CHAPTER 11

Designing Electro-Optical Systems

Anything worth doing is worth doing well.

—English proverb

Anything not worth doing is not worth doing well.

—Anonymous

Anything worth doing is worth doing badly.

—G. K. Chesterton†.

11.1 INTRODUCTION

Designing an instrument isn't just figuring out what should go where, although that's part of it. An even more important design task is to get the concept and motivation right—to think about the work in a way that maximizes your chances of success. This chapter starts off with the thinking, and then deals with the nuts and bolts. (You should approach it the same way.) By the end, you should have a framework for understanding how your instrument fits with your style, goals, and resources, and an orientation to how to proceed with the detailed design. Systems are so diverse that striving for complete generality would make it impossible to say anything concrete; thus we'll work through some examples instead.

11.2 DO YOU REALLY WANT TO DO THIS?

Before starting, do understand that every original or groundbreaking development project will nearly fail, and most of them nearly fail several times. This can be profoundly discouraging if you're not prepared for it; nonetheless, if the measurement physics is right and the technical risks have been dealt with appropriately, the project will usually

†"What's wrong with the world," P + 4, Ch. 14. Elsewhere we says, "There are some things we expect a man to do for himself, even if he does them badly." *Orthodoxy*, http://www.gutenberg.org/files/16769/16769.txt.

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

TABLE 11.1. Some Common Ways Projects Fail

Design Methodology Errors	**Designing the Wrong Thing**
Not making a photon budget	Pyramid building
Ignoring constraints (e.g., cost, size, use)	Creeping featurism
Runaway schedule and budget optimism	Runaway complexity
Not reducing technical risk fast enough	
Nobody having a clear picture of the whole	
Execution Failures	**Moral Hazards**
Not building what you designed	Fickle customers
Not verifying the design as you go	Fickle managers
System integration failure	Ignoring problems
Running out of time	Not confessing when in trouble
Running out of money	Not keeping everyone honest
	Loss of enthusiasm
	Failure of nerve

succeed. Overall, the success rate for well-conceived projects seems to be around 80%, and the failures are usually due to discouragement or delays (Table 11.1 has other ways to fail, for ready reference). You will spend time staring at the ceiling at 1 a.m., so make sure you have some answers to those wee-hours questions: “Why am I doing this?” “How sure am I that the physics is right?” “What am I missing?” “Does the prototype really match the design?” And especially, “Why doesn’t it work?”

11.2.1 Collegiality

These questions are much harder to answer if you are working alone. The principal reason is that you’ve been over the ground many, many times and have cut a rut so deep that you can’t see over the edge to where the new ideas lie. A secondary one is that personal and private failure is staring you in the face. Discouragement and fear sap creativity and enthusiasm faster than anything else, leading to little progress, leading to more discouragement and fear, until someone puts you out of your misery. This danger looks trivial to anyone who hasn’t faced it. The wisdom here is: don’t attempt a really difficult instrument project alone. That doesn’t mean you can’t be an engineering team of one, but it does mean that you need colleagues around you who are generous with their time and are willing to work to help you succeed, not just make encouraging noises. The author once had to take up an entire month of a friend’s time getting a complicated, ultrasensitive interferometric sensor to work properly, nailing down a decibel here, a partial vignetting there, and a half-wave of coma in another place. We worked elbow to elbow all month long, and although he didn’t know the system all that well at the beginning, that project would have been doomed without him.[†]

Once again: collegiality is an absolute psychological necessity for almost everyone, and the lack of it leads to severe stress and loss of enthusiasm even for good projects. If you don’t have it, don’t embark on a long and difficult project—stick to stuff with

[†]The colleague was Dr. Marc Taubenblatt, and the system was the ISICL sensor of Example 1.12.

shorter term payoffs, either data you can put on your wall, things you can sell, or a talk you can give.

11.2.2 Collegiality and Team Productivity

A person who is encouraged by the admiration and interest of colleagues and management will come in on weekends just to see how the experimental run is going, and will think about his work whenever he's not thinking about something else; his whole effort will be bent to it. It is easy to underestimate this, but the author's experience is that his productivity can easily change by a factor of 3 depending on collegiality alone.

Remember this: design teams are a bit like roses. Good ones can produce amazing results when they're properly tended—pruned, sure, but also watered, cared for, and given a place in the sunlight. Neglected ones get bedraggled very quickly. If you have people working for you, remember that they're the rosebushes and you're the gardener. Your glory comes from theirs.

Your enthusiasm and confidence are your most precious resources.

11.2.3 Choosing Projects

The 6 stages of a project:

1. Enthusiasm,
2. Disillusionment,
3. Panic,
4. Search For The Guilty,
5. Punishment Of The Innocent,
6. Praise And Honours For The Non-Participants.

—Anonymous

Is this project a smart risk? Will I be motivated by enthusiasm or by fear? Fear is a bit like amphetamines—you can stay up all night working, all right, but your integrated output will be lower, and your enjoyment will ultimately vanish. What you should look for in a project is fun, good leadership, enthusiasm, and the feeling that what you're doing is not only valuable but *seen* to be valuable, seen to be a smart risk—that is, if it fails due to bad luck, you won't suffer too much. Having management commitment that will survive failure is both rare and precious. Think about this soberly before starting a project.

11.2.4 Procedural Advice

Take Play Seriously. It is commonly said that people do their most original work before the age of 30, and this is often so. It's commonly held that the reason for this is rapid aging, but this is not so; like composers, the best instrument designers continue to improve through late middle age. It's just that most over-30 people have had their creativity flogged out of them, which is a huge waste. There's not much you and I can do about this except refuse flatly to have it happen in our vicinity, but it's worth at least naming the antidote: *play*. Play is not a waste of time that should have been productively used; it's where all the creative ideas come from, and a world of furious crank-turning can't replace it. The seed corn sitting in the barn appears wastefully unproductive, but without it, we won't eat next year. The author learned what he knows about instrument building by playing, not by being strapped to a schedule like a silent-movie heroine on her log heading for the buzz-saw. Working in the real world, we have to make real-world moves, but driving ourselves so hard that we lose our sense of fun is a disaster. Overworked engineers are the ones who burn out and get obsolete. Students have time to play, and that's why they learn fast and dream large—it isn't youth so much as *time*. Spend at least one day a week on play and head maintenance—reading journals, messing around on side projects, going for walks to think about measurement ideas, inventing things on the white board with a colleague. You will get more done and not less, and you won't burn out.

Take Counsel of the Devil. One very important step is to spend time in your customer's and (especially) your competitor's shoes. Most engineering groups have a certain hothouse mutual admiration, which is healthy and valuable, but leads us to rate the competitor's ingenuity too low and the customer's discernment too high. Have a Blue and Red team match, where you pretend to be your competitor, and try to find defects in your own design or ways to avoid your patents. This can be bruising, but it's a lot more pleasant than having your competitor do it for real. You learn a lot, and besides, you wind up with better patents and better products.

> For a successful technology, reality must take precedence over public relations, for nature cannot be fooled.
>
> —Richard P. Feynman†

Resist Overpromising. Managers in instrument companies know that schedules are usually too optimistic and have some idea of how difficult a given project actually is. On the other hand, if you're building custom instruments for some other type of organization (e.g., a large computer company), your customer is probably a lot less technical than you are and is almost certainly not an instrument designer. His expectations depend not only on what you tell him, but on what he infers from your words and your manner, as well as what he wants to believe. You have to manage his perceptions of how long it will take and what you'll deliver. Of course, your estimates of time and difficulty will be as realistic as you can make them, but nevertheless, the temptation to sound too enthusiastic is almost irresistible; if you didn't think your gizmo was wonderful you wouldn't be building it. You may well create the fatal impression in your customer's mind that the work is practically done when you get your first data plot.

This can lead to a paradox—you succeeded brilliantly, but completely disappointed your customer, because (a) he thought it was easy, and (b) you didn't leave enough slack, or look as though you were struggling hard enough. Make sure you keep him up-to-date on all your technical work, and what you think of your rate of progress.

Keep Some System Margin in Your Back Pocket. If you're building a product, you're managing several things all at once: the project, your career, and the expectations of your customers. Your design spec, be it a photon budget, a link budget, or a resolution

†Richard P. Feynman, Appendix F—Personal observations on the reliability of the Shuttle. *Report of the Presidential Commission on the Space Shuttle Challenger Accident*.

specification, is your best shot at what you ought to be able to do. That means that any surprises you encounter along the way are going to reduce performance, not increase it, which is quite natural—we aim at the highest possible point, and never quite reach it. People not directly involved in the project will normally interpret the system design as being the guaranteed specification, and will interpret any performance reduction as failure—*your* failure. This is extremely painful. Thus it is very important to keep two sets of books, and don't show the customer the real one. This is not dishonest, as in money laundering and tax evasion, but is a recognition that you and your customer have different understandings of how a system design functions, *and that yours is right and his is wrong*. He will insist on regarding it as a firm promise, whereas to you, it's an image of perfection. Hide at least 3 dB worth of buried treasure in your public version, so that when the inevitable snags occur, you have some margin to make up for it. If there's any left when you're done, use it to make yourself look like a hero. You will be.

