CHAPTER 15

Electronic Subsystem Design

> When art critics get together, they talk about style, composition, and technique. When artists get together, all they talk about is where to get the best turpentine.
>
> —Pablo Picasso

15.1 INTRODUCTION

In Chapter 14, we talked about electronic components and fundamental circuits. Chapter 18 is about building fast, quiet photodiode amplifiers. Here in the middle, we'll discuss design philosophy and strategies—how to approach the problem of synthesizing an electronic subsystem, with emphasis on the analog back end: RF signal processing, pulse detection, and analog to digital conversion. There are some digital bits, but they're simple—counters, registers, and state machines.

The most beautiful circuit and instrument designs come out of a really deep knowledge of undergraduate physics and engineering. Beauty in this sense consists not only in conceptual simplicity and high performance, but in robustness, manufacturability, and low cost.

15.2 DESIGN APPROACHES

The right approach to a design depends a lot on where you're starting from. If you have a clean sheet of paper in front of you, you have choices not available to the guy who's just putting more buttons on an existing design.

There is no substitute for experience here, but fortunately it doesn't all have to be your experience—there's a lot of design lore available. Really good analog circuit designers are fairly rare but have a subculture of their own. Have a look in some of Jim Williams's books, the ARRL handbook, and the linear applications books of National Semiconductor, Linear Technology, and Philips Semiconductor (see the Appendix). The Usenet group sci.electronics.design is a good place for detailed design questions. For help in getting started, try sci.electronics.basics.

Some of the circuit elements in this section, especially the details of the SA605 IF chip and its RSSI (meter) outputs are discussed later in the chapter, so turn there if you're unfamiliar with them.

Building Electro-Optical Systems, Making It All Work, Second Edition, By Philip C. D. Hobbs

15.2.1 Describe the Problem Carefully

The first thing you need is a good, brief description of the problem. If you're in charge of a big chunk of the design, it need not be elaborate; something like the following, which is a leaked excerpt from the system spec for an acousto-optically and galvanometrically scanned laser bug zapper (coming soon to a store near you). The system concept is to use a HeNe beam to sense flying insects, and a 50 mJ Q-switched Nd:YAG laser to vaporize them.[†] The HeNe beam makes two passes through the A-O cell, so that its frequency offset is doubled.

> **Section 2.23: Detector/Digitizer** The detector/digitizer must detect Gaussian-envelope tone bursts, 1 μs wide at $1/e^2$, arriving in a Poisson process of average rate less than 1 kHz. The peak amplitude of the bursts is 0.2 mV to 1 V (50 Ω), and the carrier frequency f_C is 100–200 MHz. The value of f_C depends on the laser tuning and system time delays, and so must be software-adjustable, but at any given time it is known in advance to an accuracy of ±5 MHz. The tone bursts are immersed in additive white Gaussian noise of −160 dBm/Hz. The system is to produce a 1 μs trigger pulse beginning within 1.5 μs of each event. Its output is to be three 14-bit serial digital words per event, corresponding to the peak amplitude ±0.2% (1σ), scan position and laser power. Efforts should be made to exceed these minimum dynamic range specifications as far as practicable, since improvements there will translate directly into improved instrument performance. The digitizer must accept external control lines DDEN' (negative-true), which enables triggering the digitizer on detection of a target, and XTRIG′ (falling edge active), which initiates an immediate conversion cycle regardless of the target status. XTRIG′ will be asserted not less than 1 μs after DDEN′ goes active.

15.2.2 Systems Engineers and Thermodynamics

Now it's time for a back-of-the-envelope calculation to determine how hard it is, and in this case we immediately find that the spec is full of problems.

Noise Floor. As we showed in Section 13.6.2, the 1 Hz Johnson noise power in an impedance-matched circuit is kT, which is −174 dBm/Hz at room temperature. The specified noise floor is already 14 dB above there. A garden variety low noise amplifier has about a 3 dB noise figure (i.e., $T_N = 300$ K). That kT per hertz and the incoming noise power add, so the fractional change is

$$\Delta P_N(\text{dB}) = 10\log\left(\frac{10^{1.4}+10^{0.3}}{10^{1.4}}\right) = 10\log 1.079 = 0.33 \text{ dB}. \qquad (15.1)$$

(This sort of move is how you should think of most things in signal processing, because the absolute levels generally aren't as significant as the ratios.) Our part of the system should thus be reasonably quiet, but needn't be as quiet as a front end amplifier—so far, so good.

[†]The optical designer is having his problems too—distinguishing bugs from aircraft, for example, and deciding what to do about a mosquito that's just landing on someone.

Frequency Plan. The wide frequency range sounds like a job for a superhet, like the AM radio of Example 13.6, because the measurement bandwidth is so much narrower than the tuning range and because the pulse envelope is what we care about, just as in AM. We need to get the LO from somewhere, and a tuning dial obviously won't do it—perhaps a DAC-controlled VCO or perhaps a low resolution frequency synthesizer, depending on how fast that "known in advance" changes. It isn't immediately obvious how we're supposed to know what the frequency is to be—do we get a reference signal, a digital word, or what? The instantaneous measurement bandwidth is at least 11 MHz, counting ±5 MHz for the carrier frequency uncertainty and a 1 MHz δf for the tone bursts ($\delta f \approx 1/\text{FWHM}$). For best SNR, we'd like to use a matched filter, whose bandwidth will be $\sim\delta f$ (see Section 13.8.10). Because the noise power goes as the bandwidth, we're giving away at least 10 dB in sensitivity by accepting that huge f_C uncertainty, at least if we do our detection in one band, and it'd be worth hearing the reason it has to be so big.

Dynamic Range. That +10 dBm (10 mW) maximum signal level looks a bit ugly. Remember from Section 13.5 that the 1 dB compression level is typically $P_{\text{LO}} - 7$ dB, and that the amount of compression is approximately linear in P_{RF} for small compressions—that is, the 0.5 dB compression point is 50% of the 1 dB point ($\approx P_{\text{LO}} - 10$ dB). That 0.2% is 0.017 dB, which is 1/60 of 1 dB. By our linear rule, the compression point has to be about 60×10 mW, a little matter of 600 mW—even ignoring the first stage gain. Using the $P_{1\text{dB}} = P_{\text{LO}} - 7$ dB rule, we'll need a diode bridge mixer with about a 3 W LO level. You can't get mixers that strong, and even if you could, there's just something wrong with having to put a heat sink on a mixer. Even if we perpetrated such an atrocity, we'd still be in the soup because (due to their insertion loss) diode mixers have about a 6 dB noise figure. Because of that, we'd need at least 10 dB of front end gain to keep the noise floor where it should be, so the formula predicts an impossible 30 W of LO injection. You might want to verify that assertion, as in (15.1). Things are definitely looking a bit stickier.

One possibility would be to attenuate the signal before mixing it, because a signal N dB smaller signal needs N dB less LO power. Unfortunately, that makes the problems worse at low P_{in}, because an N dB attenuator has an N dB noise figure if it's connected to a matched load at 300 K—which means that the weakest signals will disappear into the Johnson noise of the attenuator. Might be a job for two subranges (using range switching or two complete receivers), or possibly even direct detection at the RF frequency. Let's press on.

Digitizer. The signal power level inside a tone burst ranges from 400 pW to 10 mW (−64 dBm to +10 dBm), a 74 dB range. As we saw in Section 13.6.9, the amplitude error $\langle \Delta A \rangle / A = 1/(2 \times \text{CNR})^{1/2}$, so if we have a spec on $\langle \Delta A \rangle$, we can convert that to a CNR spec: $\text{CNR} \geq 1/(2(\langle \Delta A \rangle / A)^2)$. Meeting our specified relative accuracy of 0.2% thus requires a carrier-to-noise ratio of $1/(2 \times 0.002^2) \approx 125{,}000$ or 51 dB, so the required digitizer dynamic range is 74 dB for the signal level plus 51 dB for the accuracy, or 125 dB. Assuming that "1σ" means the $1/\sqrt{12}$ ADU quantization noise, we know from (14.31) that in a unipolar measurement, an ideal N bit ADC has a dynamic range of $6.02N + 10.78$ dB, which is 95 dB for 14 bits and 107 dB for 16 bits. Thus the signal makes it inside the 14 bit nominal dynamic range, but not by enough to give 0.2% relative accuracy anywhere near the lower limit—and that's assuming we really

have that low a noise floor in the ADC, which is starting to be dubious at that resolution. Perhaps the spec means *0.2% of full scale*, in which case we're fine. On the other hand, that needs only 9 bits, so it isn't obvious what the lower 5 bits are for in that case. Alternatively, we might need to use an ADC with enough bits to span that 125 dB, but that's a 21 bit part; these exist but are extremely slow, more like 10 ms than 1 μs, and never ever get near those dynamic range numbers in real life (see Section 14.8.3). We'll have to clarify the spec there, by thinking some more or asking the system designer.

Noise Budget. Here's where the real bite comes. With a 51 dB CNR spec at $P_{\text{in}} = -64$ dBm, the noise power in the measurement bandwidth has to be less than -64 dBm $-$ 51 dB $= -115$ dBm. Our input noise floor is -160 dBm/Hz, remember, so in our 11 MHz bandwidth, the noise power is -160 dBm/Hz $+ 10 \log(11 \text{ MHz}/1 \text{ Hz})$, which comes to -89 dBm; we're off by a good 25 dB from being feasible here, regardless of the ADC. Now it's really time to go talk to the systems person (or possibly to the mirror, depending on the size of the engineering team).

★ ★ ★ ★

That somewhat awkward interview is over, and we have a revised accuracy spec of "0.2% of full scale or ± 1 dB, whichever is less." It also became clear that that 0.2% is a bit squishy, because there's going to be an online, end-to-end calibration facility in the instrument, so what we really need is 0.2% repeatability and stability over times of a few minutes. ("So *that's* what DDEN′ and XTRIG′ were for.") The frequency uncertainty has been tightened to ± 2 MHz by making some other poor slob work harder, but that's the limit, because of the Doppler shift of the scatterer.

It turns out that the need for the wide tuning range comes from scanning back and forth. Because of the signal detection delay, by the time the detector/digitizer has produced its pulse, the HeNe beam will no longer be on the bug, so the pulsed YAG laser has to trail it slightly. The angular offset required is opposite on the two half-cycles of the scan, so we have to use an acousto-optic deflector (AOD) on the HeNe beam and change the acoustic frequency as we go. This doesn't have to track very fast (1 ms or so), so we can probably derive our LO from the AOD drive signal, although it'll have to be frequency-doubled and offset; the HeNe beam is diffracted on both transmit and receive, which doubles the frequency shift (see Section 7.10.7).

ADC Resolution Again. At the 0.2 mV end, the ± 1 dB accuracy spec means that the CNR has to be $\geq 1/(2 \times (10^{0.05} - 1)^2) \approx 15$ dB, at the very least (see Section 13.6.9), which is still pushing it a little, since it implies a dynamic range of 74 dB + 15 dB = 89 dB. Furthermore, we want each measurement to be accurate to 1 dB at worst, which means we can't use the rms uncertainty of $1/\sqrt{12}$ ADU, because that's an average over an ensemble of measurements. Perfect quantization has a maximum error of 0.5 ADU, so even with a very good DNL spec of ± 0.5 ADU DNL, we're looking at ± 1 ADU at best. Since we have to deal with the inaccuracy of the calibration, it's really more like ± 2 ADU. We're making a unipolar measurement, so we don't have to accommodate the peak–peak range of an AC signal in our ADC range (a factor of $2\sqrt{2} \approx 9$ dB). Our 14 bit A/D has 16384 ADU signal/2 ADU error ≈ 78 dB dynamic range on this basis. We'll have to compress the signal in the lower range a bit, but that isn't too hard to do

at 1 dB accuracy. The door is open to a nonlinear pulse height detector, and probably to our multiple subrange idea.

We couldn't get the front end gain cranked up any higher to loosen our noise figure spec. The maximum signal is already +10 dBm, and making that much bigger will start to cause linearity, EMI, and power consumption problems. We conceded the point reluctantly. It isn't that the expensive mixer with the 3 W LO power wasn't in our minds too, but that end-to-end calibration will greatly ease the compression problem, allowing a much more sensible LO level—we can comfortably go right up to the 1 dB compression point. We can use a low noise amplifier on the low range, where we have noise floor problems, and an attenuator on the high range, where it's saturation we're worried about.

Aside: The Old Way. In the palmy days (say, 1982), when accurate ADCs were very expensive, we'd have been thinking along different lines: "Well, allowing a factor of 2 for calibration errors, ±0.2% of full scale needs 500 codes all told, and ±1 dB for the remaining 40 dB or so needs another 40. We can do it in 10 bits, with a carefully tweaked DLVA summed with an AM detector, and maybe 9 if we pay close attention to calibration errors." Now that 14 bit ADCs are cheaper than tweaking, and computing power is cheap too, using one of those plus software calibration is a lot more reasonable.

15.2.3 Guess a Block Diagram

The next thing is to make a guess as to a good block diagram of the system. Not only does it describe how the system works, but it gives you a seat-of-the-pants feeling for how complicated it's getting. As we expected, this is a basic superhet problem, like the AM radio of Example 13.6, except we have to get the LO from somewhere. Since we can use the 50–100 MHz AOD drive to get our frequency reference, we have two basic choices: use mixers and a frequency doubler, or a PLL with a frequency offset and a ÷2 counter (see Section 15.6.2). The direct method of Figure 15.1a is fast and has no PLL acquisition funnies, but the PLL of Figure 15.1b is simple and has far fewer spurs, because we can filter its control voltage vigorously (the only way spurs can get into the oscillator output is by coming in on the control voltage line, a key PLL advantage in frequency synthesis). A 50 kHz loop bandwidth will get us fast acquisition and a settling time of less than 100 μS, which is fine. Let's draw both (Figure 15.1).

We need at least a few cycles of the carrier to get a decent envelope. A rule of thumb is that 4 is the minimum, 10 is comfortable if the timing constraints are loose, but the tighter the timing the more cycles you need. The input pulse will have at least 100 cycles, so any IF above 10 MHz or so will be fine. We could use a 70 MHz TV IF filter, which avoids the RF and LO frequencies, but the 50–100 MHz reference input crosses that IF, which will lead to nasty spurs since mixers have only so much RF rejection. Thus we're better off staying below 50 MHz. Let's choose 21.4 MHz, which is a standard frequency for wideband IFs and leaves some room for the filter to roll off.

The direct method needs three mixers, two oscillators, two filters, and a doubler, which is on the complicated side. It will also generate a weak spur exactly at 21.4 MHz ($2f_{LO1} - f_{LO2}$), which will get into our IF eventually unless we filter really well between the first mixer and the doubler. On balance, the PLL method is superior, but harder to get chips for; the highly integrated amp/mixer/oscillator chips almost always have the VCO

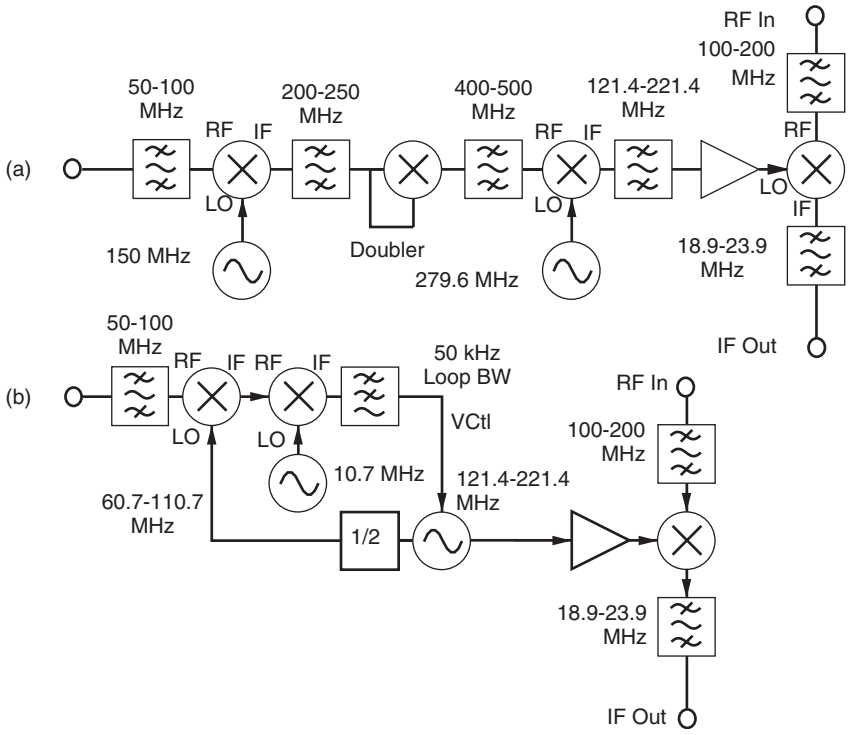

Figure 15.1. Pulse detection signal processing: (a) direct synthesis and (b) offset phase-locked loop.

connected right to the mixer, which we can't tolerate here. We could change it around to frequency-double the 50–100 MHz and then use a PLL to offset that, but that would require a tracking filter because the frequency range is a whole octave. What we'll do is to use an offset PLL at 1:1, with a doubler afterwards, as shown in Figure 15.2. The advantage of this is that as long as the offset frequency is well outside the PLL's loop bandwidth, and the VCO tuning range is restricted so that it can't lock onto the wrong sideband, we won't have any spurious problems. Figure 15.2 also shows the detection strategy: a single LO chain, with two mixer/detectors; one has a 30 dB attenuator in front of it. We sum the RSSI outputs from the two sections and take our AM output from the attenuated one. For a single digitizer, we can just sum them all, since the result will be monotonic and the (fairly serious) temperature drift will be taken out by the end-to-end calibration.

