An Introduction to the Event-Related Potential Technique

Steven J. Luck

The MIT Press

From The MIT Press

MITCogNet

© 2005 Massachusetts Institute of Technology

All rights reserved. No part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher.

MIT Press books may be purchased at special quantity discounts for business or sales promotional use. For information, please email special_sales@mitpress.mit.edu or write to Special Sales Department, The MIT Press, 55 Hayward Street, Cambridge, MA 02142.

This book was set in Melior and Helvetica on 3B2 by Asco Typesetters, Hong Kong.
Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Luck, Stephen J.
An introduction to the event-related potential technique / Stephen J. Luck.
p. cm. — (Cognitive neuroscience)
Includes bibliographical references and index.
ISBN 0-262-12277-4 (alk. paper) — ISBN 0-262-62196-7 (pbk. : alk. paper)
1. Evoked potentials (Electrophysiology) I. Title. II. Series.
QP376.5.L83 2005
616.8′047547—dc22 2005042810

10 9 8 7 6 5 4 3 2 1

8 Setting Up an ERP Lab

This chapter describes how to set up an ERP lab, including advice on selecting equipment and software for stimulus presentation, data collection, and data analysis. There are certainly many ways to set up an ERP lab, and some factors will depend on the topic area of your research. However, the suggestions in this chapter will provide a good starting point for almost anyone who is setting up an ERP lab for the first time. Also, many of the suggestions in this chapter are based on my experience, but there are often other good ways of achieving the same goals, and it is worth asking a variety of ERP researchers how they set up their labs. If you already have access to an ERP lab, you may find some good ideas for improving the lab in this chapter (see especially box 8.1 later in this chapter).

The Data Acquisition System

Computers

Figure 8.1 shows a diagram of a generic ERP data acquisition system. It is usually desirable to have at least two and possibly three computers. One computer presents the stimuli, and a second computer records the EEG. These functions are not ordinarily combined into a single computer, because each requires precise timing, and it is difficult to coordinate two real-time streams of events in a single computer. It is also very useful to have a third computer that provides real-time information about the subject's performance on the task. The stimulus presentation computer sometimes provides this

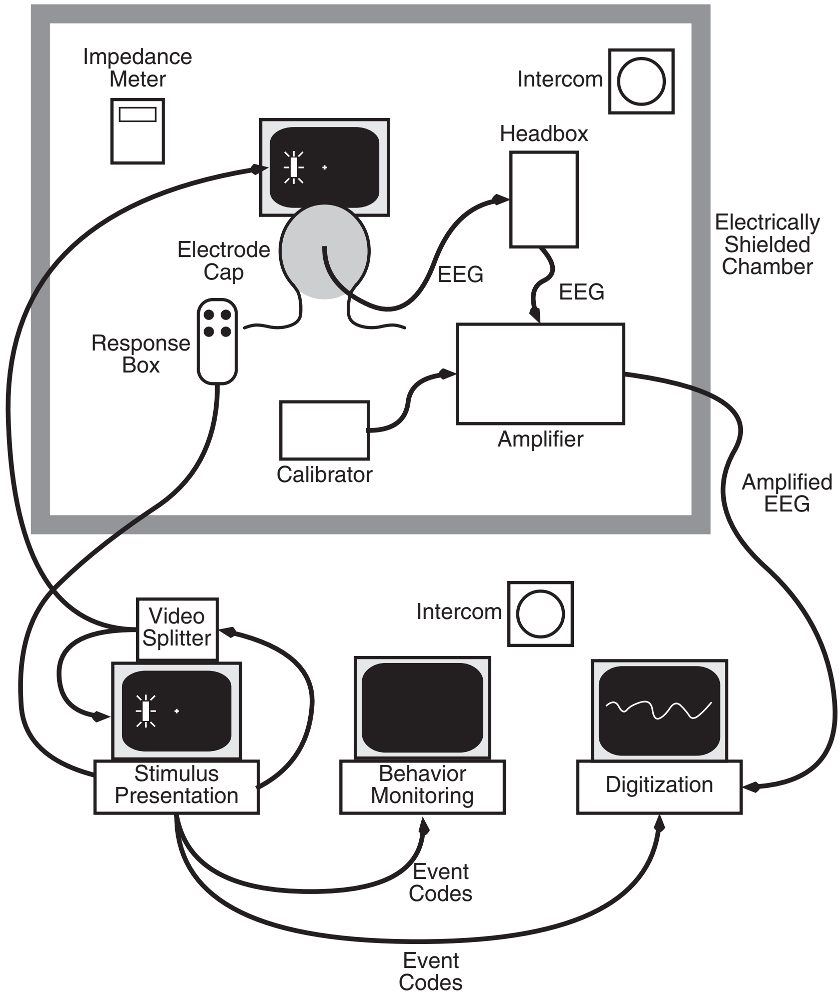

Figure 8.1 Major components of a typical ERP recording system.

function, but in this case performance information is usually summarized at the end of each trial block rather than being displayed in real time (and a real-time display is definitely better).

The computers will need some means of communicating with each other in real time so that *event codes* can be sent to the digitization computer whenever an event of some sort occurs (e.g., a stimulus or a response). These event codes are used as the time locking points for averaging, so the timing must be precise. A consistent delay is not a big problem, because you can shift the averages in time to compensate for the delay. A variable delay, however, is usually difficult to compensate for, and it will have the effect of smearing out the ERP waveforms in time, distorting the onset and offset times of the components and experimental effects (just like a low-pass filter).

The stimulus presentation computer's video output is usually directed into a video splitter that serves two functions. First, it splits the video signal so that the experimenter can see what the subject is seeing (this is essential so that the experimenter can intelligently monitor the EEG display and detect problems, such as blinks and eye movements that certain stimuli may trigger). Second, it amplifies the video signal so that one can use a relatively long cable to get the video signal into the recording chamber. It is important to use a high-quality, shielded video cable from the splitter into the chamber to minimize electrical noise inside the chamber and to avoid degradation of the video signal. As discussed in chapter 3, it's a good idea to enclose the video monitor in a Faraday cage to avoid bringing a large noise signal into the recording chamber. I provide more information about the video system later in this chapter, in the section on stimulus presentation.

Seating

In experiments using visual stimuli, the subject is seated at a specific distance from the video monitor. Even with a Faraday cage, some electrical noise may be picked up from the monitor, so the subject should not be too close to it. I would suggest a minimum distance of 70 cm, and 1–2 m is even better if the chamber is large enough.