Have People to Cover Your Back. The near-certainty of near-failure means that we all need defenders. In the bad patches, there will be people who stand back and throw rocks, saying that the project is doomed and should be killed off. Some organizations have a worse case of this than others, but the remedy is the same: don't poke the alligators, and cultivate allies by spreading the credit around, taking advice, helping out, eating lunch together, and letting them have some ownership of the system—that is, tell everybody how much the project owes to everybody, and *make it true*. If it's theirs too, they will dig in and defend it during the bad times. The most important people in this category are your management and your customer.

Confess When Your Project Is on the Skids. If you expect people to defend you when you're in difficulty, it is only fair that you tell them when it happens. Follow the airplane rule: "When lost, climb and confess." Expect the same from your colleagues, and be willing to press a bit; since designers are normally very eager to discuss their latest idea, a colleague who is evasive is very likely to be hiding a failure.

Define What Constitutes Success. Having thought about the consequences of failure, spend a little time on what is likely to happen if you succeed. It's important to think out what you want to happen then, and to reach agreement with everyone involved as to what success looks like. For a student, success usually means getting enough data to write a couple of papers and graduate, which requires a lot less engineering than producing a design to be used by thousands of people.

If you're a product designer, you might intend to go back to the lab for the next brilliant scheme, but wind up wedded to this one for your whole career. Being lead designer of a really new product in a small company will oftentimes lead to this. On the other hand, a successful development project in an area that is not your organization's core business is liable to be sold off, with you attached. If you're lucky, being sold off may make you rich, but usually it's more like being sold down the river—the job and the salary stay the same, and the pressure and insecurity get much worse. Is going that way okay with you? If not, how will you disengage, to go on to your next project? You need to be fast on your feet; leave the project about six months before the sale to avoid being caught in the undertow.

If the project is likely to remain with the organization, how will the system be manufactured? Sometimes the designer moves to manufacturing to become product manager for the system, and may not get to design another one for some time (if ever). Other times, the design is transferred to the manufacturing department, which is responsible for replicating it. This is a famous source of friction and finger pointing: the production yield is too low—whose fault is it?

Build bridges to the manufacturing people early, and make sure they have some ownership by bringing them into the design process. Explaining how it works and what it's for may seem elementary and obvious, but they probably have no idea, and most of us like to be treated like colleagues instead of cogs in the mechanism. You'll get a lot less grief from people who feel part-ownership of the system, and the transfer will work better, too. This of course presupposes that you know who will be manufacturing it.

Know Your Organization. Large companies tend to churn managers, so the expected life of your project may extend across two or even three different managers at all levels. This is because reorganizing gives the illusion of progress and is so much easier than attacking the actual problems. A new manager often feels pressure to make his mark quickly—so like a lion taking over a pride, the first thing he's liable to do is eat all the cubs. You're pretty safe if you're just incrementing an existing product, but watch out if you're doing something the new guy doesn't understand and isn't excited about. In that case, he doesn't care about success but risks getting blamed for failure, so you see the temptation. This is one big reason that large companies have trouble producing really new things over long periods; the exceptions, such as 3M, Corning, and IBM, have cultures that encourage patience.

To keep your project alive, make sure you give lots of sales pitches to managers, even ones slightly outside your chain of command, to build their enthusiasm and give them something to use to convince the new guy to let you do what you're doing. Make sure that when you do get a new manager, you make time to sit down with him to explain why what you're doing is exciting and good for the organization—managers, like most other people, want to do the right thing if it isn't too risky, so concentrate on that. Don't get so concerned with management-tending that you lose your balance, though. Management jobs attract people who are comfortable scheduling their time to the nearest nanosecond, and who equate busyness with productivity—an attitude that leads to a clean desk and zero creativity, which won't help you. Sometimes you just have to take your lumps and move on.

Don't Build a Pyramid. Everyone seems to build one pyramid per career. A pyramid is an ambitious system that one person really cares about and that winds up working well, but then just sits in the desert because nobody else cares the same way. This happens usually just after leaving graduate school.

There are two kinds of pyramids: yours and your boss's. Sometimes you get an idea into your head and sell it hard. Other times, you fall for a sales pitch about how you're the key guy in the department, how this just *has* to get done, how it will result in showers of glory, that you're the only one who can possibly do it, how you of course can't let the group down, and so on.

It's exactly the same sales pitch that officers and NCOs give infantrymen before a battle. Being unlucky in the infantry is a lot worse than in engineering, but the soldier's odds are usually better. Don't fall for a romantic but hopelessly understaffed project, no matter how big a hotshot you are; and if it's your idea, make sure you think it out very carefully. Remember, if you're in love with it, own it, and argue for it, nobody will want to tell you it's a waste of time.

Understand the Pecking Order. Without instruments people, most of science would be impossible; a major fraction of what appears to be progress in basic science is really the progress of the instrument-building art. In our sublunary world, credit is seldom bestowed really accurately, and most of us are quite used to that. Besides, gizmo building is such fun that it's worth trading off quite a bit of glory for. There are lots of interesting problems in the world, though, so you might as well work where you're appreciated. In biomedical instruments, for instance, if you aren't an MD you might as well be part of the furniture. Astronomical instruments folk are also treated like spare telescope parts. Working for a small company where your expertise is similar to that of the August Founder is usually uncomfortable unless you're the Anointed Protegé, which has its own troubles. In general, status-conscious places are miserable for everyone, and the more, the worse.

Don't Fight "Good Enough." A common error, related to pyramid building, is to build a new alternative to an existing technology that is already good enough and is in wide use. A large-scale recent example is magneto-optical storage. It was introduced in competition with magnetic storage, pushed hard for 10 years, and lost (at least for fixed-disc applications), because its then-great advantage in storage density was not good enough to overcome its bulky, massive read/write head designs, which forced the platters to be stacked too far apart and slowed down the track-to-track seek times. Magnetic storage didn't stand still, either; better media, the magnetoresistive (MR, and then giant-MR) read head, extremely small head–disc gaps (a few hundred angstroms), and improvements in servo design allowed track widths to shrink and linear recording densities to increase enormously in that 10 years. (Magnetic is now far denser.)

A new technology in a busy field needs a sheltered niche in which to grow and mature, and (what is usually forgotten) for users to become familiar with it. Mainstream technologies have huge development efforts behind them, consisting of people who have years of experience with the technology and who are backed with lots of money. Even if your idea is 10 times better, you're going to get steamrollered unless your development effort is comparable in scale.

Agree on Specifications Before Starting. Specifications are often written by people who have no idea what they're doing, but think that they have. The first task on the list should always be "specification resolution," with a customer sign-off required. That way, when they discover that the specification isn't exactly what they want, the responsibility (and financial liability) clearly lies with the customer. This is especially important when you're an independent contractor.

11.3 VERY BASIC MARKETING

11.3.1 Who or What Is Your Customer?

Before picking up a single lens, make sure you know who your customer is. For most of us, there is more than one: first, the managers, bean counters, and contract monitors we have to convince, and second, the real end user of the system.

For each class of customer, we need to know the answer to a few rude questions (it's possible to fish for the answers politely, fortunately). What do they want? Do they know what they need? Experience suggests that they often don't, but always think they do;

they are the ones with the checkbooks, though, and so can't just be ignored even if they are provably wrong. Unfortunately, they often assert their power by demanding long lists of "checklist features," which they'll never use but which your competition has. Will your market support the amount of engineering required?

11.3.2 Making a Business Case: Internal

A grubby necessity of the instrument-building trade is having to convince people with money that your latest brainstorm deserves to be made into a product, will save the company a big chunk of money, or (a much harder sell) has enormous prestige value. All trades are like that at some level, but in other businesses you're usually dealing with somebody who has some understanding of what you're doing. Unless your business manager is a technical person of some talent, he probably won't know how your instrument works, and so will have only your say-so (and perhaps someone else's evaluation) that it does. This is naturally anxiety producing for the manager, so you'll need to be prepared with a truly compelling business case. It must at least contain the following elements: a detailed design concept (what are you building?), marketing information (who's the customer, what's it worth to him, how much of that can you recover, and over what time span?), a fairly detailed plan for how to do it, including estimates of time and resources required, a levelheaded assessment of the technical and business risks (including anticipated work-arounds and how you'll go about reducing the risk quickly), and an estimate of the opportunity cost (i.e., the value of what you would otherwise do). Assuming you get taken seriously, you'll need to be able to answer questions on all these things. Doing a bit of thinking and research about how the market and the competition is likely to develop between now and when you're done will help too.

