10

Further Control Theory

So far, we have followed the trend and concentrated on developing linear theory. But in the world of mechatronics, very few systems are linear. We have already seen a simulation in Section 6.5 that shows that drive limitation can totally change the way that a system performs. Nonlinearity should be a prime consideration in designing the controller. We will also find that nonlinear elements can be very useful additions to the controller itself.

10.1 CONTROL TOPOLOGY AND NONLINEAR CONTROL

10.1.1 Feedback Topology

We have examined a position control system, a second-order system with a single input. It has a characteristic equation determined by two coefficients that are set by the position and velocity terms in the equation that determined the acceleration. Putting it another way, if we decide on the roots that we want for that characteristic equation, the feedback coefficients are uniquely determined.

When there are more inputs than one, if all the state variables can be measured, we have some freedom of choice in assigning the feedback coefficients. If the system is fourth-order and has two inputs, for example, there are eight elements in the 2×4 feedback matrix. But these determine just four coefficients in the characteristic polynomial.

Some arbitrary methods can be used to give up the freedom and make a choice, but when the response is important, the matter requires careful thought, not just concerning the roots that we might want.

Consider, for example, the pitch channel of an autopilot. There are two inputs to this axis: the elevator control surface and the throttle. There are several state variables, but the pilot's concern is with the height and the airspeed. The obvious strategy is to use the throttle to control the airspeed and the "stick" to control the height. But that is not the way a human pilot thinks of it.

If the airspeed drops, with the danger of the aircraft stalling, the pilot will first push the stick forward. The opening of the throttle is a second measure that must rely on the smooth functioning of the engine for success. It makes sense to use the throttle for controlling height instead, since opening the throttle will increase the flow of energy to the system, meaning that if the aircraft maintains constant speed, it will gain height.

We have two options for the *topology* of the controller. There is the "conventional" one of feeding the airspeed back to the throttle and the height back to the elevator, or the alternative of feeding airspeed to the elevator and height to the throttle. When the system has constraints, the topology can become even more important.

Many years ago I encountered a paper-coating process. After coating, the paper passed through a drying oven before it could be cropped and stacked. A vital factor in the operation is that the paper must not stop. If it does, a large quantity of valuable product has to be scrapped and there is a risk of fire.

Somehow the flow must be maintained while a new roll is pasted on to the tail end of the previous one, and for this the system uses a "magazine," as shown in Figure 10.1.

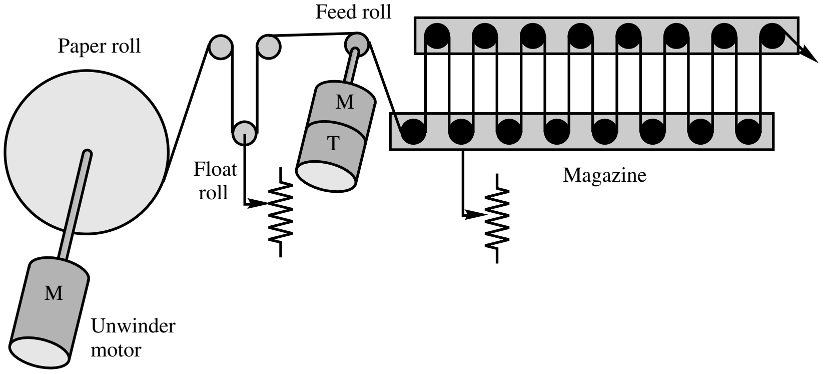

Figure 10.1 Paper coating process.

The magazine contains a hundred feet or more of paper. As the roll comes to an end, it is stopped and the tail clamped, while paper continues to flow out of the magazine. If all is well, the new roll is pasted before the magazine is empty and the roll can be brought up to speed. A motor with tacho feedback controls the speed at which paper is fed into the magazine, feeding 150% of the output speed when the magazine is almost empty and reducing to 90% when it is uncomfortably full.

So, where is the problem? The paper roll is driven by an "unwinder motor," which is in turn controlled by the state of a "float roll." This is a loop in the paper that takes up the fluctuation between the paper roll and the magazine feed motor. The entire variation of this loop might be two feet or less, and if it hits its stop, the whole process is halted. The restart process after pasting on a new roll must be performed with great delicacy.