It is very important for the subject to be seated in a comfortable position. In the early days of ERP recordings, very few electrodes were used and auditory stimuli were more common than visual stimuli. A large, padded recliner was the most sensible chair for such experiments. This kept the subject comfortable, and minimized any need for subjects to use their neck muscles to support their heads, which in turn minimized EMG noise (see chapter 4). However, when electrodes are placed over the back of the head, a recliner doesn't work as well: the head puts pressure on the electrodes, and any small head movement will cause the electrodes to move on the scalp, producing a large and sudden voltage shift.

If your experiments will require recording from electrodes on the back of the head, I would not recommend using a recliner. Instead, I would recommend a high-quality office chair that provides good lumbar support and has an easy mechanism for height adjustment. It is also preferable to use a chair that is on glides rather than casters so that subjects do not move the chair around inside the chamber. The position of the chair on the floor should be marked so that you can correctly reposition the chair if it gets moved. I would also recommend against using any kind of chin rest or other head stabilization. Although you might think that this would reduce strain on the neck muscles, my experience has been that chin rests become uncomfortable after 15 minutes or so (and most ERP experiments require at least an hour of recording time).

Behavioral Responses

Most cognitive ERP experiments use a small set of behavioral response alternatives, and only two to four buttons are usually necessary. The most convenient device for responses is usually a small, lightweight, hand-held device of some sort. My lab uses computer video game controllers. It is easy to find game controllers that are very comfortable and lightweight, and they usually offer at least four response buttons (which is enough for most experiments). Various custom response devices are also possible, but they are not usually as easy to hold and use as a game controller. A computer keyboard is not usually very good, because it is too big to rest on the subject's lap, and if it's on the table in front of the subjects, EMG noise is likely to occur as the subjects hold their arms up to the keyboard. If a keyboard is necessary because of a large number of response alternatives, the data around the time of the response may be contaminated by EMG noise, and eye movement artifacts may be seen if the subject needs to look at the keyboard to find the right key to press. However, it may be possible to design the experiment so that the data around the time of the response are not important. Also, it may be possible to use a keyboard tray to minimize the need to hold the hands upward and outward to reach the keyboard.

Devices such as keyboards and game pads are not designed for real-time electrophysiological recording, and a delay may occur between the time of the buttonpress and the time that the computer records the buttonpress. Response times are usually so variable that this small error will not matter. However, it may be a significant problem if you are doing response-locked averaging. In such cases, you will probably need some sort of custom response device with known and consistent temporal properties. One way to ensure precise timing is to use a response device with an analog output that can be recorded via the ADC along with the EEG.

Electrode-Amplifier-Computer Connections

The EEG is picked up by the electrodes and travels through cables to the amplifier system. Before they are amplified, the signals are miniscule, and any electrical noise that the cables pick up will be relatively large compared to the EEG. Consequently, the cables between the electrodes and the amplifier system should be kept as short as is practical. It is now possible to purchase electrodes with built-in pre-amplifiers, which will increase the size of the EEG signal relative to any noise picked up by the cables. This is a good idea conceptually, but I have not tested these electrodes and I don't know how well they work in practice.

Many EEG amplifier systems have a *headbox* that provides a flexible means of connecting the electrodes to the amplifier. Flexibility is important. For example, you want to be able to select a common reference electrode for most of the channels (e.g., a mastoid or earlobe reference) while allowing other channels to use a different reference electrode (e.g., eye movements are usually recorded as the voltage between electrodes on either side of the eyes). Because the output of the headbox is not amplified, the cable from the headbox to the amplifier should be relatively short. Optimally this cable should be shielded, especially if the actual amplifier is outside the recording chamber. If the amplifier is inside the chamber, an unshielded ribbon cable will probably be sufficient (a ribbon cable keeps the active, reference, and ground signals close to each other so that they pick up the same noise signals, and this allows the differential amplifier to subtract away the noise).

If the amplifier is connected to AC power, it is usually best to place it outside the recording chamber to avoid inducing noise in the electrodes. If it is battery powered, it can probably be placed inside the chamber, which is good because only amplified signals will leave the chamber, where electrical noise is much greater. Once the signals have been amplified, they are usually much larger than any induced electrical noise, so you may use longer cables to connect the amplifier to the digitization computer. However, some noise may still be picked up at this stage, and some shielding may be necessary even after the signals have been amplified. In one of the recording rooms in my lab, we found that shielded cables were necessary to avoid picking up noise between the amplifier and the digitization computer.

Recording Chamber

One of the most expensive elements of an ERP lab is the electrically isolated chamber. These chambers are usually designed as sound-attenuating chambers for hearing testing, broadcasting, or auditory research, but some manufacturers offer electrical shielding options. I have chambers made by Industrial Acoustics Company and Acoustic Systems, and they work quite well. In the late 1990s, I paid approximately US $11,000 for a medium-sized chamber (including installation costs).

Is an electrically isolated chamber really necessary? Not always. If there are not many significant sources of electrical noise nearby, you might be able to get away with using a low-pass filter with a half-amplitude cutoff around 30 Hz to get rid of line-frequency noise. If you are just starting to do ERP research and you are not sure how many experiments you will be doing, this might be a reasonable choice. However, if you plan to do ERP research for several years, the cost of a chamber isn't really that high, and it's a worthwhile investment. Remember that any significant decreases in signal-to-noise ratio will be very costly in terms of the number of trials (or subjects) needed to get reliable ERP waveforms (see chapter 3).

Don't assume that the chamber effectively eliminates all sources of electrical noise. Chambers often include AC outlets, fans, and lights that may cause substantial electrical noise. In fact, the first chamber I ever used induced so much noise that I was able to get cleaner recordings outside the chamber than inside. I eventually found the circuit breaker for the chamber, and turning it off eliminated the noise. You can use the device described in chapter 3 (see figure 3.3B) to find sources of noise inside the chamber.

Accessories

A few accessories are important in an ERP lab. First, it is usually necessary to have an impedance meter to test electrode impedances while attaching the electrodes. Some amplification systems have this function built into them, which is very useful for determining if impedance problems are occurring once you've started collecting data. Second, some sort of calibrator is necessary for measuring the actual gain of the amplifiers. Many amplification systems have a built-in calibrator, but you can purchase an external calibrator if your amplification system does not have its own calibrator. If possible, the calibrator should be able to produce square-wave pulses triggered by the stimulus presentation computer. This will help you to determine if there are any time delays in your event codes, and it will also make it easy for you to see exactly what kinds of distortion the amplifier's filters produce.