The development plan is especially fraught, because you'll often wind up being forced to invent on a schedule—a notoriously difficult task. It's worth having a couple of good ideas tested and confirmed before you make the business case, which will require stealing a bit of time from your current project to work on them. Don't hesitate to do this, because it makes you much more valuable to your organization as well as to yourself. It also helps you to avoid overpromising (see above).

Consider keeping a couple of good unannounced results (including pretty pictures) in your desk drawer, ready to pull them out on some rainy day to encourage yourself and others. There are lots of times in development projects (especially ones with lots of electronics and software) where you're nearly finished before you have much to demonstrate. It's good to have something new to show when you make yet another sales pitch.

Projects usually run in two stages: a feasibility study followed—if all goes well—by a full scale development effort. There's a sales pitch before each stage, for which you'll need some solid support: a few plots for the first sales pitch, and a working demo for the second.

11.3.3 Making a Business Case: External

Your customer often won't know what he needs, but he'll probably know what his problem is, and how much it's costing him. Thus the easiest kind of case to make goes something like this: "Here's a pile of damaged items P, which cost Y dollars to make, and could have been sold for Z dollars if they hadn't been damaged by Q (contamination, frost, broken bottles...). This happens about V times per year, and could have been avoided by using my instrument, which costs W dollars installed. Thus the instrument will pay for itself in $t = W/(VZ)$ years (where $t < 2$)." This goes down very well, provided that you really can point to the pile of damaged goods, and that you're pitching to the person who owns the pile.

It is important in making this case to ask yourself what else he could do, if he didn't use your instrument. If something simpler can do nearly as good a job, it might make sense to do that. In a bottling plant, an audio system listening for the sound of breaking glass (lots of high frequencies) might do the same job as a machine vision system, at a 10× lower cost. You'd have to make your case on the basis of quality control, which is much harder (though not impossible).

The "What else would he do?" question also means that the payback period has to be shorter for instruments than for most other investments, because there's an opportunity cost to not waiting for a possible cheaper solution. That 2 year number is about right—much longer than that and the sale gets much harder; much shorter, and you're not getting enough of the benefit, and should charge more money.

11.3.4 Figuring the Price

The usual rule of thumb is that the selling price of an instrument is about three times the parts cost, and more if it's going to need a lot of after-sales support. If this sounds high to you, it isn't really: the cost of your gizmos includes not only the parts and the facilities needed, but the people who build them, the people who fix them, the people who look after the payroll, and (last but not least) the people in Italian shoes, eating expensive lunches, who persuade people to buy them. If you can't sell it for 3× the parts cost, redesign it to use less hardware; be ready to shed the less important features. (Note that this is the real cost of the parts when you buy production volumes, not the onesie-twosie price.)

11.3.5 Budget for Market Creation

That factor of 3 is for an instrument that people already know how to use. A really new instrument often requires educating customers about how it can help them. This market-creation activity is slow and sometimes painful. Don't underestimate how difficult it is; if you are aware that it'll take a while, and don't disguise the fact, you'll have a much easier time of it when no tidal wave of orders arrives.

One piece of marketing wisdom: kind words are much easier to get than purchase orders. If your prospective customers are saying they'd like to buy your gizmo, make sure you know exactly what practical problem it will help them solve, and whether there's a genuine business case to be made for it. They'll have to make that case before the PO gets generated.

11.3.6 Budget for After-Sales Support

The education task doesn't end when the sale is made. If you're building a new class of instrument, one where the customer probably hasn't used one before, you're going to get a lot of phone calls afterwards. You'll have to pay people to answer them—good people, not phone fodder—so factor that into the price. If you're in the in situ instruments business, the cost of sale and after-sales support goes way up because of the complexity of getting your instrument into your customer's system.

11.4 CLASSES OF MEASUREMENT

What exactly do you want to measure? Do you have a clear idea of your measurement physics, or are you burrowing around in spectral, video, or time series data to find something that correlates with what you're trying to measure? Or just looking for pretty pictures? This is a somewhat rude question to ask, but the answer really determines what your design strategy should be.

11.4.1 Know Your Measurement Physics

In deciding on the feasibility of a measurement, there must be a balance between theory and experiment. Ideally, theory shows that the measurement is possible, and how big the signal should be; experiment shows whether the theory is right (often leading to improved theories) and that the apparatus can be built. A combination of theory and data, demonstrating good agreement, shows that the measurement is at least possible; the actual instrument design is less important, because there are usually a few different good ways to go about it. How well your measurement idea lives up to that ideal is the principal determinant of your strategy.

11.4.2 Crunchy Measurements

Measurements come in three textures: crunchy, squishy, and in between. Crunchy measurements are those whose measurement physics is well understood, and where the parameter being measured is identical with what you want to know, e.g., optical pyrometry for the temperature of a closed furnace, where you're measuring cavity radiation, whose properties are known in advance, and where calibration on one system is adequate for making measurements on another.

Crunchy measurements are always the best. The better you understand the measurement physics, the more robust you can make the measurement with a given amount of testing.

11.4.3 In-Between Measurements

There are lots of in-between measurements, where we have at least an arm-waving idea of why it works, and can logically connect the measurement data with what we want to know. An example of this is a fiber-bundle spectrometer looking at thin film interference fringes to measure film thickness in semiconductors, when the structure underlying the film is poorly controlled. These measurements are based on correlations between some feature of the data and the parameter you want to measure, and are as valid as your knowledge of the origin of the correlation: the better you know it, the better your measurement and the better off you are.

Making in-between measurements robust means a significant amount of experimental work in realistic conditions to establish whether they are really telling you what you need to know, and if so, how to extract the information you care about from the more or less closely related experimental data. This must be followed up by looking at a whole bunch of pathological special cases, and putting in tweaks and hacks to handle them. Measurements without much theory tend to break the first time an unanticipated situation arrives, so you have to do a lot of anticipating. An example is downstream particle counting, where an optical sensor in the roughing line† of a reactive-ion etch tool for semiconductors (say) looks for particles in the turbo pump exhaust. You care about the particles on the silicon wafer, not down the pump, but there may be a correlation you can use, since we expect a dirty chamber to have dirtier exhaust.

11.4.4 Squishy Measurements

Squishy measurements are based on a correlation between something you can measure and something you care about, where you don't really know why the correlation exists, for example, multispectral reflectance measurements of agricultural produce to indicate freshness. This is a tough business, for two reasons: first, the correlation is often unstable with time, and second, there is an enormous amount of *nuisance data*, that is, complicated changes in the background signal, which you have to be sure of rejecting (see *ground truth* in Section 11.7). Fresh tomatoes look pretty different from fresh lettuce, and Iceberg looks different from Romaine; wilting due to drying out looks different from mold, which looks different from wilting due to frost.‡

A sufficiently squishy measurement cannot be made reliable by any amount of testing and correlation whatsoever: as Deming said, "you can't test quality into a product," even if your product is a measurement technique.

Keep your measurements crunchy if you can, and if you can't, expect a lot of 3 a.m. phone calls. The moral of the story is: *good data and good theory matter more than easy data gathering*.

11.4.5 Pretty Pictures

People are busy, and most of us are intellectually lazy at least part of the time, including grant committees, managers, and thesis advisors. Almost every designer has to sell his instrument to someone at some stage. Pretty pictures help a lot—a measurement whose data generates no pretty pictures will get less attention and be a harder sell than a more pedestrian one with dazzling output. If you share the popular disdain for pretty pictures measurements, think again.

This far into this book, no one should mistake the meaning of this: it's the data that matter, the pictures are for the sales pitch.§ The pitch is important, even for a graduate student; if your thesis talk has data whose significance leaps off the screen, your defense will be a lot easier, as will your job search afterwards. For others, you know that the opportunity to build instruments for a living makes frequent sales pitches (to customers and your own management) a necessity. Have pretty pictures whenever possible, and use some ingenuity on them.

†A high vacuum system typically has a *roughing* pump (exhausting to atmosphere) to get the chamber down to the millitorr range, and then a high vacuum pump such as a diffusion or turbo pump for lower pressures. The two are arranged in series with the roughing line in between.

‡This isn't a silly example: the author was once approached with a serious proposal to use multispectral imaging of this sort for quality control in fast-food hamburger restaurants.

§Sometimes the pictures help you spot patterns in the data, but those aren't just pretty pictures.

11.4.6 Pretty Pictures Measurements

As all technical people know, the credibility of measurement data goes down as the number of colors on the chart goes up. Real data should look as different from marketing presentations as possible. Nevertheless, there is a legitimate class of measurements whose output is exclusively pretty pictures.

Pretty pictures measurements include using UV imaging to reveal counterfeit documents, proof-of-concept experiments to show that your measurement idea can actually work (e.g., the first scanning tunneling microscope picture), or writing the IBM logo in novel ways in order to get free advertising on TV. If they use existing equipment, they can be thrown together quickly, and usually should be. Good physical measurements leave the pretty picture stage behind very rapidly.