The unwinder loop is indeed a difficult control problem. Although a second-order equation links motor acceleration to the float-roll position, its coefficients vary wildly. The moment of inertia of the roll will change by a factor of 60 between full and empty. Roll speed is an unreliable measure of paper velocity, so designing a system to have a good response across the range of operation is far from easy. Perhaps a change in the feedback topology can help.

As inputs, we have the drives to the magazine feed motor and to the unwinder motor; as outputs, we have sensors to tell us the positions of the float roll and the state of the magazine, plus the tacho output from the feed motor. What should we feed back to where?

The most critical item in the system is the float roll. A relatively small error can bring disaster. Which input has the most immediate effect on this roller? It is not the unwinder, to which the float roll's signal was originally applied, but the magazine feed motor. Indeed, this motor has a tacho signal that allows the float-roll control loop to be tuned to perfection. So what of the unwinder?

The tacho gives a clear measure of the magazine replenishment speed, so this can be fed back to the unwinder, mixed with the original nonlinear demand function calculated from the magazine state. If there is a large excursion in the startup transient, it is of no importance. The magazine can absorb many tens of feet of overshoot with no problem whatsoever. Figure 10.2 shows the difference between the two alternative control systems.

10.1.2 Nonlinear Feedback and Nested Loops

The "quality" of a mechatronic control system is measured not only by the way it can respond to a change in target or set point but also by the way it can withstand disturbances and recover from them.

As we saw in Section 6.5, the presence of drive limitation can completely change the rules for setting the feedback coefficients. The choice will also depend on the maximum size of the disturbance that can be expected. By introducing a nonlinearity into the feedback, the response can avoid overshoot for any size of initial error or change in demand, but if a disturbing force exceeds the full drive of the motor, it will always win the tug-of-war.

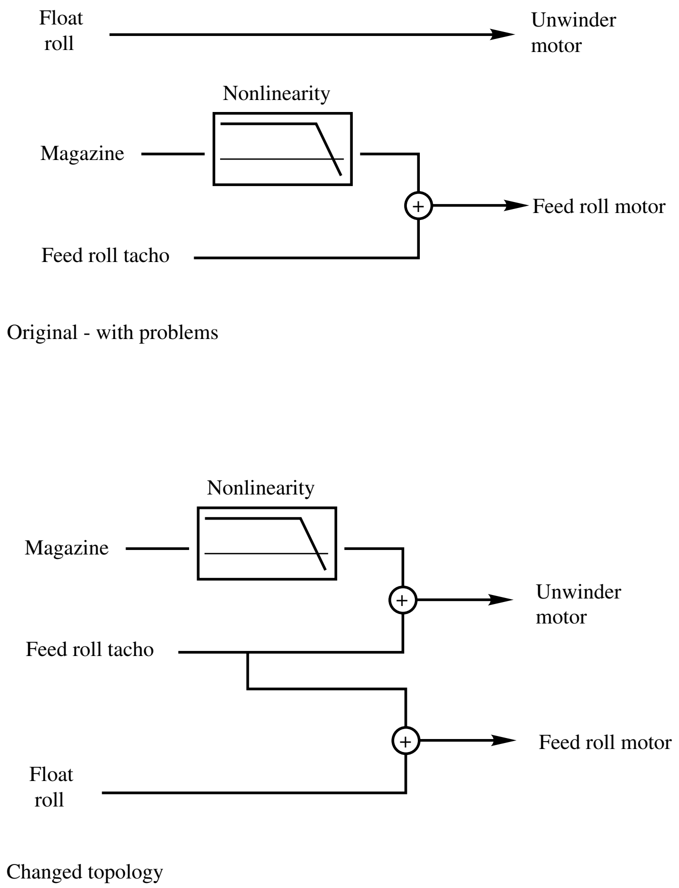

Figure 10.2 *Two control configurations.*

In Section 3.4.5 we experimented with a simple first-order system, the relationship between the drive to a motor and its tacho output. We were able to add a demand signal to the feedback, so that when

$$u = k(v_{\text{demand}} - v)$$

with k taking a large value, the motor will accelerate rapidly to reach the demanded value, applying full drive for any substantial error.