A third important accessory is an intercom system that will allow you to communicate with the subject during the recordings. When purchasing an intercom system, keep in mind that you don't want a system that is AC powered, because this will introduce electrical noise into the recording chamber. I've tried several different types of systems, and my preferred intercom system consists of a pair of microphones and powered speakers (which you can purchase at many places, such as Radio Shack). One microphone is attached to the ceiling of the recording chamber and is connected to a powered speaker outside the chamber. The other microphone is mounted near the experimenter and is connected to a powered speaker inside the chamber. The speaker inside the chamber should be powered by a battery. The advantages of this system over a conventional intercom are that a) it allows simultaneous two-way communication between the subject and experimenter (which is more natural than pressing a button when you want to talk), and b) the fidelity is quite a bit higher.

Finally, if you will be presenting visual stimuli, I strongly recommend that you enclose the video monitor in a Faraday cage to reduce electrical noise, as described in chapter 3 (see figure 3.3A). Some additional suggestions for recording high-quality data appear in box 8.1.

Box 8.1 Keeping Subjects Happy

ERP experiments tend to be long and boring, with trial after trial of the same basic task. To ensure that you are collecting the highest quality data possible, it is important to keep your subjects happy and relaxed. If they are unmotivated or become bored, they may not pay close attention to their performance, weakening your effects. Moreover, bored subjects tend to become tense and uncomfortable, leading to muscle noise and movement artifacts. By keeping your subjects happy and relaxed, you will get larger and more consistent effects, which will save you lots of time in the long run. Here are four suggestions for accomplishing this.

First, talk to the subject. As you are applying the electrodes, chat with the subject, asking about school, work, hobbies, family, sports, and so on (but stay away from potentially inflammatory topics such as politics). Make sure that the conversation revolves around the subject rather than yourself. By doing this, you will ingratiate yourself to the subject, and the subject will be more likely to try to please you during the experiment. In addition, when there are breaks in the experiment, you should continue to converse with the subject to keep the subject alert and interested. Some subjects just want to finish the experiment as rapidly as possible, and you should respect this. But if the subject is at all interested in talking, you should do so. This is also a good time to provide some positive feedback to the subject about behavioral performance, blinks, and eye movements, as well as encouraging the subject to improve if necessary. If a subject isn't doing well (e.g., in terms of behavioral accuracy or artifacts), don't be shy about telling them that, because poor performance and artifact-laden data won't be of much use to you. You should also make sure that the subject understands exactly what is going to happen and when (especially during the electrode application process). People are much more relaxed when they know what to expect.

Second, make sure that the blocks of trials are a reasonable length. If the blocks are too long, the subject's attention is likely to wane toward the end. I find that trial blocks of 5–7 minutes, separated by 1–2 minutes of rest, are optimal in most experiments. In addition, you will usually ask the subject to suppress blinks and eye movements during the recordings, and it is difficult to do this for a long period of time. Some experiments consist of brief trials separated by an intertrial interval of 1–3 seconds, and this provides a time for subjects to blink (although the offset of the blink during the intertrial may contaminate the early portion of the ERP on the next trial). If this is not possible, I would recommend providing short breaks of 15–20 seconds after every 1–2 minutes.

Third, due to the long duration of a typical ERP experiment, it is helpful to provide snacks and drinks, usually 30–50 percent of the way through the

Box 8.1 (continued)

session. Drinks including caffeine can be particularly useful in helping subjects to maintain alertness (although this may be contraindicated in some experiments). This helps to keep the subjects awake, alert, and friendly.

My fourth recommendation is something that I started doing when I first set up my own ERP lab, and I don't know if anyone else does it. Specifically, we play background music in the chamber while the subject is doing the task. In fact, we suggest to the subjects that they bring CDs of their favorite music with them (and the musical genres have included classical, pop, rock, metal, rap, country, electronic/ambient, and just about everything else imaginable). Of course, the music produces some distraction from the task, and the sounds will generate ERP activity. However, I believe that the music is less distracting than the alternative, which is mind-numbing boredom. And any ERP activity generated by the music will be unrelated to the stimuli and will just add a small amount of noise to the EEG. I suspect that any additional EEG noise created by the music is more than balanced by a reduction in other forms of EEG noise, such as alpha waves. Of course, there are situations in which background music is a bad idea. For example, it would be problematic in many studies using auditory stimuli, and it might cause too much distraction in some types of subjects. But if these circumstances don't apply to you, I would definitely recommend playing background music. I've never done a direct comparison of data collected with versus without background music, but I strongly suspect that background music leads to a moderate increase in the overall quality of the data. If you play music, make sure that you use shielded speakers and that the cables leading from the stereo to the speakers are shielded. Speaker cables are not usually shielded, but you can easily encase them in some form of shielding, such as Wiremold.

Choosing Electrodes, Amplifiers, and Digitization Software

New ERP users occasionally ask for my advice about what electrodes, amplifiers, and software to buy for recording ERPs; this section summarizes my usual advice. Given that I haven't thoroughly tested all of the commercially available systems, it would be inappropriate for me to recommend specific manufacturers. However, I can provide some general advice about how to go about selecting data acquisition equipment.

First, however, I'd like to provide a caveat. The suggestions here are my own opinions, and they may be different from the opinions of other researchers (and especially certain equipment manufacturers). Consequently, I would recommend seeking advice from a variety of experienced ERP researchers (but not from the manufacturers, who are quite naturally biased).

Electrodes My electrode recommendation is simple: Old-fashioned, low-tech electrode caps are the best option for the vast majority of new ERP users. As discussed in chapter 3, some newer systems allow you to apply a large number of electrodes in a short amount of time, whereas traditional electrodes require you to abrade the skin under each electrode, which takes quite a bit of time when applying large numbers of electrodes (e.g., an hour for sixty-four electrodes compared to 15 minutes for sixteen electrodes). However, these systems tend to be more expensive than conventional systems, and the high electrode impedances in these systems can lead to a substantial increase in low-frequency noise from skin potentials. Moreover, large numbers of electrodes do not provide a very significant advantage in most ERP studies, but they can provide a disadvantage because there are more opportunities for electrodes to misbehave as the number of electrodes increases. Consequently, I would recommend standard electrode caps with between twenty and thirty-two electrodes for most ERP studies.

Fast-application, high-impedance electrode systems may seem particularly appealing to researchers who study subjects who do not easily tolerate being poked and prodded, such as infants and young children. However, some researchers have been recording ERPs from such subjects for many years with conventional electrodes, so it can be done (although it requires good measures of both ingenuity and patience). Moreover, given that it is difficult to collect a large number of trials from such subjects, their data are far too noisy to be suitable for source localization procedures, so there is little or no advantage to having large numbers of electrodes. Thus, I would recommend using conventional electrodes even for infants and young children.