In measurements whose frank aim is pretty pictures, it is appropriate to use somewhat stronger postprocessing than you would if you were after quantitative data. For example, grey scale pictures have limited dynamic range; in turning scanning measurements into a grey scale image, we often *plane subtract*: that is, subtract out a function $f = ax + by$ for some a, b so as to flatten out the background and avoid wasting dynamic range on artifacts. In a pure pretty pictures measurement, you might take out a quadratic term in the background as well, to make it come out a visually pleasing uniform dark grey, or use false color to make it stand out better.

11.5 TECHNICAL TASTE

> Genius is 1% inspiration and 99% perspiration.
>
> —Thomas A. Edison

Edison correctly identified the *inspiriation/perspiration ratio* (IPR) as a key metric of project quality. The fact that his quoted ratio is only 1% shows that he was doing it completely wrong, of course.[†] Technical taste is a feeling for the right way to do things that develops with experience and with being steeped in a really competent and imaginative technical culture. Good taste helps you maximize the IPR; that is, the idea is to use neat ideas to minimize grunt work. Keep the IPR as high as possible, by all means, but do make sure that the neat ideas are neat because they work really well, not because they're unusual or just cool.

If you've got a tingling feeling in the back of your mouth, suggesting that there has to be an easier way, *pay attention to it*. It is one of the few indications you will get that your latest gizmo still has a lot of bric-a-brac that can be removed. There is no substitute for the process, but a few representative points can be made.

Know What Your System Should Look Like. The ideal instrument for fixed use is the size of a paperback book, with an Ethernet jack and an on-board web server for control and display. For hand-held use, it should be the size of a deck of

[†]Nikola Tesla worked for Edison for a while and was amused and frustrated at his boss's insistence on doing many, many trials without theory to guide them. Edison had excellent intuition, was a marvel of energy, worked in relatively simple technology, and eventually had a whole lot of lab assistants. Not many of us are in that position.

cards and communicate via USB or possibly an IR link. Some instruments need to be bigger, but large size costs money, and many customers won't want some big featureless rack cluttering up the place. Decent quality packaging and cabling are obscenely expensive.

Example 11.1: Cost-Reducing an Angular Displacement Sensor. A device for sensing the position of an object in 3D originally consisted of several IR LEDs mounted close together, illuminating the object, with a lens plus a two-axis lateral effect cell to sense its angular position, as shown in Figure 11.1 at the top. The total photocurrent gave its radial position. The photocurrent was dominated by ambient light, and the cost of the sensor was dominated by the lateral effect cell ($100). A quad cell wouldn't work since the angular sensitivity would depend strongly on the state of focus.

Some days after the tingling feeling started, the idea of a sundial came to mind, leading to the shadow mask detector design of Figure 3.3. Eventually (bottom of Figure 11.1), the system consisted of a small PC board, with three pairs of $1 photodiodes on it, detected differentially. A slot cut in the lid of the plastic box provided the shadow mask, and a Velostat (carbon-loaded plastic foam used for shipping ICs) gasket compressed between the board and the lid prevented the source from illuminating the detectors. The signal level and reliability increased, and the manufacturing cost went down by six

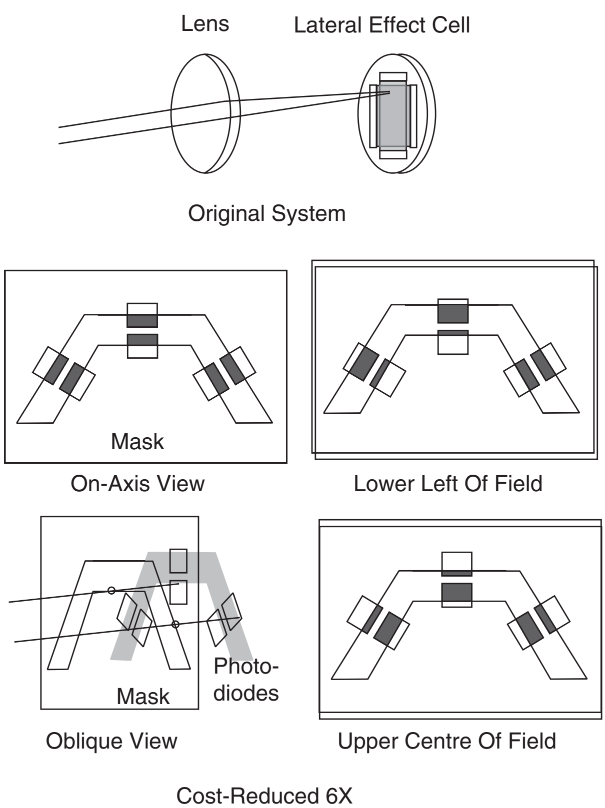

Figure 11.1. Angular displacement sensors.

times (there's more on this in Example 3.3). (Neither system is limited by diffraction—why?)

Don't Measure Anything You Don't Care About. One sign of a good measurement idea is that you're only measuring what you care about, and don't need to correct for a whole lot of large extraneous effects (keeping track of the temperature is okay). Most eye trackers used with external displays (i.e., not helmet-mounted) measure the angle of gaze with respect to the head, using a glint of light off the cornea. In order to determine where on the screen you're looking, they have to measure the position and orientation of your head in six axes, to high accuracy, as well as the gaze angle. This makes things difficult and expensive. Having to measure eight things when you care about only two is a losing proposition.

Avoid Obese Postprocessing. The trade-off between hardware and software depends on the task and the relative costs, but also on the expertise and inclinations of the people involved. In our software-heavy era, some people feel a strong temptation to try to use lots of postprocessing to squeeze a good measurement out of an inadequate sensor, which is an invitation to failure. An example is trying to use a video camera on top of a computer monitor to do accurate eye tracking, by piling on frame averagers, median filters, feature extractors, pupil-centroid-finders, saccadic-motion-blur-eliminators, and so on. Your measurement gets squishier and squishier, and so less and less reliable, for no good reason. As is repeated elsewhere, *postprocessing is not a substitute for good data*.

Have a Fallback Position. In building electro-optical instruments of any complexity, it's important to have a backup plan in case each new item fails to work. A typical project will get in trouble between two and five times before eventually succeeding.

It isn't necessary to have a recovery plan if all of your new bits fail to work; we assume that you've reduced the technical risk before going all out on the project in the first place, so that the scariest new technology is at least possible. You can't recover from total technical failure, but usually you can think of a work-around for each piece ahead of time. Pieces that don't admit work-arounds have to be nailed hard, fast. That's where potential show-stoppers lie.

Wear a Belt and Suspenders. For high performance applications, don't rely on just one layer of defense; you won't have thought of everything that can go wrong, so add as many useful extra safeguards and sanity checks as you can without overburdening the system with electro-optical bureaucracy.

The ISICL sensor of Example 1.12 uses 10 layers of protection against laser noise and time-varying speckles due to back wall and window reflections: a very low noise diode laser driver (to make the scattered light quiet), millikelvin temperature stabilization (to avoid mode hops), automatic laser current adjustment (ditto), coherent detection (to get the high gain and selectivity), a laser noise canceler (to get to the shot noise), highpass filtering (to reject the incoherent speckles and plasma fluctuations), a CFAR servo on each frequency band (which allows continuous checking of the false alarm statistics), and burst rejection in both hardware and software (to prevent buffer overrun if we hit a glint). None of these is expensive or elaborate, and the result is a reliable, shot noise limited measurement in very difficult conditions. *Defense in depth need not be expensive, but it does need thought*.

Avoid Underengineered Complexity. You can get into a lot of trouble by assuming that if the prototype works the first time, that means it's easy to do again. Make sure you know *why* it works.

You may not even get that far. A common error, especially among professors and managers, who don't have to do it themselves, is that a measurement made up of well-understood and widely used elements may be impossible due to sheer complexity. Remember the stories of the old vacuum tube computers? Although they were made of the same radio tubes that brought music to everyone's kitchen table, there were so many of them that a tube would fail every few hours, requiring laborious reloading of programs, checking of results for corruption, and so on. Even when the components are reliable, getting them all to work simultaneously can be very hard indeed.

It is a mistake to build instruments that require more than two new hard things to work at once; it's like fighting a two-front war. An interferometric readout atomic force microscope is not too bad to build and get working, but a two-wavelength combination near-field optical microscope, AFM, lateral force microscope with tunneling capability is too complicated, unless you have a shipping product (that already does most of it) to build on.

Know When It's Time for a Clean Sheet of Paper. As comforting as it may be to have a product to build upon, at some point this bird in the hand becomes an albatross around your neck. Sometimes you just have to start over. Roughly speaking, look for this possibility whenever your legacy hardware is less than 75% used in the final design, or when it reduces your performance by a factor of 2.