Aside: Sneaky Sideband Selection. At the offset frequency, where the PD operates, the upper sideband corresponds to a phase of 0 and the lower sideband to $\pm\pi$, due to the phase subtraction. This gives us another way to select sidebands. The idea is to use a phase–frequency detector like the one in a 74HC4046, which ideally has a sawtooth $V(\phi)$ characteristic. Although propagation delays prevent the step at $\pm\pi$ from being infinitely steep, its slope (and hence the loop gain) is a good 500 times larger than at 0. Jacking up a PLL's loop gain by over 50 dB is a sure-fire way to make the loop unstable, so the offset loop is unable to lock up at the wrong sideband. Thus we can choose sidebands by

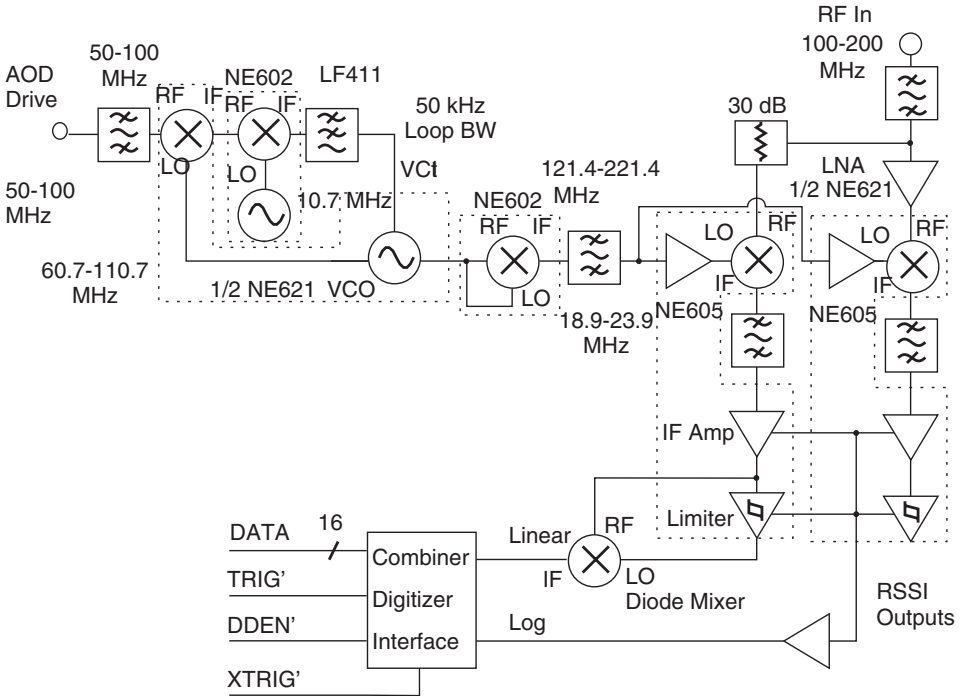

Figure 15.2. Final signal processing block diagram: a PLL similar to that in Figure 15.1 generates the LO, and two parallel single-conversion superhet receivers handle the wide dynamic range. The logarithmic outputs are summed, and the large-signal channel produces a linear output as well, which will satisfy the amplitude accuracy spec once an end-to-end calibration is done.

changing the sign of the loop gain, for example, with an exclusive-OR gate or a ± 1-gain amplifier—a pretty useful trick, especially in a loop with a small offset where the other method is difficult.

15.2.4 Getting the Gains Right

When we get down to an actual schematic with resistors, capacitors, and pin numbers, we'll need to make sure that the signal levels throughout the system are correct. In general, this will require a sprinkle of amplifiers and attenuators in various places. The radio ICs we're using have lots of gain in all their functional blocks, so we'll ignore the problem since we can put in the required attenuation very easily at the end of the design.

15.2.5 Error Budget

Once we have a block diagram, we can start apportioning the pain between the functional blocks. The way to do this depends on your application. If you're building a high-volume commercial product, start with the data sheets of the cheapest jellybean parts that look like they might do the job. Use their guaranteed worst-case specs as a starting point, and go to more expensive parts only when driven to it. This approach takes more engineering talent but reduces the per-unit cost (it's also the shortest road to wizardhood). In a research

TABLE 15.1. Bug Zapper Signal Levels[a]

	Channel 1					Channel 2			
Position	Floor (dBm)	Min Sig	Max Sig	S/N_{Min} (dB)	CC (mW^{-1})	Floor (dBm)	Min Sig (dBm)	Max Sig (dBm)	CC (mW^{-1})
Input	−92.2	−64.	+10	28.2	0	−92.2	−64.0	+10.0	0
Filter	−93.0	−64.8	+9.2	28.2	0	−93.0	−64.8	+9.2	0
30 dB Pad	−107.0	−94.8	−20.8	12.2	0				
LNA out						−75.0	−46.8	(−7.0)	2.0×10^3
Mixer in	−102.5	−94.8	−20.8	7.7	0.001	−75.0	−46.8	(−7.0)	
Mixer out	−87.5	−79.8	−5.6	7.7		−60.0	−31.8	(0.0)	1.6×10^3
Total compressibility coefficient					*0.001*			*3.6×10^3*	

[a]dBm unless noted; levels unreachable due to saturation are in parentheses.

lab, we mostly use what's in the drawer, or pick some fairly gold-plated parts that we're sure will work and which we can get in a hurry. Since no home should be without a bug zapper, Table 15.1 was calculated with jellybean parts in mind. (A real analog wizard could do most of this with discretes for less money.)

The draft design uses only six ICs, which isn't bad, and they're inexpensive FM radio chips. The BGA2011 LNA (low noise amplifier) has a noise figure of 1.6 dB, which means that the noise floor is degraded only a tiny amount. As usual, the mixer's dynamic range is the lowest in the string—it has a 4.5 dB noise figure and −13 dB input IP_3 (it's easier to make something linear than to make it be nonlinear in just the right way). We're keeping the LNA gain lowish to take maximum advantage of the mixer's dynamic range. Putting a noiseless 10 dB amp ahead of the mixer is as good as reducing the mixer noise by 10 dB, except that it reduces the system's input IP_3 by 10 dB as well—so it's easy to get too much of a good thing.

To avoid losing too much dynamic range on the high side, we cascade stages so that their third-order intercept points grow at least as fast as the accumulated gain; other things being equal, a 20 dB amplifier should have an output-referred IP_3 at least 20 dB above that of the previous stage. This is fairly permissive, because we can lose 3 dB of dynamic range per stage. We always lose something, but this rule at least avoids catastrophes. Low noise and especially high power amplifiers generally can't do this, but they're often needed for best system performance; as elsewhere in this book, rules are intended as a place to start, not as a substitute for thought.

Aside: Cost-Effective Error Budgets. Remember from Section 13.5.4 that the compressibility coefficients add, so scaling the IP_3s along with the stage gain means that each stage will contribute the same amount of compression—each will degrade linearity the same amount. This makes sense only if they cost the same. LNAs and power amps are the most expensive in general—we've paid for their performance, so we don't want to degrade it with cheap and cheesy intermediate stages (see Sections 1.7.1 and 18.8).

We assume a 6 MHz system noise bandwidth, a mixer with 4.5 dB noise figure, 15 dB (SSB) conversion gain and an output P_{1dB} of 0 dBm, and an LNA with 1.5 dB NF, 18 dB gain, and −14 dBm $P_{1\ dB}$. The right-hand column in Table 15.1, marked CC, is the compressibility coefficient (see Section 13.5.4), which allows us to calculate that the input 1 dB compression point of channel 2 is $1/(3.6 \times 10^3)$ mW or −35.4 dBm.

Channel 1 (with the pad) has a calculated $P_{1\ dB}$ of +30 dBm, and so is just beginning to compress at $P_{in} = +10$ dBm [from (13.21), $\delta = 0.23 \times (10\ mW)/(1000\ mW) = 0.23\%$] and maintains the required 51 dB SNR down to $P_{in} = -20.5$ dBm, which far exceeds the 0.2% of full scale requirement (at least from the noise point of view). Due to gain variations, it would probably be worth spending some of that extra thirty-odd decibels to increase the headroom, perhaps going to a 40 dB pad. Going too far will make the circuit more vulnerable to pickup. Channel 2 (with the LNA) maintains its required 10 dB SNR (for the ±1 dB standard deviation) down to $P_{in} = -83$ dBm, 19 dB better than spec. Incidentally, none of these ICs is more than $3.

The main thing in the detectors (the two SA605s at right) is to make sure that the RSSI outputs are not just detecting their own noise, and that there isn't a flat spot between the end of the linear range and the beginning of the logarithmic one. Making sure of that will probably require a few voltage dividers (or another two digitizer channels and a bit of software), but it's a straightforward exercise with a few pots, an ohmmeter, and an oscilloscope. In a moderately complex signal processor like this, where the conversion gains, intercept points, and other features (such as the output duty cycle and DC offset of a severely overdriven limiter) are not well specified, a good prototype is absolutely essential, so get five or so boards to hack up, besides however many you anticipate needing for the system prototype and spares. A ring-down calibrator (Section 15.7.2) will help by making the desired logarithmic characteristic into a straight line.

15.2.6 Judicious Gold-Plating

Slightly more expensive parts (e.g., Mini Circuits mixers and cascadable amplifiers) offer far better controlled gains and intercept points than these ICs. Diode mixers are also very *strong*, that is, have high intercept points, low noise, and consequently very high dynamic range; on the other hand, they need extra gain stages to provide high level LO injection and to overcome their conversion loss.

Here, where we have an online calibration, logarithmic detection, and enough dynamic range that there's room for some sloppiness, we can get away with cheap radio ICs. In a really long string of signal processing components, you'll probably want to use those better quality mixers and amplifiers rather than the IC versions, or failing that, put a bunch of DAC-controlled attenuators in the signal path and run the calibration several times with different settings to estimate what the gains and intercept points of the different stages are. That level of self-calibration probably isn't worthwhile unless you're building a lot of systems, and at that point it's worth seriously considering sampling the IF directly and doing the rest in software or in an FPGA.

15.2.7 Interface Design

We used a very simple problem description in Section 15.2.1 because there was just one designer. If more people are involved, you'll need to prespecify all the interfaces between subsystems: the protocols, data formats, timings, logic levels, power supplies, and so forth. Do this in exact detail and spill some sweat over it; more system integration nightmares come from mistakes here than you'd easily believe (Mars Climate Orbiter wasn't an isolated case). You will be amazed at the number of different ways a given spec can be interpreted too, if you don't keep talking with the designers of all the pieces you have to interface with. Contracts are very precise documents too, but lawyers spend a lot of time arguing about them even so. Neighborly cooperation is the way to go.

15.3 PERFECTION

It's often said that nothing is perfect, but components and circuits do some jobs effortlessly. These perfections come in two flavors: near-ideal component properties, such as the zero input current of MOSFETs, and conservation laws (e.g., charge conservation). Electronic designers are fond of saying "well, that may be true in *theory*, but. . . ." Don't fall for that one; there's gold to be mined out of things you know have to be perfect. Here's a short list.

Fundamental Laws

1. Things wired in parallel see exactly the same voltage, if the loops are small enough (Kirchhoff's voltage law, i.e., Faraday's law).
2. Two things in series see exactly the same current, except when there's significant radiation or capacitive coupling (Kirchhoff's current law, i.e., charge conservation: $\nabla \cdot \mathbf{J} = 0$).
3. A passive component with no voltage swing across it will draw no AC current (Ohm's law and the linearity of Maxwell's equations).
4. Devices with no dissipation exhibit no thermal noise (fluctuation–dissipation theorem).

Component Properties

5. MOSFETs draw no gate current whatever.
6. Metal film resistors contribute only Johnson noise (not shot or $1/f$).
7. A BJT's collector is a good current source; a cascoded BJT's collector is a near-perfect current source.
8. Feedback can improve the frequency response but not the SNR.
9. An RF filter can completely remove spurious products in a fixed-frequency AC measurement.

Let's look at a couple of related points in some detail.

1. *Flying capacitors can add and subtract perfectly.* Good maps have a scale of distance down in one corner, which is nice but not enough by itself, because the two points whose separation we care about never lie on the scale. Accordingly, we use chart dividers (like sharp-pointed calipers) to transfer a replica of the line segment down to the scale, and read it there.

Voltage measurements are the same way; usually neither terminal of the voltage source is exactly at the amplifier's idea of ground. Instrumentation amps transfer the voltage difference to ground with precision resistor strings and some op amp circuit hacks, but we can use the transfer-a-replica trick here too. A good quality capacitor functions the same way as the chart dividers: we charge it up across the voltage source, then disconnect it and reconnect it across the input of the amplifier. We do it with CMOS analog switches, and it really works; CMRs of 120 dB are typical, even with $\pm 10\%$ capacitors. There are special circuits (e.g., the LT1043) for this, but if you don't mind a constant offset voltage caused by charge injection, you can use ordinary switches too, as in Figure 15.3. It is

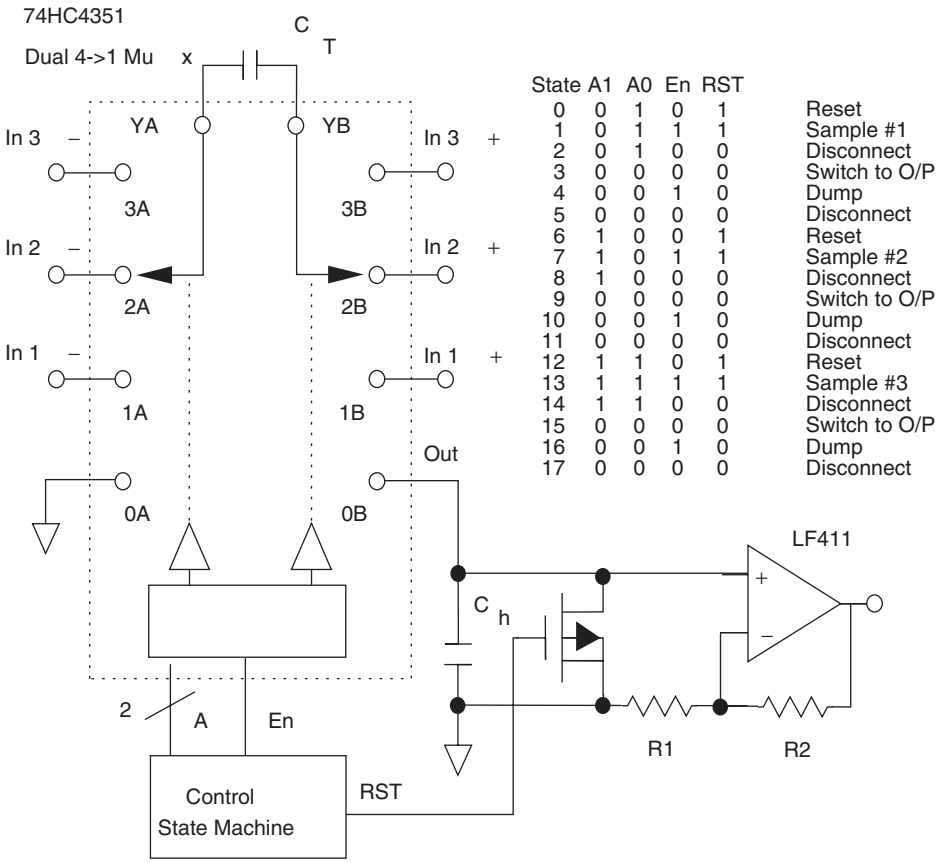

Figure 15.3. Three-way flying capacitor sampler made from a dual 4-into-1 analog multiplexer.

very important that the switches finish disconnecting before they reconnect to the other set of terminals—the *break-before-make* condition. Most inexpensive switches can't be relied on to do this, so we use a simple state machine such as a 4017 CMOS 1-of-10 counter and a few gates to do the sequencing, leaving the switches open-circuited for a whole clock cycle between disconnecting from one set and reconnecting to the other. In the state table in Figure 15.3, it may look tempting to eliminate disconnect states 2, 8, and 14, so that the state table can be implemented with 4 bits. You might be able to get away with it, but that depends on the inner workings of the 4351. The 4351 is not guaranteed to be break-before-make, so it may cause transient shorts between inputs, leading to hard-to-find errors. Building that sort of glitch into your circuit is a mistake, especially when an extra flip-flop will guarantee it doesn't happen.

2. *Independent noise sources are really uncorrelated.* There's another example in Section 17.11.6: the two-point correlation technique, where the power in a small AC signal is measured with two noisy voltmeters in parallel; their voltage noises are uncorrelated, so if the voltage measurements are $v_1(t)$ and $v_2(t)$, the signal power is $\langle v_1(t) \cdot v_2(t) \rangle / R$. Here's where the perfect component comes in: i_N is a real current that comes out on the input lead, so the current noises of the two voltmeters add, and you can't

separate that out by cross-correlating. Fortunately, we can easily get MOSFETs with femtoampere input currents and noises of a few hundred electrons/s/Hz$^{1/2}$, so we just use two MOSFET buffers in front of the voltmeters, and we're in business. (That kind of MOSFET doesn't have gate protection diodes, so we have to be careful not to blow them up.)

15.4 FEEDBACK LOOPS

Negative feedback is one of those concepts that is so deeply imbedded in all our technologies that it may come as a shock to realize that the feedback amplifier was invented as late as 1927.

There are earlier examples, for example, James Watt's flying ball governor for steam engine speed, or the tank valve of a flush toilet, but they are rare and isolated by comparison. The inventor, Harold S. Black of Bell Labs, had a great struggle to get it through the patent office. Why would anyone want to make an expensive amplifier with 40 dB more gain than it needed—two extra tubes in those days—and then just throw it away by closing the loop? Patentable inventions have to have utility, and that didn't seem so useful, it seems.

As Black probably explained to the patent examiner (in words of one syllable or less), the reason was that doing so gave an amplifier of unprecedented linearity and flat frequency response. That was good enough then, and it's good enough now. A cascade of N identical amplifiers whose gain ripple is m dB exhibits mN dB ripple overall. This was requiring the telephone network to use unwieldy cascades of stagger-tuned amplifiers, so that the peaks of one would coincide with its neighbor's valleys, reducing the ripple. Furthermore, the distortion due to nonlinearity accumulated regardless of such maneuvers. (Readers using optical amplifiers as fiber repeaters will recognize both problems.)

Given an amplifier like an LF356, whose gain varies from 100 dB near DC to 1 at 5 MHz, subject to big swings with temperature, supply voltage, and common-mode voltage, negative feedback using two resistors will produce a 30 dB amplifier flat from DC to 150 kHz; the essence is that when there's lots of excess gain, and certain stability criteria are met, a feedback system's behavior is governed entirely by the feedback network, which is made of extremely well-behaved components such as metal film resistors.

Our interest in this section is not so much learning how to build op amp circuits. You can read about the intuitive method in Horowitz and Hill and the rigorous method in Dostal. What we're aiming at here is learning how to make more complicated feedback systems that use op amp circuits as loop amplifiers and filters—things like temperature controllers, feedback laser stabilizers, and phase-locked loops. Accordingly, we won't spend a lot of time on what happens up near the op amp's unity gain crossover f_T, because we almost never build complicated loops that fast. There's more on pushing those speed limits in Chapter 18.

