Z transforms visualized on the complex plane

Joe Dellinger, Clement Kostov, and Fabio Rocca

ABSTRACT

Z transforms provide a unifying framework for designing filters and understanding their properties. Unfortunately, Z transforms are not very intuitive. To remedy this situation for students at SEP, we have developed a program to display Z transforms on the complex plane. We display several different filters, and show how their properties are visible in the plot.

INTRODUCTION

Definition and basic properties

The *Z transform* of a sequence

$$\{a_k\}$$

is

$$A(Z) = \sum_{k=-\infty}^{k=+\infty} a_k Z^k .$$

Note that $A(Z)$ is not a discrete time series but is instead a complex-valued function of a complex variable. (The Z transform may be infinite or undefined for some values of Z.)

Why is this transform useful? For one thing, just as with the Fourier transform, convolution ($\{a_i\}*\{b_j\} = \{c_k\}$) maps into multiplication ($A(Z)B(Z) = C(Z)$). (Put another way, polynomial multiplication convolves the polynomial coefficients.) This is not surprising since the Z transform is closely related to the Fourier transform via the relation

$$\mathrm{FT}(a_k)|_\omega = A(Z)|_{Z\,=\,e^{i\omega}} .$$

The slice of the Z transform lying on the unit circle gives the Fourier transform.

Poles and zeroes

Any polynomial

$$X(Z) = \sum_{k=0}^{k=\infty} x_k Z^k$$

is uniquely specified up to some constant multiplicative factor by knowing all of its "zeroes" (values of Z for which $X(Z) = 0$). Gauss's theorem tells us how many zeroes there are to look for.

In general, Z transforms have terms in both positive and negative powers of Z. Such a series $A(Z)$ can be written in the form

$$A(Z) = \frac{N(Z)}{D(Z)},$$

where $N(Z)$ and $D(Z)$ each contain only positive powers of Z. $A(Z)$ becomes zero if $N(Z) = 0$. $A(Z)$ becomes infinite if $D(Z) = 0$. The zeroes of $D(Z)$ are called the "poles" of $A(Z)$.

If you know all the zeroes and poles of $A(Z)$, you know much about the properties of the time series $\{a_k\}$.

THE PROGRAM

Our program accepts as input the two complex causal time series $\{n_k\}$ and $\{d_k\}$, corresponding to $N(Z)$ and $D(Z)$ mentioned before. It then plots 4 different things. (You should now refer to the first pair of plots, Figure 1, as you read along.)

In the upper left it plots the central portion of the time series $\{a_k\}$, with the zero'th lag in the center. Since in general the series is complex, it plots the Real part in Blue[1] and the Imaginary part in Red.

In the upper right is the log spectrum of the time series. The plot shows deviations from unity, so amplified frequencies plot above the axis and attenuated frequencies plot below. Zero frequency is at the center, +Nyquist is at the right, and −Nyquist is at the left. The phase is encoded by the color. A cyclical color scheme is used which completes one cycle from $-\pi$ to π:

+Real	→	Cyan
+Imaginary	→	Blue
−Real	→	Magenta
−Imaginary	→	Red

[1]Your eyes contain cells sensitive to three different colors: red, green, and blue. These are the primary additive colors. Combinations of two of these at a time make the primary subtractive colors: red and green make Yellow, red and blue make Magenta, and blue and green make Cyan. For some reason everybody knows what yellow is but magenta and cyan languish in obscurity. If you live in the city the sky is probably cyan colored; magenta is a purplish red, and has its name after Magenta, a town 15 Km from Milano where a bloody battle was fought in 1859.

In the lower left the amplitude of $A(Z)$ over a portion of the Z plane is shown. Higher amplitudes are brighter, lower amplitudes are darker. Thus a pole shows up as a bright spot, and a zero shows up as a dark spot. A cyclical color variation has been superimposed on the basic grey scale so that amplitude "contours" may be perceived. The actual colors themselves are arbitrary. The axes and the unit circle are shown to give a reference frame.

In the lower right the phase of $A(Z)$ over a portion of the Z plane is shown. The phase is encoded with the same color scheme used for the "Log Spectrum" plot.

EXAMPLES

All of our examples will consist of pairs of plots showing two different filters. Compare the two to see how they relate to each other and how they differ.

Figure 1: Single zero with causal inverse

We start with two very simple filters, a filter consisting of only a single zero

$$\text{(Upper plot)} \quad 1 - .9Z$$

and its inverse, a filter consisting of only a single pole

$$\text{(Lower plot)} \quad \frac{1}{1 - .9Z}.$$

$1 - .9Z$ obviously represents the time series $\{1, -.9\}$. What does $1/(1 - .9Z)$ represent? Noting that

$$\frac{1}{1 - X} = 1 + X + X^2 + X^3 + \ldots,$$

we let $X = .9Z$ and find that

$$\frac{1}{1 - .9Z} = 1 + .9Z + .9^2 Z^2 + .9^3 Z^3 + \ldots,$$

an exponentially decaying causal series.