(In principle, the value of k can be infinite, switching the drive from one extreme to another. When the control is discrete time, however, as in computer control, a requirement for a stable response will put a limit on the feedback value that will depend on the sample rate.)

We were then able to give v_{demand} a value proportional to the position error to arrive at a closed-loop position control system. By limiting the value of

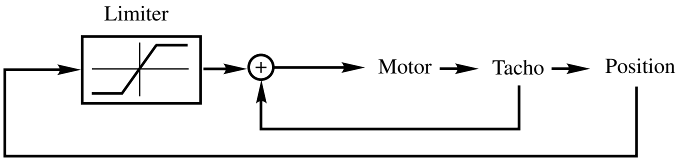

Figure 10.3 *Nested loops.*

v_{demand}, we were able to obtain a response without overshoot for any size of disturbance.

The control appears as two "nested loops" as shown in Figure 10.3.

In Chapter 3, we met the problem of balancing an inverted pendulum. The equations are almost identical with those that describe a bicycle that is being ridden to follow a line. Although designing a controller for a bicycle might not have great practical merit, it makes a very interesting design example.

There are four state variables that concern us. First is the distance from the line, which we will measure from the rear wheel and call x. Next is the angle of the bicycle to the line in radians, α. An important requirement is to remain upright. The angle of lean can be termed θ, and its rate of change is given the label ω. All the variables are *positive to the right.*

The input to the system is the handlebar angle u. The control task is to devise a feedback arrangement that will express u in terms of x, α, θ, and ω and give "good" control.

Let us suppose that the bicycle is proceeding at constant speed V and that each angle is small enough that its sine can be approximated to its value in radians.

First let us set up the state equations. The component of velocity perpendicular to the line is $V \sin\alpha$, so, making the approximation $\alpha = \sin\alpha$, we have

$$\dot{x} = V\alpha$$

A little geometric study will show that the rate of change of α is given in terms of V, u, and the length L between the wheels as

$$\dot{\alpha} = \frac{V}{L}u$$

When we consider the lean of the bicycle, for the first equation we have, of course

$$\dot{\theta} = \omega$$

but the second is less obvious.

The force exerted by the bicycle on the rider has vertical component *mg*, where *m* is the mass of the rider. The horizontal component will be *mg* tan *θ*, so the horizontal acceleration of the rider in the direction of *x* will be *g* tan *θ*. The acceleration of the point where the rear wheel touches the ground is $\ddot{x}$. If we assume that the rider is a point mass a height *h* from the ground, the angular acceleration is related to the difference between these two accelerations as

$$h\ddot{\theta} = g\tan\theta - \ddot{x}$$

From the first two equations involving *x* and *α*, we see that

$$\ddot{x} = V\dot{\alpha} = \frac{V^2}{L}u$$

so, making the usual approximation concerning angles, we have our fourth equation:

$$\dot{\omega} = \frac{g}{h}\theta - \frac{V^2}{Lh}u$$

In matrix form these four state equations become

$$\begin{bmatrix} \dot{x} \\ \dot{\alpha} \\ \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & V & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & g/h & 0 \end{bmatrix} \begin{bmatrix} x \\ \alpha \\ \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ V/L \\ 0_2 \\ V^2/Lh \end{bmatrix} u$$

We can substitute our feedback value for *u*

$$u = ax + b\alpha + c\theta + d\omega$$

to obtain the matrix for the closed-loop system. Then we can find the characteristic equation as in Section 8.1.2. It might be surprising to find that *a*, *b*, *c*, and *d* must all be positive. To move to the left, we must first turn the handlebar to the right.

The various strategies can be tried out on a simulation. To give it numerical values, set *h* and *L* both to 1 m/s and *V* to 2 m/s.

When we come to consider the practical implementation of a controller, we see again that the effect of constraints cannot be ignored. First there must be a limit on the handlebar angle, either from considerations of hitting the rider's knees or from the danger of skidding. The limit might be taken as 0.5 radian.

The lean angle is limited in a different way. If it is too great, the bicycle will skid and the rider will hit the ground. This is a condition that the control must seek to avoid, rather than the sort of constraint that "stabilizer wheels" would impose.