Beware of Signal Processing Fads. There has been a lot of snake oil peddled in the signal processing world over the years. The latest and greatest technique will allegedly pull clean signals out of ugly backgrounds so effectively that all previous methods are immediately obsolete. Image processors will take blurred pictures of ugly people and turn them into sharp pictures of pretty people—you know the tale. Nowadays the medicine shows concentrate on techniques like genetic algorithms, neural networks, maximum likelihood estimators, and so on. Many of these techniques are genuinely useful, but none is revolutionary—they're just tools, like Crescent wrenches.

When you're fixing a car, you need a set of combination wrenches, a socket set, some pliers and screwdrivers, and maybe a meter or two. If you need a special tool, like a gear puller, you go out and buy it specially. Doing the work yourself is partly an excuse to collect a nice set of tools with the money you saved.

Signal processing is like that too. If you know how to build and use amplifiers, filters, mixers, and A/D converters, you can make a good analog signal processor. Put that together with some simple postprocessing, and you're most of the way there. If you need something special like a phase-locked loop or a Kalman (adaptive) filter, you can look up how to make one. Besides solving the problem, you've acquired a useful skill. Do things a bit differently when you have an opportunity—that's a good excuse for collecting signal processing tricks. Deep and broad expertise can be acquired this way, providing you don't treat the things you look up as mere recipes to be followed. Spend a little time figuring out what's going on in each one, and you will be richly rewarded.

As a rule of thumb, it takes half as long to become really familiar with a new technique as it does to debug one you half-understand.

Beware of Optical Fads. There are a lot more electronic and computer people than optical people, so there are more fads in the electronic and computer businesses. However, there are some optical ones to avoid as well. A few years back there was a big vogue for "nondiffracting beams," which allegedly exhibited greatly increased depth of focus for a fixed wavelength and numerical aperture. The trick was to build up the beam from plane wave components that all had the same value of k_z, so that their relative phases would be constant along the axis of the beam, and so they would never go out of focus.

Such a beam is typically generated by sending a collimated beam through an *axicon*, a cone-shaped prism pointing along the beam axis, that refracts the beam into an annular cone.The pupil function of the beam is a ring δ-function, $\delta(\rho - a)$, where $\rho = (u^2 + v^2)^{1/2}$, and the Fourier transform of that is a Bessel function, $2\pi a J_0(2\pi ar/\lambda)$.

It's perfectly true that the 3 dB intensity radius of such a beam is very narrow and is not a strong function of defocus; what was glossed over, however, is that only a small fraction of the beam energy was inside the 3 dB intensity radius, because J_0 falls off asymptotically as $r^{-1/2}$, so its energy density goes as r^{-1} and the total beam power is infinite—asymptotically, each individual diffraction ring contributes 28% *more* power than the central spot. In instrument terms, that means that even if we chop off the rings at some point (as we must), most of the signal comes from outside the central spot. The apparently improved resolution/depth of focus trade-off is illusory, just another instance of specsmanship.†

There are other examples: diode laser feedback measurements, bright-field intracavity measurements, and some types of fiber sensor.

11.6 INSTRUMENT DESIGN

> Nothing is more dangerous than an idea, when it is the only one that you have.
>
> —Émile Chartier‡

Okay, so we know what the aim of our measurement is, and what crunchiness class it's in. Now what do we do?

Know the Problem. One of the most basic laws of engineering is that you can't solve what you don't adequately understand. Do a conceptual design from end to end, first, as we did in Chapter 1, and play around with it to find the best trade-offs. Once youhave a rough cut at the system design, you can identify the hard parts.

For example, if you want to build an imaging system, you need to know the resolution requirements in advance; these will of course depend on other system parameters, such as camera to subject distance, field of view, and so on. Use rules of thumb (see Table 11.2) to spread the pain out equitably among subsystems and among designers. This will normally lead to the lowest cost system using a given technology mix. Be

†One can argue that lithographic applications can benefit, because with the high contrast of photoresist, you can adjust the exposure to make the central spot print but not the rings. Since the rings of J_0 alternate in amplitude, that means you'd need a phase-shift mask, and the author is still dubious.

‡Quoted in Miller and Friedman, *Photonics Rules of Thumb*. McGraw-Hill, New York, 1996, p.187.

TABLE 11.2. Reasonable Limits for Common Design Parameters

Field angle: <30° half-angle
Pixel rates: <20 Mpel for scanning systems
Bandwidth: 10 Hz $\leq$ BW $\leq$ 100 MHz
Mirror flatness: 1/10 wave @633 nm
Etalon fringes vs. bandwidth and spatial coherence: 1% p-p, slope 15%/GHz collimated
Polygon alignment: 15 arc seconds
Mechanical backlash: 5 μm
Motor speeds: <3000 rpm best
Eye-safe lasers: 1 mW visible, 100 μW IR
Diffraction limit: λ/4 rms wavefront error $\rightarrow$ 0.8 Strehl ratio
Diode laser feedback tolerance: 3×10^{-7} in power
Diode laser current tunability: 1 cm^{-1} @ 800 nm, 0.5 cm^{-1} @ 650 nm between mode jumps
Diode laser temperature tunability: 10 cm^{-1}
CW laser noise: 0.1% for gas lasers, 1% for N_2, excimer, 50 dB above shot noise for small YAGs
Diode laser linewidth: 50 MHz for a single-frequency diode laser, 500 MHz for VCSEL
Laser pointing stability: 10 arc seconds
Diode laser lifetime: 50,000 hours at 25 °C
Silicon PIN diode stability: 10^{-3}/°C with windows, 10^{-5}/°C windowless
PMT/APD gain stability: 1% without too much difficulty, 0.1% if you really work at it
Departures from Gaussian noise: outliers often begin near 4σ
Electronic component stability: 100 ppm/°C
Highest reasonable impedance: $\|Z\| < 60$ kΩ/f(MHz)
Coefficient of thermal expansion: 10^{-5}/K metal and glass, 10^{-4}/K plastic
Temperature coefficients of index: 10^{-5}/K glass, 10^{-4}/K plastic
Thermoelectric cooler temperature drop: 40 °C for 1 stage, 90 °C for four stages
Small heat sinks: 5 °C/W without fan, 1 °C/W with fan
Tungsten bulb life vs. temperature: life $\propto \exp(10{,}500/T)$
Surface leakage vs. humidity: 10^{14} Ω/square dry, 10^{9} Ω/square at 95% humidity
Inductor Q: <80

aware that a shift of emphasis may give an offbeat, but robust and cheap solution; for example, an imaging spectrophotometer system might use a lens with bad lateral color but good longitudinal color; the lateral color is a change in magnification with wavelength, which can be removed by using several spectral bands and interpolating between pixels to change the images to a common scale. The trade-off there would be more CPU power and possibly more pixels required versus better resolution at fixed focus and a cheaper lens.

Mess Around with the Tools. Leave lots of time at the beginning of the project for playing. Diddling around with the new laser, drawing Lissajous figures with the scanning system, and looking at the trees outside the window with your new IR viewer are all things that may seem unproductive, but are actually important for a couple of reasons: first, seeing what cool junk we have to work with builds enthusiasm, and second, messing about this way builds technical taste faster than any other activity except debugging. Whimsy is a highly practical thing in a designer.†

†See Jim Williams, The zoo circuit, in Jim Willams, ed., *Analog Circuit Design: Art, Science, and Personalities*. Butterworth-Heinemann, Woburn, MA, 1991.

Understand the Sources of SNR Limitations.

Good, fast and cheap: pick any two.

—Anonymous

Increased measurement accuracy is a perennial goal. A new scheme that yields better accuracy is like money in the bank—it can be traded for improved speed, reduced cost, or a quieter life, so even if your accuracy is OK for now, keep your eyes peeled for improvements.

To improve a measurement, you have to know what limits it. In Chapters 1, 3, and 18, we talk about shot noise, which is the noise of the light itself, and Johnson noise, the noise contributed by the front end amplifier, as well as some less common types, for example, generation–recombination noise in photoconductors. In Sections 3.10.1 and 19.10.12, we talk about source noise. Although we talk a lot about shot noise limited measurements, what really matters is the SNR, understood broadly as the ratio of signal power to true noise plus spurious signals.

Other kinds of noise crop up all the time. The simplest ones are additive noise from the background signal, for example, shot noise and 100 or 120 Hz spurs from room lights.

Many systems (e.g., disc drives, video microscopes, thin film measurements with pattern below, structured light, and speckle interferometry) are not limited by true noise, but rather by motion or random variations in the medium or in the object being measured. These sources of uncertainty produce complicated data sets, with a lot of structure that is not relevant. Understanding how to pull a decent measurement out of this sort of thing requires some basic physics and a lot of head scratching and trial and error. Make sure you find out what the limiting noise source is going to be before you begin serious design.

Look for Other Constraints. Lots of measurement problems are easy in principle but hard under the circumstances. The canonical example is the World War II radar proximity fuze, which was able to survive being fired from a cannon (remember, this was before the transistor—it used collapsible vacuum tubes). A more common constraint is an *in situ* sensor, which usually has to be small and noninvasive, and must accept all sorts of background light, noise, vibration, and probably a nonideal measurement geometry as well.

