CHAPTER

4

Signal Averaging

4.1 INTRODUCTION

Data analysis techniques are commonly subdivided into operations in the ***time domain (or spatial domain)*** and ***frequency domain***. In this chapter we discuss processing techniques applied in the time (spatial) domain with a strong emphasis on ***signal averaging***. Signal averaging is an important technique that allows estimation of low-amplitude signals that are buried in noise. The technique usually assumes that:

1. signal and noise are uncorrelated,
2. the timing of the signal is known,
3. a consistent signal component exists when performing repeated measurements, and that
4. the noise is truly random with zero mean.

In the real world all these assumptions may be violated to some degree; however, the averaging technique has proven sufficiently robust to survive minor violations of these four basic assumptions. A brief overview of other frequently used time domain techniques can be found in Section 4.8.

4.2 TIME-LOCKED SIGNALS

Averaging is applied to enhance a time-locked signal component in noisy measurements. One possible representation of such a signal is as measurement x consisting of a signal s and a noise component n, with the underlying assumption that the measurement can be repeated over N trials. In the case where each trial is digitized, the k-th sample-point in the j-th trial (Fig. 4.1) can be written as:

$$x_j(k) = s_j(k) + n_j(k) \tag{4.1}$$

Signal Processing for Neuroscientists, Second Edition
https://doi.org/10.1016/B978-0-12-810482-8.00004-7

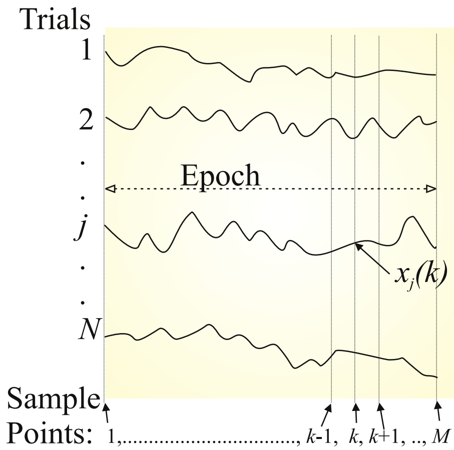

FIGURE 4.1 A set of N raw trials composed of a signal and significant noise component can be used to obtain an average with an enhanced signal-to-noise ratio. Each full epoch consists of a series of individual sample points $x_j(k)$, with $k = 1, 2, \ldots, k, \ldots, M$.

with k being the sample number ($k = 1, 2, \ldots, M$). The *rms* value of s can be several orders of magnitude smaller than that of n, meaning that the signal component may be invisible in the raw traces. After completion of N repeated measurements, we can compute an average measurement for each of the k sample indices:

$$x(k)_N = \frac{1}{N}\sum_{j=1}^{N} x_j(k) = \frac{1}{N}\sum_{j=1}^{N}\left[s_j(k) + n_j(k)\right] \tag{4.2}$$

The series of averaged points (for k from 1 to M) obtained from Eq. (4.2) constitutes the average signal of the whole epoch. In the following we explore some of the properties of signal averaging in a simulation.

The following MATLAB® routine pr4_1.m is a simulation of the averaging process.

```
% pr4_1
% averaging
clear

sz=256;
NOISE_TRIALS=randn(sz);              % a [sz × sz] matrix filled with noise
```

```
SZ=1:sz;                                  % Create signal with a sine wave
SZ=SZ/(sz/2);                             % Divide the array SZ by sz/2
S=sin(2*pi*SZ);

for i=1:sz;                                        % create a noisy signal
   NOISE_TRIALS(i,:) = NOISE_TRIALS(i,:) + S;
end;

average=sum(NOISE_TRIALS)/sz;                      % create the average
odd_average=sum(NOISE_TRIALS(1:2:sz,:))/(sz/2);
even_average=sum(NOISE_TRIALS(2:2:sz,:))/(sz/2);
noise_estimate=(odd_average-even_average)/2;

figure
hold
plot(NOISE_TRIALS(1,:),'g')
plot(noise_estimate,'k')
plot(average,'r')
plot(S)
title('Average RED, Noise estimate BLACK; Single trial GREEN, Signal BLUE')
```

As shown in the simulation result depicted in Fig. 4.2, the averaging process described by Eq. (4.2) results in an estimate of the signal. As compared with the raw (signal + noise) trace in Fig. 4.2, the averaged noise component is reduced over 256 trials. When averaging real signals, the underlying component may not always be as clear as it is in the example provided in Fig. 4.2. In these cases the averages are often repeated in search of consistent components in two or three replicates (e.g., see the superimposed SEP waveforms in Fig. 1.4). The idea here is that it is unlikely that two or more consistent averaged results will be produced by chance alone. A specific way of obtaining replicates is to average all ***odd*** and all ***even*** trials in separate buffers (see the superimposed odd_average and even_average in Fig. 4.2). This has the advantage of allowing for comparison of the even and odd results from interleaved trials. An average of the odd and even averages (i.e., addition of the odd and even results divided by 2) generates the complete averaged result, while the difference of the two constitutes an estimate of the noise (see Section 4.4 for details on such a noise estimate).

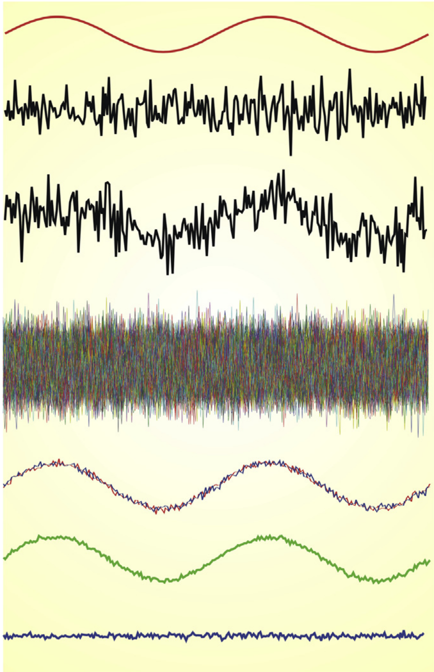

FIGURE 4.2 Signal averaging of a signal buried in noise (Signal + Noise). This example shows 256 **superimposed** trials of such a measurement and the average thereof. The average results of the odd and even trials are shown separately. The sum of all trials (Signal Average) shows the signal with a small component of residual noise. A ± signal average is shown as an estimate of residual noise. These example traces are generated with MATLAB® script pr4_1. An example of a recorded average of the brain's response to electrical nerve stimulation is depicted in Fig. 1.3.