The nested-loop topology can be used to good advantage in devising a controller, assuming that we can measure or estimate all the state variables.

The first requirement is to remain upright, so this loop is closed first. But a demand signal is added into the loop as well:

$$u = c(\theta - \theta_{\text{demand}}) + d\omega$$

It is clear why c and d must be positive. If the bicycle falls to the right, the handlebars must be turned to the right. Their values can be tuned to give a rapid and well-damped correction to any disturbance, while taking the handlebar limits into consideration.

The next loop concerns the bicycle's angle to the line, α. To turn the bicycle, we require it to lean, and the lean loop takes care of the handlebars. To turn to the left, we lean to the left

$$\theta_{\text{demand}} = p(\alpha_{\text{demand}} - \alpha)$$

and to avoid disaster, we must limit θ_{demand} to, say, 0.25 radian. It is θ_{demand} that must be limited in our simulation, not θ.

If we are off the line, we demand an angle that will bring us back to it:

$$\alpha_{\text{demand}} = q(x_{\text{demand}} - x)$$

We need yet another limiter. However far we are off the line, there is a limit to the angle at which we wish to approach it.

If the system has been well designed, we can see the limits taking effect in turn as we follow a large initial offline error to the right.

First we will see the handlebars twitch to the right as the bicycle is required to lean into a turn to the left. For a short while the lean angle will be to the left at the maximum value allowed, while the handlebars are also turned to the left and the bicycle follows an arc of a circle. As the maximum heading angle is approached, the bicycle becomes upright and steers in a straight line. As the target line is approached, the bicycle leans to the right, turning to settle on the target line.

The control loop has a variable structure as each limit comes into play. Constant lean control is only a second-order system. Constant heading control is third-order and it is only when errors are small that the full four orders take effect. Provided the coefficients of θ and ω are well tuned, falling over should not be an option. A wrong choice of a and b will see the bicycle swooping from side to side, but never leaning beyond the value of the demand limit.

I first encountered the nested-loop approach when working on the design of the roll channel of an autopilot many decades ago, but the principles are still true today.

In the roll channel of a rate–rate autopilot, the tightest loop is the feedback around the aileron servomotor. The high-gain velocity control loop will cause the motor to be driven to subdue any disturbances that wind gusts might cause to the aileron control surface. A signal added into this loop will constitute a velocity demand signal.

The next loop is based on the signal from a rate gyroscope. This measures roll rate, and when fed into the aileron loop as a velocity demand, it causes the control surface to move at a rate proportional to the roll rate. The loop is closed through the response of the aircraft to the aileron control, so that a signal added into the loop becomes a roll rate demand.

Now, a passenger airliner has some serious requirements that limit the allowable maneuvers. The passengers would certainly be unhappy if the aircraft rolled at greater than 3° per second, so it is important that a demand signal injected into this loop be limited.

A position gyro that measures roll angle is the sensor for the next loop. There are a few complications associated with creating the roll rate demand from the roll angle error, but this loop now has an input that is the roll angle demand. And, of course, since to roll to an angle greater than 30° would make the passengers decidedly uncomfortable, there has to be a limiter on the demand.

When an aircraft banks (rolls), it flies in a circle. It changes its heading at a rate proportional to the roll angle, so when the pilot wishes the aircraft to fly on a compass heading, the error is fed into the roll demand.

When making the approach for a landing, a radio "localizer" beam results in a signal representing the distance off the centerline of the runway. This signal is added into the heading loop to perform the control. But when the aircraft is "acquiring" the beam, the error is large, so we wish to limit the heading change that it will cause. There is our final limiter. The resulting scheme, somewhat simplified, is shown in Figure 10.4.

On my last day with the firm before leaving for doctoral studies, I experienced a test flight with the autopilot. It worked.

10.2 PHASE PLANE METHODS

We have seen how easy it is to set up a computer simulation of a system and include constraints with a simple "IF" statement. Even so, it is useful to have non-computer-based methods for "back of an envelope" scheming.

