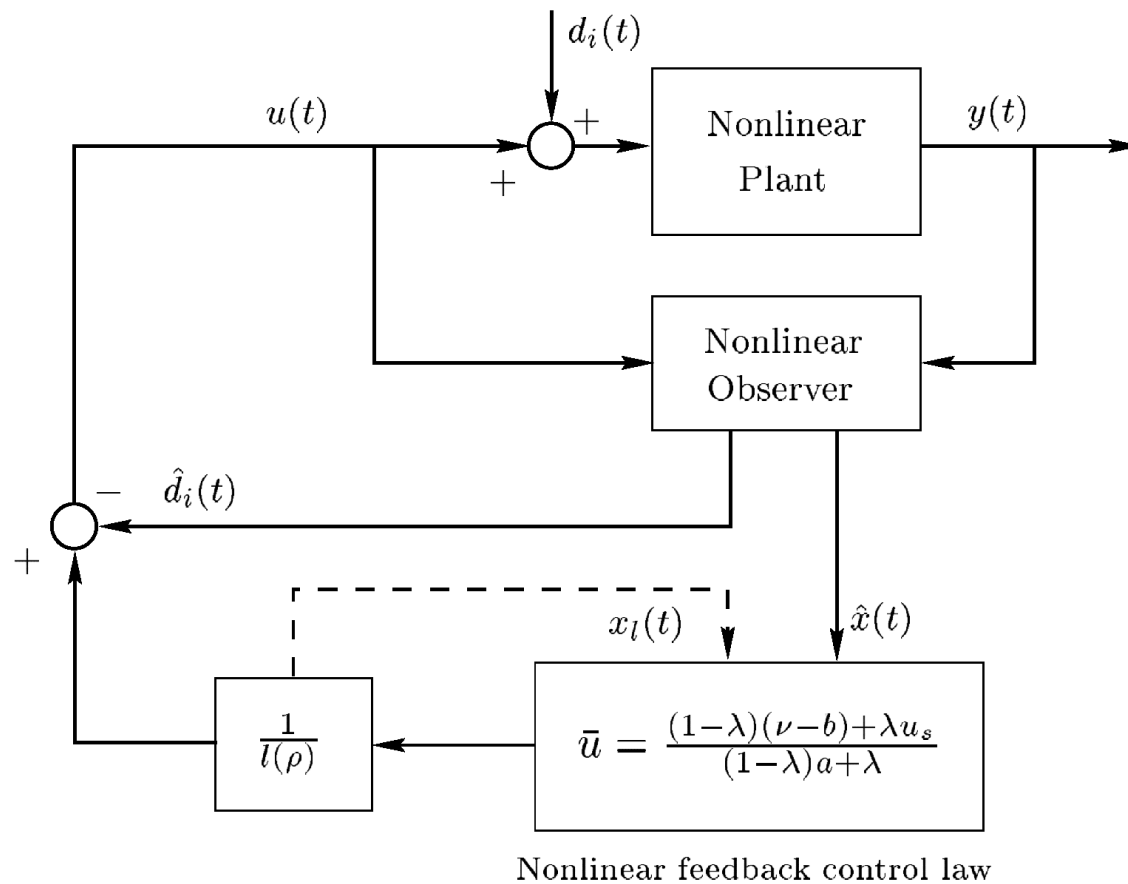
Observer designed via linear output injection:

$$\dot{\hat{x}}(t) = f(\hat{x}) + g(\hat{x})[u(t) + \hat{d}_i(t)] + J_1[y(t) - h(\hat{x})]$$

$$\dot{\hat{d}}_i(t) = J_2[y(t) - h(\hat{x})]$$

This leads to the Generalized Feedback Linearization scheme shown on the next slide.

Input disturbance case:



The above scheme has been designed assuming an input disturbance. However, referring back to Section 19.8, we recall that for nonlinear systems, one needs (*in principle*) to treat input and output disturbances differently. Thus, on the next slide, we suggest a nonlinear observer that might be used when the system is perturbed by a constant output disturbance.