4.3 SIGNAL AVERAGING AND RANDOM NOISE

The noise in Eq. (4.2) is a 0-mean random process: $\langle x(k)\rangle = \langle s(k)\rangle + 0$. Here, $< \ldots >$ denotes the average computed over a large number of trials ($N \to \infty$), equal to the true value of the enclosed variable. Therefore, the general idea of signal averaging is to reduce the noise term $\frac{1}{N}\sum_{j=1}^{N} n_j(k) \to 0$ for large N such that $x(k)_N \to s(k)_N$. Because in the real world $N << \infty$, we will not reach the ideal situation where the measurement x exactly equals the true signal s; there is a residual noise component that we can characterize by the variance of the estimate $\overline{x(k)_N}$. To simplify notation we will indicate $Var\left(\overline{x(k)_N}\right)$ as $Var(\overline{x})$. The square root of $Var(\overline{x})$is the standard error of the mean (SEM). We can use Eq. (3.11) to estimate$Var(\overline{x})$:

$$Var(\overline{x}) = E\left\{(\overline{x} - \langle x \rangle)^2\right\} = E\left\{\overline{x}^2 - 2\overline{x}\langle x \rangle + \langle x \rangle^2\right\}$$

Taking into account that $\langle x \rangle$ represents the true average value of x (therefore $E\{\langle x \rangle\} = \langle x \rangle$ and $E\left\{\langle x \rangle^2\right\} = \langle x \rangle^2$), we may simplify:

$$E\left\{\overline{x}^2 - 2\overline{x}\langle x \rangle + \langle x \rangle^2\right\} = E\{\overline{x}^2\} - 2\langle x \rangle E\{\overline{x}\} + \langle x \rangle^2$$

Further, we note that the expected value of the average of x ($\overline{x}$) is equivalent to $\langle x \rangle$ (i.e., $E\{\overline{x}\} = \langle x \rangle$), the above expression can be simplified further leading to:

$$Var(\overline{x}) = E\{\overline{x}^2\} - \langle x \rangle^2 \tag{4.3}$$

Combining Eqs. (4.3) and (4.2), we obtain:

$$\begin{aligned} Var(\overline{x}) &= E\left\{\left[\frac{1}{N}\sum_{i=1}^{N} x_i\right]^2\right\} - \langle x \rangle^2 = E\left\{\left[\frac{1}{N}\sum_{j=1}^{N} x_j\right]\left[\frac{1}{N}\sum_{i=1}^{N} x_i\right]\right\} - \langle x \rangle^2 \\ &= \frac{1}{N^2}\sum_{j=1}^{N}\sum_{i=1}^{N} E\{x_j x_i\} - \langle x \rangle^2 \end{aligned} \tag{4.4}$$

The two summations in this expression represent all combinations of i and j, both going from 1 to N and therefore generating $N \times N$ combinations. This set of combinations can be separated into N terms for all $i = j$ and $N^2 - N = N(N-1)$ terms for $i \neq j$:

$$Var(\overline{x}) = \underbrace{\overbrace{\frac{1}{N^2}\sum_{j=1}^{N} E\left\{x_j^2\right\}}^{N \;\; terms}}_{for \;\; i=j} + \underbrace{\overbrace{\frac{1}{N^2}\sum_{j=1}^{N}\sum_{i=1}^{N} E\{x_j x_i\}}^{N(N-1) \;\; terms}}_{for \;\; i \neq j} - \langle x \rangle^2 \tag{4.5}$$

The separation in Eq. (4.5) is useful because the properties of the terms for $i = j$ and for $i \neq j$ differ significantly. As we will see in the following, by exploring this product $x_i x_j$, we will be able to further simplify this expression as well as clarify the mathematical assumptions underlying the signal averaging technique. As these ***assumptions surface in the text they will be presented in bold and italic.*** Using Eq. (4.1) we can rewrite a single contribution to the summation of N terms with $i = j$ in Eq. (4.5) as:

$$E\left\{x_j^2\right\} = E\left\{[s_j + n_j]^2\right\} = E\left\{s_j^2\right\} + 2E\{s_j\}E\{n_j\} + E\left\{n_j^2\right\} \tag{4.6}$$

Assuming that the noise component n_j ***has zero mean and variance*** σ_n^2, i.e., $E\{n_j\} = 0$ and $E\left\{n_j^2\right\} = \sigma_n^2$:

$$E\left\{x_j^2\right\} = E\left\{s_j^2\right\} + \sigma_n^2 \tag{4.7}$$

The variance of the signal component (σ_s^2) is given by $\sigma_s^2 = E\left\{s_j^2\right\} - \langle s\rangle^2$, which we may substitute for the first term in Eq. (4.7), producing:

$$E\left\{x_j^2\right\} = \sigma_s^2 + \langle s\rangle^2 + \sigma_n^2 \tag{4.8}$$

Combining Eqs. (4.1) and (4.5), the expression for one of the $N(N-1)$ cross terms can be written as:

$$E\{x_j\, x_i\} = E\{[s_j + n_j][s_i + n_i]\} = E\{s_j s_i\} + E\{n_j n_i\} + E\{s_j n_i\} + E\{s_i n_j\} \tag{4.9}$$

If we assume that ***all noise terms and the signal are statistically independent within a given trial, and also across trials***, i.e., independent between trials i and j. Recall that s_i and s_j each include a signal component noise. Therefore, the first term in Eq. (4.9) becomes:

$$E\{s_j s_i\} = E\left\{[\langle s\rangle + n_{s_i}]\left[\langle s\rangle + n_{s_j}\right]\right\}$$

$$E\{s_j s_i\} = E\left\{\langle s\rangle^2 + \langle s\rangle n_{s_i} + \langle s\rangle n_{s_j} + n_{s_i} n_{s_j}\right\} = \langle s\rangle^2$$