These constraints will largely determine not only the packaging but the measurement principle as well; an *in situ* IR spectrometer probably won't be a moving-mirror FTIR, and an interferometer attached to a vacuum chamber with a turbopump or cryopump won't be working at DC.

Write a Specification. Any complicated system needs a detailed system-level specification. The system spec is not merely a list of operational requirements, for example, signal-to-noise ratio, tuning range, and gas mileage, but includes detailed specs of every interface in the system, including, for example, the dynamic range of the detector, the backplane of the card cage, the API and user interface of the software, and the purpose and signal levels of every interconnection between subsystems. This is especially necessary if more than two people are going to work on the system, because otherwise you'll have complete chaos, but even if you're working alone it forces you to think all that stuff out in advance.

TABLE 11.3. System Properties

	Passive Features
Layout:	aperture, field of view, resolution, etalon fringes, stray light
Wavelength range:	wideband, single line, monochromatic
Polarization:	*s* or *p*; ellipticity, purity, birefringence, effects of coatings and mirrors, topological phase
Fidelity:	image quality, aberrations, CTF, OTF, Strehl ratio, pupil functions, illumination functions
Efficiency:	étendue, radiometry, photometry, materials, components
Alignment:	mechanical design, adjustments, verniers, locking, pointing
Component Quality:	quality specification, testing, vendor reliability
Coatings:	efficiency, durability, cost, yield, test environment, wavefront degradation, blooming, absorption, thermal effects
Stops and Baffles:	flare, type (field, aperture, hybrid), ghosts
Filters:	wavelength, bandwidth, tuning, collimation, stability with time, temperature, humidity
Detectors:	photon budget, type, quantum efficiency, gain, background, capacitance, noise, cooling
Front End:	bandwidth and noise vs. capacitance and signal level
Signal Processing:	SNR, spurs, $1/f$ noise, trade-offs between subsystems, frequency plan, bandwidth, pulse response, overload, data rate
All Components:	cost, availability, second sources
	Active Features (Includes All of the Above)
Stimulus Geometry:	size, 3D shape, divergence, beam control
Choice of Source:	thermal, laser, flash
Other Source Properties:	power, intensity stability, spectral stability, pointing stability, intensity noise, FM noise, lifetime
Coherence:	spatial and temporal source properties, mode structure, speckle, coherence fluctuations
Stability:	stray light, laser isolation, etalon fringes, drift, mode partition noise, coherence fluctuations

Example 11.2: Laser Shutoff. In a laser-based instrument controlled by a computer, you'll need to make sure that the laser turns off if the computer crashes or loses power. If you've thought about the software in advance, you'll realize that all the exception handlers and exit list processing in the world won't do this adequately, and that a very little additional hardware (e.g., a monostable that turns off the laser unless it's reset once a second by a software-generated heartbeat pulse) will make a robust design.

Include All Relevant Parameters. A partial list of system properties to think about, loosely modeled on Smart,[†] is presented in Table 11.3.

Solve Stupid Problems by Overkill. Once in a while you'll encounter a really stupid problem that stops you in your tracks. An AM detector is a good example; few people who have never designed one know how hard it can be to get it both fast and linear. If it's a small part of a big system, bring out the big guns: use a $100 commercial module

[†]Anthony E. Smart, Folk wisdom in optical design. *Appl Opt*. 33, 8130–8132 December 1, 1994.

that's guaranteed to work, or use three cheaper ones with three different amplifier gains, and digitize them all—one's sure to be in its sweet spot that way.

The number of warts your system grows in this way is nearly insignificant compared with the schedule delays caused by not doing it. If they really bother you, get rid of them in an engineering change after the product is shipping.

Make Trade-offs Early. Some trade-offs affect the basic strategy of your measurement. For example, a sensitivity/cost trade-off will show that a 3 dB increase in sensitivity in a shot noise limited system will require 2× the laser power or $\sqrt{2}\times$ the collector diameter—which is 2.8× in weight and probably 2× to 3× in cost. This may be able to be saved by improving the signal processing, or coatings, or using a different measurement approach (e.g., TDI).

Identify Show-Stoppers Early. It's worth making a list of all your assumptions, no matter how stupid and obvious, and what would happen if they weren't true—in any complex measurement, at least one will probably be wrong (see Section 19.4).

Once you have your design specification done, hold a beer check. Get all your technical friends to gather round a big table for a couple of hours as you lay out the specs, and offer them a bounty of one beer for every mistake they find. Don't be bashful about it—this is an excellent use of everyone's time. Everybody learns something, and on average, the amount of waste and delay saved is much greater than six people times the two hours it takes.

Do Your Tool Building Early. There are usually tool-building tasks to be done, such as writing a data display program that will produce histograms, maps, and time series data for your customer. That program is probably extremely useful for hardware debugging too; it's much more informative to have graphs and reports than just screens full of numbers (or worse, files full of binary data). By one-third of the way through the schedule, you'll be feeling deadline pressure already, so the temptation is to do testing and characterization work using highly manual methods such as sitting there with a clipboard watching something that software could do better and faster. This temptation gets stronger rapidly as the deadline approaches, so if any tool building is going to occur, it has to start *right away*, even at the risk of building the wrong thing.

The amount of time those tools will save you during development is easy to underestimate; if you can spot what's wrong immediately, you can fix it quickly. Speeding up the rate at which you can iterate is the real key to getting your product to market efficiently, and that needs good tools.

Know What Limits You're Pushing and Why. As a corollary, you need to know what things are hard. Measurement complexity is one part of it, but things get difficult as you approach the limits of current engineering practice, as well. Table 11.2 has reasonable limits on a number of common parameters. These are not hard limits, and experts can certainly exceed most of them, but try to stick within them if you can; life gets more difficult out there. If you really need higher performance than this, budget extra time and money to make sure that those hard things can be done reliably.

Use Standard Parts. Design your optical system with standard parts (especially standard lenses) wherever possible. Make sure you know which ones are really standard (i.e., kept continuously in stock) and which are just there to decorate the catalog. A lot of catalogs have *diner syndrome*; there are 200 things on the menu but only 15 are available fresh.

Make Good Drawings. You may know exactly how to build your system, but unless you want to build every one yourself, and have an infallible memory, make engineering drawings good enough that a technician of average skill can easily read and understand them. Especially don't scrimp on the interfaces: drawing schematics of backplanes is very tedious but very necessary.

Annotate the drawings well, too; notes about what is normal and what to watch out for in testing and assembly are enormously useful in preventing people from sweeping blunders under the rug, intentionally or unintentionally. Some examples are what the signal and bias voltages should be, what the bandwidth and data rate are just here, what the purpose of this lens is, how pure the polarization should be here, or where on the base plate a nice reflection that can be used for focusing happens to hit. Bob Pease quotes Milligan's Law (named for an acquaintance of his): "If You Notice Anything Funny, Record Amount of Funny." Having the right value on the drawing makes this easy.

Some organizations discourage this, because the notes may become out of date with engineering changes, so that they may be misleading. This is a rotten excuse; the same is true *a fortiori* about comments in source code, and nobody is arguing for leaving those out. You just have to have the discipline to maintain the notes as well as the rest of the design.

It Isn't Finished Until the Test Stand's Done. A measurement consists of data plus ground truth; if there is no *a priori* way of knowing your data are good, you can't assume that they are. A cardinal principle of life, in engineering, mathematics, and cookery, is this: *If it isn't tested, it's broken*. A corollary is that your design isn't done until the test stand's done. Build good ones; for example, an integrating sphere may be expensive, but how many days' schedule slip are you risking, having to cobble something together that won't work as well?

Don't leave the test stand until the end; very often a minor change in the instrument, such as a well-thought-out test plug, will let you put the test software right inside the instrument where it belongs. That way, when a unit is assembled, it gets stuffed into the jig and turned on; a green LED says it passed, and a red LED says it failed. Calibration data are stored in EEPROM right on board, along with revision level, serial number, and other parameters that will come in handy later.

11.7 GUIDING PRINCIPLES

There's a whole lot in Chapter 10 about what measurement principles are good for which jobs, but it's worth some discussion here about how to decide on your own. The main goals are to get the best possible data with the most robust and cost-effective instrument, in a reasonable time; this requires maximizing the IPR.

Trust Freshman Physics. If you have some simple approximation that says the gizmo should be perfect, it'll usually be very good. There are lots and lots of examples, from the massively parallel Fourier transforming action of a lens, to the way bipolar junction transistor (BJT) differential pairs split current with perfect linearity. We'll use a mechanical example.

