A general framework for defining evoked potential paradigms

HOWARD L. KAPLAN and NANCY E. NOLDY
Addiction Research Foundation, Toronto, Ontario, Canada

We have developed a framework for defining a range of evoked potential paradigms through the completion of screen forms rather than through the modification of program code. Within this framework, an experimenter can specify stimulus events and their timing, randomization of different stimuli, collection of responses, rejection of artifacts, and detection of peaks within the averaged waveforms. Printed copies of the screen forms and of timing diagrams derived from them can later be incorporated into the documentation of the experiment.

The evoked potential methodology can cover a wide range of experimental designs. In all of these designs, potentials generated by brain activity are sampled before and after scheduled stimulus presentations, and the responses to stimuli in the same class are combined by an appropriate averaging strategy.

Taken one at a time, many of the stimulus-scheduling strategies are straightforward to program: one stimulus every 2 sec, one stimulus every 1 to 3 sec distributed uniformly over that range, a pair of clicks separated by 100 msec every 4 sec, and so forth. However, effective computer programs to implement these strategies must also include large volumes of other code to provide such services as digitization of analog data, real-time graphic display of data, postexperimental manual adjustment of cursors indicating peak responses, detailed printed reports, and so on. When the experimental paradigm is changed, this larger body of support code might at least need to be recompiled or relinked and might need to be revised to accommodate the new paradigm. If there are several paradigms in active use, each one implemented as a separate version of a master program, then the task of propagating changes and improvements through this set of programs can become a tedious and error-prone exercise in software configuration management. It would not be surprising to find that any programmer responsible for maintaining such a collection of paradigms would seek an alternative strategy.

To reduce the demands on the programmer role and to give experimenters the flexibility to modify their paradigms without waiting upon programming services, we have tried to design a general framework for defining evoked potential studies. Within this framework, a series of five screen forms lets experimenters specify the nature and timing of stimuli, the relative odds with which the different stimuli are presented, the criteria for rejecting and rescheduling trials, the duration and sampling fre-

quency of analog response collection, and the rules for extracting peaks and averages from the summary records.

The Hardware and Software Environment

The evoked potential procedure described here executes under the general control of a program called the Scheduled Measurement System (SMS; Kaplan, 1985, 1987). SMS executes on an IBM-PC class computer equipped with a CGA video card, an analog-to-digital converter (we use the Data Translations 2801), and an 8087 coprocessor; for practical evoked potential work, a fast processor (PC-AT class or equivalent), a hard disk, and at least 512K of RAM are required. SMS provides task scheduling, analog-to-digital sampling, video display, form management, and data management services for a variety of tasks, including manual tracking, memory, multiple-choice questionnaires, spontaneous EEG, and evoked potentials. The timing grain for stimulus presentation and for duration of response sampling is 16.67 msec, or $\frac{1}{60}$ of a second, synchronized with the computer's CGA video display refresh cycle. This timing base was chosen for tasks other than evoked potentials; our visual evoking stimuli are not displayed on a video monitor. Responses on up to eight simultaneous channels can be sampled at any multiple of 60 Hz, up to 960 Hz per channel. Because of this limited sampling rate, this software is suitable only for the slower paradigms; it cannot, for example, be used for brainstem auditory evoked potentials.

Timing Diagrams

Each paradigm consists of a sequence of discrete trials, with each trial divided into three parts. Figure 1 shows the timing diagram of the simplest kind of paradigm—response to a single click. The first part of the trial is the nonsampling time between trials; this can be of either fixed or varying duration. In Figure 1, it is shown as ranging from 2,000 to 2,500 msec, with the dotted portion of the axis representing the range of uncertainty. The next part is the prestimulus period, during which potentials are sampled but no stimuli are presented (shown as 300 msec in

Correspondence may be sent to Howard L. Kaplan, Addiction Research Foundation, 33 Russell St., Toronto, Ontario M5S 2S1, Canada.

Figure 1. The timing diagram for a simple paradigm, a click repeated at random intervals.