10.2.1 Meet the Phase Plane

Many of the problems we encounter will be second-order. If we have just two state variables, such as position and velocity, we can plot the state as a point on graph. As time goes on, the state variables will change smoothly and the

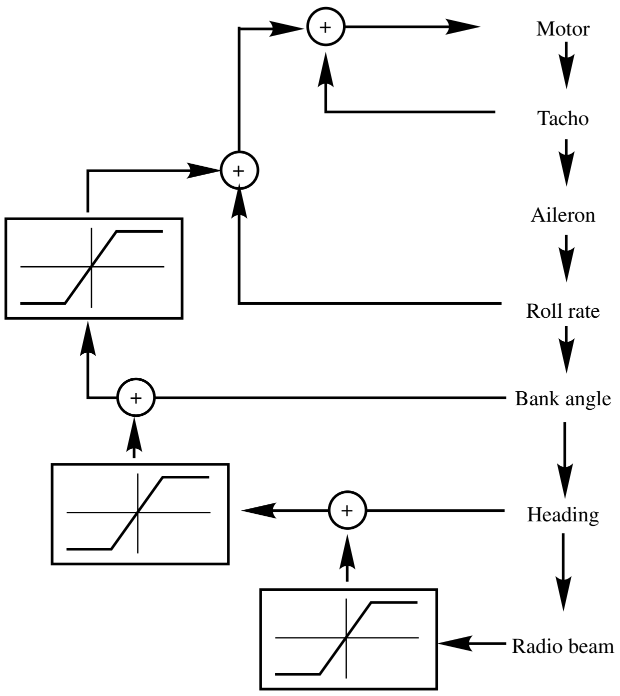

Figure 10.4 *A nested autopilot.*

point will move along a curve, a *trajectory* representing the response of the system.

If all such trajectories lead to the origin of the error–velocity plane and stay there, the system will be *asymptotically stable*. If any trajectory heads off toward infinity, we have an instability problem. But there is a third possibility. A trajectory can form a closed loop, cycling in a *limit cycle* oscillation. This can be annoying, but might not be fatal to the system meeting the design requirements. If there is a region of the plane into which all trajectories lead and from which no trajectory leaves, we have *bounded stability*.

How do we construct these trajectories? We start at some point $(x,\dot{x})$ and begin to draw the trajectory—but in which direction?

We need to know its slope. We need to know $d\dot{x}/dx$, the rate of change of the velocity with respect to the position, not to time. But maybe there is a relationship between this derivative and the time derivative.

It can be shown that if f and x are both functions of time, then

$$\frac{df}{dx} = \frac{df}{dt}\frac{dt}{dx}$$

so this is also true of $\dot{x}$, and we can write

$$\frac{d\dot{x}}{dx} = \frac{d\dot{x}}{dt}\frac{dt}{dx} = \ddot{x}\frac{1}{\dot{x}} = \frac{\ddot{x}}{\dot{x}}$$

The slope of the trajectory is equal to the acceleration divided by the velocity.

The differential equation will give us an expression for the acceleration at any value of position and velocity, so by dividing this by the velocity, we will have an expression for the slope of the trajectory through any point in the plane.

Let us try it out on a second-order system that we have met before:

$$\ddot{x} + 5\dot{x} + 6x = 0$$

Now

$$\ddot{x} = -6x - 5\dot{x}$$

so

$$\frac{\ddot{x}}{\dot{x}} = -6\frac{x}{\dot{x}} - 5$$

Tracing a trajectory by working out its slope at the starting point, drawing a small segment, working out its slope at the next point and so on threatens to be a tedious task—but fortunately there is a shortcut.

At the point (0,1) the slope will be –5. At the point (0,2) the slope will be –5. In fact, anywhere on the line $x = 0$ the slope will be –5. The line $x = 0$ is an *isocline*, a place where the slopes are all the same.

We can easily spot any number of isoclines—in this case any line on which $x/\dot{x}$ is constant, in other words, any line through the origin.

We can start straight in with the two axes. On the x axis, the slope will be infinity—the trajectories cross it at right angles. On the line $x = \dot{x}$ the slope is –11, while on $x = -\dot{x}$ the slope is 1.

We can draw these lines and mark them with small ticks in the direction of the trajectories, a sort of unfinished spider's web (see Fig. 10.5). Then we can draw the trajectory from some starting point, bending it to obey the slope as it crosses each isocline.

We need some more isoclines to get an accurate plot, especially three particular isoclines in this case.