Consider an active vibration isolation table, where a small optical breadboard is mounted on a few actuators (typically piezoelectric) to make it insensitive to vibrations coupled through the supporting structure. You might want to be able to null out a 100 nm amplitude vibration up to 10 kHz bandwidth, with a breadboard weighing 15 kg. Wiggling a large mass fast takes a lot of power even for such a small displacement: the mechanical power required to jiggle a mass m is $P = \mathbf{F} \cdot \mathbf{V}$, and $\mathbf{F} = m d\mathbf{V}/dt$. For a sinusoidal disturbance of amplitude s, the peak power is

$$P = \tfrac{1}{2} m s^2 \omega^3 \tag{11.1}$$

or 12.4 W, which for a piezo is really whomping. Besides, the breadboard will certainly resonate way below that, and who knows how much stuff will be loaded on it?

In fact, it isn't necessarily as impossible as all that, because what we're trying to do is keep the breadboard in an inertial frame of reference, and as we know from freshman physics, that requires zero force. Down at DC, the table has to follow the Earth, so very low frequency forces have to be transmitted, but not high frequency ones. Thus to isolate at high frequencies, all the piezos have to do is to move *themselves* at that rate, to keep the table from pushing on the breadboard. Not only is this much easier, but if it is done properly (e.g., mounting the piezos on the table instead of the breadboard), there should be no force on the breadboard to excite its resonances. This won't help forces applied directly to the breadboard itself, for example, from moving stages or vibrating fans, but it sure makes conducted noise easier to handle, and what's more, the attenuation needn't be a strong function of how much mass you put on the table, since it is being kept still.

Where it will make a difference is in the transfer function of the piezo actuators, which will make feedback loops impossible to design. Put an accelerometer under each piezo, to measure the forcing function a known time before it reaches the piezo, and use feedforward (see Section 15.12.7). Use an adaptive digital filter that adjusts its tap weights to minimize the residual vibration.

Believe Your Photon Budget. Your photon budget is not a lucky rabbit's foot, to be kept in your pocket, but a map of your way home in unfamiliar territory: it's vital to keep looking at it, and tracing your way as you go. You can waste a lot of time lost in some pretty unpleasant places if you don't make careful, *numerical* measurements of signal strength, noise floor, pulse width, and so on. We've belabored this and belabored it some more, but it's astonishing how many designers don't do it, even when their livelihood depends on the results.

Reduce the Background. Reducing the background signal is a very important way of improving the sensitivity and stability of your measurement. Doing measurements at AC (e.g., chopping, heterodyning, time-resolved pulses) will get you out of the $1/f$ noise region, providing that your modulation scheme doesn't move the background up in frequency along with your signal, or add significant artifacts of its own (e.g., beats between signal and chopping frequencies). We talked about that extensively in Section 10.2.

Don't Yearn for Greener Pastures. Electro-optical system design nearly always makes us do some things we're not good at or we're not sure will work. There's an insidious tendency for the problems in unfamiliar fields to look small. Software people sometimes seem to think that the way to do any optical thing is to stick a video camera on a frame grabber. Optics people, conversely, will sometimes try to shift off onto software what is best achieved with a pot, for example, fixed gains and offsets on a high speed data stream, or will rely on calibration to remove a nonlinearity that should have been designed out in the electronics. Get some good advice, backed up with more than just professional opinion (i.e., prejudice) before adopting a design that relies on a technology you don't understand thoroughly.

Try to stick to what you have a comparative advantage in. When faced with two nearly equally good ways of doing a measurement, choose the one that is a better fit to your skills. If you're a gifted electronic designer, start off building stuff with lots of electronics and only enough of everything else to achieve a good signal-to-noise ratio, and branch out into more complicated optics as your expertise broadens.

Model It. Get into the habit of modeling what you're planning to build. This doesn't mean building full 3D electromagnetic models of everything, or even finite-element models of the response of the system to vibration. It does mean calculating how the aberrations of the system propagate to the business end, what the SNR is going to be, how the system performance varies with filter bandwidths, scan speeds, dark current, and so on. Go through your signal processing strategy in exact detail, for example, how you're going to separate closely spaced peaks, and how the time/bandwidth behavior of your filters will affect the measurement. You can build confidence this way, and it will help you to understand any anomalies that surface later. Finding problems early on gives you time to work around them before they become large and hairy.

Note that the model doesn't have to involve computers; math programs are convenient and popular, but an analog model or a simple physical model can be very helpful (e.g., the pad stack capacitance model of Section 16.2.6), and of course for generating lots of intuition in a short time there's no substitute for grinding out an analytical model manually. Some combination of these is usually best, but don't fall into the trap of relying solely on computer modeling, lest you become a one-trick pony.

Get Ground Truth. In the remote sensing business, you have a satellite that can take immense amounts of data, for example, hyperspectral images of the entire Earth at 10 m resolution, with coverage once a week at the same sun-time each day. This is a wonderful thing—imagine all the information in those pictures. When you get them to look at, though, it's just this oddly colored map of surface reflectance averaged in some way over that 10 m diameter area. What in the world is really down there? Someone has to travel there and look, and look, and look, to correlate the imagery with what's there on the ground—*ground truth*, it's called, and the usefulness of your measurement depends utterly on the uniqueness of the mapping between it and the imagery. Satellite imagery isn't much use looking for deforestation if forest and mud have the same spectral signature with your apparatus. The squishier your measurement gets, the more ground truth you need, and the luckier you have to be to find a unique mapping. This is actually a very general problem, from microscopy to astronomy.

Keep It Simple. People buy or use things because they make life easier enough to make the pain of buying or using worthwhile. If you are the only person who can use a system, you will be the only person who does. Instruments have to be good enough to ignore. For example, if your instrument uses expendable items such as tungsten bulbs or small amounts of inert gas, consider having online spares that can be activated automatically, allowing maintenance to be done at some convenient time rather than *right now*. Try to achieve graceful degradation, where a stressed or damaged system functions as well as it can until replaced or repaired, and gives an intelligent indication of what state it's in, in real time.

Consider Moving the Goal Posts. One good way to improve the IPR is redefining the problem to make it easy. There are no hard or fast rules, but when you find yourself murmuring "This would be a lot easier with more light," or something of that sort, always ask "Well, so why can't I have more?"

Consider scanning or time gating to improve SNR in Johnson noise or background limited systems, or use retroreflective tape to improve signal levels. Noise cancelers or feedback stabilizers can dramatically reduce laser noise. Single-sideband mixers are 3 dB quieter than ordinary ones, since they get rid of the noise at the image frequency. Homodyne detection (IF at DC) has the same advantage, but is phase sensitive.

A tunable diode laser gas spectrometer with etalon fringe problems can be improved by two orders of magnitude by taking one measurement with the sample, subtracting a second one taken after admitting room air to the sample cell; the sample absorption will be pressure-broadened into nonexistence, but the etalon fringes will be closely similar.[†]

Build the Test Fixture into the Instrument. Nowadays most of the effort in designing test fixtures goes into their software, and the instrument can hold that just as well as the test stand. Building the testing software in costs little and ensures that there are no problems with incompatible versions of the instrument and tester. Instruments that can calibrate themselves online have hugely less maintenance cost, can correct for drifts and measurement errors in near-real time, and can tell you reliably about their own health.

Keep Gross Errors Obvious. Keep zero signal on-scale. Bring out the DC photocurrent so you can see how much light you've got. Put in a pilot light. Add a viewer. Make failures obvious and alignment easy, and you've got an instrument whose maintenance is painless and hence cheap. It's very frustrating for your customers to have to send an instrument back to the manufacturer. The author once had a shiny new shear-plate measuring interferometer whose Nitinol-flexure actuator broke. The distributor insisted on having it sent back, whereupon they installed the new part grossly misaligned. (Would you believe 15°?) It took the best part of a month, and the author had to realign the instrument anyway. If your customer is competent, and your instrument is simple, just ship him the part and the manual page. If he screws it up, it's fair to make him pay for the replacement; if he doesn't, he's happy and you're happy.

11.8 DESIGN FOR ALIGNMENT

Systems whose alignment is easy and stable don't appear by accident. As you're laying out the system, figure out the alignment algorithm and how you'll test it. Make sure

[†]At least they will if the cell is short enough; for larger cells, you can do several sweeps as the cell bleeds up to atmosphere, and correct for the etalon fringes' phase shift with air pressure.

each subsystem can be aligned and verified independently. Sections 12.8 and 12.9 have a fair amount on alignment, and you should know something about it before beginning a design.

Use Corner Cube Interferometers. The best alignments are the ones you don't have to do. Using corner cubes instead of flat mirrors in interferometers makes alignment the optician's problem, and a stock of corner cubes is like a stock of perfectly aligned interferometers. Use them whenever you can afford to; they do cost money and photons. Watch out for the polarization shift in TIR cubes.

Use the Poor Man's Corner Cube: Retroreflective Tape. As we saw in Section 7.8, retroreflective tape is an excellent solution to alignment worries in many systems. It is commonly used in fiber-fed bulk optics sensors, where it greatly simplifies the problem of how to get the light back into a single-mode fiber. You can use it to enormously increase the sample space accessible from a small sensor and illuminator, at least for optically thin paths where diffuse scatter doesn't dominate.