Figure 2: Single zero with anti-causal inverse

Next we try a seemingly slight variation on the previous example,

$$\text{(Upper plot)} \quad 1 - 1.1Z$$

and

$$\text{(Lower plot)} \quad \frac{1}{1 - 1.1Z}.$$

Note that while the time series on the top plot has hardly changed from the previous example, the time series on the bottom plot is now anti-causal.

The identity we used before no longer applies (it diverges) and we have to use instead the identity

$$\frac{1}{1-X^{-1}} = -X - X^2 - X^3 - \ldots .$$

Letting $X = 1/1.1Z$, we find that

$$\frac{1}{1-1.1Z} = 0 - 1.1^{-1}Z^{-1} - 1.1^{-2}Z^{-2} - 1.1^{-3}Z^{-3} - \ldots ,$$

an exponentially decaying anti-causal series.

A causal series with a causal inverse like the one in example 1 is called *minimum phase.* Since a pole inside the unit circle causes a series to be non-causal, a series is minimum phase if and only if all of its poles and zeroes (which are poles in its inverse) are outside the unit circle.

Figure 3: Notch and all-pass

In this example we show two filters that can be made from a single pole-zero pair. The filters are a "notch" filter,

$$\text{(Upper plot)} \quad \frac{1-Z}{1-.5Z}$$

and an "all-pass" filter,

$$\text{(Lower plot)} \quad \frac{1-2Z}{1-.5Z}.$$

A pole and zero are exactly opposite in effect; a pole and zero in the same place annihilate each other like anti-particles. It follows that such a pair only affects a nearby portion of the Z plane; further away the distance between them becomes insignificant and they cancel out.

A notch filter strongly attenuates frequencies around some $\omega = f_0$, but leaves the rest of the frequency spectrum nearly untouched. To kill frequencies around f_0 we place a zero on the unit circle at $Z = e^{if_0}$. We balance out the zero for other frequencies by placing a pole next to the zero and just outside the unit circle. The closer the zero and pole are together, the sharper the notch is.

If a pole and zero are placed at reciprocal positions on either side of the unit circle, an all-pass filter results. This filter has a constant amplitude over the unit circle, but a varying phase.

Figure 4: Autocorrelation Functions

To autocorrelate a sequence, convolve it with a time-reversed and conjugated copy of itself. If a sequence is time-reversed and conjugated, the amplitude spectrum is left the same but the phase spectrum changes sign. When this is convolved with the original sequence the result must have zero phase, ie, the spectrum must be purely positive real.

All sequences that are autocorrelations must be symmetric, but not all symmetric sequences are also autocorrelations. All symmetric sequences have a purely real spectrum, but the spectrum may be both positive and negative real.

In this example we show a filter that is an autocorrelation,

$$\text{(Upper plot)} \quad .4Z^{-1} + 1 + .4Z$$

and one that is not,

$$\text{(Upper plot)} \quad .6Z^{-1} + 1 + .6Z.$$

The spectrum in the upper plot is purely positive real (cyan), while the spectrum in the lower plot contains a negative real region (magenta) around the nyquist. Where these regions meet there must be a zero, and so we know the two zeroes on the lower amplitude plot must actually be *exactly* on the unit circle.

What happens as we perturb the lower case into the upper one, always keeping the filter symmetric? The zeroes must slide down the unit circle like beads on a wire, until they collide with each other at the nyquist and can finally be pulled off sideways. Sequences that are autocorrelations may only have double (or quadruple etc.) zeroes on the unit circle.

Note that in the previous example we saw a filter that had constant *amplitude* over the unit circle; in this example we have seen a filter with constant *phase* over the unit circle.

Figure 5: Many zeroes ⟺ one pole

As we saw before, a single pole corresponds to a decaying exponential as a time series. What if we truncate the exponential? Such a finite series can have only zeroes. How can many zeroes look like one pole, given that zeroes and poles are opposites?

To test this, for the upper plot we will use

$$\text{(Upper plot)} \quad \frac{2}{1 - .8Z},$$

and for the lower one

$$\text{(Lower plot)} \quad 2(1 + .8Z + \ldots + .8^{10}Z^{10}).$$

As you can see, we get 11 evenly spaced zeroes, except that one is missing. Inside the ring of zeroes the missing one looks like a pole! Mathematically,

$$1 + .8Z + \ldots + .8^{10}Z^{10} = \frac{1 - .8^{11}Z^{11}}{1 - .8Z}.$$

The numerator has zeroes at the 11th roots of unity, but one of them gets canceled out by the pole in the denominator. This is just another guise of the same identity we used back in examples 1 and 2. For Z outside the ring of zeroes the identity breaks down and the two plots look quite different.