The line

$$5\dot{x} + 6x = 0$$

is special because the acceleration there is zero. The trajectories cross it with zero slope, parallel to the x axis.

But what about the line $\dot{x} + 2x = 0$? The slope on this line is $-5 - (-6/2)$, which gives a result of –2.

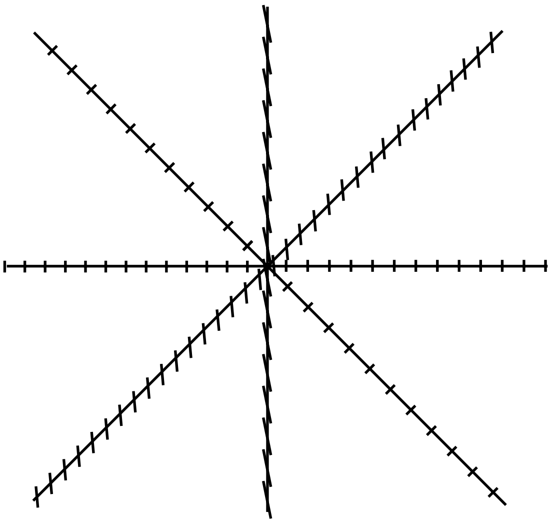

Figure 10.5 *Axes and diagonals with ticks.*

In other words, on the line with slope –2, the trajectories also have slope –2. Any trajectory reaching or starting on this line will be "glued" to it.

We find that the same is true for the line

$$\dot{x} + 3x = 0$$

After drawing these onto the diagram, a good sketch of the phase plane can be created (see Fig. 10.6).

Let us take another look at these "special" isoclines. If the trajectory follows

$$\dot{x} + 2x = 0$$

this is not just the equation of a line; it is a differential equation. It tells us that

$$x = x(0)\mathrm{e}^{-2t}$$

which should not be surprising, since we have already found the general solution of this differential equation to be

$$x = A\mathrm{e}^{-2t} + B\,\mathrm{e}^{-3t}$$

The two special isoclines are the special cases where A or B is zero. But we can learn a little more. As time advances, the e^{-3t} term will decay faster than the other term, so the trajectories will become asymptotic to the line

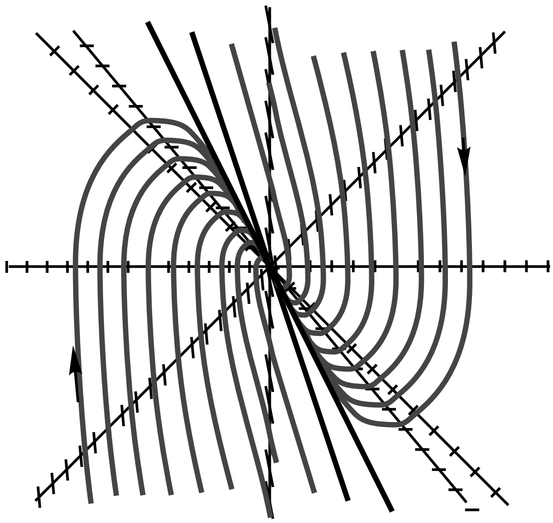

Figure 10.6 Phase plane sketch.

$$\dot{x} + 2x = 0$$

On the other hand, if we trace the trajectories backward into the past, the e^{-3t} term will become dominant, so the slopes will become asymptotic to –3.

As an exercise, sketch the phase plane for the system

$$\ddot{x} + 3\dot{x} + 6x = 0$$

When you solve the characteristic equation to find the "special" isoclines that are asymptotes, you will find that the roots are complex. These isoclines do not exist. The system is underdamped and the trajectories perform spirals around the origin as the system "rings."

10.2.2 Dealing with Constraints

We seem to have devoted considerable effort to deal with a system that we had already analyzed analytically. But the analytic method is tailored only for linear systems. The phase plane comes into its own when there are constraints and other nonlinearities.