Figure 1). Time zero marks the end of the prestimulus period and the start of the poststimulus period (1,500 msec in this example); stimulus events can occur at or after this point. The intertrial interval, or ITI, is measured from the previous trial's time zero to the current trial's time zero; in this example, it ranges from 3,800 to 4,300 msec, being the sum of 2,000 to 2,500, 300, and 1,500 msec. The timing diagram also shows, under the heavy line indicating the sampling duration, lighter horizontal lines that indicate the windows within which the program looks for peak responses.

Figure 2 shows the timing diagram for a more complex kind of paradigm. In this paradigm, a single click is presented every 2,000 to 3,000 msec, uniformly distributed. However, on one fifth of the trials, that click is preceded 250 msec earlier by a 100-msec warning-light flash; on the remaining four fifths of the trials, no such warning is given. In such a paradigm, the question of interest is how the response to the click is modified by the presence of the earlier warning light. We will call this paradigm 20% LIGHT and use it as an example as we explain the software in more detail.

The paradigms are described by filling in screen forms; examples of such forms are shown in Figures 3 through

7. Field names are shown in an ordinary font, and field contents are shown in reverse video, with the length of the reversed area indicating the field length. Standard cursor control keys allow the user to navigate the form, moving from field to field to edit their contents.

After the experimenter completes these screen forms, the program automatically represents their contents as a timing diagram for the experimenter's review. Both the forms and the diagrams can be printed as hard copy. These documents, being in an experimenter's language rather than in a programmer's language, form a useful and easily understood record of the differences among experiments. The timing diagrams are also useful in subsequent publications, particularly when describing novel paradigms.

Paradigm Definition

To simplify the experimenter's task, part of the paradigm definition is separated out as stimulus definitions that can be reused in different paradigms. The two stimulus definition forms for the example paradigm are shown in Figure 3. Each form describes a sequence of two or more events occurring at the output interface, where each change in the output bit mask is considered

Figure 2. The timing diagram for the paradigm 20%LIGHT, in which a click that occurs on every trial is preceded by a 100-msec warning light on 20% of the trials. When there are two or more stimuli, as in this example, we have adopted the convention of dividing the height above the time line into several bands, one for each stimulus. In this case, the band adjacent to the time line represents the click, and the more remote band represents the light.

a separate event. For example, to turn on a stimulus for 50 msec, one event consists of setting the bit controlling the stimulus and a second event consists of resetting that bit. For designs involving stimuli that might be presented simultaneously, several bits can be changed at once. When external devices have their own timing controls, the output duration specified here becomes arbitrary and is normally chosen to be the minimum that can be generated within the system timing grain, 17 msec.

There is not necessarily a stimulus event happening at time zero. In some paradigms, for example, there may be one tone fixed at 1,000 msec after time zero and a prior tone that, on different trials, occurs at time zero, 500 msec later, or 750 msec later than time zero, or equivalently, at 1,000, 500, or 250 msec before the second tone.

Once a stimulus has been defined, it is not limited to use in a single paradigm; a number of different paradigms can all reference it by its name, such as CLICK or 2T/500. These names are generally chosen to describe both the

modality and the timing; the latter example might be two tones presented 500 msec apart.

In the paradigm 20%LIGHT, the two stimuli are called NCL250 (nothing, followed by a click 250 msec later) and LCL250 (a light, followed by a click 250 msec later). The timing diagrams for these are shown in Figure 2. Figure 3 shows how the events are specified. In both stimuli, the third event is the onset of interface bit 0 (the rightmost bit) at 250 msec, and the fourth event is its offset at 267 msec; this causes external hardware to generate a click. In stimulus LCL250, the first event is the onset of interface bit 1 at 0 msec, and the second event is its offset at 100 msec. For consistency, the first two events also occupy space in the definition of stimulus NCL250, but no bits change at the interface. This allows the click onset to be uniformly referred to as event 3.