Amplifiers When selecting an EEG amplifier, there are a few key specifications that you should examine and several features that you may wish to have (depending on your budget). The most important specifications are the input impedance and the common mode rejection (see chapter 3 for definitions of these terms). The input impedance should be at least 100 KΩ, and the common mode rejection should be at least 100 dB. The essential features are (a) the ability to select a common reference for many sites but separate references for other sites, and (b) a wide range of filter settings. In particular, I would recommend having a low-pass filter that can be set to half-amplitude cutoffs of 20–40 Hz, 80–120 Hz, and 300–700 Hz, and a high-pass filter that can be set to cutoffs of approximately 0.01 Hz (for most recordings) and approximately 0.1 Hz (for use with especially troublesome subjects). In some amplifiers, the lowest half-amplitude cutoff is 0.05 Hz, which is five times higher than the 0.01 Hz that I would recommend; this is right at the border of what I would consider acceptable. It would also be worth asking the manufacturer to provide the impulse response function of the filters so that you can assess the time-domain distortions that the filters will produce (if they can't provide the impulse response functions, then that should be a warning sign about the manufacturer). I would recommend against DC amplifiers that don't have a high-pass filter, unless you plan to record very slow voltage shifts.

Here are some features that are useful, but not absolutely necessary:

1. Multiple gain settings. It can be useful to increase or decrease the gain for testing purposes or to match the input range of the analog-to-digital converter.
2. Impedance checking. It's convenient to be able to test impedances without disconnecting the subject from the amplifier. It's even better if the system can automatically warn you if a channel's impedance exceeds some user-defined level.
3. Calibration. It's useful for the amplifier to have a built-in calibrator.

4. Notch filter. Although notch filters are generally a bad idea, they are sometimes a necessary evil (see chapter 5).
5. Independent or linked filter settings. You will usually want to use the same filter settings for every channel, but it's occasionally useful to have different filter settings for a few of the channels.
6. Computer-controlled settings. If a computer can set the amplifier's settings on the basis of a user-defined profile, this will decrease the probability that another user will unintentionally change the settings.

Digitization Software Once you have electrodes and an amplifier, you will need a computer and software for digitizing the EEG. In some cases, the electrodes will dictate the choice of amplifier, and the amplifier may have an associated digitization program. Most digitization programs have the basic features that you will need, so you should choose whatever seems convenient and will work with the other components of your system. The essential features of a digitization program (and associated analog-to-digital converter) are as follows:

7. The analog-to-digital converter must have at least twelve bits of precision. Many now offer sixteen bits of precision, but given the poor signal-to-noise ratio of the EEG signal, this is not a very big advantage. A sixteen-bit converter allows you to use a lower amplifier gain, reducing the probability that the amplifier will saturate. However, saturation is usually caused by large, slow voltage shifts that will contaminate your data even if the amplifier does not saturate.
8. There must be a convenient means of sending event codes to the system. Some systems merely use one channel of the analog-to-digital converter to encode a pulse whenever an event occurs, and the actual nature of the event is stored on another computer (usually the stimulus presentation computer). This makes averaging the stimuli less convenient, so it is better for the digitization computer to receive codes indicating the precise nature of each

stimulus. This is usually done with eight-bit digital codes (it is convenient to have even more than eight bits for some experiments).

9. The system must allow continuous EEG recording. Many ERP paradigms involve stimuli that are separated by long periods of time with no ERP-eliciting events, and it is therefore possible to record just the EEG segments surrounding the stimuli, pausing during intertrial intervals. This is called *epoch-based recording*. However, you may later wish that you had recorded a longer epoch. For example, you may find an effect that is still growing in amplitude at the end of the epoch, or you may want to evaluate prestimulus overlap more thoroughly. Epoch-based recording does not allow this. The alternative (*continuous recording*) is to record the EEG continuously and extract whatever epochs are necessary during the averaging process. All of the data are saved, so it is possible to re-average the data with a different epoch at a later time. This requires more storage space, but storage space is so cheap now that it's not a problem to record continuously. Thus, you will want a system that can do continuous recording. The option to do epoch-based recording may sound attractive, but I would recommend against epoch-based recording because of the loss of flexibility.
10. The system must allow a convenient way to view the EEG in real time as it is being recorded. Real-time monitoring is essential so that the experimenter can identify problems with the electrodes, high levels of artifacts (especially ocular artifacts and muscle activity), and subject weariness as indicated by high levels of alpha activity (weariness can be a significant issue because ERP experiments usually last several hours). It is essential to be able to adjust the vertical scale of the signal to match the size of the EEG signal, and it is also useful to be able to adjust the horizontal (time) scale. The system should also include some means of seeing the arrival of event codes. It is very convenient if the event codes appear in a position that is synchronized to the EEG display so that you can view the relationship between event codes and artifacts.

Some additional options that may be useful are as follows:

11. The ability to view the EEG in the frequency-domain as well as the time domain. This makes it easier to determine the level of various noise signals, such as line-frequency electrical noise, and to notice when alpha levels are getting high (which is often a sign of drowsiness).
12. The ability to save the data in multiple formats, including text files. Text files are huge, but they have the advantage of being easy to convert into other formats.
13. The ability to do channel mapping, in which the amplifier channels are sampled in an arbitrary, user-controlled order. This makes it easy to put the data into a useful order, skipping channels that are not necessary for a given experiment.
14. The ability for the user to define the spatial layout of the EEG display. This may make it easier to see patterns in the EEG that are indicative of artifacts. Some systems can even display maps of EEG frequency bands in real time.
15. The ability to do simple on-line averaging. In some experiments, for example, subjects may tend to make small but systematic eye movements that are difficult to see on individual trials but can be easily seen in averaged waveforms. By doing on-line averaging, it may be possible to identify and correct artifacts such as this during the recording session rather than throwing out the subject's data afterwards.

Basic digitization software is fairly straightforward, so you don't need to worry too much about getting software that will work well. The only tricky aspect of digitizing the EEG is that both the EEG and the event codes must be acquired in real time, with no temporal errors. Modern computer operating systems make this difficult, so I would not recommend that you write your own digitization software unless you have considerable expertise with real-time programming.

Once you have purchased a digitization system, it is very important that you test it very carefully. The most common problem is that delays will be introduced between stimuli and the event codes stored on the digitization computer. In fact, my lab recently found that one of the most common commercial digitization systems introduced delays that increased gradually over the course of a trial block. When we contacted the manufacturer, they told us that this problem arises when their software is used with a particular brand of computer, and the problem disappeared when we switched to a different computer. But we never would have noticed the problem if we hadn't tested the system extensively. You cannot simply assume that your software works correctly.