Note that in this case, in contrast to the evaluation of $E\left\{s_j^2\right\}$ in Eqs. (4.7) and (4.8), all noise terms vanish. This result and the independence assumption allow us to rewrite all combined expectations in Eq. (4.9) as the product of the individual expectations:

$$\begin{aligned}
E\{s_j s_i\} &= E\{s_j\}E\{s_i\} = \langle s\rangle \times \langle s\rangle = \langle s\rangle^2 \\
E\{n_j n_i\} &= E\{n_j\}E\{n_i\} = 0 \times 0 = 0 \\
E\{s_j n_i\} &= E\{s_j\}E\{n_i\} = \langle s\rangle \times 0 = 0 \\
E\{s_i n_j\} &= E\{s_i\}E\{n_j\} = \langle s\rangle \times 0 = 0
\end{aligned} \tag{4.10}$$

In the above we repeatedly use the property that the expectation of a product of two independent variables can be replaced by the product of the expectation of the individual variables (Appendix 4.1).

Substituting from Eq. (4.8) for the N $i = j$ terms and from Eqs. (4.9) and (4.10) for the $N(N-1)$ $i \neq j$ terms into Eq. (4.5), we obtain the following expression for the variance:

$$Var(\overline{x}) = \frac{1}{N^2}\left[N\left(\sigma_s^2 + \langle s\rangle^2 + \sigma_n^2\right) + \left(N^2 - N\right)\langle s\rangle^2\right] - \langle x\rangle^2 \tag{4.11}$$

Finally, again using the assumption that $\langle n\rangle = 0$, the true value of the measurement x is the averaged signal, i.e., $\langle x\rangle = \langle s\rangle$. This allows us to simplify Eq. (4.11):

$$Var(\overline{x}) = \frac{1}{N^2}\left[N\left(\sigma_s^2 + \langle s\rangle^2 + \sigma_n^2\right) + \left(N^2 - N\right)\langle s\rangle^2\right] - \langle s\rangle^2 \tag{4.12}$$

This expression simplifies to:

$$\boxed{Var(\overline{x}) = \frac{\sigma_s^2 + \sigma_n^2}{N}} \tag{4.13}$$

Eq. (4.13) quantifies the variance of the average ($\overline{x}$), showing that the estimate of the mean improves with an increasing number of repetitions N. In our example the variances σ_s^2, σ_n^2 are generated by two independent sources. In this case the compound effect of the two sources is obtained by adding the variances, similar to the combined effects of independent sources on v_{eff} in Eq. (3.15). The square root of the expression in Eq. (4.13) gives us the SEM; therefore we conclude that the noise in the average $\overline{x}$ decreases with a factor of $1/\sqrt{N}$.

4.4 NOISE ESTIMATES

The ultimate reason to perform signal averaging is to increase the signal-to-noise ratio (Chapter 3). The estimate of residual noise can easily be established in a theoretical example illustrated in the simulation in pr4_1, where all the components are known. In real measurements the noise and signal components are unknown and the averaged result is certain to contain both signal and residual noise (as in Fig. 4.2). In practical applications, there is a number of techniques one might use to estimate the residual noise in the averaged result, and the following sections discuss three of these techniques.

4.4.1 Prestimulus Noise

One might estimate the residual noise by using the prestimulus epoch if there is a reason to assume that the time-locked signal only occurs after the trigger for the signal average, such as an evoked potential. Without a clear indication of a poststimulus response, such as activity surrounding the occurrence of a spike in a spike-triggered average (Chapter 20), this approach obviously won't work. It should also be noted that the prestimulus epoch is not necessarily reliable in the case of a stimulus-evoked potential. Since this type of average is obtained by repetitive stimulation, the late component of the response to a stimulus can still be ongoing in the prestimulus epoch of the next stimulus. In this case, the noise estimate of the prestimulus average will include the effect of the late component of the response. One could attempt to mitigate this problem by using larger interstimulus intervals, removing the late/slow components by high-pass filtering (Chapters 15—18), or using an alternative noise estimator.

4.4.2 Bootstrap

Bootstrapping is another method that works well, especially in off-line average procedures where the signal average is produced with signals that were previously recorded and stored. In this case, a control-average can be produced by picking random triggers rather than the triggers used for producing the "true" average. By picking the trigger randomly, the time-locked aspect of the epochs producing the average is destroyed. This procedure will produce a control-average that still includes the effects of the time-locked signal. However, since the averages are not aligned with the real trigger, the time-locked signal will not be enhanced in this control-average. Obviously, this bootstrap method will work well if the power of the time-locked component is small with respect to the noise that embeds it. Fortunately, this is usually the case because it is a principal motivation to employ signal averaging in the first place!

One can produce a series of control-averages by using the bootstrap multiple times. In this case, you can compare the statistics of the control-averages with the "true" average and use this comparison to validate the "true" average result. The disadvantage of this method is that it requires multiple averages: one for the "true" average and several to produce the control-averages. In addition, the epochs used in the control-averages are not the same as the epochs in the "true" average.

4.4.3 ± Average

One efficient way of establishing the amount of residual noise using the same epochs as those in the "true" average is by using so-called ± averaging.