Model including constant output disturbance:

$$\dot{x}(t) = f(x) + g(x)u(t)$$

$$y(t) = h(x) + d_o(t)$$

$$\dot{d}_o(t) = 0$$

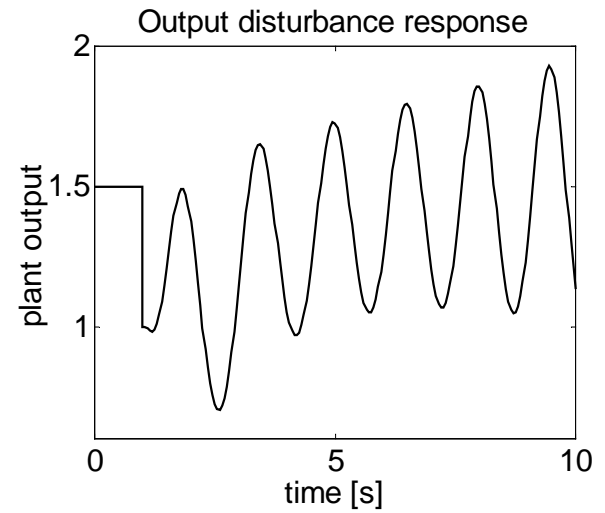
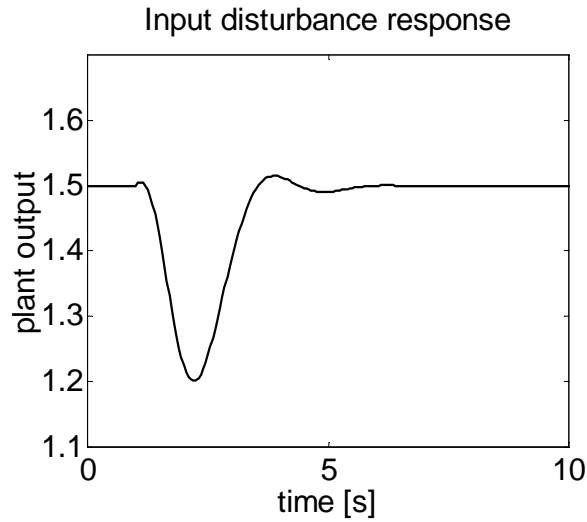
Corresponding observer designed via linear output injection

$$\dot{\hat{x}}(t) = f(\hat{x}) + g(\hat{x})u(t) + J_1(y(t) - h(\hat{x}) - \hat{d}_o)$$

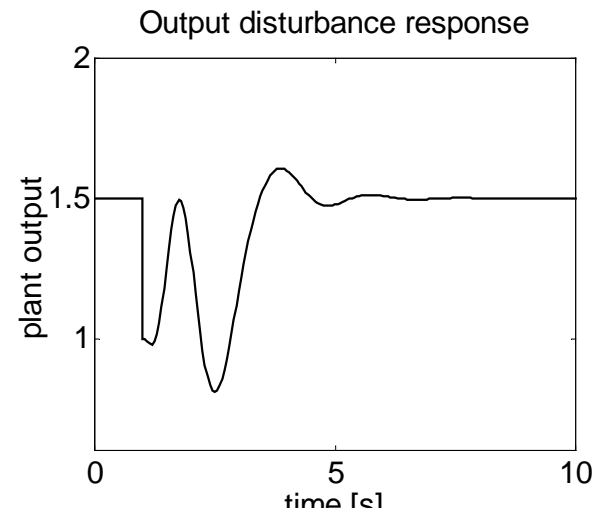
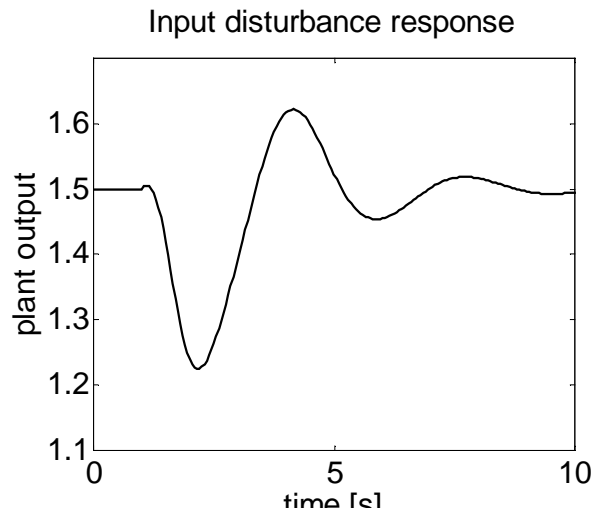
$$\dot{\hat{d}}_o(t) = J_2(y(t) - h(\hat{x}) - \hat{d}_o)$$