15.4.1 Feedback Amplifier Theory and Frequency Compensation

A feedback network is shown schematically in Figure 15.4. For now, we'll assume that the components are linear and time invariant, and the amplifier's output impedance is zero and its input impedance infinite; as usual, we can fix things up afterwards when

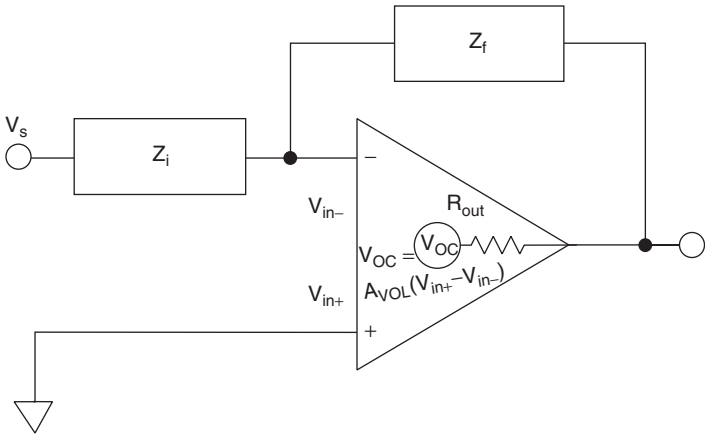

Figure 15.4. A feedback amplifier.

they aren't, quite. The amplifier has a frequency-dependent gain $A_{\mathrm{VOL}}(f)$, and the input and feedback impedances are Z_i and Z_f. Since this gain is always spoken of in the frequency domain, it's usually called the transfer function, which is what it really is. All amplifiers are differential, but some are better than others. This one produces an output voltage

$$V_o(f) = A_{\mathrm{VOL}}(f)(V_{\mathrm{IN}+} - V_{\mathrm{IN}-}). \tag{15.2}$$

The feedback network forces V_i to obey

$$V_i = V_s \frac{Z_f}{Z_i + Z_f} + V_o \frac{Z_i}{Z_i + Z_f}. \tag{15.3}$$

Requiring that both of these hold together, we get the closed-loop gain A_{VCL} of the whole circuit,

$$A_{\mathrm{VCL}-}(f) \equiv \frac{V_o}{V_s} = \frac{-Z_f}{Z_i}\left(\frac{\beta A_{\mathrm{VOL}}}{\beta A_{\mathrm{VOL}} + 1}\right), \tag{15.4}$$

where the *feedback fraction* β is the gain from the output back to the inverting input,

$$\beta \equiv \frac{\partial V_{\mathrm{in}-}}{\partial V_o} = \frac{Z_i}{Z_i + Z_f}. \tag{15.5}$$

If we use the noninverting pin as the input, we get the noninverting closed-loop gain,

$$A_{\mathrm{VCL}+}(f) = \left(1 + \frac{Z_f}{Z_i}\right)\left(\frac{\beta A_{\mathrm{VOL}}}{\beta A_{\mathrm{VOL}} + 1}\right). \tag{15.6}$$

Since the amplifier is linear, and there's nothing special about ground, the total output voltage is the sum of the two contributions,

$$V_o = A_{\mathrm{VCL}+} V_+ + A_{\mathrm{VCL}-} V_-. \tag{15.7}$$

If $|\beta A_{\mathrm{VOL}}| \gg 1$, these simplify to

$$A_{\mathrm{VCL}-} \approx -\frac{Z_f}{Z_i}, \quad |\beta A_{\mathrm{VOL}}| \gg 1, \tag{15.8}$$

$$A_{\mathrm{VCL}+} \approx 1 + \frac{Z_f}{Z_i}, \quad |\beta A_{\mathrm{VOL}}| \gg 1, \tag{15.9}$$

which is what we mean by saying that the feedback network takes over control of the amplifier's performance. Departures from this ideal condition are expressed by the error factor, the last term in parentheses in (15.4) and (15.6), and are clearly controlled by the *loop gain* $A_{\mathrm{VL}} = \beta A_{\mathrm{VOL}}$. There's nothing special about voltage dividers—transformers work just as well at AC, and there are more complicated networks too. They're cheap and convenient, and they stabilize the DC bias as well as the gain, but it's β that matters.

The amplifier A_1 doesn't have to be an op amp, of course. In a complicated system such as an AGC loop, where the gain of a receiver's IF amplifier is controlled to keep the DC voltage from the detector at a constant value, A_1 is the IF amplifier with its input signal, the IF filters, detector, and AGC error amplifier; any and all of these may have their own local feedback loops, too. The open-loop gain is then the product of all the gains of the string. Its frequency dependence of A_{VOL} is a bit complicated in that case, but finding it is a straightforward matter of calculation or measurement, after which these equations apply.

15.4.2 Loop Gain

The response of a feedback loop to an external disturbance, for example, somebody sneaking in and putting a battery in series with its output (inside the loop), is also governed by the loop gain, which is the amount of gain thrown away when we close the loop. If we put a signal source in series with the amplifier output, we find that its effect is reduced by a factor of A_{VL}. Since we assume that the amplifier is stable when the loop is open, and $\beta \neq 0$, there is no opportunity for instability unless $A_{\mathrm{VL}} \approx 1$, when the denominator can go through 0.

The open-loop gain has to roll off somewhere below the frequency of daylight, however, so the high gain limit is not the only interesting place. We saw in Chapter 13 that a stable network has no poles in the lower half f plane, so the denominator of (15.4) has to be nonzero there.

The simplest way to be sure we satisfy the stability condition is to ensure that $A_{\mathrm{VOL}}\beta$ has less than 180° phase delay at the frequency where $|\beta A_{\mathrm{VOL}}| = 1$. The difference between the actual phase delay and 180° is called the *phase margin* and shouldn't be allowed to get below 45° in most cases, because gain peaking, overshoot, and ringing get much worse for phase margins less than that. We do this by constructing a *Bode plot* like the one in Figure 15.5, which is a plot of the asymptotes to the open-loop transfer functions and (especially) A_{VL} on a log–log plot with phase on the other Y axis. In Chapter 18, we'll come back to this point in more detail.

15.4.3 Adding Poles and Zeros

If you're making a fast amplifier, then all you usually need are resistors R_i and R_f, and perhaps a small capacitor C_f across R_f to cancel the effects of the input capacitance of the op amp. The manufacturer's data sheet will tell you what you need to know.

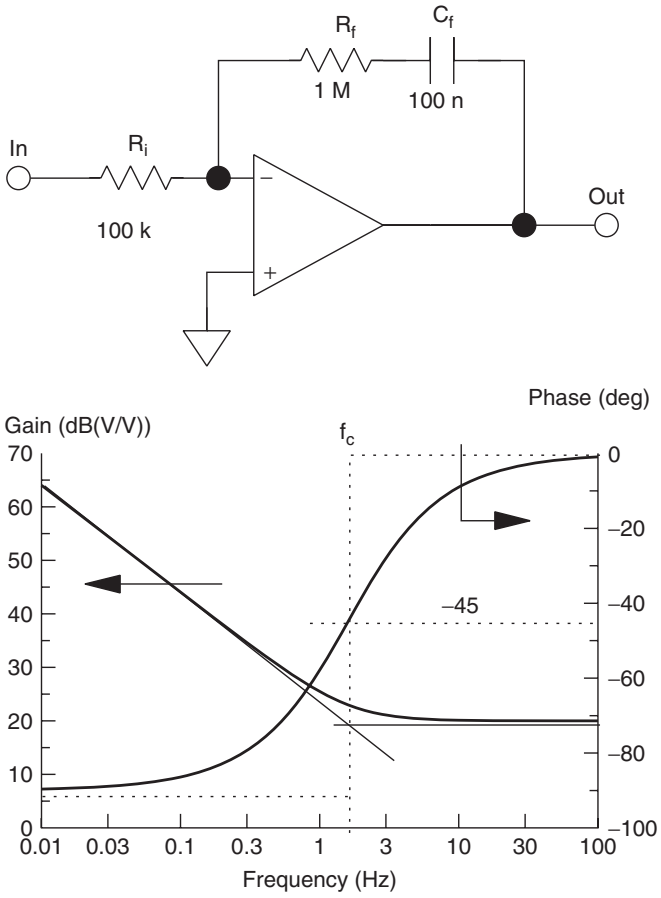

Figure 15.5. Op amp lead–lag network and its transfer function. Due to C_F, the amplifier's bias is unstable, so this circuit fragment has to be wrapped in another feedback loop.

Right now, as very often, we're building some much more complicated thing such as a motor control loop. The loop bandwidth is then usually far below f_T of the op amp, so the high gain equations (15.8) and (15.9) apply.

At that point, it becomes duck soup to put poles and zeros just where you like, because the zeros of the transfer function are at poles of Z_i and zeros of Z_f, while its poles are at poles of Z_f and zeros of Z_i. We chuck in a few series or parallel RC networks, and we can make our loop gain do nearly anything we want.

Example 15.1: Lead–Lag Network. The op amp circuit of Figure 15.5 has a *lead–lag* characteristic. At low frequency, series combination $R_f + C_f$ looks capacitive, so the circuit looks like an integrator. At f_c the corner frequency of R_f and C_f, it starts looking like a flat 20 dB amplifier with resistive feedback. When used as part of a complicated loop such as a PLL, this allows a two-pole (-40 dB/decade, 180° phase lag) rolloff where the loop gain (of the big loop) remains high, changing to a one-pole rolloff (-20 dB/decade, 90° phase lag) before the loop gain gets to 0 dB, to preserve a good phase margin. This allows us to use a much higher low frequency gain, for much

lower static errors, at the price of a longer settling time. We'll use a similar trick in Section 18.4.1 to get more bandwidth from a transimpedance amp.

Aside: Nonlinear Instability. High order feedback networks can be stable with $|A_{VL}| > 1$ and huge phase shifts near DC (270°). The problem is that when a nonlinearity occurs (e.g., on signal peaks or on power-up), the nonlinearity will reduce $|A_{VL}|$ and the stupid thing will oscillate. Good overload recovery circuitry is a vital necessity there.

15.4.4 Integrating Loops

Some circuit elements have an integrating response already built into them; the canonical example is a PLL's VCO and phase detector, where the control voltage sets the frequency, which is $d\phi/dt$, but the detected output is ϕ. From the formula (1.40) for the Fourier transform of an integral, the feedback fraction β is then proportional to $1/f$, instead of being a constant ratio as we're used to from op amps.

A very tightly coupled heater and thermal mass are a close approximation under some circumstances. A given heater power $\dot{Q}$ will eventually raise the sensor temperature by $\dot{Q}R_{TH}$ degrees above ambient (where R_{TH} is the thermal resistance), so the response is constant at large t (near DC). The heat takes a while to diffuse from the heating element to the sensor, and this thermal diffusion dominates the response at short times (i.e., high frequencies). If the sensor is intelligently placed, there's a region in the middle where the transfer function goes as $\dot{Q}/(M_{TH}f)$, where M_{TH} is the thermal mass. Another example is a current-sensing motor speed controller, where the motor current controls the torque Γ, which controls the angular acceleration $\dot{\Omega} = \Gamma/I$, where I is the moment of inertia.

In that sort of region, increasing the loop gain moves the loop corner frequency f_c up, and so increases the speed of the loop without greatly affecting the shape of its transient response. You can play this game up to the point where the excess phase shift due to other poles, to thermal diffusion, or to losses in the motor becomes important.

Integrating loops such as PLLs are intrinsically more difficult to compensate, because your loop filter has only 45° to work with; you really have to calculate these ones carefully, because it's virtually impossible to get a good result otherwise. When you think you've finished, apply Pease's Principle: Bang On It. Do it electrically, by looking at the turn-on transient under all sorts of load conditions, from no load to above maximum; apply an adjustable-frequency square wave to the control voltages, and look for overshoot or odd peaks and nulls in the response curve. Put a brake on the shaft, cold spray on the temperature sensor.

If the loop gain is nonlinear, as it is in heaters and usually in PLLs, test it over the entire range of gains, and put in a safety factor for unit-to-unit variations.

15.4.5 Settling and Windup

Besides loop stability, we usually care about the transient response. Depending on the application, a step response that overshoots may be harmless; it may cause measurement errors, as in a front end amplifier; or it may cause destruction of the whole system, as in a heater or a motor controller. Systems with slow transducers, such as servomotors, are prone to *windup*. At power-up, or when a command to move arrives, there is a large error signal. The control loop starts ramping the motor speed, which increases until the

position encoder reaches the set point. Unfortunately, due to the motor's inertia, it goes flying past the set point and has to be brought back again—at which point it winds up again, and has to be brought back again, and so on. It's basically a disease of slow transducers with integrating control loops. There are two main ways of dealing with this: adding some speed feedback (i.e., a derivative term) into the error signal, and using some nonlinear element for windup control, such as an analog switch that freezes the integrator contribution when the error signal is large.

Good windup control is important for another reason: the long-time settling behavior of any amplifier is dominated by its lowest frequency open-loop pole or zero, so to get good settling, we don't want the the open-loop transfer function to have corners. Integral-only loops settle to high accuracy faster than if there's a lead–lag or derivative network. That's where a well-designed nonlinear network can be a big help—it controls the windup during slew, but goes away during settling. Tim Westcott's book on control systems[†] has lots of good advice on windup control among many other things.

15.4.6 Speedup Tricks

After seeing how long it can take a feedback loop to recover from saturation when there's a slow integrating time constant in it, the temptation to limit the range of the integrating part is nearly irresistible—we inevitably try shoving in some back to back diodes or perhaps a cleverly connected FET across the capacitor, across the resistors, wherever they look nice. Done right, anti-windup and speedup tricks can help a lot, but they won't do anything good unless you're using some simulations of how the loop will behave. One exception is the so-called baby-scale loop, a slow one-pole, unity gain noninverting lowpass filter, with high and low value input resistors controlled by an analog switch and a window comparator; whenever $0.95V_{\text{in}} < V_{\text{out}} < 1.05V_{\text{in}}$, the high value resistor is in, for wriggle rejection, and otherwise the low value one is used, to help the slew rate, and keep the babies from getting cold waiting. This is pretty problem-free in first-order loops, but not so easy otherwise.

15.4.7 Output Loading

There are two major effects of nonzero output impedance in the op amp: feedthrough and sensitivity to output loading. Feedthrough is signal current going through the feedback network to the output and causing a voltage drop in the output resistance that appears as output voltage. A rail-to-rail op amp with a 1000 ΩR_O, and a feedback network made up of a 1k R_i and 10k R_f will have a small-signal gain of 1000/12,000 (−21.5 dB) even with the output railed, due to finite output impedance alone. This happens often in op amp based analog switching applications, so it isn't an imaginary problem. If the output is connected to some noisy place like a motor, junk can arrive at the input from the output, which is sometimes pretty serious, especially if it can get into other circuit nodes from there.

A finite R_O makes the insertion phase of the amplifier depend on the reactance of the load; a capacitor on the output puts a pole in the open-loop transfer function at $f = 1/(2\pi R_O C_L)$, which adds phase shift that can destabilize the amplifier. The classical way to fix this is to isolate the amp from the capacitance with a series resistor, with

[†]Tim Westcott, *Applied Control Theory for Embedded Systems*. Newnes, 2006.

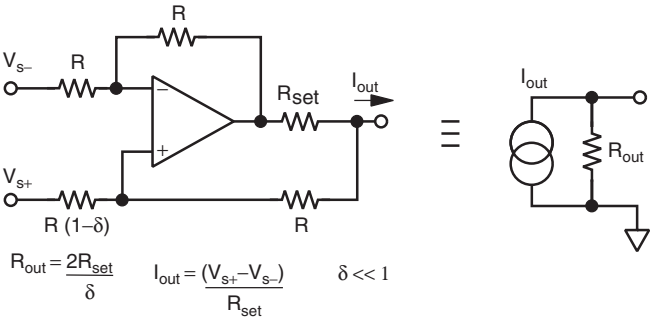

Figure 15.6. Modified current source for low impedance loads.

feedback taken directly from the op amp output. A slightly modified version of this uses an *RC* network to take low frequency feedback from the load and high frequency feedback from the amplifier output, which will improve the accuracy for low impedance resistive loads, at the price of significantly poorer settling time.

For cable capacitance, we can use a beefy op amp or buffer, and terminate the cable in its characteristic impedance—either series terminated at the buffer's output or (if you can stand the power dissipation) shunt terminated at the cable's output. We can even be model citizens and do both, but then must be prepared for a 6 dB loss as well as high dissipation.

Another strategy, in cases where some loss of speed is acceptable, is to use an LM7332 or equivalent op amp that can drive lots of capacitance. A slightly unbalanced op amp current source (see Figure 15.6) can be made to have a 50 Ω output impedance without needing a 50 Ω series resistor, which helps with the signal loss. Watch the high frequency behavior of this circuit, though, and don't make the series resistor *too* small.

As we saw in Chapter 14, rail-to-rail output op amps have very high open-loop output impedances, and so are far more susceptible to instability due to output loading.

Aside: Oscillating Integrators. Given the 90° phase lead from the feedback network in an integrator, it takes real talent to make one oscillate, but it's possible. Use an op amp with a high output resistance, a low R_i, and a huge C_f, and you can make an integrator that rings like a bell.

15.5 SIGNAL DETECTORS

We spent a fair amount of time on the basic principles of signal detection in Chapter 13, and one thing we concluded was that detectors are much of a muchness at large SNR, but differ substantially at low SNR, which is what limits our measurement performance. Here we'll delve a bit deeper into how to make good ones.

15.5.1 AM Detection

We've already encountered the two basic types of AM detector, namely, rectifying and synchronous, in Section 13.9.2.

The simplest AM detector is a half-wave rectifier made out of a diode with an *RC* lowpass on its output, but as we saw there, it doesn't work too well, because diodes

are not great voltage switches. For small signals, the output voltage is roughly proportional to the input power (the so-called square-law region), but then as the RF amplitude rises, it gradually changes to being proportional to the RF voltage, as you'd expect from a rectifier. The level at which this happens depends on loading, temperature, and frequency. As a linear detector, the half-wave rectifier has about a 20 dB dynamic range. This may be adequate for some purposes, for example, AGC detectors, where a servo loop controls the IF amplifier gain to keep the detected voltage constant.

Backing up to what we know about diodes, they are nearly perfect current switches. It thus might occur to us to drive the diode detector with a current instead, as in Figure 15.7; any current imbalance will cause the collector of Q_1 to swing as far as it has to, to send that current through one of the detector diodes. This looks like a complete solution, but of course it isn't—the slewing becomes slow at small signals, and the collector impedance isn't infinite. Still, it's good for a 40 dB dynamic range up to 50 MHz or so, and maybe 60 dB at 10 MHz, which is a great improvement.