To test a digitization system, you need to be able to record a reference signal that is triggered by your stimulus presentation system. The easiest way to do this is to use a square-wave calibration signal with an external trigger input (if your amplifier doesn't have such a calibrator, you can buy one). You can trigger the calibrator with your stimulus presentation software and see if the calibration pulse occurs at the same time as the event codes in the EEG data file. The square wave should start at the same time as the event code, and you should test this over a period of many minutes to convince yourself that the timing does not drift over time. Depending on the nature of your system, you may occasionally see an offset of one sample period; this can occur occasionally even if the timing only off by a microsecond or two. But if this happens on more than 10 percent of event codes, or if the errors are more than one sample period, you should contact the manufacturer.

The Data Analysis System

My lab uses a terrific package of custom ERP analysis software, but this software is not available to the general public. Thus, my advice in this section is based on what I would do if I did not have access to this system.

I have taken a look at some of the commercial ERP analysis systems that are available, and I haven't been favorably impressed by them. They are full of bells and whistles, but they are expensive and they don't seem to do exactly what I need them to do. One reason for this is that scientific research is generally focused on doing things that have never been done before or that need to be done in a way that depends on the details of the research domain. This makes it hard for software companies to write ERP analysis software that suits the specific needs of the diverse set of researchers and can anticipate future needs. They also try to make the software easy to use, but ease of use is usually inversely related to power and flexibility.

Instead of buying commercial software, you may want to write your own ERP analysis software. I've done a fair amount of this over the years, and it's very time consuming. Flexible data analysis programs that can be used in many experiments takes a very long time to write, especially in general-purpose programming languages such as C, Pascal, and BASIC. Special-purpose programs written for a single experiment take less time to write, but your research will go slowly if you have to write new programs for the analysis of every experiment.

Perhaps the best compromise is to write your own software, but to do it in a development system that is designed for numerical processing, such as MATLAB. MATLAB is designed for performing mathematical computations on arrays and matrices of numbers, and this is exactly what ERP waveforms are. That is, in a typical experiment, the averaged data can be described as a time × electrode × condition × subject matrix of voltage values. MATLAB contains built-in routines for quickly performing mathematical operations on matrices, and these routines are accessed from a relatively simple but powerful programming language.

To make data analysis with MATLAB even easier, two groups of researchers have developed free, public-domain MATLAB libraries for analyzing EEG/MEG and ERP/ERMF data. One of these packages is called EEGLAB (Delorme & Makeig, 2004, 〈http://sccn.ucsd.edu/eeglab/〉). This package includes routines for basic functions, such as artifact rejection and averaging, and it also includes excellent support for some advanced functions, particularly frequency-based analyses and independent components analysis. And it can import EEG data files from a variety of data acquisition systems. The second package is called BrainStorm (Baillet et al., 1999, ⟨http://neuroimage.usc.edu/brainstorm/⟩). It focuses primarily on source localization techniques.

Both of these packages provide a graphical user interface, which is very useful for new users. However, it is also possible to type written commands, and this makes it possible to generate scripts, which are extremely valuable for experienced users. Moreover, once you learn a little bit of MATLAB, you can add your own functions. I would definitely take this approach if I were setting up an ERP lab and didn't have access to my current data analysis package.

Assuming that these free software packages will not fill 100 percent of your needs, this approach requires that you are (a) reasonably competent at computer programming, (b) willing to spend a considerable amount of time learning computer programming, or (c) willing to hire someone else to do some computer programming. If you are a graduate student and you are planning to do ERP research extensively in graduate school and beyond, it would be worth learning some programming. If you are already an advanced researcher and you are planning to augment your research with a modest number of ERP experiments, you probably won't have time to learn to program and write the programs you need.

If you can't do the programming yourself, and you can't pay someone else to do it for you, then you'll have to buy one of the commercial ERP analysis systems. I can't tell you what package to buy, but I can make a few suggestions about some features that are important:

16. Easy methods for averaging the data on the basis of sophisticated criteria, such as sequences of stimuli, correct versus incorrect responses, and responses with various different reaction times. Response-locked averages must be possible.

17. A broad set of artifact rejection functions, such as those described in chapter 4. You should be able to set the parameters on the basis of visual inspection of the EEG. It is also useful to be able to do artifact rejection manually on the basis of visual inspection.
18. The ability to filter the raw EEG and the averaged ERP waveforms with a variety of half-amplitude cutoffs. The impulse response functions of the filters must be clearly described, and the inclusion of filters with gaussian impulse response functions is highly desirable (see chapter 5).
19. A broad set of component measurement routines should be available (see chapter 6).
20. The ability to plot topographic maps showing the distribution of voltage (or current density) over the scalp, preferably using the spherical spline interpolation algorithm (see Perrin et al., 1989).
21. It must be easy to perform mathematical operations on the ERP waveforms, including re-referencing the data and forming difference waves.
22. The ability to import and export data in different formats, particularly text files, is very useful.
23. The ability to automate processing can be very useful. For example, you may find that you need to re-average the data from each subject in an experiment with a longer time epoch, and you will save a lot of time and effort if you can automate this process.
24. Convenient statistical analyses are good to have. You can use standard, general-purpose statistical packages instead, but it is useful to be able to automate the process of measuring and analyzing the data, which is difficult with most general-purpose packages.
25. A convenient but flexible means of plotting the ERP waveforms is essential. You must be able to specify the line types and line colors used for overlapping waveforms, the placement of these sets of overlapping waveforms, and the formatting of the time and voltage axes (see the section on plotting at the beginning of chapter 6). In addition, the system must be able to save the plots in a vector-based file format that can be imported by standard vector-based

graphics packages (bitmapped outputs are not usually suitable for creating figures of ERP waveforms for publication).

The Stimulus Presentation System

Timing of Event Codes

In general, the key to good ERP stimulus presentation is precise timing of the event codes with respect to the stimuli. Modern computer operating systems are not designed to operate in real time, and programs are often interrupted briefly so that the operating system can conduct various housekeeping tasks. These interrupts are not usually noticeable during normal computer usage, but they can introduce timing errors that are large relative to the time scale of ERP recordings (as much as 200 ms). Real-time programming and precise timing was actually easier to achieve with more primitive systems, such as MS-DOS and the Apple II's operating system. Consequently, you should not attempt to create your own stimulus presentation software unless you really know what you are doing (or have access to software development systems that simplify real-time control).