Put in a Viewer—You Can't Align What You Can't See. As a corollary, try to have as many dials as you have knobs. Aligning a system with too many interacting adjustments and not enough independent readouts is very frustrating indeed. An example is doing initial alignment of a multimirror laser, where nothing at all happens until you get very close, and then every adjustment seems to make it worse. For such a system, an auxiliary alignment system such as a HeNe laser plus a set of cross hairs or iris diaphragms and a viewing system is a great help.

Instruments requiring fine alignment should have a microscope trinocular head† designed into them; since these expect parallel light coming in, it's very simple—pick a convenient pupil (i.e., the Fourier transform plane of whatever you're trying to align), and just put in a beamsplitter to shine some of the returned light into the trinocular head, which you mount somewhere convenient. It looks a bit odd having microscope heads sticking out of your setup, but it surely makes alignment easy. Try it just once and you'll do it forever. In some cases, you may need to put in a field lens (see Section 12.3.14).

Note: Don't do this in a laser system without great care—be sure that there's enough attenuation so that the laser light cannot exceed 100 μW CW per eyepiece, even in fault conditions, for example, a tilted sample that sends the main beam right toward the eyepieces. Visible light that bright will make your eyes hurt, and IR brighter than that may damage them. Filter out all the invisible radiation just to be on the safe side.

Use Verniers. If you need a fine adjustment, put in a vernier, such as a full wave plate which can bewobbled to get exact correction of small polarization shifts, a couple of wedges that can be rotated for fine beam steering, or a glass plate glued to a single ball bearing pivot‡ for x-y beam position adjustment. Don't put a full range adjustment on an expensive mount and expect to do as well. Make sure you consider the effects of such a change, however; most full wave plates based on birefringence don't work well in white-light systems, pivoting plates will produce fringes, and rotating wedges make the pupil move.

†The part that the eyepieces slide into.

‡As used in those ugly pen-and-pencil desk sets for the-person-who-has-everything.

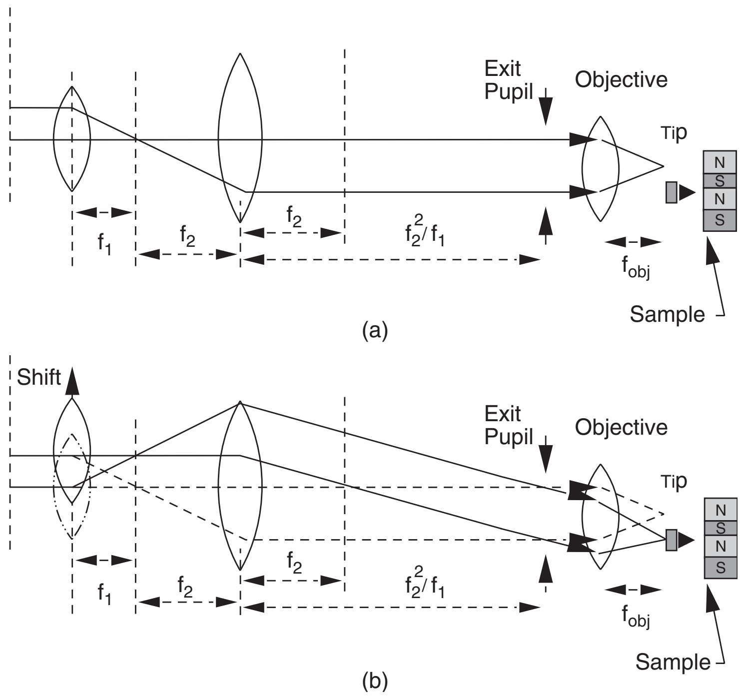

Figure 11.2. Shifting lens *A* moves the focused spot without misaligning the interferometer, making tip changing a lot easier.

Adjust the Right Thing. It is frequently possible to tailor the adjustment to the purpose. Consider a magnetic force microscope using cantilever tips made from fine nickel wire, electropolished down to 5–10 μm diameter. If we want to use a heterodyne Mach–Zehnder interferometer to detect the vibration of the cantilever, the beam must be focused to a spot less than 2 μm in diameter with a microscope objective. The cantilever has to be changed often and is not easy to position that accurately, but we can't stand to realign the interferometer every time.

One good solution is the auxiliary two-lens system of Figure 11.2. Shifting lens 1 sideways moves the beam around in angle in the pupil of the objective, which causes the focused spot to move telecentrically (i.e., the cone of light doesn't tilt as it moves). Moving lens 1 in and out a bit gives a fine focus adjustment. Because the cone is always normal to the cantilever, light reflecting from the tip retraces its path regardless of where exactly the cantilever winds up.† Thus moving the beam around doesn't misalign the interferometer. All you have to do is get the beam onto the cantilever and focus to get a good signal, and as long as there's a viewer, that takes about 10 seconds.

Watch Out for Temperature Gradients. Gradients are much worse than uniform variations with time. They cause things to *bend*, so that the error builds up and builds up with distance. Objects bend, and the temperature coefficients of length and refractive index mean that beams bend too—from the Schlieren effect and from the thermal

†There's a small amount of vignetting, but not enough to notice.

distortion of the figures of lenses and mirrors. See the chapter problems (available on the Web at http://electrooptical.net/www/beos2e/problems2.pdf) for more.

Usually Follow the Leader. There are a lot of nonobvious potholes in optical instruments. If something in someone else's instrument looks much too complicated for what it does, it probably is; but it might not be, and the reason for it may be very subtle. Do not be too quick to label other designers idiots, but find out why they did it that way. For example, a Savart plate has unequal phase delay in the two beams, unlike a Wollaston. A nonzero path difference interferometer is vulnerable to frequency noise in the laser; even a small fraction of a coherence length will produce a large increase in the noise of a bright-field measurement if you're not careful (see Section 19.1.1).

Don't Always Follow the Leader. A good design hack can overcome some pretty ugly limitations. Two-photon Doppler-free spectroscopy is an excellent example (see Example 1.1), and there are lots of others.

11.9 TURNING A PROTOTYPE INTO A PRODUCT

With the emphasis placed on prototyping elsewhere in this book, it is easy to assume that a working prototype translates into a working instrument by itself. This is of course not true, but by avoiding the most common mistakes, you can improve your odds enormously.

11.9.1 Be Very Careful Of "Minor" Optical Design Changes

> He [Ferdinand Porsche] is a very amiable man but let me give you this advice. You must shut him up in a cage with seven locks and let him design his engine inside it. Let him hand you the blueprints through the bars. But for heaven's sake don't ever let him see the drawings or the engine ever again. Otherwise he'll ruin you.
>
> —Ernst Heinkel, *He 1000*

There really isn't any such thing as a minor optical design change, because each optical element does so many things, and we only know some of them. This means that a mildly redesigned optical system has to be tested as though it were an entirely new optical system, that is, breadboarded and run through the whole range of tests (see Section 19.2.1). Failure to understand this can be very expensive; the liquid-borne particle sensor of Section 19.1.1 was derailed by replacing two elements, a HeNe laser and a Nomarski prism. Ironically, it would have worked with either of the two changes, just not both, and a breadboard would have shown the problem before the trouble got started.

11.9.2 Don't Design in Etalon Fringes

In turning a prototype into a limited production product, there is a temptation to take all the optical mount designs and turn them into one big complicated custom mount that can be made in one step on a CNC milling machine. This saves money by reducing the amount of handling and assembly, but if it is done naively, all the incidental minor misalignments that protected your prototype from etalon fringes will be zeroed out; the resulting fringes will blow you right out of the water—just like Microbench, only worse. Not only that, but you're at the mercy of the tolerances of the components you're using;

for example, lenses often come with a 5% tolerance on their focal length. Make provision for strategically chosen adjustments and baffles, and watch those stray beams.

11.9.3 Handle Demo Karma Gracefully

Besides pretty pictures, your sales job will include demonstrating your system in action (unless you make sensors for explosions). Prototypes are a bit flaky, always. Demonstrations of prototypes are dangerous; the stakes may be high and preparation time limited, which makes for jitters. Jittery people break things, so the death rate of prototypes is ironically highest during demos, leading to the phenomenon known as *demo karma*, where you break the prototype at 5 p.m. the day before. Even working all night to fix it will usually not gain you enough merit to get it going in time.

Resist the temptation to cut corners in preparing for demos. You know, hot plugging boards and cables, wiping lenses on your shirt instead of cleaning them properly, stuffing loose bits in for test purposes without securing them, generally doing things in a hurry. Your system will be more likely to survive if you get it in shape a few days in advance, in a lab with spares for all critical parts, and then spend the three days before the demo polishing up your talk, cleaning your desk, or writing a paper. Unfortunately, this requires more willpower than a pastry chef on a diet. It remains an unsolved problem.