15.5.2 Emitter Detector

Figure 15.8 shows an emitter detector, which is nothing more than an emitter follower with a slow *RC* in series with the emitter. It functions as a half-wave detector that doesn't load the circuit the way a diode does. This detector is mainly useful when two or more of them are strung together, to make full-wave emitter detectors and DLVAs. For small signals, where the transistor stays conducting, its rise and fall times are $r_E C$, while for a big negative excursion, the transistor cuts off and leaves only the slower *RC* falloff.

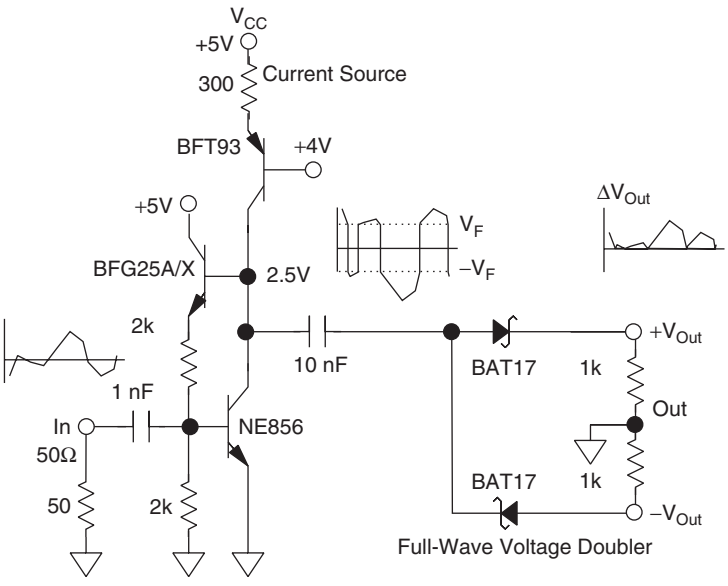

Figure 15.7. Linear rectifying detector. The common emitter amplifier and current source load function as a bipolar current source, so that its output voltage skips across the diodes' dead zones to source current to the load resistors.

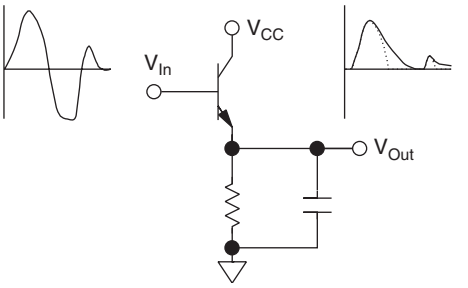

Figure 15.8. The emitter detector works like a combination of a half-wave diode and an emitter follower; the rise time and the small-signal fall time are set by r_E and the large-signal fall time by R. (Watch out for oscillations if the source impedance is very low.)

15.5.3 Synchronous Detectors

A better approach is the synchronous detector, where you put the signal into a balanced mixer whose LO has the same frequency and phase as the signal. If you're providing the carrier signal, this is how you should do it almost every time (it's how we did it in the bug zapper example). If all you have is the input signal, you can produce the LO by shoving the signal into a limiter or a comparator. This works pretty well at all signal-to-noise ratios, providing the signal is always large enough for good limiting. At low SNR you wind up detecting the noise as well as the signal, because of course the circuit cannot distinguish the two. AM–PM conversion can be controlled by slight adjustment of the phase, so that the slight slowing of the limiter for large signals doesn't move the phase too far from 0°.

Limiters with AC coupling between stages work better than barefoot comparator ICs, because DC offsets and possible comparator oscillations become less troublesome. Comparator ICs usually need some sort of hysteresis to prevent oscillation at low signal levels, and this will seriously degrade the small-signal operation of the detector.

15.5.4 High Performance Envelope Detection

We encountered vector (I and Q) modulation in Section 13.8.7 as a way to convert a signal to baseband without folding the sidebands. In detection problems, it allows us to use a fixed oscillator to drive the phase-sensitive detector instead of having to track the phase of the incoming signal. The problem is then to convert the resulting quadrature pair into amplitude and possibly phase too. One technique is to digitize both and use a lookup table, which is a good way to proceed if you need phase information, but it isn't the only possibility.

In radar, they use a sleazy trick to get the envelope magnitude from I and Q, as shown in Figure 15.9: take absolute values, then pick the largest out of $|I|$, $|Q|$, $(|I| + |Q|)/\sqrt{2}$. This is equivalent to using eight phases, spaced evenly at 45°; the worst-case scallop loss is only 0.9 dB. Using more phases works even better (you win quadratically), at the cost of one op amp per time: using $|I|$, $|Q|$, $|I|\sin(\pi/6) + |Q|\cos(\pi/6)$ and the same for $\pi/3$ (leaving out the $\pi/4$ one) gives a scallop loss of only 0.3 dB.

Using absolute values saves parts, because it confines us to the first quadrant. Those rectifying amplifiers limit our speed, though. By omitting them, we can go much faster,

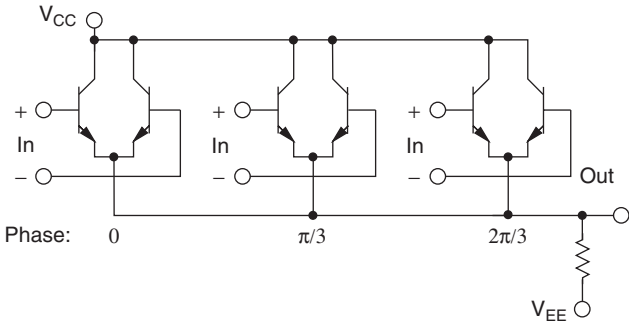

Figure 15.9. High performance envelope detection. Here three phases are shown; the six transistors correspond to positive and negative half-cycles of all three.

tens to hundreds of megahertz. We use resistor networks and an inverting amp or two to generate the phases independently all round the circle: I, $-I$, Q, $-Q$, $(I + Q)/\sqrt{2}$, $(I - Q)/\sqrt{2}$, and so on, and put an emitter follower on each one. The phase selection is done by wiring all the emitters together, wire-OR fashion, to a single current source load. Although this scheme shares the small-signal nonlinearity of the emitter detector (see Section 15.5.2), this actually helps a bit at larger signals, since it occurs only when two phases are producing nearly the same output, and it will thus help to fill in the scallops.

15.5.5 Pulse Detection

The classical peak detector is an op amp with a diode in its feedback loop and an *RC* lowpass network on its output, with feedback taken from the hold capacitor (Figure 15.10 shows an improved version). It's a bit like a track/hold, except not as good. Peak detectors are slow, inaccurate, or both, due to overshoot and slew limiting of the amplifier used. As shown in Figure 15.11, a peak detector that overshoots is frozen forever at the peak of the overshoot waveform, which depends on time, temperature, frequency compensation, and the fine details of the pulse shape. You can do a decent job on a pulse whose top is smooth and whose rise time is at least 1 μs, provided that the range of pulse heights does not exceed 30 dB.

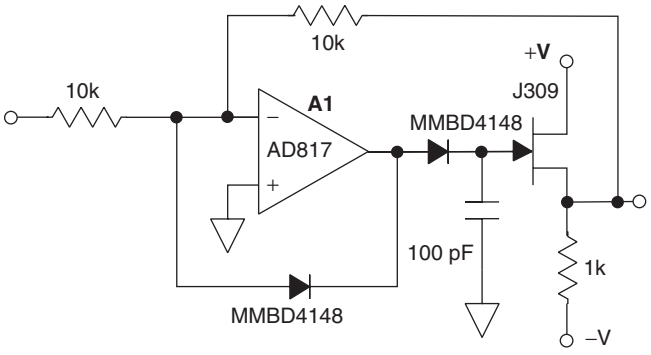

Figure 15.10. Peak detector.

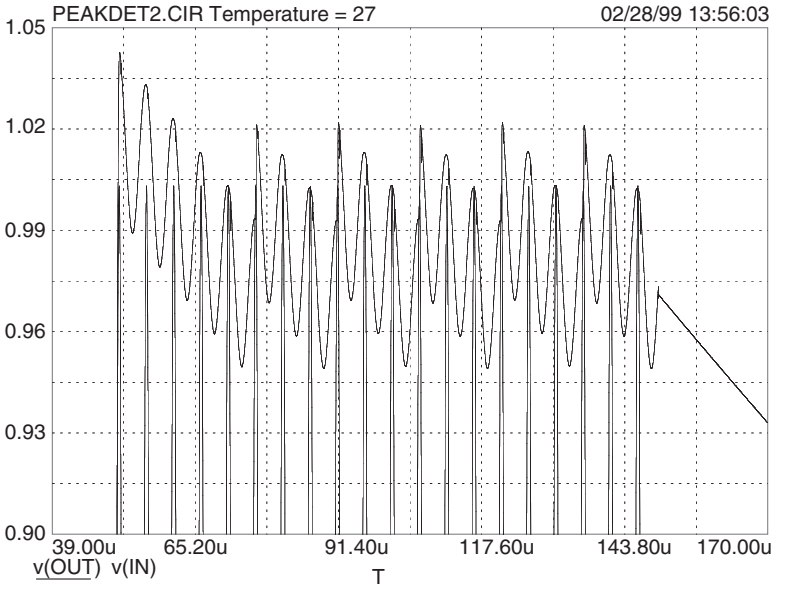

Figure 15.11. Peak detector pathology.

Aside: Constant Fraction Triggering. Getting a low jitter trigger from pulses with variable amplitude is a pretty generic problem, especially with equivalent-time (stroboscopic) sampling. Unless you already have a really low jitter timing reference, you have to design around the problem.

A common error is to set the trigger voltage somewhere on the leading edge and hope for the best. The reason this doesn't work is that the leading edge isn't infinitely sharp, so if your amplitude uncertainty is $\langle dA \rangle$, the resulting timing jitter is

$$\langle dt \rangle \approx \langle dA \rangle / (dV/dt). \tag{15.10}$$

If the pulse is round-looking at all, this is pretty bad. The trick is shown in Figure 15.12: without slowing down the edges too much, modify the pulse so that it has positive and negative lobes, and trigger on the zero crossing in the middle, which is independent of pulse height. We aren't quite done, because a trigger set at 0 V will keep triggering on noise. If you're using a scope with fancy triggering, you can tell it to look for a threshold crossing followed by a falling edge. If you're building it into an instrument, you can combine the zero crossing with an ordinary threshold signal, as in Figure 15.12b, which will reject the noise. The combination will give a stable, low jitter trigger signal more or less independent of amplitude variations. (You can also apply the threshold signal to the trigger comparator in analog, using different delays; also, there's nothing sacred about the coax stub—all you need is an impulse response with one positive and one negative lobe and fast edges.)

15.5.6 Gated Integrators

Most of the time, the shape of the pulse does not change much with its amplitude, so that the area under the pulse is a reasonable surrogate for the pulse height. Gated integrators

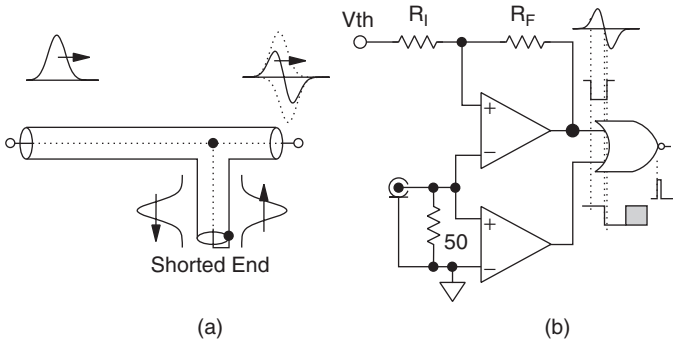

Figure 15.12. A constant-fraction trigger eliminates trigger jitter caused by pulse height variations: (a) a shorted coaxial stub produces an inverted copy of the input pulse, and the sum of the two produces a zero crossing somewhere near the peak of the input pulse; (b) two comparators and a NOR gate produce a stable trigger edge. Hysteresis provided by R_I and R_F ensures that the upper comparator remains active long enough to generate a good trigger pulse but not long enough to let through random junk from the lower one when the input has returned to 0.

are much easier to make than peak detectors. Because the integrator slews much more slowly than the peak detector, and can recover from overshoot rather than being frozen, gated integrators make much better pulse height detectors than peak detectors do. You do need a trigger, though, which as we just saw can be a nuisance. An excellent alternative is the *ping-pong gated integrator*: use two gated integrators running continuously at about a 75% duty cycle, but 180° out of phase (so that half the time, both integrators are active at once), which ought to guarantee that any signal pulse that shows up will be completely inside one of them. You can pick which output samples you want more or less at your leisure. Ping-ponging is useful in all sorts of places.

15.5.7 Peak Track/Hold

If you really must detect peak height, use a peak track/hold (T/H) instead of one of those ugly things with the diodes in its feedback loop. A peak track/hold is a track/hold whose sampling signal is controlled by a comparator, as in Figure 15.13. The comparator compares the input and output of the T/H, and as soon as the output drops below the input, it switches to hold mode. Ideally the comparator should be faster than the T/H, or sample the signal somewhere slightly upstream, so that errors caused by the signal delay are reduced. It's still hard to do better than 1% with this arrangement, and it will give the wrong answers systematically for pulses with long tails (slower than the T/H droop).

15.5.8 Perfect Rectifiers

The exponential dependence of $I(V)$ for a forward-biased diode leads to serious nonlinearity in simple rectifier–diode detectors. Because the conductivity of the diode depends on the drive level, low load impedances make them even worse. This effect is worse than you would expect, because the conduction occurs on the signal peaks, and this small duty cycle magnifies the effect. The spec for this is the video resistance, which is usually around 1–10 kΩ.

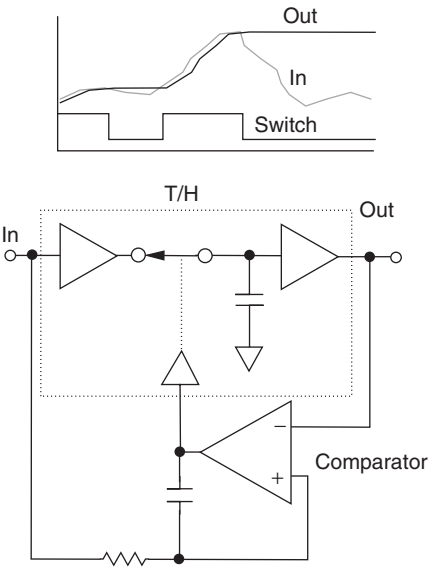

Figure 15.13. Peak track/hold circuit.

One approach to fixing it is to put it in a feedback loop, just as we did with the peak detector. This can be done well, or it can be done fast, but not both—the perfect rectifier problem has a lot in common with peak detection, since after all an AC waveform is just a string of closely spaced pulses. The fast approach is to use an emitter follower stage instead of the diode; several can be wire-ORed together to make a maximum detector, as we saw in Section 15.5.4. This is not a bad deal but does tend to have glitches caused by the finite slew rate; the output has to slew very rapidly across the forward voltages of the two diodes (or diode and transistor). They tend to be worse at small signal levels, where there is less overdrive available to make the output slew rapidly. Thus for signals of any speed, we're back to a quadratic problem.

You can also use the emitter follower with the op amp output connected to its base and the inverting input connected to a voltage divider between base and emitter. This is a diodeless version of the perfect rectifier plus catch diode trick. This works pretty well, a lot better than the catch diode version, but is also nonlinear at small signals because the output impedance of the emitter follower changes with its load current, so that the switchover between the two transistors is gradual at the level of 20 millivolts or so.

The closest approach to a *perfect* perfect rectifier is a switching detector (e.g., a Gilbert cell) driven by a pure current, and switched by a really high gain, fast limiter, whose gain is high enough to switch cleanly on the noise of the input in the absence of signal.

15.5.9 Logarithmic Detectors

Optical measurements can reach shot noise limited sensitivities on the order of 1 part in 10^8 intensity change in 1 second, so their dynamic range is often enormous. That 160 dB dynamic range cannot be accommodated by any A/D converter whatever; unless we can afford to throw away dynamic range, we have to use range switching or logarithmic detection schemes to fit it all into our ADC.

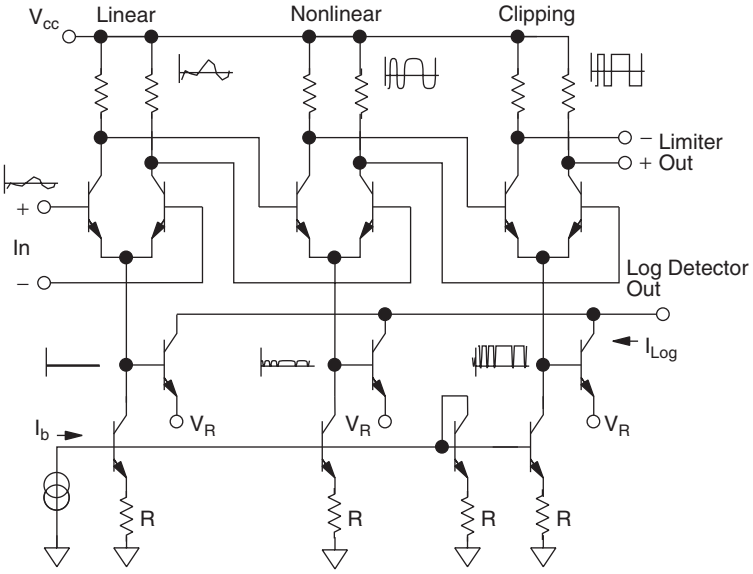

Figure 15.14. Successive detection logarithmic AM detector.

Logarithmic detection has several advantages: a level change of x dB translates into the same number of ADU anywhere in the dynamic range, the sensitivity of subsequent stages to interference and noise is greatly reduced, and the ease of achieving an accurate logarithmic characteristic makes them stable and predictable. Furthermore, we don't lose any data by being caught on the wrong range setting. The trade-offs are lower resolution at higher signal levels and moderate errors in the logarithmic characteristic (0.5–1 dB usually).

The classical method of logarithmic signal detection is *successive detection*, as shown in Figure 15.14. In this clever scheme, N amplifiers of equal gain A are cascaded, each with some sort of AM detector, whose outputs are summed. Here it's differential pairs, doubling as amplifiers and full-wave emitter detectors; as you can see, the DC voltage at the emitters of each stage goes up with the signal, until the *previous* stage starts clipping.