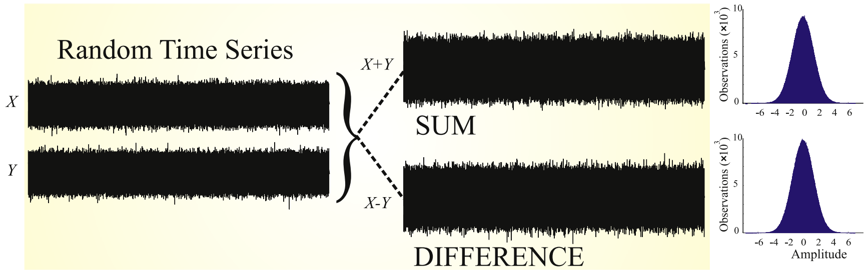

FIGURE 4.3 Random noise traces X and Y, their sum $(X+Y)$ and difference $(X-Y)$ waves. The two amplitude distributions on the right are similar for the sum and difference signals, suggesting that they can be characterized by the same PDF. For random noise, an addition or subtraction creates different time series (i.e., $X+Y \neq X-Y$), but does **not** create different statistical characteristics. This property of random noise is used when considering the ± signal average (Fig. 4.2 bottom trace) as the basis for estimating the rms of the residual noise in the averaged result.

This is a procedure in which measurements from every other trial are inverted prior to creating the averaged result. This technique removes any consistent signal component by alternating addition and subtraction. However, the residual noise is maintained in the end result (Fig. 4.3). The *rms* value of the noise component estimated from the ±average is the same as that produced by the standard average because random noise samples from the inverted traces have the same distribution as those from noninverted trials. A demonstration (not a proof) is provided in the example in Fig. 4.3, where a pair of random signals X and Y are added and subtracted. The similarity of the amplitude distributions of $X+Y$ and $X-Y$ confirms that the sum and difference signals have the same statistical properties.

4.5 SIGNAL AVERAGING AND NONRANDOM NOISE

The result in the previous section depends heavily on a noise component being random, having zero mean, and being unrelated to the signal. A special case occurs when the noise is not random. This situation may affect the performance of the average and even make it impossible to apply the technique without a few critical adaptations. The most common example of such a situation is the presence of hum (50 or 60 Hz noise originating from the power lines, see Chapter 3 and Fig. 3.4). In typical physiological applications an average is obtained by repeating a standard stimulus of a certain sensory modality and recording a time-locked epoch of neural (of neuromuscular) responses at each repetition. Usually this series of stimuli is triggered at a given stimulus rate dictated by the

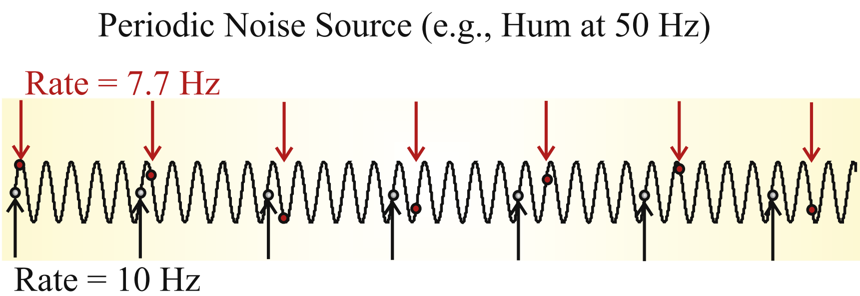

FIGURE 4.4 The stimulus-rate and a periodic component (e.g., a 50-Hz or 60-Hz hum artifact) in the unaveraged signal can produce an undesired effect in the average. An average produced with a 10-Hz rate will contain a large 50-Hz signal. In contrast, an average produced at a 7.7-Hz rate will not contain such a strong 50-Hz artifact. This difference is due to the fact that a rate of 10-Hz results in a stimulus onset that coincides with the same phase in the 50-Hz sinusoidal noise source (black dots), whereas the noninteger rate of 7.7 Hz produces a train of stimuli for which the relative phase of the noise source changes with each stimulus (red dots).

particular purpose of the experiment. It is critical to understand that in this scenario the time-locked components evoked by each stimulus will be enhanced in the average result, but also periodic components with a fixed relation to the stimulus rate (Fig. 4.4). For example, if one happens to stimulate at a rate of exactly 50 Hz, one enhances any minor 50 cycle noise in the signal. (The same example can be given for 60 Hz.) The situation is worse, because any stimulus rate r which divides evenly into 50 will have a tendency to enhance a small 50 cycle noise signal (for example the 10-Hz rate represented by the black dots in Fig. 4.4). This problem is often avoided by either ***randomizing the stimulus interval*** or by using a ***non-integer stimulus rate*** such as 3.1, 5.3, or 7.7 Hz (red in Fig. 4.4).

Although the above consideration with respect to periodic noise sources seems trivial, averaging at a poorly chosen rate is a common mistake. I have seen examples where expensive Faraday cages and power supplies were installed to reduce the effects of hum, while with normal averaging procedures, a simple change of the stimulus rate from 5.0 to 5.1 would have been much cheaper and, usually, more effective.

4.6 NOISE AS A FRIEND OF THE SIGNAL AVERAGER

It seems intuitive that a high-precision ADC combined with signal averaging equipment would contribute significantly to the precision of the end result, i.e., the average. In the following example it is shown that ADC precision isn't necessarily the most critical property in such a system and that noise can be helpful when measuring weak signals through averaging. Noise is usually considered the "enemy," preventing one from measuring the signal reliably. Paradoxically, the averaging process, made to reduce noise, may (in some cases) work better if noise is present. As we will see in the following examples, this is especially true if the resolution of the ADC is low relative to the noise amplitude. Let's assume an extreme example of a 1-bit ADC, i.e., there are only two levels: 0 or 1. Every time the signal is >=0, the ADC assigns a value of 1; every time the signal is <0, the ADC assigns a 0. In this case a small deterministic signal without added noise cannot be averaged or even measured because it would result in the same uninformative series of 0s and 1s in each trial. If we now add noise to the signal, the probability of finding a 1 or a 0 sample is proportional to the signal's amplitude at the time of each sample. By adding the results of a number of trials, we now obtain a probabilistic representation of the signal that can be normalized by the number of trials to obtain an estimate of the signal ranging from 0 to 1.

We can use the individual traces from the simulation script pr4_1.m to explore this phenomenon. Let's take the elements in the matrix NOISE_TRIALS, which is used as the basis for the average, and replace each of the values with 0 if the element's value is <0 and with 1 otherwise. This mimics a 1-bit converter where only 0 or 1 can occur.