The remainder of the paradigm definition requires four forms, shown in Figures 4 through 7. The first one, shown in Figure 4, describes some of the hardware details of the port and names the stimuli that are generated at that port

Stimulus description NCL250						
N of events	+4					
Time in msec	0	17	250	267		
New bit mask	0	0	1	0		

Stimulus description LCL250						
N of events	+4					
Time in msec	0	100	250	267		
New bit mask	2	0	1	0		

Figure 3. The stimulus definition forms for the two stimuli, NCL250 and LCL250, used in paradigm 20%LIGHT.

and are used in the paradigm. Logically, all of this hardware information might be placed on the stimulus forms instead; it is perhaps arbitrary to place it here, but this does save repeating it for each of the stimuli. The first two lines simply define the port addresses. The last line of the form assigns names to the various stimulus bits; these names are used only in drawing the timing diagram.

The third line of this form, port preset logic, indicates whether or not there is special hardware to guarantee the synchronization of stimulus presentation and response collection. Normally, each screen refresh cycle ($1/60$ of a second) begins with the processing of an interrupt; the analog-to-digital converter is also synchronized with this interrupt. While the first analog samples of the refresh cycle are being collected, the computer's interrupt processing routine is executing, performing tasks that might take 1 or 2 msec. If a scheduled stimulus event were not to oc-

cur until the evoked potential program regained control of the CPU after this interrupt, then it might be presented up to 2 msec after its intended time. This is perhaps unimportant in the time scales involved in these experiments, but it seems safer to avoid the problem than to ignore it. To circumvent the problem, some interface bits are equipped with port preset logic, consisting of a latch that delays the output of any change in the port until the next screen refresh interrupt. Using this logic, an event scheduled to occur at 500 msec after time zero is actually sent to the port at any point in the 17-msec interval preceding the 500-msec point, and this change is copied to the latch's output bit at exactly 500 msec, regardless of what the CPU is doing at that time.

The second paradigm form, shown in Figure 5, describes the timing and randomization parameters. The items that define the sampling periods and the ITIs are

Paradigm description 20%LIGHT, port selection					
Data port address	+300				
Control port address	0				
Port preset logic?	TRUE				
Bit names	CLICK	LIGHT			

Figure 4. The paradigm definition form, part one of four, used to describe the hardware environment for paradigm 20%LIGHT.

Paradigm description 20%LIGHT, stimulus section						
Pre-stimulus msec	250					
Post-stimulus msec	1250					
Base & usual trials	10	100				
Constrained	TRUE					
Stimulus classes	2					
Stimulus name	NCL250	LCL250				
Condition name	NO LIGHT	LIGHT				
Base count	8	2				
Minimum ITI, msec	2000	2000				
Maximum ITI, msec	3000	3000				

Figure 5. The paradigm definition form, part two of four, used to describe the stimulus presentations for paradigm 20%LIGHT.

straightforward; because of the basic system timing grain, they must be rounded multiples of $\frac{1}{60}$ of a second; that is, they must end in 00, 17, 33, 50, 67, or 83. The paradigm includes one or more stimulus classes, where responses are averaged within a class but not across classes. The classes are allowed, but not required, to have different stimuli and different ITIs. In Figure 5, the two stimuli of paradigm 20%LIGHT share an ITI of 2,000 to 3,000 msec. If the same stimulus is named two or more times with identical ITIs, the separate averages in the several classes provide a rough estimate of the variability of the response statistics. If the same stimulus is named two or more times with different ITIs, the condition name fields allow the responses to be separated on the output; otherwise, the stimulus names are simply applied to the corresponding averages.

Randomization of the presentation of the stimulus classes is controlled by the overall base pool and each condition's base counts. The base pool consists of the total odds, and each stimulus class's base count consists of the odds of presenting that class. If sampling is constrained, then sampling occurs without replacement until all stimuli have been presented as many times as their odds count, at which point the pool is replenished and sampling re-

sumes. In this example, the base pool size is 10 and the two classes have base counts of 8 and 2. Therefore, out of each consecutive block of 10 trials accepted (not rejected for response artifact), there are exactly 8 presentations of stimulus NCL250 and 2 presentations of stimulus LCL250, in a random sequence. It is also possible to sample the odds without replacement, should that be desired.