We present four simulations below for this unstable, nonstable, invertible example under different scenarios and different disturbances.

*Input
disturbance
design*



*Output
disturbance
design*

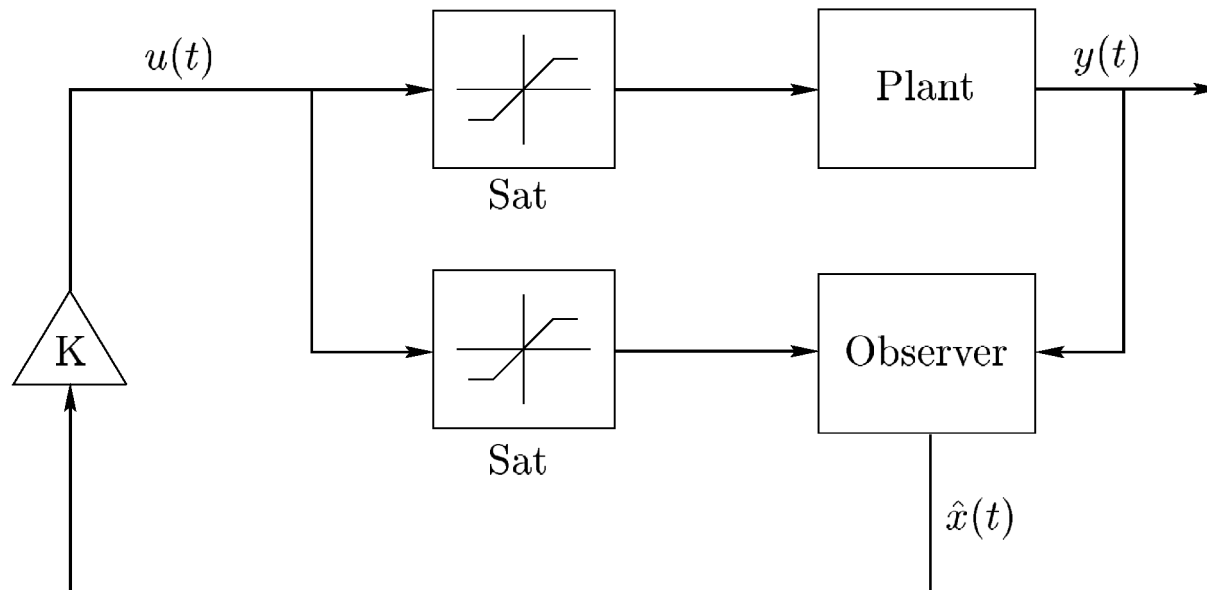


We see that the generalized feedback linearizing control law has performed well when the disturbance injection point is correctly modelled. Less satisfactory performance (*indeed instability*) results when the input disturbance point is incorrectly modelled.