First run the script pr4_1 (!!) and then type in the following or use script pr4_3.m:

```
for k=1:sz;
  for m=1:sz;
    if (NOISE_TRIALS(k,m) < 0);      % Is the element < 0 ?
      NOISE_TRIALS(k,m)=0;           % if yes, the simulated ADC result=0
    else;
      NOISE_TRIALS(k,m)=1;           % if not, the simulated ADC result=1
    end;
  end;
end;

average2=sum(NOISE_TRIALS)/sz;
figure
plot(average2)                       % Signal between 0 and 1
```

The figure generated by the above commands/script shows a digitized representation of the signal on a scale from 0 to 1. In Fig. 4.5 we compare

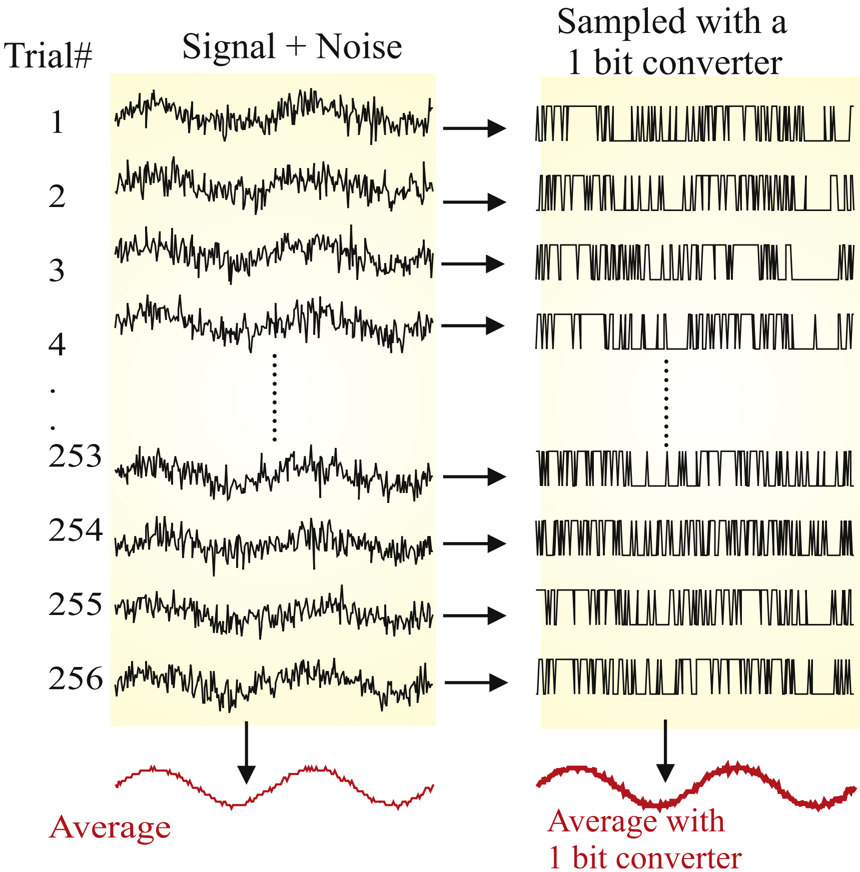

FIGURE 4.5 Signal averaging of the 256 traces generated by pr4_1.m is shown in the left column. The right column shows individual traces that were digitized with a 1-bit ADC using the MATLAB® commands in pr4_3. The averaged result of the traces in the right column is surprisingly close to the average obtained from the signals in the left column. It can be seen that the relative noise component of the 1-bit average is large compared to the standard result shown in the left column. Because the 1-bit converter only produces values between 0 and 1, all amplitudes are normalized to allow comparison.

the averaging result obtained in our original run of pr4_1 and the result obtained here by simulating a 1-bit converter.

The above example shows that reasonable averaging results can be obtained with a low-resolution ADC using the statistical properties of the noise component. This suggests that ADC resolution may not be a very critical component in the signal averaging technique. To explore this a bit further, let's compare two signal averagers that are identical with the exception of the ADC precision: ***averager A*** has a ***4-bit resolution ADC*** and ***averager B*** a ***12-bit ADC***. ***Let us say we want to know the number of trials N required to obtain an averaged result with a signal-to-noise ratio of at least 3*** (according to Eq. 3.13 ≈9.5 dB) in both systems. Further, let's assume we have a ±15 V range at the ADC input and an amplification of 100,000×. In this example we consider an *rms* value for the signal of 5 μV and for the zero mean noise component of 50 μV. For simplicity we assume a consistent signal, i.e., the variance in the signal component is zero. The signal-to-noise ratio at the amplifier input of both (Eq. 3.13) is $20\log_{10}\left(\frac{5}{50}\right) = -20$ dB (our target is therefore a $9.5 - (-20) = 29.5$ dB improvement in signal-to-noise ratio). At the amplifier output (= the ADC input) of both systems we have:

$$\begin{aligned} &rms_{signal} \quad 5\ \mu\text{V} \times 100,000 = 0.5\ \text{V} \\ &rms_{noise} \quad 50\ \mu\text{V} \times 100,000 = 5\ \text{V}. \end{aligned} \tag{4.14}$$

The quantization noise q_A and q_B in ***systems A and B*** are different due to the different resolution of their ADC components. At the output of the systems, the range of this added noise is:

$$\begin{aligned} &\text{Averager A:} \quad q_A = \pm 15\text{V}/2^4 = 30/16 \approx 1.88\ \text{V} \\ &\text{Averager B:} \quad q_B = \pm 15\text{V}/2^{12} = 30/4096 \approx 7.30\ \ 10^{-3}\ \text{V} \end{aligned} \tag{4.15}$$

The variances $\sigma^2_{q_A}$ and $\sigma^2_{q_B}$ associated with these quantization ranges (applying Eq. 3.26) are:

$$\begin{aligned} &\text{Averager A:} \quad \sigma^2_{q_A} = (30/16)^2 \Big/ 12\ \ \text{V}^2 \\ &\text{Averager B:} \quad \sigma^2_{q_B} = (30/4096)^2 \Big/ 12\ \ \text{V}^2 \end{aligned} \tag{4.16}$$

Combining the effect of the two noise sources in each system (using Eq. 3.15), we can determine the ***total noise*** at the input of the ADC as the combination of the original noise (Eq. 4.14; creating an ms of 5^2 V^2) and that produced by quantization:

$$\begin{aligned} &\text{Averager A:} \quad 5^2 + \sigma^2_{q_A} = 5^2 + (30/16)^2 \Big/ 12\ \ \text{V}^2 \\ &\text{Averager B:} \quad 5^2 + \sigma^2_{q_B} = 5^2 + (30/4096)^2 \Big/ 12\ \ \text{V}^2 \end{aligned} \tag{4.17}$$

According to Eq. (4.13) these noise figures will be attenuated by a factor N_A or N_B (number of trials in systems A and B) in the averaged result.