In addition, you should not assume that commercially available stimulus presentation software has the level of precision necessary for ERP recordings. About 15 years ago, I tested one of the most popular stimulus presentation programs of the time, and I found very significant timing problems (it was truly shocking!). Thus, whatever system you use, you should verify the precision of its timing (I'll explain how to do this in the next paragraph). And keep in mind that precision is the key, not accuracy (as these terms are technically defined). That is, a constant, known delay between the stimulus and the event code is fine, because it can be subtracted away by shifting the averaged waveforms by the delay factor. An unpredictably varying delay is not acceptable, however, because there is typically no way to compensate for it.

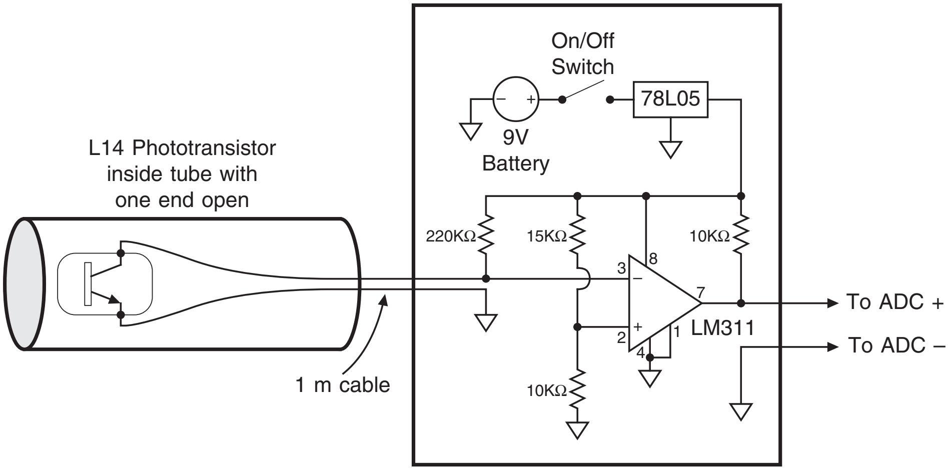

Figure 8.2 A circuit that can be used to measure the light emitted by a small region of a video display. The output can be connected directly to the analog-to-digital converter of your EEG digitization system (assuming that the voltage range is correct). Thanks to Lloyd Frei for designing this circuit.

The best way to test the timing of a stimulus presentation system is to somehow record the stimulus with your data acquisition system. With auditory stimuli, for example, you can place a microphone in front of the speakers and record the microphone's output as if it were the EEG. You can then use your data analysis software to see if the onset of the stimulus occurs exactly at the time of the condition code. This is a bit more difficult with visual stimuli, but figure 8.2 shows a circuit for a simple device that can be used to measure the light being generated by a video monitor. You can point this device toward the part of the screen where the stimuli appear and record the output on your data acquisition system. When you do this kind of testing, I would recommend using a higher-than-normal sampling rate to record the signals (1000 Hz or greater). This will make it easier to see these signals, which may contain very fast transitions. In addition, you should record a large number of stimuli so that you can test for the presence of occasional large timing errors. If you find timing errors, you may be able to eliminate them by disabling any unnecessary hardware and software systems on the computer. For example, you can make sure that no other programs or operating system extensions are running in parallel on the computer, and you can disconnect the computer from networks, unnecessary hard drives, printers, and so on.

Stimulus-Related Artifacts

Whatever stimuli you use, you should make sure they will not cause electrical artifacts that will be picked up by the electrodes or cause reflexes that will contaminate the data. For example, most audio headphones operate by passing a current through a coil to cause the movement of a membrane with respect to a magnet, and this can induce a current in the electrodes. Thus, shielded headphones may be important for auditory ERP studies. In addition, sudden noises can elicit a muscle twitch called the post-auricular reflex, which the EEG electrodes can pick up. Somatosensory stimulation is most often produced with small shocks, and these can also induce artifacts in the electrodes and elicit reflexes.

Stimulus Timing on CRTs

The remainder of this chapter will focus on presenting visual stimuli on a computer-controlled cathode-ray tube (CRT) monitor (see Brainard, Pelli, & Robson, 2002 for a more detailed discussion). There are three reasons for focusing on visual stimuli. First, visual stimuli are used much more commonly than stimuli in other modalities. Second, the presentation of visual stimuli on a CRT involves some subtle timing issues that are particularly important in ERP experiments. Finally, I just don't have much experience with stimuli in other modalities, so I can't provide much detailed information. You can learn about the details of non-visual stimulation from the literature or by asking experts.

CRT Basics CRTs pose special problems for stimulus timing, and you need to understand how computer-controlled CRTs operate in order to avoid timing errors. As figure 8.3A illustrates, CRTs operate by passing a narrow electron beam through a magnetic field, which directs it to a specific location in a phosphor layer at the front of the monitor. When the electron beam hits the phosphor, the phosphor emits light that is proportional to the intensity of the electron beam. The location that is struck by the electron beam at a given time corresponds to a single pixel on the monitor. A color CRT uses three beams pointed at slightly different locations containing phosphors that emit red, green, and blue light; for the sake of simplicity, however, we will assume a single electron beam except as noted.

The electron beam activates only one pixel at a given moment. The magnetic field is modulated moment by moment to control which pixel is being activated. In an oscilloscope, the electron beam can be moved around in any desired pattern. A CRT, however, uses a *raster* pattern, and the electron beam is called a *raster beam* (see figure 8.3B). The raster beam starts in the upper left corner of the screen and illuminates all of the pixels on the top row of the monitor, one by one, with an intensity that is adjusted individually for each pixel. It then shifts down to the next row and draws the next set of pixels, and so on until it reaches the bottom right corner, at which point every pixel on the monitor will have been illuminated by some amount. The beam is then turned off briefly and moves back to the upper left corner of the monitor, and the whole process is repeated. The rate of repetition (the *refresh* rate) is usually between 50 and 100 Hz, corresponding to a time between refreshes of between 20 and 10 ms, respectively.