As the input signal level increases from 0, the last detector produces an output proportional to $A^N v_{\text{in}}$, until the last stage clips. When that happens, the last detector's output remains constant, and the output slope is proportional to $A^{N-1} v_{\text{in}}$. Every time the input level goes up by a factor A, the gain drops by a factor A, so that

$$\frac{\partial V_{\text{out}}}{\partial V_{\text{in}}} \propto \frac{1}{V_{\text{in}}}, \tag{15.11}$$

whose solution is proportional to $\log(V_{\text{in}})$. Purpose-built successive detection circuits are often known as detector-log video amplifiers, or DLVAs. These can be very fast and do not suffer from the linearity and speed limitations of diode detectors at low signal levels. There are also logarithmic amplifiers, made from bipolar transistor pairs, but these tend to be slow and have serious parametric effects such as signal-level-dependent slew rates and bandwidth.

You can get dedicated DLVA parts, such as the MAX4003 (which has a 45 dB range), or use FM IF ICs for consumer radio receivers as we did in the bug zapper. These use DLVAs for the signal level meter output (called RSSI, for received signal strength indicator). The meter outputs are often very useful in instruments, since they typically cover an 80 dB range with ±1 dB accuracy, and the ICs cost only $2 or so. There aren't as many of these as there used to be, but NXP still has several. "High speed" parts have op amps on the RSSI output, but the older ones just bring out the detected current. You can actually get quicker response from the "slow" current-output RSSIs, as fast as 40 ns if they're loaded with a low enough impedance (see Section 18.4.4). Besides that ±1 dB error, these devices are normally not temperature compensated too well, so calibration and compensation will be required. We saw in Section 14.6.6 that BJT saturation is slow and badly behaved, so if you're designing your own, make sure that clipping occurs due to cutoff instead.

15.5.10 Phase-Sensitive Detectors

Diode bridge mixers are the phase-sensitive detectors of choice at high frequencies, from a few megahertz up to 30 GHz or so, because they work extremely well for such simple devices. The output swing is about ±300 mV for the garden variety to ±1 V for dedicated ones like the Mini Circuits RPD-1. Their amplitude fidelity is pretty good; until the signal is lost in the DC offset of the mixer, or starts to saturate it, a diode bridge phase detector will provide good linearity in amplitude for a fixed phase anywhere in its range. It works best at $\theta = 0$ or π.

Its phase accuracy is mixed. On one hand, the AM–PM performance of a phase detector is usually excellent near null; if θ is near $\pm\pi/2$, turning the signal amplitude up and down causes only very small changes in the position of the null until the RF level becomes comparable to the LO and this starts controlling the DC offset. On the other hand, though, the detailed shape of the function $V_{IF}(\theta)$ for constant A_{RF} is not all that accurately sinusoidal, and its exact shape depends fairly sensitively on the RF signal amplitude. Another way of looking at this is that all those spurious responses we spent so much time avoiding in Section 13.7.2 land right on top of us when the IF is at DC. Though their frequency is 0, their phases still go as multiples of $\Delta\phi$.

Like mixers, phase detectors are normally used with lots of LO drive to reduce their sensitivity to LO amplitude variations. For a pure phase application, such as FM detection, phase detectors work best with both inputs driven hard.

Gilbert cell multipliers (see Figure 14.16) have much the same characteristics as diode bridges. They are not as resistant to overdriving the RF input, are noisier, and require more ancillary circuitry. On the other hand, they exhibit gain, and if the LO is driven gently, they can have very low spurious products, owing to the accurately bilinear multiplication possible with bipolar differential pairs. Even driving the LO hard requires much less LO power than a diode bridge. Because of their complexity, they are less often used in discrete circuitry than diode bridges, but are a nearly universal choice in bipolar ICs.

Below 1 MHz or so, the CMOS transmission gate approach is best unless the dynamic range is very large. The main problems with these are duty cycle errors, which cause the nulls to be in the wrong places; switching glitches; and charge injection in the switches, which causes a level-dependent DC offset. The large-signal performance of CMOS switches is excellent, so the gradual compression of the output level that afflicts diode mixers is not present.

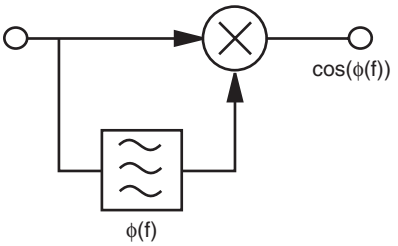

Figure 15.15. A discriminator is a frequency-dependent phase shift and a phase-sensitive detector.

15.5.11 FM Detectors

Good FM detectors work by taking two copies of the input signal, applying a frequency-dependent phase shift to one of them, and then running them into a phase-sensitive detector (Figure 15.15). Typical ways of getting the phase shift are delay lines (often just a chunk of coax) or resonators (e.g., *LC* tank circuits, quartz crystals, or SAW devices). The phase shift should be as linear as possible, and the total phase shift over the band of interest should be small enough to stay within the linear range of the phase detector (sometimes the nonlinearity of one can compensate that of the other, as in double-tuned quadrature detectors in FM radios). Linearity aside, there is an obvious trade-off between sensitivity and bandwidth for a given output voltage swing and noise level.

The other approach is to turn the FM into AM by sitting on the skirt of a filter, and detect the result with an AM detector. We're often stuck with that sort of thing in optics (e.g., the Pound–Drever technique for locking a diode laser to an etalon), but there's no reason for it in electronics.

15.5.12 Delay Discriminator

The simplest way of getting the phase shift is to use a chunk of coax or a serpentine microstrip trace, some odd multiple of $\lambda/4$ long. In that case, the phase is very linear with frequency, and the resulting signal is approximately

$$V_\phi \approx A \cos\left(\frac{2\pi f L}{v}\right). \tag{15.12}$$

Delay discriminators are very wideband, which is their best feature; if we can accept 5% nonlinearity at the phase extremes, a nominally $\lambda/4$ delay line works over a range of ± 0.55 radian, a bandwidth of more than an octave. The downside is that this makes them pretty insensitive. Delay discriminators are really useful in the lab, because you can wire one up with a Mini Circuits mixer, hybrid splitter, and two patch cords of different length, as in Figure 15.16.

An unequal-path interferometer is a delay discriminator too. If there are exactly two paths, and no spatial phase variations, (15.12) still holds; if as usual there are more, we have to put the right-hand side inside a double sum over all the pairs of paths. This isn't all that difficult, although for continuous distributions and multiple scatter (e.g., double Rayleigh scattering in fibers), it does become a bit subtle analytically. Usually, though, we have one path that is much stronger than the others, and the predominant effect is

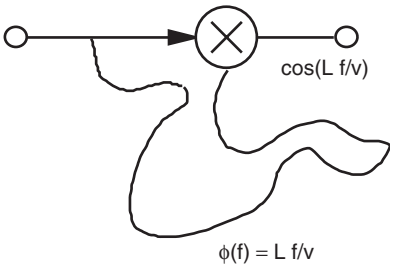

Figure 15.16. Delay-line discriminator.

interference of stray paths with that one. In that instance, a single sum is enough, and (15.12) becomes

$$V_\phi \approx \sum_{i=1}^{N} A_0 A_i \cos\left(\frac{2\pi f L_i}{v}\right). \tag{15.13}$$

15.6 PHASE-LOCKED LOOPS

Since we can measure phase electrically and can control it with a voltage-controlled oscillator, we can make a feedback loop controlling the relative phases of two signals, a so-called phase-locked loop (PLL), as shown in Figure 15.17. We don't have too much space for them, but PLLs can do some very powerful things, including pulling good data out of a signal buried in noise, and doing detection in a bandwidth much narrower than the uncertainty in the carrier frequency. You can think of a PLL as a lock-in amplifier that makes its own reference signal by following the input.

Crudely understood, a PLL generates a VCO signal whose phase follows the input signal's within the PLL bandwidth f_0, and ignores it outside. A narrowband PLL follows only the average frequency, and so the phase detector output can be used as a PM detector for modulation frequencies above f_0. A wideband one follows the small dips and wheels of the instantaneous frequency, so the VCO control voltage can be used as an FM detector for sufficiently slow modulation.

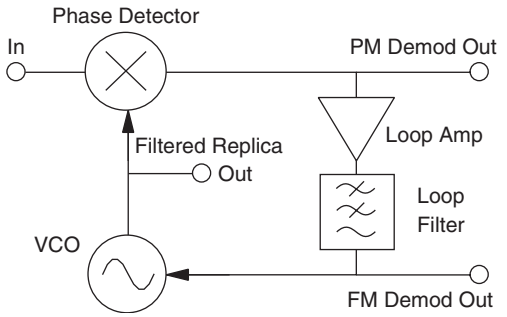

Figure 15.17. Phase-locked loop.

If you're going to build your own PLLs, read Gardner's *Phaselock Techniques*, and watch out for three things: acquisition of lock, which is difficult in wide-range loops with narrow loop bandwidths; severe reference frequency ripple due to the loop amplifier and filter having too little attenuation at f or $2f$; and, of course, frequency compensation in the face of gain variations in the VCO (K_{VCO} in rad/s/V) and the phase detector (k_ϕ in V/rad), which change with center frequency and input signal level.

15.6.1 Loop Design

In Sections 15.4.1 and 18.4.1 we talk about frequency compensation of amplifiers. PLLs are slightly more complicated because if the VCO's radian frequency sensitivity is K_O rad/s/V, its transfer function is

$$A_{\mathrm{VCO}}(f) = \frac{K_O}{j\omega} = \frac{1}{jf}\frac{df}{dV}, \tag{15.14}$$

which is an ideal integrator. The reason is that we control f but measure $\Delta\phi$, and phase is the integral of frequency. (It's usually less confusing to stick to radians for PLL calculations.) The PD characteristic is roughly $K_\phi \sin\phi$, but all that matters is the slope near the operating point (i.e., 0 V), so its gain is assumed to be constant and frequency independent. If the loop filter gain is $A_F(\omega)$, the overall loop gain is

$$A_{\mathrm{VL}}(\omega) = \frac{K_\phi K_O}{j\omega} A_F(\omega), \tag{15.15}$$

so all we have to do is choose A_F. Since we want a phase margin of at least 45°, and the integrator contributes 90°, we have 45° of phase shift to work with when $A_{\mathrm{VL}} > 1$.

The phase detector produces a lot of ripple at $2f$, and usually some at f as well. Diode bridge phase detectors are generally best for high stability applications, due to their very low noise and predictable ripple. Phase–frequency detectors of the 74HC4046 type ideally produce no ripple at zero phase error, and don't need aided acquisition because the output is railed when the loop is out of lock, rather than producing a beat note like other PD types. Unfortunately, they exhibit a dead band right at the zero-ripple point, so you have to bias them a bit off center, which makes their claimed ripple advantage more apparent than real.

Generally, the best loop filters are op amp integrators with lead–lag networks, as shown in Figure 15.18 (assume $\mathrm{R_a} = 0$ and $\mathrm{R_b} = \infty$ for now), because their high DC gain makes the static phase error very small. For our purposes, the best simple method for compensating the loop is first to decide on the desired loop bandwidth f_L. For a reasonable phase margin, set the zero $1/(2\pi R_f C_f) \sim f_L/2$, and choose $R_1 + R_2$ and R_f so that the gain drops to 1 at f_L (remember that the phase detector has gain in V/rad as well). Then you can put in C_1 to roll off the ripple at f and $2f$. If necessary, you can use a notch filter to increase the ripple rejection, but that requires careful attention to phase shifts. Either way, you'll want to use a math program or SPICE to verify that your phase margin is at least 45°. Another source of phase shift is the phase detector itself; linear multipliers allow the output to react to an input phase shift immediately, but saturated or XOR PDs can't react until the next transition, which is a quarter-cycle delay, and a PFD can't react until the next cycle, which averages a half-cycle delay. Make sure you account for these effects if you're building a wideband PLL.

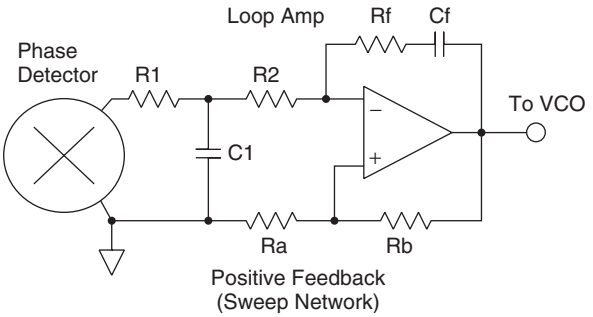

Figure 15.18. Narrowband PLL loop filter with automatic acquisition: when the loop is unlocked, the overall loop gain drops to 0 and the DC component of the phase detector output becomes very small. The positive feedback network *Ra–Rb* takes over, turning the loop amplifier into a slow triangle wave generator ($\dot{f} \ll f_c^2$), which sweeps the VCO frequency until the loop acquires lock.

15.6.2 More Complicated PLLs

So far we've talked only about 1:1 PLLs, where the VCO frequency is the same as the input frequency. If you put a frequency divider between the VCO and the phase detector, you can make a frequency multiplier, and if you put dividers on both the input and the VCO, you have a *frequency synthesizer* that can in principle make any rational multiple of the input frequency. In practice, this hits limits due to noise multiplication [the phase noise goes up by a factor of 20 $\log(f_{\text{out}}/f_{\text{ref}})$], and due to the requirement that $f_L \ll f_{\text{ref}}$ for ripple rejection. Offset loops, in which f_{VCO} is mixed down to a lower frequency f_{IF} before phase detection, are also frequently useful—especially when using acousto-optic deflectors, where they allow the detected signal to be mixed down to a fixed IF (see Section 15.2.1).

15.6.3 Noise

There are some pretty poor PLL chips out there. For high SNR applications, avoid anything with a digital-logic phase detector or an *RC* VCO. (Chips like the CD4046 are great for low performance applications.) Keep the resistor values low, as you would in a low noise amplifier, and use a high output diode bridge phase detector such as a Mini Circuits MPD-1. Those have nanovolt output noise densities and about 500 ohm output impedance, so use really quiet bipolar op amps such as the LT1028A. Use good quality VCOs, such as a VCXO for narrowband applications or a varactor-tuned *LC* VCO with automatic level control for wider ranges. The phase noise of the VCO is suppressed by the loop gain, so try to keep the bandwidth wide unless the oscillator is really quiet. High comparison frequencies are your friends, because they reduce noise multiplication. For more on PLL noise, see Gardner.

15.6.4 Lock Detection

It's obviously important to know when the loop is in lock. A phase–frequency detector makes this easy; PFDs usually come with a lock detection output, and (more usefully) they rail when they lose lock, so a window comparator on the VCO control voltage line

will tell you when the loop is unlocked or near the edges of its range. Lock detection is more of a puzzle when using diode or XOR phase detectors, because their DC output is near zero whether they're locked or unlocked. The usual approach is to use a second phase detector, driven 90° out of phase, and look for a large DC level there to tell us we're in look. We're often interested in the AM information on our signal, so we'd likely be needing an in-phase output anyway (see Section 13.9.3).

15.6.5 Acquisition Aids

Slow PLLs take a very long time to acquire lock, and if they're sufficiently far off, they may never make it. Figure 15.18 shows a high gain PLL loop filter for use with a low noise diode bridge phase detector (e.g., a Mini Circuits MPD-1). This filter has a built-in acquisition aid—when the loop loses lock, it becomes a triangle wave oscillator that sweeps through its entire tuning range until it reacquires. Choose Ra/Rb to be a couple of times V_{OS}/V_O, where V_{OS} is the maximum offset voltage of the phase detector and V_O is the maximum output voltage of the amplifier. (Split supplies are assumed.) Alternatively, you can use a phase–frequency detector as an acquisition aid, but that's a bit more involved because PFDs work at 0° rather than the diode bridge's 90° (a divide-by-4 walking-ring counter is often useful). The only drawback of the triangle wave approach is that the hysteresis forces the phase detector to work slightly off null, which slightly degrades the AM rejection of the loop. If that's a problem, and you have a lock detector, you can use it to control an FET switch that grounds the + input.

15.7 CALIBRATION

A detection system is composed of a detector and a calibration; a barefoot detector isn't much use for quantitative data. Make sure you know how your detector is going to be calibrated before you go too far with the design. One particularly thorny issue you should wrestle with early is whether you can afford to do a two-dimensional calibration; I/Q measurements need to have the I and Q detectors calibrated over a wide range of amplitudes and phases—their output isn't just $(A\cos\phi, A\sin\phi)$. A clever strategy can accomplish this, but it doesn't happen by accident. The alternative is to use a phase-insensitive envelope detector plus a nulling phase detector, which needs only a 1D calibration.

Calibration is often made much easier through the process of linearization, which not only reduces the size of the corrections but often eliminates their cause; for example, putting a matched CMOS switch in the feedback loop of an op amp driven by an analog mux can correct for switch resistance, nonlinearity, and temperature drift all at once (see Section 15.10).

15.7.1 Calibrating Phase Detectors

Phase detectors are easy to calibrate, since phase is easy to turn into time and frequency. You can use two divide-by-N digital counters, driven by a crystal oscillator or synthesizer, to produce two phase-coherent signals for the signal and reference phase inputs of the phase detector (a mixer and filter arrangement can shift them up to frequencies too high for the counter outputs). Make one a pulse-swallowing counter, whose output phase

can be shifted by $2\pi/N$ under software control, and you can take N phase points at each amplitude level. Cubic spline interpolation will do an excellent job for a 12 bit nulling-type phase detector with $N = 100$. The major things to avoid, as in all time and frequency measurements, are leakage of one phase into another, multiple reflections, anything that will cause the phase steps to be unequally spaced. Use lots of shielded boxes, attenuators, and isolation amps in your calibrator, and test it by sitting on the pulse-swallowing button (to shift one channel's frequency slightly) while looking at the outputs on a spectrum analyzer. That'll show up spurious coupling pretty fast.

15.7.2 Calibrating Amplitude Detectors

It isn't that easy to calibrate an amplitude detector to the accuracy of your ADC, mainly due to the difficulty of getting accurate calibration levels. Probably the best way is to use a quartz crystal, excited by a long tone burst, and terminated in a very low, resistive impedance (Figure 15.19). When the burst cuts off, the crystal will ring at its natural frequency f_s (i.e., its series resonance), with exponentially decaying amplitude. Once the ring-down is started, samples taken at regular time intervals will have accurately known relative amplitudes and (even better) will produce a linear ramp from a logarithmic detector. Crystals have Qs of 10^4 to 10^6, which means the decay is nice and slow; a decent 80 MHz crystal will decay at around 1 dB/ms, which is very convenient.

A stable amplitude detector of limited dynamic range (e.g., a temperature compensated diode detector) can be used to measure the initial amplitude and the time constant, from which our *a priori* knowledge of the envelope's functional form allows a good calibration over a very wide range. The main opportunities for corruption here are nonlinearity in the buffer amplifier following the crystal, overdriving the crystal with the tone burst, or allowing wideband noise to dominate the signal at late stages. Keep the rms current in the crystal between 100 μA and 1 mA during the tone burst.