Using the signal-to-noise ratio $rms_{signal}/rms_{Total\ Noise}$ and including our target (a ratio of 3 or better), we get:

$$\begin{aligned}
&\text{Averager A:} \quad \frac{0.5}{\sqrt{\dfrac{5^2+\sigma_{q_A}^2}{N_A}}} = \frac{0.5}{\sqrt{\dfrac{5^2+(30/16)^2/12}{N_A}}} \geq 3 \\
&\text{Averager B:} \quad \frac{0.5}{\sqrt{\dfrac{5^2+\sigma_{q_B}^2}{N_B}}} = \frac{0.5}{\sqrt{\dfrac{5^2+(30/4096)^2/12}{N_B}}} \geq 3
\end{aligned} \tag{4.18}$$

Solving for the number of trials required in both systems to get this signal-to-noise target, we find that $N_A = 911$ and $N_B = 900$. From this example we conclude that in a high noise environment (i.e., with a high noise level relative to the quantization error q), the precision of the ADC doesn't influence the end result all that much; in our example a huge difference in precision (4 vs. 12 bit, which translates into a factor of 256) only resulted in a small difference in the number of trials required to reach the same signal-to-noise ratio (911 vs. 900, a factor of ~1.01). The example also shows that in a given setup, improvement of the signal-to-noise ratio with averaging is best obtained by increasing the number of trials; from Eq. (4.18) we can determine that the signal-to-noise improvement is, as expected, proportional to $\sqrt{N}$.

4.7 EVOKED POTENTIALS

Evoked potentials (EPs) are frequently used in the context of clinical diagnosis; these signals are good examples of the application of signal averaging in physiology (Chapter 1). The most commonly measured evoked potentials are recorded with the EEG electrode placement (Fig. 1.2A) and represent neural activity in response to stimulation of the auditory, visual, or somatosensory system (AEP, VEP, or SEP, respectively). These examples represent activity associated with the primary perception process. More specialized evoked potentials also exist; these record the activity generated by subsequent or more complex tasks performed by the nervous system. One example is the so-called oddball paradigm, which consists of a set of frequent baseline stimuli, occasionally (usually at random) interrupted by a rare test stimulus. This paradigm usually evokes a centrally located positive wave at 300 ms latency in response to the rare stimulus (the P300). This peak is generally interpreted as representing a neural response to stimulus novelty.

An even more complex measurement is the contingent negative variation (CNV) paradigm; here the subject receives a warning stimulus (usually a short tone burst) that a second stimulus is imminent. When the second stimulus (usually a continuous tone or a series of light flashes) is presented the subject is required to turn it off with a button press. During the gap in between the first (warning) stimulus and the second stimulus, one can observe a centrofrontal negative wave. Relative to the ongoing EEG the CNV signal is weak and must be obtained by averaging; an example of individual trials and the associated average is shown in Fig. 4.6. Here it can be seen that the individual trials contain a significant amount of noise, whereas the average of only 32 trials clearly depicts the negative slope between the stimuli (note that negative is up in Fig. 4.6). The ± average provides an estimate for the residual noise in the averaged result. The original trials are included in the material that can be downloaded from http://booksite.elsevier.com/9780128104828/ (single_trials_CNV.mat).

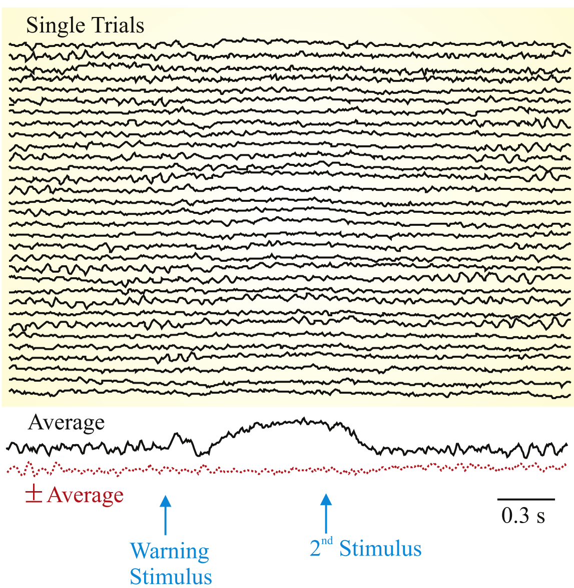

FIGURE 4.6 The contingent negative variation (CNV) measured from C_z (the apex of the scalp, Fig. 1.2A) is usually made visible in the average of individual trials in which a subject receives a warning stimulus that a second stimulation is imminent. The second stimulus must be turned off by a button press of the subject. The lower pair of traces shows the standard average revealing the underlying signal and the ±average as an estimate of the residual noise.