The intensity of the raster beam for each pixel is determined by a series of numbers stored in the video card's *frame buffer*. The frame buffer is just a contiguous set of memory locations, where each location contains the intensity value for a given pixel (or three values in the case of a color CRT, one each for the red, green,

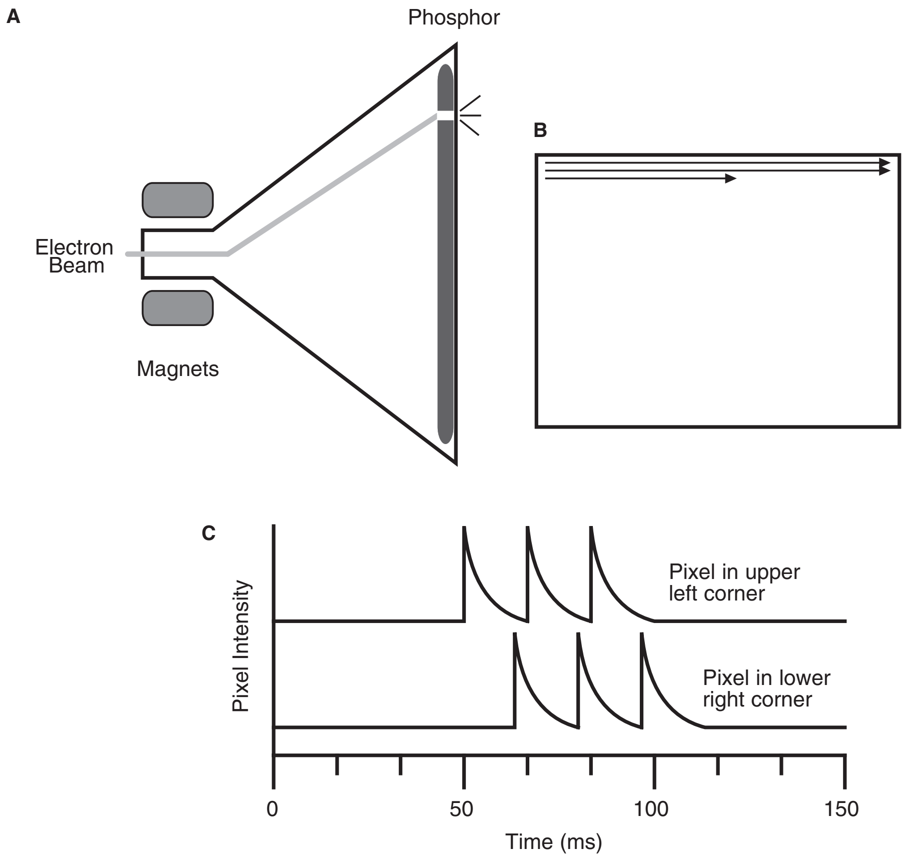

Figure 8.3 Operation of a CRT monitor. (A) The electron beam is directed by a magnetic field to a specific location at the front of the monitor, causing a small spot in the phosphor layer to become illuminated. (B) Raster pattern of the movement of the electron beam over time. (C) Intensity of the luminance from pixels on two different locations on a CRT monitor.

and blue intensities). When a program draws an image to the monitor, it is really just updating values in the frame buffer. No change occurs on the monitor until the raster beam reaches the set of pixels corresponding to the updated values, at which point the values are read from the frame buffer and the pixels are activated accordingly. This is a big part of why timing is so tricky for CRTs: the time at which something actually appears on the screen is determined by both the contents of the frame buffer, which your program directly controls, and the position of the raster beam, which your program does not directly control.

CRT Timing As illustrated in figure 8.3C, a pixel's intensity increases almost instantaneously when the raster beam hits the phosphor, and then the intensity falls fairly rapidly toward zero when the raster beam moves away As an example, consider a scenario in which the refresh rate is set at 60 Hz (16.67 ms per refresh), the raster beam starts at the upper left corner at time zero, and the entire screen is to be turned on at 50 ms, remain visible for 50 ms, and then be turned off. The top trace in figure 8.3C shows the intensity of the pixel at the top left corner of the screen for this scenario. The intensity of the raster beam will be low when it hits this pixel at times 0, 16.67 ms, and 33.33 ms, and then transition to high when it hits the pixel at times 50, 66.67, and 83.33 ms, returning again to a low value at 100 ms. The pixel's illumination will spike upward at these times and then decay downward in the intervening periods. We don't perceive it as flickering, however, because the retina integrates information over time.[1]

Less than a microsecond after the pixel in the upper left corner is illuminated at 50 ms, the next pixel to the right will be illuminated, and then the next one, and so on until the entire top row has been illuminated (drawing one row of pixels will take approximately 120 μs with a 640 × 480 resolution). The pixels in the next row will then be sequentially illuminated, and then the next row, and so on until the entire display has been illuminated (which will take approximately 16 ms with a 60-Hz refresh rate). The bottom trace in figure 8.3C shows the actual illumination that would be observed for a pixel near the bottom of the display. The illumination of this pixel is delayed by approximately 16 ms compared to the pixel in the upper left hand corner of the display. Thus, the display is not drawn in a single instant, but instead is gradually painted over a period of many milliseconds.

Timing Errors If your stimulus presentation program draws an object at some random time, it may cause a large, visible artifact known as *tearing*. This happens when the frame buffer is being updated in the same set of locations that are currently being drawn to the display by the raster beam. It is important to avoid this artifact, which can be very distracting.

To avoid tearing and permit precise timing, video cards send a signal to the computer called a *vertical retrace interrupt* or *video blanking interrupt* just after the raster beam has finished a cycle of drawing and just before the next cycle is about to begin. In most cases, the best way to ensure correct timing is to use the following sequence of events in the stimulus presentation program: (a) wait for the interrupt signal, (b) send an event code, and (c) draw the stimuli in a manner that ensures that any changes in the intensity values stored by the frame buffer occur after the interrupt but before those values are read by the raster beam.

By following this sequence, you can be sure that your event code is linked to the position of the electrode beam and therefore to the moment at which the illumination of the monitor actually changes. There will be a delay between the event code and the illumination of a given pixel, and this delay will depend on the location of the pixel (i.e., it will be larger for lower rows than for higher rows). However, this delay is constant for a given location, and you can determine the amount of delay by recording the illumination at a given location with the device shown in figure 8.2C. A known, constant delay is a minor annoyance, whereas an unknown, variable delay can be a major problem.

The tricky part of this sequence is the last part, making sure that you do not change values in the frame buffer after those values have already been used to control the raster beam on a given raster cycle. There are two common ways this can occur. First, if you are drawing a complex stimulus or have a slow computer or software system, it may take longer than one raster cycle to draw the stimulus. Second, you may not draw from top to bottom, and a portion of the stimulus display may be drawn to the frame buffer after the raster beam has already drawn this portion of the display. For example, imagine you are conducting a visual search experiment in which twenty-four squares are drawn at random locations across the display. If you wait for the vertical retrace interrupt and then start drawing the squares to the frame buffer in random order, you may end up drawing a square near the top of the frame buffer 5 ms after the interrupt. By this point, the raster beam will have already drawn the top portion of the frame buffer, so you may see tearing artifacts, and the square will not appear until early in the next raster cycle.