The third form, shown in Figure 6, describes all of the response processing other than that involved in detecting individual peaks. Input channels are numbered 0 through 7, with a (possibly empty) initial subset consisting of channels used to reject trials for the presence of EOG artifact. Each channel is given a name indicating the placement of electrodes on the scalp. If an EOG channel exceeds its threshold, or if any other channel exceeds a secondary, possibly different, threshold, then the data from that trial is ignored and the trial is put back into the pool for rescheduling later. Other items on this form note calibration information and control the formatting of graphic displays.

The last form, shown in Figure 7, defines the parameters for automatic peak detection. Up to 10 peaks can be detected, with their synchronization and windows

Paradigm description 20%LIGHT, response section				
Sampling rate	-500			
Last channel	2			
Last EOG channel	0			
Channel name	EOG	FZ A2	CZ A2	
Usual gain (uV/mm)	7			
Calibration uV	50			
Usual rejection uV	75	75		
Connect?	FALSE			
Graphs per 1/3 page	3			

Figure 6. The paradigm definition form, part three of four, used to describe the response collection and rejection for paradigm 20%LIGHT.

specified separately. Each peak is of one of three types: positive, negative, or mean. Positive and negative peaks are simply the point at which the waveform achieves its maximum or minimum voltage within the specified window. Means are not really peaks, but they fit into the peak-defining framework and are simply the average voltage over a temporal window. Such means have two uses: (1) they can serve as secondary baselines for measuring subsequent peaks and (2) in the measurement of some slow components, they are interesting dependent variables in their own right.

Each peak is defined with reference to a stimulus-event number, which is assumed to be equally relevant across all stimuli. The window during which the peak is detected is then taken relative to that stimulus event. In the example paradigm, responses to the initial warning tone are ignored, and only responses to the click, which is event 3, are processed. Any peak that is specified to have synch event 3 is detected in a window that has a fixed time relationship to the onset of the click, regardless of whether a tone preceded the click. (In this example, the click always occurs 250 msec after time zero, but responses can

also maintain their synchronization to events that occur at different time points in different stimuli. For example, if each stimulus consisted of two clicks with a varying interval between them, this could track responses to the second click regardless of when it occurred.) For the computation of secondary baselines, the windows in which peaks are measured can precede the associated stimulus events, that is, have negative starting and stopping times.

Any peak can be measured with respect to any earlier peak or mean. This allows, for example, the direct measurement of such statistics as a negative-positive peak difference or the difference between two successive peaks of the same polarity. In this context, peak 0 is not a true peak; instead, it is the mean voltage during the prestimulus period preceding time zero, which is available as a reference for measuring subsequent peaks.

Because we cannot always conduct enough trials to remove the noise inherent in evoked responses, we have the option of applying a straightforward n -point smoothing to the data before the detection of peaks. This can be specified on a subject-by-subject basis after the data are collected, but the initial smoothing can also be specified

Paradigm description 20%LIGHT, analysis section					
Display range	+10				
Number of peaks	4				
Peak name	BASE	N100	P200	PN DIFF	
Pos/Neg/Mean	MEAN	NEG	POS	POS	
Peak synch event	3	3	3	3	
Peak start msec	-150	75	150	150	
Peak stop msec	0	120	240	240	
Reference peak	0	1	1	2	
Usual smoothing	1				

Figure 7. The paradigm definition form, part four of four, used to describe the peak detection logic for paradigm 20%LIGHT.

on this last paradigm definition form. In the example, 1-point smoothing is specified; this presents the data in their raw form.