One problem is that the tone burst has to be at $f_s \pm f_s/(2Q)$ or so, or the crystal won't ring very much. This is pretty easy to solve in the lab, because you can just tune your synthesizer until the output peaks, but is more awkward for an online calibration. In that case, you can make the crystal into an oscillator. It isn't as easy to get a good

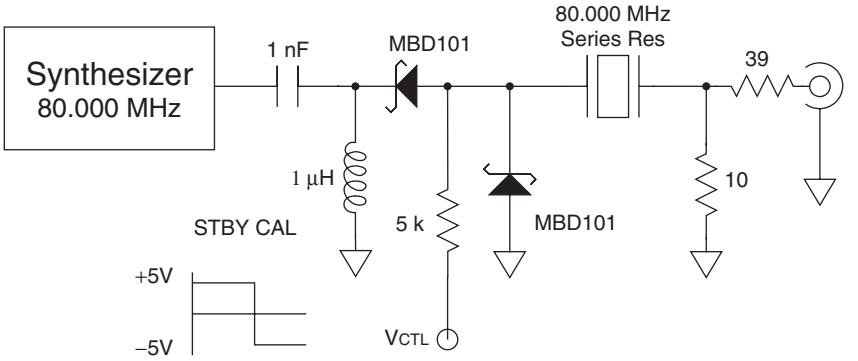

Figure 15.19. Crystal ring-down calibrator: For low oscillation amplitudes, the rate of decay of the oscillation is accurately exponential until it becomes comparable to the noise level in the measurement bandwidth. Since timing is easy to do accurately, this provides an excellent linearity calibration for amplitude detectors and limiters.

result, because you have to get the crystal to oscillate right at f_s, which requires external series inductance, and you have to use diode switches to ground the transistor end of the crystal to cut off the burst.

15.7.3 Calibrating a Limiter

We've talked about AM–PM conversion in limiters, which also requires calibration. The crystal ring-down method will produce the AM–PM characteristic in radians/dB as a function of time, which is very convenient. Putting a calibrated limiter ahead of your phase detector is another way to reduce the calibration to two 1D problems instead of one 2D problem.

15.8 FILTERS

In Section 13.8.5 we talked about filtering in general. Here we'll go into the gory details. A cascade filter is just a bunch of resonant sections strung together in a row. If we put buffers in between them so that they don't interact (as in an op amp active filter), we can calculate their individual transfer functions from the series and parallel impedance formulas, and then just multiply them together to get the filter transfer function, as we did building feedback loops in Section 15.4.3, but in the pure *LC* case it's a bit harder because the sections detune each other and so the design equations become complicated. Thus we usually start from some canned design, transforming it to fit roughly, then running a numerical optimizer and tweaking component values and tolerances to get the final design. *Filter etiquette note:* People sometimes talk about "selectivity" when they mean "stopband attenuation," leading to embarrassing stares when they mention their "high-selectivity *RC* filter."

15.8.1 *LC* Filters

An *LC* ladder filter is a cascade of series and shunt sections. (We'll specialize to the low-pass case for now, so it's series inductors and shunt capacitors, as shown in Figure 15.20.) Such filters have transfer functions that are rational functions (i.e., ratios of polynomials) in f. This is because the element impedances themselves are polynomials or rational functions of frequency, and putting them in series or parallel is equivalent to making rational functions of the impedances. Thus there is no way out of the class of rational functions for these filters' frequency responses. It is in principle possible to make an adequate approximation to any causal transfer function with enough *LCR* sections and enough delay, but in practice tuning highly complex filters is a serious problem.

Rational functions exhibit poles and zeros in the complex plane. These are nothing more than the roots of the denominator and numerator polynomials, respectively. The positions of these poles and zeros determine the form of the transfer function, so transfer functions are designed by moving poles and zeros around, as we saw in Sections 13.8.5 and 15.4.3.

Historically, people have cared more about amplitude than phase, so the usual procedure is to pick a convenient even-order rational function for $|H(f)|^2$, then factor the numerator and denominator polynomials to find the poles and zeros (picking the stable ones, of course). You'll probably want to look the values up in Zverev instead. (One comforting fact is that all ladder filters are minimum-phase networks.)

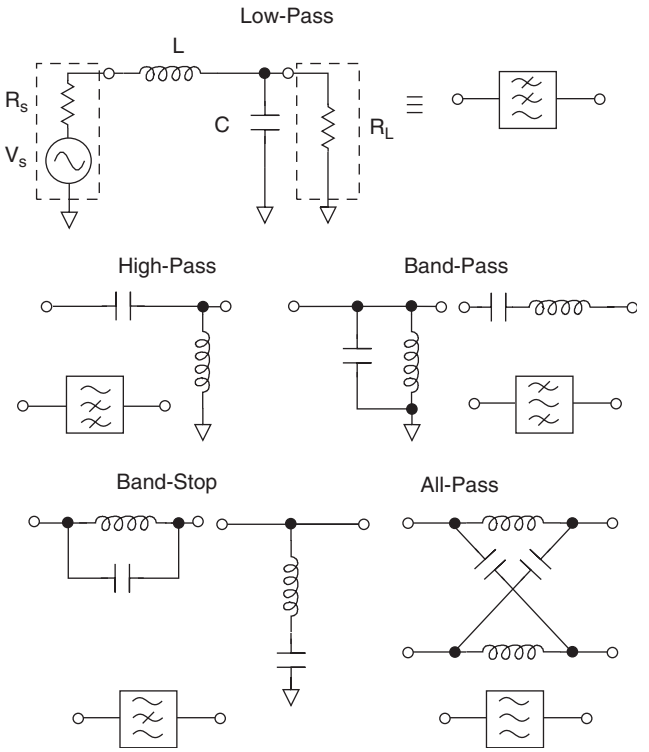

Figure 15.20. *LCR* filter sections; the all-pass is a balanced lattice section, the rest unbalanced ladder sections. Band-pass and band-stop filters have both series and shunt realizations.

15.8.2 Butterworth Filters

The simplest *LC* filter is the Butterworth, which has

$$|H(f)|^2 = \frac{1}{1 + (f/f_c)^{2n}}, \tag{15.16}$$

which is very flat near 0 (the vastly overrated *maximally flat* property), but rolls off continuously through the passband. Butterworths are easy to design and have only moderate group delay problems, but have few other redeeming features.

15.8.3 Chebyshev Filters

Chebyshev polynomials have uniform-sized ripples for $x \in [-1, 1]$, and steeply rising monotonic behavior outside that. A lowpass Chebyshev filter comes from picking

$$|H(f)|^2 = \frac{1 + \epsilon^2 \mathrm{T}_n^2(0)}{1 + \epsilon^2 \mathrm{T}_n^2(f/f_c)}. \tag{15.17}$$

There will be a small amount of passband ripple, approximately $\pm 10\epsilon^2/\ln(10) \approx 4.3\epsilon^2$ decibels. Obviously there is a trade-off between how fast the filter falls off and

how big the ripples are; in practice, we rarely use filters with more than 0.5 dB of ripple, and 0.1 or 0.2 is more common. Chebyshev filters distribute the passband error more uniformly and gain sharper skirts in the process, which is pretty worthwhile.

15.8.4 Filters with Good Group Delay

As we saw in Section 13.8.9, if we want good group delay we have to accept either very gradual skirts and some passband bowing (like a Butterworth but even worse), or use a group delay equalizer to even it out. Inside a feedback loop it is important to have minimum phase, so in that case look for Bessel, Gaussian, or equiripple phase filters.

15.8.5 Filters with Good Skirts

The filters we've considered so far are *all-poles* filters, that is, the numerator is a constant, so there is a nonzero response at all frequencies and a monotonic falloff in the stopband. By putting a higher order polynomial in the numerator, we can make the skirts fall off more rapidly, at the cost of poorer attenuation deep in the stopband. The *elliptic function* or *Cauer* filter is the classical design, based on elliptic function theory, but any filter with zeros at real frequencies is usually called an *elliptic filter* anyhow. An example is a lowpass filter with inductors in the series legs and series *LC*s in some of the shunt legs. These can be made with highly asymmetric skirts, which is great if you have some ugly close-in spur to get rid of. Look these up if you need to use one. The group delay of these things is not a sight for the faint-hearted.

15.8.6 Lowpass to Bandpass Transformation

Bandpass filters are generated from the lowpass prototype by a simple transformation, in three steps.

1. Choose a lowpass filter with the same bandwidth.
2. Find the component values for the prototype lowpass from tables (*e.g.* Zverev).
3. Resonate each element at $f_0 = (f_1 f_2)^{1/2}$, the design center frequency. For a bandpass filter, you want the impedances at f_0 to be correct for the center of the passband, that is, the equivalent of DC for the lowpass: series-resonate the series elements, and parallel-resonate the shunt elements. For a bandstop, you want f_0 to be the center of the stopband, so you do it the other way.

This transformation is an example of a *conformal map*, in the bandpass case $f' \rightarrow f - f_0^2/f$. The bandpass and bandstop filters designed this way have symmetrical responses when plotted against $\log(f)$. The imaging of the filter response from lowpass to bandpass results in the section reactances changing twice as fast with frequency, because the inductive reactances are going up and the capacitive ones are going down. The full bandwidth stays the same, because although each side is half as wide, we get an image on each side of f_0. The settling time is doubled compared with the lowpass prototype. If you're trying anything the least bit fancy with this filter (*e.g.* constant group delay), you should stick a numerical optimizer on the design when you're done. Remember from Section 13.8.8 that the nonlinear frequency warp makes this transformation useless for linear-phase filters.

Aside: Component Limitations. The design procedure is general enough to let you design a 100 Hz wide Butterworth bandpass filter at 1 GHz, but unfortunately that would need 30 millihenry inductors and 30 attofarad capacitors, both with Q values of 10^8, and you won't find those at Radio Shack. You can do the equivalent with a superhet, as we saw in Chapter 13. When your component values become extreme, you're reaching the limits of realizability of a filter.

15.8.7 Tuned Amplifiers

A tuned amplifier stage has a tuned circuit in its input, output, or both. A tank circuit in the collector of a common-emitter stage is a common example, and so is a photodiode amp with the diode capacitance resonated away with a series or parallel inductor, as in Section 18.5.1. Tuned amps are useful in reducing the number of filters you need, reducing the broadband noise, and giving you flexibility in the impedance levels you use. They have to be adjusted manually, so stick to low Q values (usually below 10).

15.8.8 Use Diplexers to Control Reflections and Instability

You can compensate a series RC by putting a series RL in parallel with it; the Rs have to be equal, and the L is R^2/C. This actually gives constant, pure resistance equal to R at all frequencies. The nice thing about this is that the high frequency energy goes to one resistor and the low frequency energy to another; you can filter the output of some termination-sensitive device like a diode mixer without an impedance mismatch, sending a low frequency IF to the LC arm and the image and high frequency spurs to the RC arm. This is a simple example of a *diplexer*.

15.9 OTHER STUFF

15.9.1 Diode Laser Controllers

Diode laser power supplies need to have very low noise, preferably below the shot noise. This isn't hard, with care. The idea is to make it extremely quiet at AC, then wrap a slow feedback loop around it to make it stable at DC, while keeping very careful control of its transient response into weird loads. We saw in Sections 14.6.5 and 18.4.6 how to make a very quiet current source, and an op amp sensing the collector current and comparing it with a heavily filtered voltage reference finishes the job, as shown in Figure 15.21. Make sure you supply a mechanical relay contact across the outputs to protect the laser against electrostatic discharge when it's off.

15.9.2 Digitizing Other Stuff

So far, we've stuck with converting our signals to voltages and digitizing them afterwards. This is the usual way of doing things, but not always the best. For example, consider the phase digitizer of Figure 15.22. A state machine called a *successive approximation register* uses a DAC driving a voltage-controlled phase shifter to perform a binary search of the DAC's code space to find the null at the phase detector output. This keeps the phase detector right at 90° all the time, so the calibration only needs to cover errors due to amplitude, and not phase nonlinearities changing with signal level. The state machine

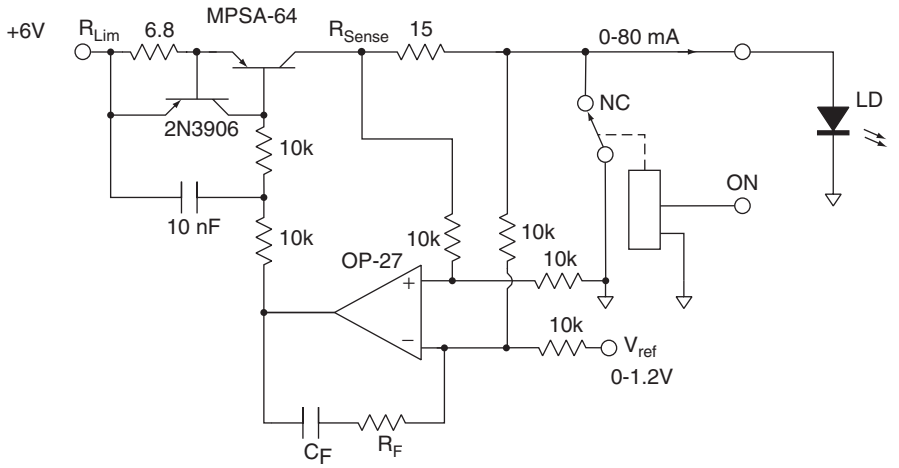

Figure 15.21. Diode laser controller design.

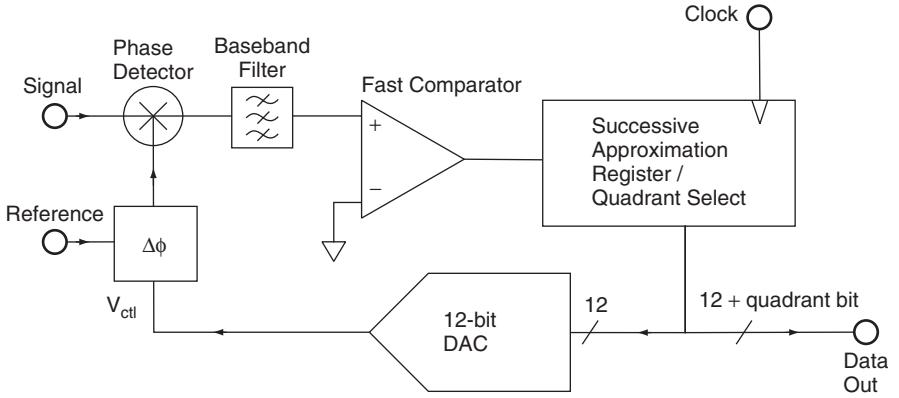

Figure 15.22. Successive approximation phase digitizer: the successive approximation logic finds the correct null by a modified binary search.

needs an extra flip-flop to make sure it's shooting for the right null—there are two, of course, and only one is stable. (Why?)

15.9.3 Use Sleazy Approximations and Circuit Hacks

This far into this book, it isn't necessary to repeat it too loudly, but there's usually a circuit hack to fix a seemingly intractable problem, as long as it's technical and not fundamental. For example, we can make really, really quiet amplifiers, but can't reduce the shot noise of starlight except by gathering more. It's worth having a good look at the temperature-compensated breakpoint amp (R. Widlar, National Semiconductor Application Note AN4) and the low phase error amplifier of Section 15.10 (Analog Devices Application Note AN-107). Some of the oscilloscope front end tricks in Jim Williams's books are really worth reading too. Chapter 18 has an extended example of

how to get through a difficult circuit problem (a shot noise limited front end for low light), and talks about the laser noise canceler, which is another neat circuit hack for doing ultrasensitive detection with noisy lasers.

15.9.4 Oscillators

We often talk about the characteristics of RF signals without much attention to where they come from, but unless you're listening to a pulsar with a crystal radio, all must come from some oscillator ultimately. An oscillator consists of an amplifier and a frequency-selective network, which feeds back the oscillator's output to its input. The network is often a resonator, but this is not always the case; the criterion for oscillation is that the total phase shift around the loop should be an exact multiple of 2π, while the gain equals 1. This condition sounds very delicate, but in fact is not; the frequency continuously adjusts itself to make the phase condition true, and if the gain is initially larger than 1, the amplitude of the oscillation will increase until circuit nonlinearity reduces it to an average value of exactly 1, the *self-limiting* condition. This intrinsic nonlinearity causes the low frequency noise of the amplifier to intermodulate with the oscillation, producing noise sidebands at low modulation frequency, which is why the best oscillators use ALC instead of self-limiting (see Sections 15.6.3 and 15.10.5). Oscillators have noise in both amplitude and phase, with the phase noise generally being much more objectionable, because it's hard to get rid of; amplitude noise can be eliminated with an external limiting stage, but reducing the phase noise requires redesigning the oscillator or phase locking it to some quiet reference source.

Oscillators are divided into categories, depending on the nature of their feedback networks and how the amplitude limiting is done. Stable and quiet oscillators result from quiet amplifiers, in feedback loops based on stable resonators whose phase shifts vary rapidly with frequency, and amplitude limiting schemes that minimize the nonlinear mixing of the amplifier noise with the oscillator signal.

They aren't that hard to build, but they need lengthy explanation, so for the purposes of this book, oscillators are something that you buy, or build from oscillator ICs such as the LMC555 astable or SA601 RF mixer/oscillator.

15.10 MORE ADVANCED FEEDBACK TECHNIQUES

15.10.1 Put the Nonlinearity in the Loop

We saw that changes in A_{VOL} get suppressed by a factor of A_{VL}. It isn't difficult to use this linearizing property to correct for other circuit elements. There are two ways to do this; we'll use a FET switch as an example. The first way is to put the switch in series with the output, with another one to keep the amp from railing when the first one opens, as in Figure 15.23a. This relies on the stability of the summing junction for its OFF isolation, and consequently isn't great at high frequency, but works on unmatched switches and gains other than -1. You can also rely on monolithic matching and use a constantly closed switch in a unity gain inverter's feedback loop to provide a matched nonlinearity, as in Figure 15.23b (see also Section 15.12.5). It's possible to combine the two tricks ju-jitsu style, to make the op amp's input errors track themselves out, as in Figure 15.23c. This puts some stress on the amplifier's frequency compensation, so it's worth simulating it before using, just to make sure it does what you expect. (This is one place where even op amp macromodels might come in useful.)