Let us again consider the system

$$\ddot{x} = u$$

where

$$u = -5\dot{x} - 6x$$

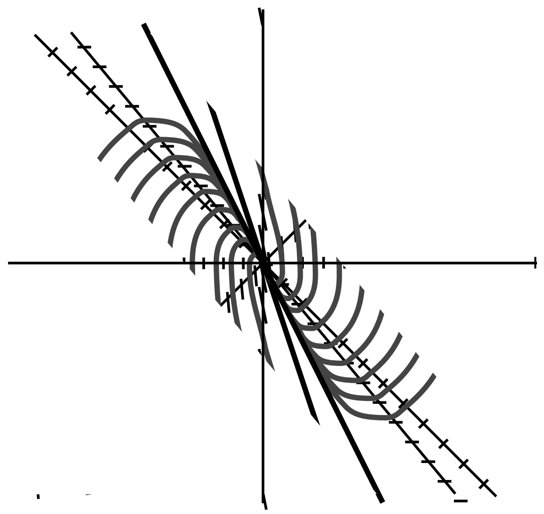

Figure 10.7 *Linear region of phase plane.*

But this time we have a limit on u, namely, $|u| \le 4$.

Close to the origin the drive is not saturated and the phase plane is just as we have drawn it, but outside the linear region the system equation is

$$\ddot{x} = 4$$

or

$$\ddot{x} = -4$$

The boundary of the linear region will be the two lines

$$-5\dot{x} - 6x = \pm 4$$

These are parallel to the line on which the drive is zero, so the linear region appears as shown in Figure 10.7.

In the two other regions, the slope is given by

$$\frac{d\dot{x}}{dx} = \frac{4}{\dot{x}}$$

and

$$\frac{d\dot{x}}{dx} = -\frac{4}{\dot{x}}$$

The isoclines in both cases will be lines of constant $\dot{x}$, parallel to the x axis.

The trajectories will be parabolas, and the complete phase plane will resemble Figure 10.8.

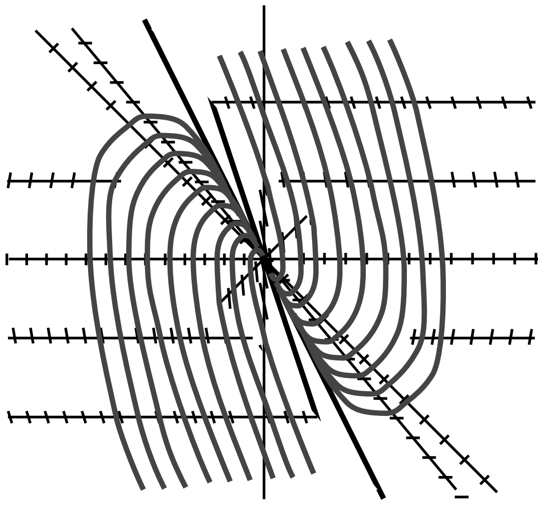

Figure 10.8 *Composite phase plane.*

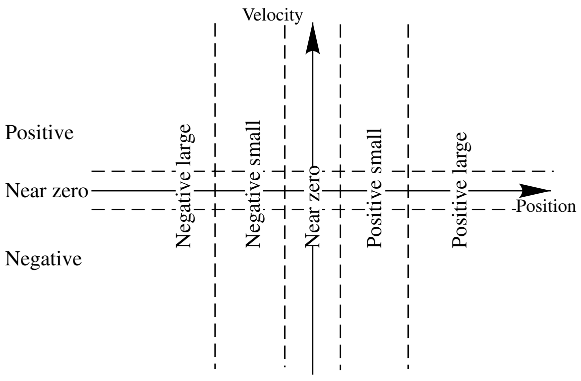

Figure 10.9 *Fuzzy logic phase plane.*

The phase plane can deal with a wealth of nonlinearities, including friction and deadband and, with a little ingenuity, backlash. It can also be used to try out a variety of nonlinear feedback strategies.

One approach that has been fashionable is termed "fuzzy logic." Instead of a precise measurement of position and velocity, their values are simply reported in terms such as "near zero," "positive small," "positive large," and so on. These divisions will divide the phase plane into a tartan pattern of combinations of position and velocity ranges similar to that shown in Figure 10.9. In each rectangle, the control designer can put any available value of drive. On one hand this will overcome the weakness of linear feedback strategies, but on the other hand there are some serious limitations on performance.