Typing the following MATLAB® commands will display the superimposed 32 original traces as well as the average of those trials.

```
clear
load single_trials_CNV
figure
plot(single)
hold
plot(sum(single')/32,'k+')
```

4.8 OVERVIEW OF COMMONLY APPLIED TIME DOMAIN ANALYSIS TECHNIQUES

1. *Power and related parameters*
   Biomedical applications often require some estimate of the overall strength of measured signals. For this purpose, the variance (σ^2, Chapter 3) of the signal or the mean of the sum of squares $\frac{1}{N}\sum_{n=1}^{N} x^2(n)$ are frequently used. Time series are also frequently demeaned (baseline corrected) before further analysis, making the mean of the sum of squares and the variance equivalent. Another variant is the *rms* (root mean square, Chapter 3).
   Hjorth (1970) describes the signal variance σ^2 as the ***activity index*** in EEG analysis. In the frequency domain, activity can be interpreted as the area under the curve of the power spectrum. To this metric he adds the standard deviations from the first and second derivatives of the time series, σ_d and σ_{dd}, respectively. On the basis of these parameters, Hjorth introduces ***mobility*** $\frac{\sigma_d}{\sigma}$ and ***complexity*** $\frac{\sigma_{dd}/\sigma_d}{\sigma_d/\sigma}$ parameters. In the frequency domain, mobility can be interpreted as the standard deviation of the power spectrum. The complexity metric quantifies the deviation from a pure sine wave as an increase from unity.
2. *Zero-Crossings*
   The 0-crossings in a demeaned signal can give an indication of the dominant frequency component in a signal. For example, if a signal is dominated by a 2-Hz sine wave, it will have four 0-crossings per second; i.e., the number of 0-crossings/s divided by 2 is the frequency of the dominant signal component. The lengths of epochs in between 0-crossings can also be used for ***interval analysis***. Note that there are two types of 0-crossings, from positive to negative and vice versa. Zero-crossings in the derivative of a time series can also be used to find local ***maxima*** and ***minima***.

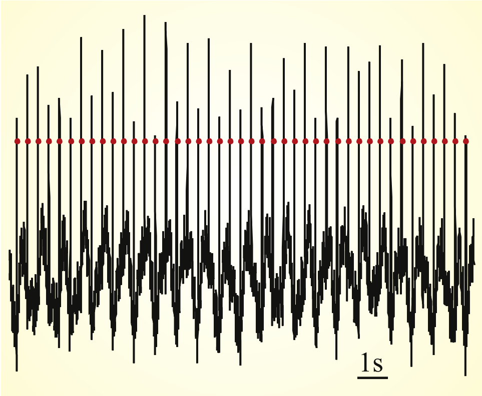

FIGURE 4.7 An ECG signal (see also Fig. 1.4) from a human neonate and the detected QRS complexes (red dots).

3. *Peak Detection*
   Various methods to detect peaks are used to locate extrema within time series. If the amplitudes between subsequent local maxima and minima are measured, we can determine the amplitude distribution of the time series. In the case of peak detection in signals consisting of a series of impulses, the peak detection procedures are used to calculate intervals between such events. This routine is frequently used to detect the events in signals containing spikes or in the ECG to detect the QRS complexes (Chapter 1, Fig. 1.5). An example of QRS complex detection in human neonates is shown in Fig. 4.7. The general approach in these algorithms consists of two stages: first pretreat the signal in order to remove artifacts and then detect extreme values above a set threshold.

   The following part of a MATLAB® script is an example of a peak detector used to create Fig. 4.7.
   NOTE that this is a part of a script! The whole script pr4_4.m and an associated data file (subecg) can be downloaded from http://booksite.elsevier.com/9780128104828/.

```
% 1. pre-process the data
[c,d]=butter(2,[15/fN 45/fN],'bandpass');          % 2nd order 15-45 Hz (XIII)
subecgFF=filtfilt(c,d,subecg-mean(subecg));         % use filtfilt to prevent
                                                       phase-shift

% 2. detect peaks
```

```
% In this routine we only look for nearest neighbours (three subsequent
    points)
% adding additional points will make the algorithm more robust
threshold=level*max(subecgFF);              % detection threshold
for i=2:length(subecgFF)-1;
  if (subecgFF(i)>threshold);               % check if the level is exceeded
    % is the point a relative maximum (Note the >= sign)?
    if((subecgFF(i)>=subecgFF(i-1))&(subecgFF(i)>=subecgFF(i+1)));
      % if yes, is it not a subsequent maximum in the same heartbeat
      if (i-i_prev > 50)
        D(n)=i;                             % Store the index in D
        i_prev=i;
        n=n+1;
      end;
    end;
  end;
end;
```

4. *Level and Window Detection*
   In some types of time series (such as in extracellular recordings of action potentials) one is interested in identifying epochs in which the signal is within a certain amplitude range. Analog- or digital-based window and level detectors are available to provide such data processing.
5. *Filtering (see Chapters 15—18)*
   The filters we will consider in later chapters are both analog and digital implementations. For analog filters we will focus on RC circuits, digital implementations will cover infinite impulse response (IIR) and finite impulse response (FIR) versions.
6. *Real Convolution (see Chapter 13)*
   Convolution plays an important role in relating input and output of linear time invariant systems.
7. *Cross-Correlation (see Chapter 13)*
   Cross-correlation is related to convolution and can be used to quantify the relationship between different signals or between different parts of the same signal (termed "auto-correlation").
8. *Template Matching*
   In some applications signal features are extracted by correlating a known template with a time series. Wavelet and scaling signals can be considered as a special type of template (see Chapters 21 and 22).
9. *Miscellaneous*
   In some cases the task at hand is highly specific: e.g., detection of epileptic spikes in the EEG. In these instances a specially

developed metric may provide a good solution. For example, in EEG spike detection, a "sharpness index" works reasonably well (Gotman and Gloor, 1976).

APPENDIX 4.1 EXPECTATION OF THE PRODUCT OF INDEPENDENT RANDOM VARIABLES

The development used to show that averaging reduces the noise component is based on the use of the expectation formalism, $E\{\ldots\}$ (see also Eq. 3.9). In this appendix we summarize a few of its properties that are frequently used.