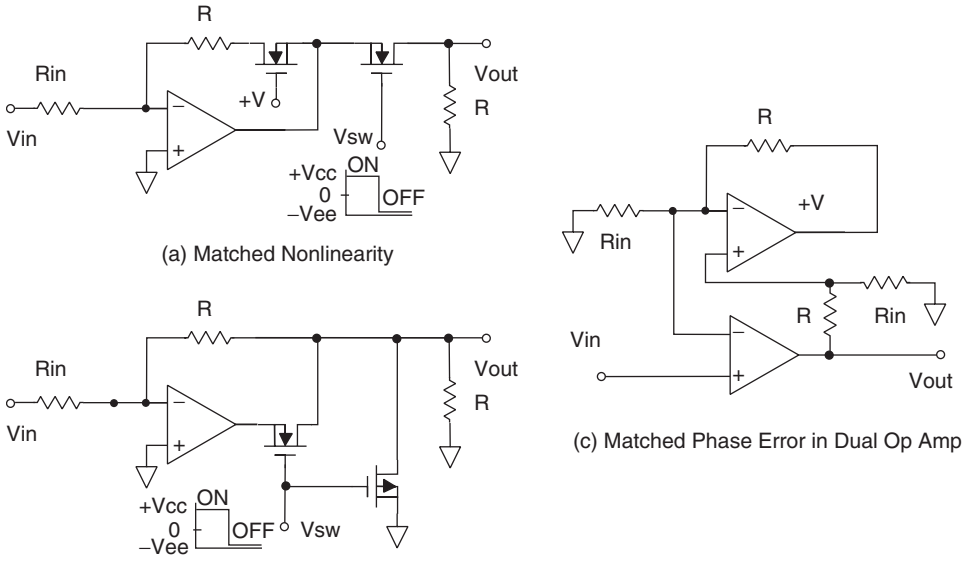

Figure 15.23. Eliminating nonlinearity: (a) matched on-resistance nonlinearities cancel; (b) nonlinearity tracked out by the loop (extra switch needed for off-state isolation; and (c) combining (a) and (b) phase error due to op amp frequency response can also be tracked out, although some amplitude peaking occurs (see Analog Devices Application Note AN-107).

15.10.2 Feedback Nulling

One interesting method of suppressing background is to use a servo loop that detects the background plus signal, and nulls out the whole works. Of course, this helps only when the desired signal is a transient, so that the loop doesn't suppress the signal as well as the background. Dithered systems using Bragg cells are especially suitable for this (see Section 10.8.3).

15.10.3 Auto-zeroing

A related technique is auto-zeroing, where we measure the background at a time when the signal is known to be 0, and subtract it out in analog. An example is correlated double sampling in CCDs (see Section 3.9.4), where the kTC voltage uncertainty is different for each pixel clock cycle, but we can measure it separately and so eliminate it completely. We can use it to subtract off the huge sloping baseline in current-tunable diode laser measurements, by adding in an adjustable amount of the scan signal, using a multiplying DAC. These examples may seem like sleight of hand, but it can help a lot; unless your system is taking data continuously at all times, the local servo loop will do a better job than postprocessing for this, and the demands on the system's dynamic range are enormously reduced. In the diode laser example, the laser power variation can easily be 50%, and if we're trying to detect absorptions in the 10^{-4} range, that's 74 unnecessary decibels.

Auto-zeroing at a rate of f_s suppresses noise at lower frequencies, and puts some funny ripples in the noise spectrum up to a few times f_s due to switching transients and redistribution of noise power. Signals at integer fractions of f_s are nulled perfectly, and intermediate frequencies go roughly as f/f_s. This is valuable in eliminating the horrible $1/f$ noise of CMOS amplifiers and is widely used in very low level DC amps.

15.10.4 Automatic Gain Control

It's tempting to extend this idea to automatically changing the gain as well as the offset, but that's not usually a good idea. AGC is used in radios with great success, but there the exact signal level is of no concern whatever, which is utterly unlike the situation in instruments. Linearity and absolute accuracy tend to get compromised pretty badly with AGC control, and fast AGC loops (faster than $\sim$1% of the IF bandwidth) can produce parametric effects, such as asymmetric pulse decay, or phase modulation as a function of AGC voltage. It is often best to use a combination of range switching, redundant detectors, and digital scaling. One exception is in PLL phase demodulators, where optimum low SNR performance requires a combination of wideband and narrowband AGC (see Gardner).

15.10.5 Automatic Level Control

On the other hand, AGC's twin brother *automatic level control* (ALC) helps enormously in oscillators and frequency synthesizers, by allowing them to operate in a highly linear regime while having enough excess gain to start up reliably—see Section 15.9.4.

15.10.6 Feedback Loops Don't Have to Go to DC

It's a good idea to use DC coupling when you can, because it saves blunders. Things that are obvious when the DC is there (e.g., a photocurrent of 0 mA or a front end whose output is pegged) are easily missed in a purely AC-coupled system. On the other hand, feedback loops can sometimes do remarkable things when you don't DC couple them. This is usually because of the perfection of capacitors in subtracting off DC. One example is the capacitance multiplier of Example 14.1. Another is the high voltage supply quietener of Figure 15.24, which jiggles the bottom of a high voltage supply to keep the top still, as measured via a capacitor. Ten volts of 120 Hz ripple can be brought down to the 100 μV level this way for about $5, which really helps the stability and noise of piezoelectric actuators and PMTs.

Aside: Motorboating. Instability usually results from too much phase lag at high frequency, but AC-coupled feedback loops can also exhibit instability at low frequency due to accumulated phase *lead*. This instability is called motorboating, an apt description of how it sounds over a speaker.

15.11 HINTS

Invert When Possible. If you can build your circuit inverting, do it. Op amps are not as happy when there's a big common-mode swing on their inputs, and their protest shows up in nonlinearity and offsets. Inverting operation keeps the inputs still, somewhere near

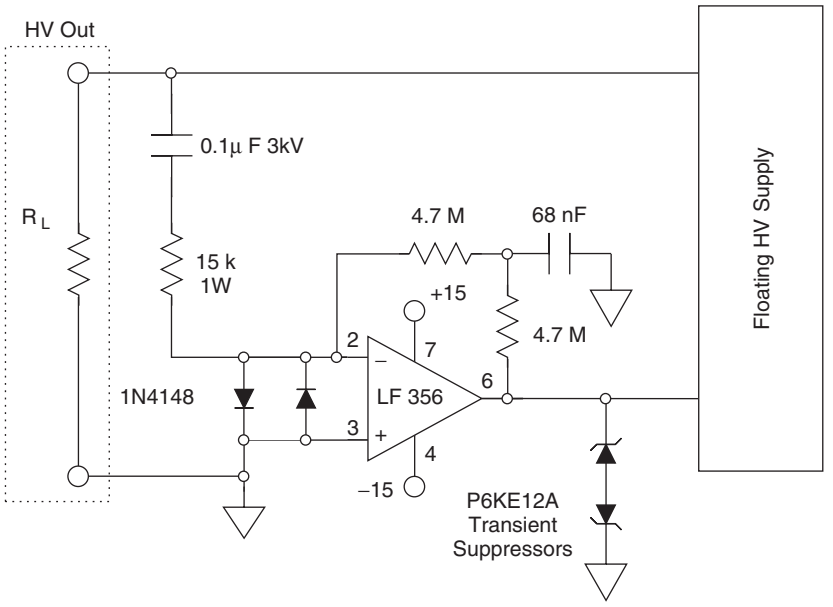

Figure 15.24. High voltage power supply quietener.

the middle of their range, and that helps a lot; you also never run into the problem of somebody's wide output swing exceeding the next guy's input common-mode range that way. In general, inverting stages with similar gains will be noisier than noninverting ones, because we often need to use higher value resistors in inverting stages, and because an amp with inverting gain $-A$ has a noninverting gain of $1 + A$, so you get more signal for the same noise by not inverting.

Watch for Startup Problems. There are lots of circuits that don't start up well. An LM7805 regulator, which is of course a byword for good behavior, will fail to start if you drag its output below ground while its input is coming up; its output remains at a high impedance, so it'll never pull itself back above ground. This can easily happen if the negative supply starts up quickly, but a rectifier connected from output to ground will prevent it. Circuits can also latch up, for example, old technology CMOS chips, which made quite reasonable SCRs if you took their inputs outside the supply range.

Besides failure to start, circuits can also be damaged by turn-on transients. For example, consider a 0–10 V multiple DAC driving a low breakdown voltage device such as a micropower Gilbert cell mixer used as a DAC-controlled gain source, or a ±15 V op amp driving the base of a transistor with its emitter near ground, as in Section 18.6.3. These devices can be damaged by momentary overvoltages, and those are almost inevitable on startup unless you use a voltage divider, current limiting resistor, or protection diode. You can normally ignore this problem when using ICs running off the same power supplies, but not always. Note: For simulation devotees, SPICE often misses this sort of problem.

Subtract, Don't Divide. Since all the photocurrents in most instruments are exactly proportional to the source power, it is very tempting to measure the source power

independently and then divide it out, either with an analog divider chip or in software, yielding a beautifully normalized result. Except that it doesn't usually work very well.

Most of the time, we're measuring a small signal on a big background (the proverbial grass on top of the Empire State Building), so we have a little dynamic range problem. Let's say our light source has a $1/f$ noise δ of 0.2%/decade, and that our signal is a 5 mV rms ripple on a 5 V background. In 1–10 Hz, that's $5\ \text{V} \cdot \delta = 10$ mV of noise from the background, plus noise intermodulation of $5\ \text{mV} \cdot \delta = 10\ \mu\text{V}$ from the signal itself. An ideal divider fed a faithful replica of the source power will completely eliminate both. But let's go on carefully. Analog dividers with a ± 10 V range have flatband noise of about 1–2 $\mu\text{V/Hz}^{1/2}$ with a full-scale denominator, and a $1/f$ corner of 10 Hz or so—about 60 dB worse than a good op amp. By the time we've thrown in that -60 dB signal-to-background ratio, we've given up 120 decibels of dynamic range before we even start. In our 1–10 Hz bandwidth, the noise is

$$\begin{aligned} e_N^2 &= \left(2\mu\text{V/Hz}^{1/2}\right)^2 \cdot \left(9\ \text{Hz} + \int_{1\ \text{Hz}}^{10\ \text{Hz}} \frac{10\ \text{Hz}}{f} df\right) \\ &= 2\ \mu\text{V} \cdot \sqrt{9 + 10\ln(10)} = 11.3\ \mu\text{V rms}, \end{aligned} \tag{15.18}$$

which limits us to a 54 dB SNR. If the measurement intrinsically has a shot noise limited CNR of 140 dB in 10 Hz, then even with the background taking up 60 dB of that, we've lost 26 dB to the divider, which is just plain silly.

It's even worse if we digitize first, because even a noiseless, perfect 16 bit digitizer has only 107 dB; combining two measurements loses us 3 dB, and the 60 dB leaves us a maximum SNR of 44 dB, a 36 dB loss solely to the digitizer.

The right way to do it is to subtract instead. A op amp plus an 8 bit multiplying DAC, (with the source intensity feeding the reference) will get rid of the background down to the 10 mV level, including almost all of the additive noise; after that, digitizing the remainder and dividing by the source intensity in software will do the rest of the job.[†] Because of the high accuracy required, you have to do the subtraction right at the detector, and good background suppression requires great accuracy in the ratios of the two arms. For measurements in white light or with incoherent sources with lots of low frequency noise, this quasidigital method reduces the dynamic range to something your digitizer can probably handle. If you're doing a laser measurement, you can do a great deal better than this using a laser noise canceler (see Section 10.8.6).

15.12 LINEARIZING

In this day of high precision ADCs, fast computers, and cheap memory, why do we care if a measurement is linear? We're going to calibrate it anyway.

There are a number of reasons. Severely nonlinear measurements are a waste of good bits, because their resolution is widely variable over the input range. Nonlinearity inside a feedback loop changes the loop bandwidth with signal, which leads to weird parametric effects, poor settling, and possible oscillation. A measurement that is inherently linear will usually have a much more stable calibration curve, which helps a lot with real-world

[†]This of course assumes that the two are sampled simultaneously. If you're using a data acquisition card, they probably aren't, and it matters.

accuracy—the calibration can be simpler and less frequent. Harmonic and intermodulation distortion of the signal is much reduced by linearizing, which allows simple and extremely sensitive system tests using sine waves and filters.

The basic techniques for linearization are circuit balancing, matched errors, feedback, feedforward, and bootstrapping.

15.12.1 Balanced circuits

We've encountered lots of balanced circuits, for example, diode bridge mixers and transistor differential pairs. The symmetry of these devices suppresses even-order harmonics, because the positive and negative half-cycles are treated the same way. Since the coefficients of the distortion polynomial (Section 13.5) are usually steeply decreasing with order, this makes a big difference to the circuit linearity.

15.12.2 Off-stage Resonance

A varactor phase shifter can be linearized, and its range extended, by wrapping series and parallel inductances around it, as shown in Figure 15.25. A nearby resonance speeds up the variation of reactance with tuning voltage at the ends of the range, where the varactor tuning curve flattens out. The MMBV432 series-resonates with the 45 nH inductors just beyond the high capacitance (2.8 V) end of the range, and parallel-resonates with the sum of the 43 and 45 nH just beyond the low capacitance (10V) end. Proper placement of these resonances leads to a reactance range from near 0 to effectively ∞, so we can get very nearly the theoretical range of π radians per section. Its linearity is much better than with just the varactor.

15.12.3 Waveform Control

Phase detectors have output characteristics that depend on the input waveform. If the waveforms both have flat tops (e.g., square waves or clipped sine waves), then until

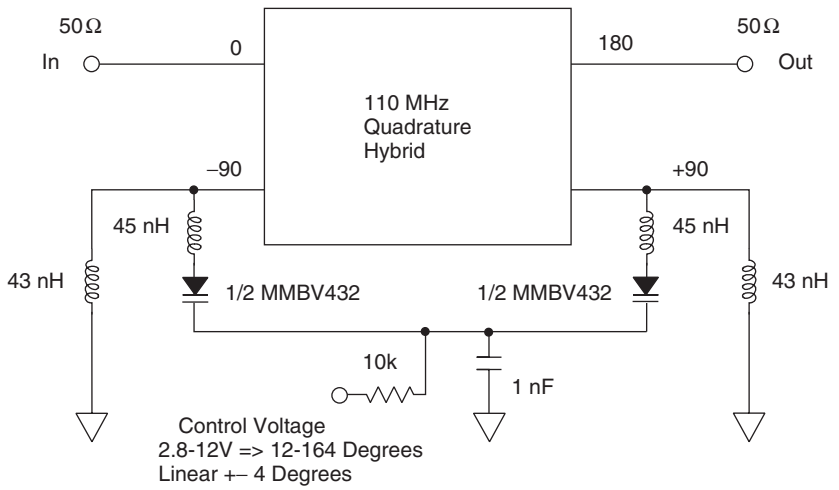

Figure 15.25. Linearized varactor phase shifter.

the phase shift gets large enough that the edges start to overlap, the phase detector's $V(\phi)$ curve will be linear. Only the ends of the curve will depend on the details of the waveform edges. Thus a Gilbert cell phase detector can be linearized by driving both the LO and RF ports hard enough to switch the transistors completely, and a diode bridge phase detector (which doesn't like being overdriven) can be driven from logic gates.

15.12.4 Breakpoint Amplifiers

A last resort is to make an amplifier whose distortion is the inverse of your circuit's. This is so tempting and so nearly worthless that we won't discuss it much. The origin of the nonlinearity of most circuits is some parameter that drifts with temperature and time, so that tweaking a breakpoint amplifier to fix it is a complete waste of time. One important exception is resistance heaters, whose output power is quadratic in the input power, so that the loop gain will vary all over the map with set point and ambient temperature; a breakpoint amp can limit the range, and since it's a low accuracy application and inside the feedback loop, it will work okay. You can stick diodes in the feedback loop of an op amp, which drifts like mad with temperature, or try Widlar's temperature-compensated, ultrasharp breakpoint amp idea from National Semiconductor's Application Note AN4, which is much more accurate but still slow. Good luck.

15.12.5 Feedback Using Matched Nonlinearities

You can also rely on monolithic matching and use a constantly closed switch in a unity gain inverter's feedback loop to provide a matched nonlinearity. If you have a nonlinearity that is inherently matched to your circuit's, then all is easy; you can just put the matched nonlinearity in the feedback loop of an op amp. A typical example of this is fixing the nonlinearity of a CMOS switch by putting another one, closed, in series with the feedback resistor of the op amp, as shown in Figure 15.23a. The same trick is often used with dual optocouplers, but since they are not monolithic devices it is somewhat less successful.

Still another variation uses one op amp's input error to null out another's: the low phase error amplifier of Figure 15.23 uses the input error of one-half of a dual op amp to null out the error of the other half, operating at exactly the same gain. Assuming perfect matching, this compensation makes A2's input errors quadratic in f rather than linear, so that the closed-loop phase shift is below 0.1° for $f < 0.11 f_c$ instead of $0.002 f_c$ as it would be otherwise. The price is a moderate amount of gain peaking (3 dB) near f_c, and having to work at a high enough closed-loop gain that the circuit doesn't oscillate. (It's not clear that production variations and layout strays will let you reach 0.1° in real life, of course.)

15.12.6 Inverting a Linear Control

One good way of getting linearity is to use a device known to be very linear, and putting it in a feedback loop controlling the nonlinear one. An example is wide range control of LED brightness by shining part of the LED output on a photodiode and using an op amp to force the photocurrent to equal an externally supplied reference current. This is useful with voltage-to-frequency converters, photodiodes, and heaters.

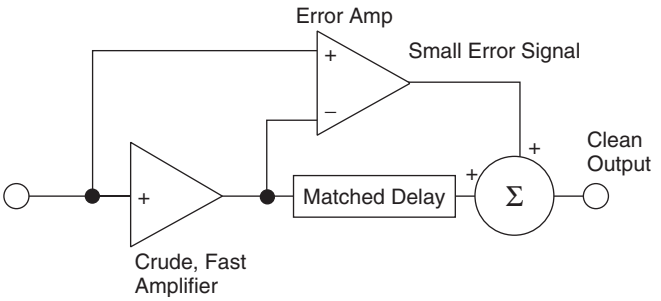

Figure 15.26. Feedforward system.

15.12.7 Feedforward

Feedback control is very useful, but it has its limitations, mainly caused by the unavoidable phase shifts and the attendant limitations on bandwidth, loop gain, and transient response. Open-loop control doesn't have these problems to the same degree, but it is inaccurate. The solution to this dilemma is *feedforward*, shown in Figure 15.26. The error signal is applied (in open-loop fashion) to the output of the amplifier instead of its input, with a suitable delay inserted to make the uncorrected signal and the correction sync up in time.