The system can come to rest anywhere in the near-zero range of x with no further attempt at correction, leaving a standing error. To avoid this, the near-zero range can be omitted, leaving the zone boundary as the axis. But now we can be left with a limit cycle or at best multiple overshoots as the drive switches to and fro.

If the velocity is added to the position as continuous signals before the values are cropped into ranges, we can, of course, have a crisp nonovershooting response. But this is not how the game is usually played.

10.3 OPTIMIZATION

If the requirement is simply one of stability, there are untold possible variations in the feedback parameters. The "by the book" control system designer would like to find a unique solution that is somehow "the best." This is optimal control.

10.3.1 Least Squares

If we are seeking the best, we must have a measure of the quality that we are trying to optimize. The problem is described in the form of a *cost function*, and the design task becomes one of minimizing that cost function. The cost function could be something explicit, such as settling time or fuel consumption, but a textbook favorite is *least squares*.

For the second-order system

$$\ddot{x} = u$$

we might choose a cost function

$$C = ax^2 + b\dot{x}^2 + cu^2$$

After some mathematical manipulation, we discover that the controller that minimizes the integral of C is based on proportional feedback. Indeed, if $b = 0$, so that the cost involves only the position and the input, the solution has a damping factor of 0.707.

Examples can be found in process control, where "gentle" adjustment is in order. The controller acts as a "regulator," keeping the process at an optimal setpoint while countering any disturbances.

However, we have already seen that if the input is limited, as in a servomotor, the design should depend heavily on that limit. Selecting a quadratic cost function as the basis from which to design the controller loses any logical reason when the coefficients of the cost function have to be "fiddled" to give an acceptable response.

10.3.2 Time-Optimal Control

When we have a "real" cost function, such as the time to reach zero error with all derivatives at zero, the solution is usually *bang-bang* control. The input is at all times at one or the other limit until the target is reached. Techniques such as *dynamic programming* and the *maximum principle* can define the nature of the switching function, but it takes more ingenuity to find the actual switching times that will bring all the errors simultaneously to zero.

For our simple example of acceleration control, the *maximum principle* deduces that just one change of sign is required to bring the system to the target. But when? Imagine that you are driving from one traffic light to the next in minimum time, in a vehicle with just one gear and insufficient power to "burn rubber." As the light turns green, you must put the pedal to the floor. At some point before hitting the next red light, you must apply full brakes. When?

For minimum time, you must hit the brakes at the last possible moment from which you can actually stop before the light. Your *time-optimal trajectory* consists of a period of maximum acceleration switching to a period of maximum deceleration, coming to rest at the target when zero drive is applied. The "quality" of your control depends on your ability to model the braking process accurately with some sort of switching curve. If your estimate of your braking power is overoptimistic, or if there is any disturbance that pushes the car on its way, then an overshoot is unavoidable.

If, on the other hand, the braking deceleration is underestimated, the settling time will be slightly increased but there will be leeway to account for mishaps. The nature of the control will be *sliding*. Brakes will be applied as the switching line is crossed, but the greater-than-predicted deceleration will take the state across the line again and acceleration will be applied. The drive will switch rapidly to and fro, causing the state to follow the switching line. There is a simulation example at www.essmech.com/10/3/2.htm.

Time optimization has much in common with the task of fuel optimization in a lunar lander. Many years ago, our Cambridge group received a visitor from Moscow. He told of the computational task of calculating the control to bring the first unmanned lunar probe to rest with minimum fuel consumption.

The nature of the solution was the same as that for time-optimal control. The probe is allowed to fall freely until the last moment, when continuous full drive will just bring it to rest as it touches the surface. Unfortunately, at that time the first two or three probes had landed far from softly.

At the time that the final burn is initiated, the probe might be falling at a mile per second. A one-second error will leave the probe irrevocably heading on a trajectory that would end a mile beneath the surface, if an impact did not intervene!

My suggestion, that a deliberate underestimate of the thrust would cause a minimal increase in the fuel actually consumed, was passed on very tactfully by our professor. The next probe landed successfully.

Optimization might serve the purpose of giving a unique solution that can be claimed to be "right," but it is seldom the best in practice. The function that needed to be optimized in this particular case was the probability of a "successful" landing. Any fuel that remained after the landing was of no value whatsoever.