In most cases, the best way to avoid these problems is to predraw each stimulus display in an offscreen memory buffer. That is, prior to each trial of the experiment, you can predraw the stimulus displays in an area of memory that simulates the frame buffer but is not actually visible. When it is time to display the stimulus, you will wait for the vertical retrace interrupt, send an event code, and then copy the simulated frame buffer into the actual frame buffer. As long as the copying is done from top to bottom and is faster than the raster beam, this will guarantee that your timing is perfectly precise (although pixels near the bottom of the display will still be illuminated later than pixels near the top of the display). Alternatively, some video cards contain multiple frame buffers, and it is possible to predraw the stimuli in one frame buffer while another frame buffer is being displayed. The predrawn frame buffer can then be displayed by telling the video card to switch the displayed location, which is virtually instantaneous. No copying from memory into the frame buffer is necessary in this case.

Software Packages Various stimulus presentation programs are available, and some are designed expressly for use in ERP experiments. Before purchasing a program, you should inquire about the nature of the video timing. Every vendor will tell you that the timing is accurate, but you should find out exactly how the program works. In particular, you should find out whether the program (a) synchronizes stimulus presentation and event codes to the vertical retrace interrupt, (b) predraws each display to an offscreen memory buffer, and (c) might be occasionally interrupted by other programs or the operating system, which may cause large timing errors.

If you are inclined to write your own visual stimulus presentation programs, I would highly recommend using MATLAB and a set of routines called the PsychToolbox. This toolbox was developed by two highly respected vision researchers, David Brainard and Dennis Pelli (Brainard, 1997; Pelli, 1997), and they have rigorously tested its timing. MATLAB provides an excellent environment for the relatively rapid development of new experiments, and the PsychToolbox makes it easy to implement the sequence of procedures described here.

CRTs versus LCDs At the time of this writing, CRT monitors are quickly being displaced by LCD monitors, and this change may obviate much of the information provided so far. Unfortunately, LCD technology has not stabilized enough for me to provide useful information about how to achieve precise timing with LCDs.

LCD displays operate by using electrical signals to change the polarization of display elements, which in turn influences the transmission of light. No raster beam is necessary, so LCDs can potentially make instantaneous changes to pixels in any position on the screen. However, LCDs are being integrated into computer systems that were originally designed with CRTs in mind, and information from the video card's frame buffer is typically transmitted serially to the LCD display. Thus, it is still important to consider carefully the timing of the stimuli with respect to the event codes.

The optical properties of LCDs can also be problematic. In the context of ERP recordings, the biggest problem is that the transmission of light by LCDs changes relatively slowly. As a result, stimulus onsets and offsets will be more gradual than on CRTs. Fortunately, LCD manufacturers are motivated to reduce this problem because of the growing use of LCDs for television displays, video games, and other high-speed applications. Thus, LCDs may soon have better temporal properties.

To examine LCD timing firsthand, I used the circuit shown in figure 8.2 to measure the output of a high quality LCD monitor, and I recorded this signal with an ERP digitization system. Rather than using a digital LCD interface, I connected the LCD display to the computer's analog VGA output, which is designed for CRTs. Most current LCDs have both analog and digital inputs, and the digital

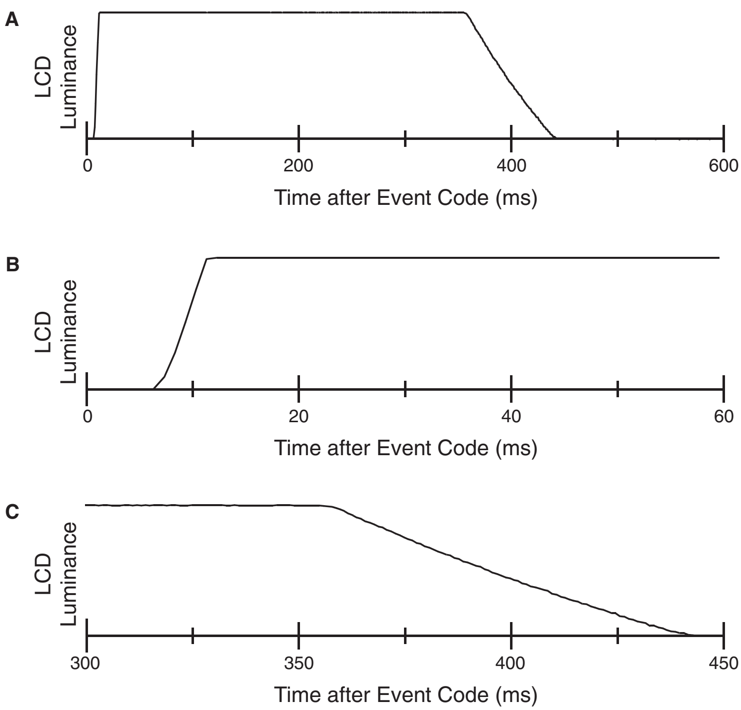

Figure 8.4 Luminance recorded from an LCD monitor in response to a white square presented on a black background. The same luminance signal is shown on three different time scales.

input will usually have higher fidelity (but may have unknown timing properties). Figure 8.4 illustrates the output of the LCD at the location of a white square that was presented on a black background. The square was supposed to onset at 0 ms (the time of the event code) and offset at 350 ms (and this is exactly what would have happened if I had used a CRT monitor).

Figure 8.4A shows the entire time course of the LCD output; the onset and offset appear on expanded time scales in figures 8.4B and 8.4C, respectively. The figure illustrates three potential problems with LCDs. First, the LCD's output is slightly delayed: The onset of the luminance change does not begin until approximately 7 ms after the event code, and the luminance does not begin to decline until approximately 357 ms. I suspect that this is caused by the analog VGA connection; the analog signal must be converted back into a digital form by the LCD monitor. Second, the onset is not instantaneous, but instead builds up over a period of approximately 5 ms. This is probably fast enough for the majority of ERP experiments, but this particular LCD display was chosen for its rapid onset time, and other LCD displays ramp up over a substantially longer period. Third, the offset of the display is very gradual, requiring almost 100 ms to reach the baseline luminance value. This won't be a significant problem when stimuli are presented on a black background. However, if the stimuli are presented on a white background, the onset of a stimulus will be achieved by a decrease in luminance, and the luminance will change in the slow manner shown in figure 8.4C. This could be quite problematic for experiments in which precise timing is important.

LCDs do have a very significant advantage over CRTs: They don't pump a stream of electrons directly toward the subject's head. Consequently, LCDs should produce less electrical noise than CRTs, and a Faraday cage may not be necessary. LCDs may therefore become the best choice for most experiments once the technology reaches maturity. In the meantime, you may wish to try using LCD monitors, but you should carefully test the timing of an each display before using it (different models may have radically different temporal properties).