An example of this is the beam steering and focusing electronics in synchrotron accelerators, to control both systematic and random position and focus errors in the beam. The particle beam consists of individual bunches of particles going round in a circle. Bunch N, which has some position error, goes round the loop at almost the speed of light and can't be slowed down; however, the error signal goes across a chord, whose shorter path length compensates for the limited speed of the electronics and magnets—the correction arrives just in time to fix the error on bunch N, the same one that the measurement was done on. This is an excellent application of feedforward; a feedback design would have used a measurement on bunch N to control bunch $N + 1$, which in the face of random packet-to-packet errors is obviously inferior. This example is not pure feedforward, because the same bunch will come round to the sensor again, be measured again, and corrected again, which is a feedback process. If the system behavior drifts a bit, a supervisory program that occasionally estimates and tunes the feedforward parameters will usually do a good job.

15.12.8 Predistortion

Another feedforward-like technique is *predistortion*, in which the nonideal behavior of some amplifier or transmission medium is compensated in advance. A filter can boost high frequencies to compensate for dispersion and high frequency rolloff in high speed cable links, for example.

15.13 DIGITAL CONTROL AND COMMUNICATION

You'll spend a significant amount of design time just figuring out how to make all the data and control bits twiddle at the right times, so make it easy on yourself. The control and

communications tasks are not normally the places you should spend all your effort—they may influence the price of your instrument, but not the specifications. The general philosophy here is to use PCs (or single-board computers running DOS) to control instruments and network with the outside world, and build small amounts of special-purpose hardware to do the things the PC is bad at (e.g., sequencing). If your instrument is going to be built in tens of thousands, it starts making sense to use embedded microcontrollers, but in that case you'll have help (and need it, too, unless that's your field).

Real-time operating systems for embedded processors exist, but you don't want to open that can of worms unless you really have to.

For communication between the PC and instrument, choose (best to worst) a bidirectional printer port, an RS-232 port (if you can find either of them nowadays) a purchased bare-bones USB or TCP/IP communications module, a frame grabber, a data acquisition card, or a custom designed plug-in card.

15.13.1 Multiple Serial DACS

One of the pleasant consequences of Moore's law is a steady improvement in DACs and ADCs. One example is the octal serial DAC, a $2 device that replaces eight pots and allows software control. There are also EEPots, which are multiplying DACs with EEPROMs remembering the setting, and are good for infrequently required adjustments. An octal DAC and a really good understanding of what your system does can make automatic tweaking of the operating parameters very attractive.

It is a simple matter to connect a serial DAC, or a string of them, to a PC parallel port, and twiddle bits to load them (you can make up your own data protocol with complete freedom on parallel ports, because every bit is under your control). Make sure you connect the data out pin back to the port, and verify that the correct data got loaded by reading it back.

15.13.2 Data Acquisition Cards

The author must make a confession: though he has bought the occasional data acquisition card over a span of 20 years or so, he really hates them. It isn't that there are no good ones, but the vast majority are really suitable only for undemanding low speed use. The main problems are lack of time coherence, short product lifetimes, lack of detailed schematics, and poor quality Windows software that hides bugs. The combination leads to flaky measurements and reasonable-looking wrong answers. There are honorable exceptions, but they are few, and the many bad ones have sowed confusion through two generations of experimentalists.

15.13.3 Nonsimultaneous Sampling

The sampling theorem states that we can reconstruct a band-limited function exactly from equally spaced samples. The key is "equally spaced." Most N-channel A/D cards have one actual sampling digitizer and an N:1 multiplexer. This leads to lots of flexibility and great-sounding specifications at low cost, but rarely to good measurements. Say you're sampling four inputs at 10 kHz. What is the time relationship between them? You don't know. Some cards take the samples as close to each other as they can, but others space the acquisitions out equally over a sampling period, regardless of what the period is,

so you can't improve the time coherence even if you slow to a crawl. If you use the manufacturer's software, you will get them plotted as though they were really coincident in time—in other words, the software constantly tells you lies about your data.

It's especially bad if there are high frequency components in the data you're sampling. You might have a measurement scheme such as subtraction or division that ought to be very stable regardless, but even small sampling time errors will function like a delay discriminator (see Section 15.5.12) and bring those out-of-band components right into your measurement where you don't want them.

Besides the multiplexing, how regular is the sampling, really? PC sound cards are quite good at this, and more modern A/D cards often have hardware timing as an option, but it isn't automatic.

15.13.4 Simultaneous Control and Acquisition

Most instruments don't just sit there and take data, like an old-fashioned chart recorder. They go somewhere, take data, move, take data, move, recalibrate, check the motor speed, take data,. . . , with control commands and data acquisition interleaved. They have to be interleaved correctly, too, observing all the timing restrictions, *especially* the requirements for perfectly periodic sampling and of simultaneous sampling of data that are to be combined later.

Doing this well, that is, with coherent timing, is usually hard with ready-made analog I/O cards, and finding out whether the settings you picked are actually doing what you think they are is even harder. If the card depends on the PC for any of its timing, interrupt latency and multitasking will completely disrupt the show by introducing delays of tens of milliseconds at unpredictable intervals. This makes many kinds of measurement impossible. Pretty-face GUI programs—even very expensive measurement suites—make this even worse, because there's no way to find out what is really going on under the covers.

To be sure of good performance, it's usually better to wire up something simple yourself, so you know exactly what it's doing and why. If you're building a commercial product for volume production, choose a major microcontroller family and write the code yourself. You can sit the processor in a housekeeping loop, and use one of the internal timers to interrupt the processor at each data point. This can be done surprisingly fast, and the timing will be very good (with a few flip-flops sprinkled around for resynchronizing, it can be very good indeed).

If you're building lab stuff, relax. Just use a couple of PLDs and maybe a FIFO buffer such as a 74ACT7804 to make a finite state machine. It'll effortlessly go 500 times faster than a PC-based solution, have very low jitter, and not bring your PC to its knees. Alternatively, consider using a small MCU module such as those from Parallax, perhaps with a bit of hardware assist, or a single-board computer running DOS. Either way, once you get good at it and have a drawer full of parts, it only takes a day or so to reconfigure the hardware. That's a good deal if your experiment is going to last awhile.

Put the complicated stuff like detailed housekeeping, communication, and data reduction on the PC side, where you can write it using a well-upholstered C or C++ compiler and debugger. Another approach is to use a flash EPROM with a counter generating addresses sequentially; you basically just store the whole timing diagram of the system in the flash, and the counter steps you through it. You may need to resynchronize the

output of the flash memory with a register (e.g., a 74HC374) so that it only changes state synchronously with the clock edge. An in-circuit programmable flash will allow you to control the sequence on the fly from your PC. This is a poor man's arbitrary waveform generator. For more complicated things, you can use a small single-board computer running Linux, but that involves a bit longer learning curve for most people. Still, for $300 or so you get a pretty whizzy set of tools—ports, flash, Ethernet, video, and lots of memory and CPU power, all with open-source tools.

A bidirectional printer port (just about all printer ports are bidirectional, if you can still find one) can be used for all manner of control and data acquisition tasks. All the control and data lines are software programmable, so you can make up your own rules as you go along (you need a small library to allow your code to twiddle the hardware directly, but you can download several). This is really a sweet solution to the state machine problem. Of course there are also FPGAs if you're good at that.

15.14 MISCELLANEOUS TRICKS

15.14.1 Avalanche Transistors

There are lots of situations where you have to make something switch quickly with minimal jitter. If the voltages involved are small, say, less than 2 V, you can generally do it with logic parts such as fast CMOS or picosecond ECL. When higher voltages are involved, e.g. switching longitudinal-mode Pockels cells or driving a high speed magnet, and ordinary power MOSFET switches aren't making it, there are three ways of proceeding: thyratrons, spark gaps, and avalanche transistors. Of the three, spark gaps are simple and fast, and can stand high voltages, but require a lot of maintenance and have poor jitter, because the initial few ionization events are stochastic; thyratrons are powerful but slower; and avalanche transistors are lower powered, but very fast and much more repeatable than the others. Avalanche-rated transistors are available from Zetex, along with a lot of good application notes. Their devices tend to be quite a bit slower to avalanche than the old standby 2N2369, which can easily do 300 ps with picosecond jitter (see Figure 15.27). Interestingly, it's the old, slow, diffused-junction parts that avalanche fast. You can also get few-kilovolt pulses as short as 100 ps from the reverse recovery of rectifier diodes, if you hit them hard enough.[†]

15.15 BULLETPROOFING

You have to expect failures. It's critically important to design your system so that it can fail without melting down. For example, make sure your power supplies have well-defined current limits, that there's a thermal cutout on all power circuits and a limit switch on all mechanical motions. Most of all, make sure as far as possible that no hardware damage or human injury can occur as a result of software failures, turn-on transients, or any conceivable abuse of the inputs and outputs. It is proverbial that you can't make something idiot-proof, but you can at least make it idiot-resistant.

[†]See the publication of I. V. Grekhov et al., High-power subnanosecond switch. *Electron. Lett*. **17**, 422–423 (1981).

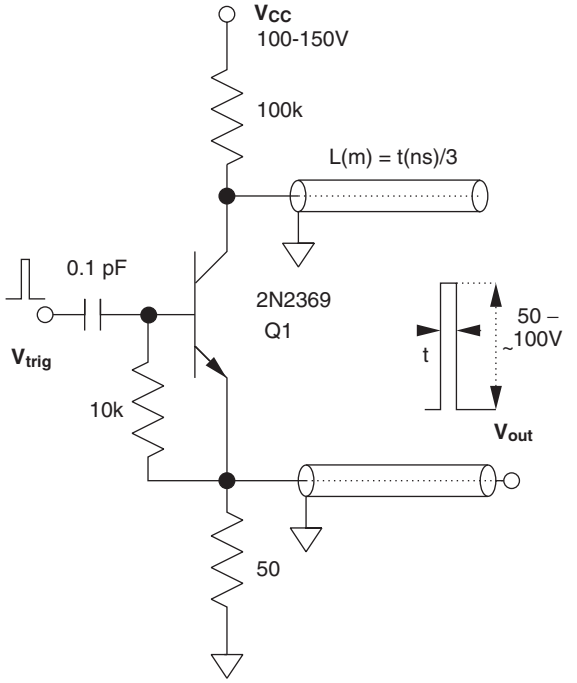

Figure 15.27. Avalanche transistor flat-top pulse generator. When Q1 avalanches, the upper transmission line discharges a constant current into the lower one. With a sufficiently light load, the open-circuit reflection doubles the voltage at the far end of the lower line. You can also play other transmission-line games, such as using a balun to get higher output voltage, or increasing the emitter resistor to get higher peak voltage at the expense of worse reflections.

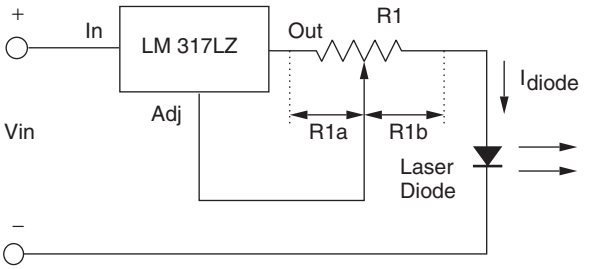

Figure 15.28. This current driver will blow up the diode laser when the wiper of $R1$ opens momentarily.

Example 15.2: How to Kill a Diode Laser with a Pot. At the risk of getting somewhat ahead of ourselves, consider Figure 15.28. The LM317LZ is a cheap but good voltage regulator. It keeps a constant 1.25 V drop between its output and adjustment terminals. Putting a voltage divider between the output and ground, with the adjustment pin connected at the middle, makes a positive voltage regulator adjustable from 1.25 V up. It works quite well for this, since the adjustment pin sources a small and reasonably constant 50 μA. Here it is used as a not-too-great constant current source (far above shot

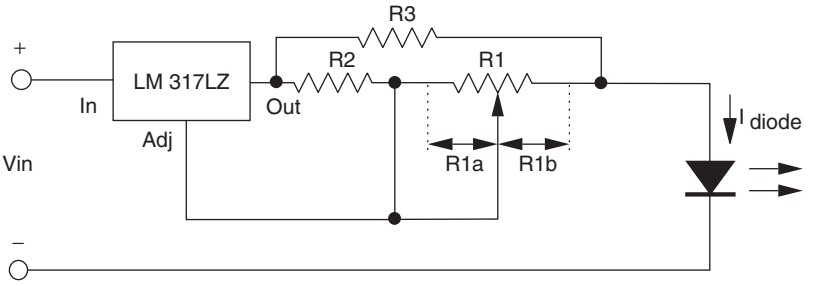

Figure 15.29. Provided that $R2$ is chosen correctly, this diode will survive failure of $R1$. $R3$ is there to handle most of the current, rendering $R1$ more reliable as well.

noise); it keeps a constant 1.25 V across R_{1a}, thus keeping the diode current constant at $I_{\text{diode}} = 1.25\text{V}/R_{1a}$. (This circuit may have problems on startup, but we'll ignore them for now.) The circuit works fine unless the wiper of R_1 opens up momentarily (100 μs is long enough). If that happens, the adjustment pin can't source its 50 μA, and the output rises to $V_{\text{in}} - 2$ V or so, destroying the diode laser instantly.

A better way to do this is shown in Figure 15.29. Here if R_1's wiper opens, the diode current rises only to a maximum value set by R_2. R_3 is there to take most of the current, which otherwise would come through R_1's wiper. Another advantage of this scheme is that you can't blow up the diode by turning R_1 too far. Remember: pots do get noisy eventually, and you don't want your instrument to die on the first blip.

15.15.1 Hot Plugging

The most common cause of death for prototypes is losing ground or supply connections. Lethal scenarios depend on how many supplies of each polarity you have, and whether there are any loads connected across the supplies (e.g., heaters or motors). You have to be systematic about what can happen, by checking each possible order in which the supply and ground connections can open. The actual fatal fault is always either transient overvoltage or supply reversal. If the person doing the plugging is ham-handed about it, it is also possible to short a supply pin to somewhere and blow up the on-card regulator or the somewhere.

It is rare and usually stupid to connect a load between supplies of the same polarity, for example, a load running on the 7 V between +5 and +12. If the load is a heavy one, then if whatever uses most of the +5 gets unplugged, the +5 V line will get dragged away upwards and probably blow something up. It is less rare and more intelligent to connect a load (e.g., a motor or heater) between positive and negative supplies, and this is also easier to protect against. When one supply is lost, the other one will try to drag it through ground and beyond, which is bound to kill something. Fix this with Schottky rectifiers between each supply lead and ground (reverse biased, of course). That way they can't be dragged more than about 0.4 V in the wrong polarity.

If ground is lost first, the usual failure mode is that one supply draws much more current than the other (positive, usually), and so the "ground" lead gets dragged rapidly toward the positive rail as the positive supply's bypasses get drained down. This puts the whole supply range (30 V for a ±15 V system) across the negative rail, which is

often enough to blow something up. A transient-suppressing Zener diode across the lower current supply fixes this. The Zener should be able to take the full supply current of the opposite supply, plus discharge all its bypass capacitors. ON Semiconductor sells good ones cheaply—check out the P6KExxxx series.

Otherwise, the main failure modes are that the loss of ground before the signal leads causes big voltage excursions on the signal connections, sometimes blowing up logic parts. Low impedance signal connections (i.e., those without >1 kΩ resistors in series with them) should have some sort of protection, such as diodes to the power supplies.

Plugging in has some dangers as well. Most of them are the same as for unplugging, but there are one or two unique ones. Boards are often plugged in slightly cocked, so that one end of the connector mates a little before the other, leading to the supplies and signals coming up in different orders and at different times. Some voltage regulators don't start under weird loads, especially foldback limiters and 7800 series devices, which refuse to start up if their inputs are brought below ground. When the supply contact is suddenly made, all the bypass caps charge up very rapidly, requiring a lot of current. If the supply contact doesn't get made soon enough, this current may wind up coming through the signal leads. A few 100 Ω resistor arrays, wired in series with your signal inputs and outputs, can save headaches here. This also protects inputs and outputs from damage when connected but unpowered, which happens a lot in prototypes and is especially nasty because they're so precious and usually harder to repair. If you're using card edge connectors, it's possible to arrange the pads so that the ground connections are made first, which helps a lot. On a multipin connector (e.g., DIN Eurocard), you can put grounds at both ends of the connector instead. This isn't as good, because small amounts of crud on the pins can cause delayed contact, but it helps.

15.15.2 It Works Once, How Do I Make It Work Many Times?

This is a really generic problem, and a major source of difficulty in putting a design into production. There are two basic causes: first, it is a corollary of Murphy's law that all prototypes turn out to contain the normally unobtainable perfect component (at least one, sometimes several). Lenses with no decentering, transistors with infinite beta, op amps with no offset current, AR coatings that really have zero reflectance at your wavelength, the list goes on. Prototypes invariably seem to depend critically on these perfect parts, leading to pain in replication. Second, most R&D people (including the author) have a tendency to work on the irritating problems until they're no longer irritating, then stop. This is fine for graduate study, but a disaster in a technology development or transfer situation. The problem is that faults which do not irritate the designer may render the instrument unusable by others. This leads to difficulty handing the system off, either to customers or to a product division, and so to heartburn on all sides. Early customer involvement is a good way to avoid this problem. Complex systems that really work are invariably found to have developed out of a close interaction between one or a few gifted designers and a small, highly involved community of users.

The users involved should include a few querulous curmudgeons, too, but it is best to pick ones who are unlikely to be loud and public in their criticism. If the users are given work to do, their level of commitment increases, because they gain a feeling of ownership. That way, they'll be more likely to stick by you when the project gets in serious trouble for a while (most successful projects do this at least twice).

15.15.3 Center Your Design

The perfect component problem is best handled by simulation if possible. The author is very sceptical of the uncritical use of simulations, but one place they are frequently indispensable is in making sure that the design is *centered*—that its components are chosen in such a way that normal component variations are allowable on both sides of the nominal design. It is common to design a circuit or instrument in such a way that (for example) a 5% shorter focal length is a disaster, leading to an inability to focus at infinity, while a 5% longer one is no problem, leading only to a barely noticeable difference in the closest focus distance. This will hurt when your lens supplier ships a bunch of slightly short lenses, so if the specified tolerance is ±5%, the nominal focal length should be made slightly longer to reduce the danger. Simulation can help in avoiding this sort of trouble.

In a mixed-technology system, getting simulations right is considerably more difficult than it is in lens design or digital circuit design. Because of the labor involved, it should be used judiciously.