Anti-windup Protection

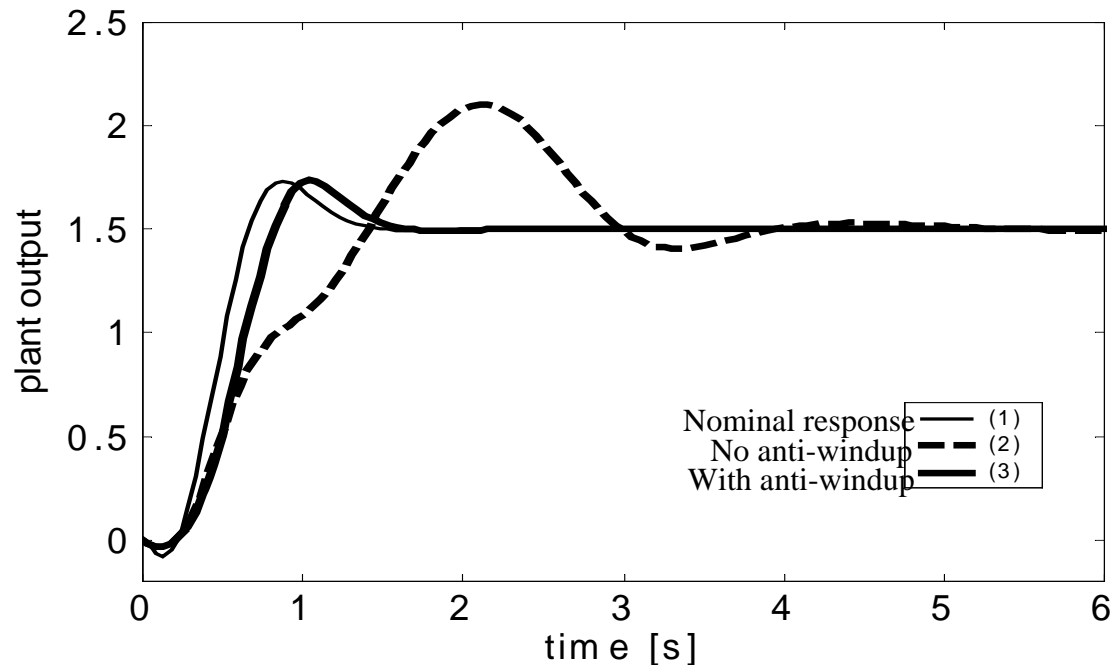
Finally, we ask a very hard question - *What should we do if the input reaches a saturation limit?* Actually, the form of nonlinear controller that we have described above it is immediately compatible with the state space anti-windup scheme described in Section 18.9. An ad-hoc extension of this idea to the nonlinear case is shown below.

-
- ❖ Prevent the integrator from winding up by using an appropriate ad-hoc anti-windup strategy.



We repeat the earlier simulation but here we saturate the input. The results are shown on the next slide.

The implementation of the GFL strategy using state observers allows the inclusion of a form of anti-windup protection.



We see that this nonlinear controller appears to work well for this difficult system which has the following features:

- open loop unstable
- nonstably invertible
- input saturations limit

All of the designs described above have been essentially based on *engineering* considerations. Of course, we have not provided a formal proof of stability of the resulting system. This is a major drawback. Hence, if the reader wanted to use these kinds of schemes in practice, then we would suggest that extensive simulations should be carried out prior to final implementation.

Analysis of Stability for Nonlinear Systems

Having to test stability via extensive simulations is somewhat unsatisfactory. Fortunately in some cases, it is possible to analyze stability using formal methods. The book describes two methods.

- ◆ Lyapunov Stability Analysis
- ◆ Function Space Stability Analysis

We will focus here on the Lyapunov approach.

Lyapunov Stability

The basic idea of Lyapunov stability is to show that there exists a positive definite function (*similar to a measure of total energy*) of the states that is decreasing along trajectories of the system. The positive definite function is usually called a *Lyapunov function*.

Lyapunov functions can be used to assess different types of stability. We formally define the concept of *Global Asymptotic Stability* below:

Definition: Consider a discrete-time system of the form

$$x[k + 1] = f(x[k]); \quad x[k_0] = x_o$$

We say that the system is globally asymptotically stable, if, for all initial states $x[k_0]$ and for any $\epsilon > 0$, there exists a T such that $\|x[k_0 + \tau]\| < \epsilon$, for all $\tau \geq T$.

(Basically, this states that given any initial condition, if we wait long enough, the size of the state will fall below any given number ϵ .)

We next show how a Lyapunov function can be used to assess Global Asymptotic Stability.

Formal Requirements on a Lyapunov Function

If we can find a function $V(x) \in \mathbb{R}$ (*a Lyapunov function*) having the following properties:

(i) $V(x)$ is a positive definite function of x : i.e. $V(x) > 0$ for all $x \neq 0$, $V(x)$ is continuous and is a strictly increasing function of $|x|$, and $V(x)$ is radially unbounded - i.e., $|V(x)| \rightarrow \infty$ for all $\|x\| \rightarrow \infty$.