Note that the expectation of a constant c (with distribution $\delta(x-c)$) is simply the constant itself: $E\{c\}=c$. Since $E\{\ldots\}$ is a linear operator, the sum of two random variables is the sum of the individual expectations: $E\{x+y\}=E\{x\}+E\{y\}$. The procedure we need to follow for the evaluation of the expectation of the product of two variables $E\{xy\}$ depends on whether they are independent or not. It is simply $E\{x\}E\{y\}$ if the joint distribution $p(x\,y)$ of x and y can be written as the product of two independent distributions, i.e., $f(x)g(y)$. In this case we may write

$$E\{x\,y\} = \int\int xy\,p(x,y)\,dxdy = \int x\,f(x)dx \int y\,g(y)dy = E\{x\}E\{y\} \tag{A4.1-1}$$

This property is repeatedly used in the derivation leading to Eq. (4.13) and multiple other cases, for example in the development for the Poisson–Wiener kernels in Appendix 26.1.

In the following we show a simulated example that for two independent random variables x and y we may indeed use $E\{x\,y\}=E\{x\}E\{y\}$. Let's create two normally distributed variables x and y with mean values of 1 and 3 and a variance of 1.

To do this, type in the MATLAB® command window:

```
x=randn(1,10000)+1;
y=randn(1,10000)+3;
```

If you want you can plot these variables to make sure they are really different, or even better you can plot the variables against each other with:

```
figure;
plot(x,y,'.')
```

and you will see a cloud of points around coordinates (1,3), reflecting the absence of a clear relationship between x_1 and x_2. Now recall that in an ergodic process the mean values of x and y are $E\{x\}$, $E\{y\}$, so we can estimate the product $E\{x\} \times E\{y\}$ by typing:

```
mean(x)*mean(y)
```

The answer will be a value close to 3. The outcome will usually not be exactly equal to 3, because we simulate the procedure using data sets of random numbers with limited size.

Now we can compute $E\{x\,y\}$ by computing the mean of the product of x and y using:

```
X=x.*y;
sum(X)/length(X)
```

As we expect, the answer will again be a value close to 3.

You can repeat the procedure with another pair a and b that are **not** independent. For example:

```
a=randn(1,10000)+1;
b=a*2;
```

By following the same procedure we used above for x and y, you will now find that the plot of the variables against each other shows a clear nonrandom relationship, and that Eq. (A4.1-1) does not hold, i.e.,: $E\{ab\} \neq E\{a\} \times E\{b\}$.

EXERCISES

4.1 You want to apply the signal averaging technique. For reasons of efficiency you plan to monitor the progress of improving your signal-to-noise ratio (SNR) by estimating the SNR after each repetition (trial). You decide that you will try to accomplish this by plotting the ratio [rms(signal)/rms(residual noise)] versus trial number.

Note: From the average, you cannot compute the rms(signal) directly but only the rms (signal + residual noise). You may, however, obtain an estimate by assuming that the signal and noise are independent and that the mean values of the signal and noise are zero so that (see Eq. 3.15) ms(signal + residual noise) = ms(signal) + ms(residual noise). From this relationship you can estimate rms(signal) as the square root of ms(signal + residual noise), obtained from your average, minus ms(residual noise), obtained from your ±average.
An alternative practical solution to monitor SNR is to compute rms (signal + residual noise)/rms(residual noise).

Modify program pr4_1.m to evaluate averages at $n = 1, 2, \ldots, 1024$ repetitions.

a. Depict the average and ±average for $n = 1, 10, 100, 1000$.
b. Calculate the ms of the noise for each value of n.
c. Calculate a value for the SNR for each value of n.
d. Plot the values in (b) and (c) against n.
e. Relate your finding to Eq. (4.13).
f. Comment on the noise reduction in your simulated results by comparing it with the analytically obtained estimate of the SNR improvement as a function of the number of repetitions.

4.2 Now assume that in Exercise 4.1, the sine wave you averaged is not the signal but noise originating from another instrument. So, in your average, you want to get rid of both the sine wave as well as the random noise component.

a. Explain how you can accomplish this.
b. Write a Matlab® script or make modifications to pr4_1.m to show how this works.
c. Plot the ms of the noise versus n.
d. Relate your finding to Eq. (4.13).

4.3 A signal averager's output generates Gaussian noise with zero mean and variance σ_0^2. This output noise is added to any output and it is independent of the averager's input signal s, input noise n, sample point k ($1, \ldots, M$), and number of trials N. (See also Fig. 4.1 and Eq. (4.1) for the definitions of these variables.)
The evoked signal s is identical across all trials (e.g., its variance is zero: $\sigma_s^2 = 0$). The input noise n is zero mean with variance σ_n^2. Derive an expression for the signal-to-noise ratio
$G = SNR_{out}/SNR_{in} > 0$

4.4 Load ActionPotential.mat, a recording of action potentials, into MATLAB®.

a. Create a peak detection script that detects the action potentials.
b. Validate your detection by plotting the original signal and the detected peaks in a single graph (similar to Fig. 4.7).

4.5 Assume a measurement that includes a noise-free stimulus-evoked signal with a mean square (ms) value of 5 μV^2 and additive Gaussian noise with an ms value of 50 μV^2. Both signal and noise have zero mean.

a. What is the SNR of this measurement in dB?

b. How many repetitions N of this measurement are required to obtain a value for SNR $>$20 dB?
c. What is the ms in the ±average associated with your result in question b?
d. Check your answers in questions (a), (b), and (c) in a MATLAB® simulation employing a sine wave with an ms value of 5 arbitrary units (AU) contaminated with noise with an ms value of 50 AU (use the randn function to generate the noise).
(Hint: for the sinusoidal signal see Appendix 3.3.)
e. Why do we use the MATLAB® randn instead of rand function?

References

Gotman, J., Gloor, P., 1976. Automatic recognition and quantification of inter- ictal epileptic activity in the human scalp EEG. Electroencephalogr. Clin. NeuroPhysiol. 41, 513–529.
The first paper describing a successful automated time domain analysis to detect epileptic spike activity in clinical recordings. Although relatively simple, this method is still being used in clinical equipment today.

Hjorth, B., 1970. EEG analysis based on time domain properties. Electroencephalogr. Clin. Neurophysiol. 29, 306–310.
Introduction of activity, mobility, and complexity parameters in EEG analysis.