Real-Time Data Collection

The emphasis of this presentation is on the paradigm definition logic, so the real-time data collection will be described only briefly. The usual facilities for viewing input waveforms, checking calibration, and adjusting artifact rejection thresholds for individual subjects are present. During the run, responses are averaged in stimulus classes, and the experimenter can choose to view either the raw input waveforms or the running class averages. Trials rejected for artifacts are always displayed in raw form, with the rejection thresholds also shown on the screen as dashed lines (Figure 8). Only averages are retained; neither individual trials nor variance statistics are kept with these averages. An initial display at the run's end shows the automatically detected peaks on each channel separately, and the experimenter can review the data later to alter the display smoothing factor (the averages are always kept on disk in unsmoothed form), to review

the new peak cursor placements after resmoothing, and to manually adjust the cursors when the automatic rules do not provide what the experimenter thinks is an appropriate detection of the peak (Figure 9).

Capabilities and Limitations

One of our major research interests deals with the topics of excitability and recovery, that is, how the response to a stimulus is affected by the time elapsed since the preceding stimulus. Our software is particularly well suited for designing such experiments. For studies in which the interstimulus intervals are long, on the order of 750 msec to several seconds, the stimulus definition is a simple tone occurring at time zero, and the classes of trial differ only in their ITIs. In fact, we repeat each ITI in two separate but identical stimulus classes, using the separate averages collected as a measure of the response consistency. When we are interested in shorter interstimulus intervals, on the order of 200 to 700 msec, we create several separate stimulus definitions, each one consisting of two closely spaced tones, we let the ITI be consistent, and we measure the response to the second tone in each stimulus,

Figure 8. A simulated run-time display of a trial rejected for EOG artifact, showing the screen layout. (Actual eye movement artifacts do not resemble damped sine waves.)

much as we described above for the experiment 20%LIGHT. Differences in stimulus modality, duration, frequency or wavelength, and other physical parameters would likewise be handled on the stimulus definition screens, after hardware is constructed that can generate the necessary stimuli in response to the state of a single 8-bit output port.

There are, however, some common paradigms for which the current software is not suitable. Chief among these are studies in which the evoked potential response must be classified according to whether or not the subject makes another, overt response to classify the stimulus. Currently, there is no provision to collect such other responses. On the other hand, if subjects are simply asked to count stimuli of one class, the software can be used. The software also is unsuitable for some designs in which a mixture of stimuli (different frequencies, modalities, durations, etc.) is used, and it is necessary to classify each response according to both the current and previous stimulus class.

Concluding Remarks

The development of this software was motivated by the desire to avoid maintaining more than one version of a

very large program. There never was a version of this software that ran only a single paradigm; therefore, we cannot estimate how much of the current size is caused by the versatility per se. However, we can give some statistics about the current complexity of the program. The source code, written for compilation under Turbo Pascal 3.0, occupies about 150K in its usual form, and about half of that volume consists of comments and white space used solely to improve the readability of the source code. The compiled object code occupies about 50K, divided among several overlays. These statistics refer only to the code specific to evoked potentials; the source code for the SMS support services, which are shared among a number of measures, is about four times as large as the source code specific to evoked potentials.

In software of this magnitude, it is important to find ways to generalize the program's operation at critical points to allow for future expansion and new ideas. To take a very simple example, the ideas of "turn on a stimulus" and "turn off a stimulus" can be combined into "update a stimulus status bit." Once this is done, the program need not concern itself with the difference between responses to stimulus onsets and offsets. Similarly, the idea of a response window from 1,000 to 1,200 msec from

Figure 9. A display of an average waveform from an actual experiment, with cursors indicating the peaks and with a table of derived statistics. The experimenter can use the keyboard to change the smoothing factor or to adjust the peak cursors, in which case the statistics table automatically reflects the changes.

the start of the trial can be generalized to a response window whose reference is the n th stimulus event, where n is a parameter of the window. It is the repeated application of these generalizations that has allowed us to move so much of the paradigm definition from code to data, where it is conveniently under the experimenter's control.

REFERENCES

- KAPLAN, H. L. (1985). When do professional psychologists need professional programmers' tools? *Behavior Research Methods, Instruments, & Computers*, **17**, 546-550.
- KAPLAN, H. L. (1987). Incorporating good programs into the larger laboratory context. *Behavior Research Methods, Instruments, & Computers*, **19**, 210-214.