(ii) V is decreasing along trajectories - that is,

$$-(V(f(x)) - V(x)) \quad \text{is positive definite}$$

We then have, the following theorem, due to Lyapunov.

Theorem 19.1: (*Lyapunov Stability*). The solution of $x[k + 1] = f(x[k])$; $x[k_o] = x_o$ is globally asymptotically stable if there exists a Lyapunov function for the system satisfying properties (i) and (ii) above.

Proof: See the book.

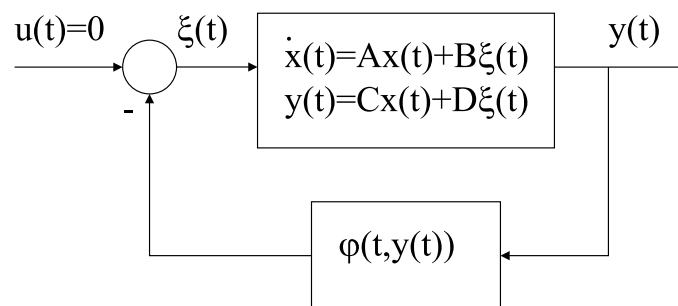
An application of Lyapunov methods will be given in Chapter 23, where we use this approach to prove stability of a general nonlinear-model predictive-control algorithm.

The basic idea of Lyapunov stability can also be used to determine simple stability results that hold in special cases. We illustrate this below by describing tools for assessing stability of a control loop containing a single static (*memoryless*) nonlinearity.

Circle Criterion

The Lyapunov approach to nonlinear stability is a powerful tool. The main difficulty, however, is in finding a suitable Lyapunov function. One class of problems for which an elegant solution to the issue of nonlinear stability exists is that of a feedback system comprising a linear dynamic block together with static (*or memoryless*) nonlinear feedback. This is often called the Lur'e problem - see Figure 19.21.

Figure 19.21: *The single-input single-output Lur'e problem*



To analyze this situation, we can use a very neat result (*known as the circle criterion*). To prove this result, we will utilize a particular Lyapunov function that is matched to this special problem. We also need the technical result given on the next slide.

Kalman-Yacubovich Lemma

Lemma 19.1: Given a stable single-input single-output linear system, $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$, with (\mathbf{A}, \mathbf{B}) controllable, and given a real vector, v , and scalars $\gamma \geq 0$ and $\varepsilon > 0$, and a positive definite matrix \mathbf{Q} , then there exists a positive definite matrix \mathbf{P} and a vector q such that

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} = -qq^T - \varepsilon \mathbf{Q}$$

and

$$\mathbf{P} \mathbf{B} - v = \lambda^{1/2} q$$

if and only if ε is small enough, and the scalar function

$$H(s) = \gamma + 2v^T (s\mathbf{I} - \mathbf{A})^{-1} \mathbf{B}$$

satisfies

$$\Re\{H(j\omega)\} > 0, \quad \text{for all } \omega$$

where $\Re\{\cdot\}$ denotes the real part.

We can now state the following stability result:

Circle Criterion

Theorem 19.2: Consider the Lur'e system illustrated in Figure 19.21. Provided

- (i) the linear system $\dot{x} = Ax + B\xi; y = Cx$ is stable, completely controllable, completely observable, and has a Nyquist plot that lies strictly to the right of $-\frac{1}{k}$, $k > 0$, and
- (ii) the nonlinearity $\phi(t, y)$ belongs to the sector $(0, k)$ in the sense that

$$0 \leq y\phi(t, y) \leq ky^2 \quad \forall y \in \mathbb{R}, \forall t \geq 0$$

then the feedback loop of Figure 19.21 is globally asymptotically stable.

Proof: See the book (uses the Kalman-Yacubovich Lemma together with a Special Lyapunov function)

The previous result can be extended to the case when the nonlinearity lies in a sector (k_1, k_2) in the sense that

$$k_1 y^2 \leq y\varphi(t, y) \leq k_2 y^2$$

For example, we have the following corollary for the case $0 < k_1 < k_2$:

Corollary 19.1: Consider the Lur'e system illustrated in Figure 19.21 with $G(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$. Provided ...

(i) The linear system $\dot{x} = Ax + B\xi, y = Cx$ has n_c unstable poles, is completely controllable, completely observable and has a Nyquist plot that does not enter a circle of center $\frac{-(k_1+k_2)}{2k_1k_2}$ and radius $\frac{(k_2-k_1)}{2k_1k_2}$ but encircles it n_c times counterclockwise. Then, the loop is globally asymptotically stable.

We illustrate the application of the above idea to one of the systems used earlier to illustrate nonlinear control system design. Earlier we stated (*without proof*) that the system was open loop stable. Here we establish this fact.

Example 19.7

Consider the nonlinear system

$$\dot{x}_1 = 10x_1 - 10x_2$$

$$\dot{x}_2 = 16.925x_1 - 16x_2 + 0.1 \tan^{-1}(x_2) - 0.1u$$

$$y = x_2$$

The system can be put into the Lur'e structure, with

$$\mathbf{A} = \begin{bmatrix} 10 & -10 \\ 16.925 & -16 \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} 0 \\ -0.1 \end{bmatrix}$$

$$\mathbf{C} = [0 \quad 1]$$

$$\varphi(y) = \tan^{-1}(y)$$

It is readily seen that $0 \leq y\varphi(y) \leq y^2$

Also, it is readily verified that the Nyquist plot lies to the right of the point -1. Stability of the system follows immediately from Theorem 19.2.

Summary

- ❖ So far, the book has emphasized linear systems and controllers.
- ❖ This chapter generalizes the scope to include various types of nonlinearities.
- ❖ A number of properties that are very helpful in linear control are not - or not directly - applicable to the nonlinear case.
 - ◆ *Frequency analysis*: The response to a sinusoidal signal is not necessarily a sinusoid; therefore, frequency analysis, Bode plots, etc., cannot be carried over directly from the linear case.
 - ◆ *Transfer function*: The notion of transfer functions, poles, zeros, and their respective cancellation is not directly applicable.

-
- ◆ *Stability* becomes more involved.
 - ◆ *Inversion*: It was highlighted in Chapter 15, on controller parameterizations, that, regardless of whether the controller contains the inverse of the model as a factor and regardless of whether one inverts the model explicitly, control is fundamentally linked to the ability to invert. Numerous nonlinear functions encountered, however, are not invertible (*such as saturations, for example*).
 - ◆ *Superposition* does not apply; that is: the effects of two signals (such as set-point and disturbance) acting on the system individually cannot simply be summed (*superimposed*) to determine the effect of the signals acting simultaneously on the system.
 - ◆ *Commutativity* does not apply.

-
- ❖ As a consequence, the mathematics for nonlinear control become more involved, solutions and results are not as complete, and intuition can fail more easily than in the linear case.
 - ❖ Nevertheless, nonlinearities are frequently encountered and are a very important consideration.
 - ❖ Smooth static nonlinearities at input and output
 - ◆ are frequently a consequence of nonlinear actuator and sensor characteristics
 - ◆ are the easiest form of nonlinearities to compensate
 - ◆ can be compensated by applying the inverse function to the relevant signal, thus obtaining a linear system in the precompensated signals. (*Use caution, however, with points singular such as division by zero, for particular signal values.*)

-
- ❖ Nonsmooth nonlinearities cannot, in general, be exactly compensated or linearized.
 - ❖ This chapter applies a nonlinear generalization of the affine parameterization of Chapter 15 to construct a controller that generates a feedback-linearizing controller if the model is smoothly nonlinear with stable inverse.
 - ❖ Nonlinear stability can be investigated by using a variety of techniques. Two common strategies are
 - ◆ Lyapunov methods;
 - ◆ function-space methods.

-
- ❖ Extensions of linear robustness analysis to the nonlinear case are possible.
 - ❖ There also exist nonlinear sensitivity limitations that mirror those for the linear case.