Figure 6: Windowing

What happens to the poles and zeroes of a function when it is windowed? In Figure 6 we show the Hilbert operator windowed two different ways:

(Upper plot) Hilbert transform with boxcar window,

and

(Lower plot) Hilbert transform with raised cosine window.

The frequency spectrum of the perfect, unwindowed Hilbert operator is $+i$ for $\omega > 0$ and $-i$ for $\omega < 0$. Since the spectrum of a boxcar is a sinc, we should not be surprised at the sinc-like oscillations superimposed on the basic pattern. The raised cosine avoids these oscillations by moving the zeroes away from the unit circle while still keeping the effect of each doublet the same.

A common misconception when looking at this figure is that the zeroes at $\omega = 0$ and $\omega = \pi$ are double. It looks like the inner and outer rings of zeroes briefly merge at those two points. Actually, both these zeroes are single; if they were double the spectrum could not abruptly change sign there.

Figure 7: Levinson recursion

For any given filter, several other filters can be found that have the same autocorrelation. For instance, $Z - Z_0$ and $Z - 1/Z_0^*$ have the same autocorrelation. A simple way to obtain a new filter with the same autocorrelation as another is to convolve it with your favorite unit-gain all-pass filter. Among the set of all filters that have the same autocorrelation function, however, there is only one that has the property of being minimum phase. This unique minimum phase filter is particularly important for numerical applications, because it is the only filter that has a causal inverse.

Levinson recursion is an efficient algorithm for finding the unique minimum-phase sequence corresponding to a given autocorrelation function. Given N lags of an autocorrelation function, the Levinson recursion determines in N steps a minimum phase sequence of length N. An important property of the Levinson recursion is that the filters generated at each intermediate step of the process are guaranteed minimum-phase.

We computed the autocorrelation of a window of 1024 data samples from a seismic trace, and applied the Levinson recursion to determine the minimum-phase filter of length 50 that has the same first 50 lags for its autocorrelation. The zeroes of the filter at the 25th iteration are displayed in the upper part of Figure 7; the zeroes for the filter at the 50th iteration are displayed in the lower part of the same figure. Although some of the zeroes are very close to the unit circle, all are outside.

Figure 8: Conjugate gradients algorithm

Alternatively, the system of normal equations for the computation of the minimum-phase filter with a given autocorrelation can be solved using a general-purpose minimization algorithm such as the Conjugate-Gradient algorithm. An advantage of the conjugate-gradient algorithm is that in many cases it provides a very good approximation of the solution after only a few iterations. This property makes it particularly attractive for large problems.

In Figure 8 we show the results of doing the same problem as in the previous example by using conjugate gradients instead of Levinson. In this case, there is no guarantee that the iterates will be minimum phase, even though the starting point is known to be minimum phase, and the end point should be minimum phase as well since theoretically the final result should be the same as in the previous example. Of course, the final results did not turn out to be the same due to different handling of numerical errors by the two different algorithms. As before, we display the filter at the 25th iteration in the upper part of Figure 8, and the filter at the final 50th iteration in the lower part. The question is, are these filters minimum-phase?

Figure 9: Close up of zeroes near the unit circle

We applied the Levinson recursion to the filters calculated at the final iterations of the previous two examples. A given filter is minimum phase if and only if the reflection coefficients computed at each step of the Levinson recursion are less than one in absolute value. Thus the Levinson recursion is a sensitive numerical test of whether a given filter is truly minimum phase or not. We found that the filter obtained by the Levinson recursion was indeed minimum phase while the one obtained by Conjugate Gradients was not.

Since this observation can not be made without ambiguity by looking at the plots in Figures 7 and 8 (since many zeroes are so close to the unit circle that it is impossible to tell whether they are actually inside, outside, or even exactly on it), we produced an enlargement of a portion of the Z plane where the zeroes clustered particularly closely to the unit circle. The result is shown in Figure 9: in the upper part we show the final iteration of the Levinson example from Figure 7, and in the lower part we show the final iteration from the conjugate-gradients example from Figure 8.

If the zeroes obtained by the conjugate-gradients algorithm are within the unit circle, they are only barely so! Yet this is enough to cause the conjugate-gradients result to have a non-causal inverse. It is interesting to note that the error functions, the spectra, and the distribution of zeroes are all similar in the two algorithms. The time series, however, differ significantly.

Go reread "Fundamentals of Geophysical Data Processing" and see if you can understand Z transforms better now!

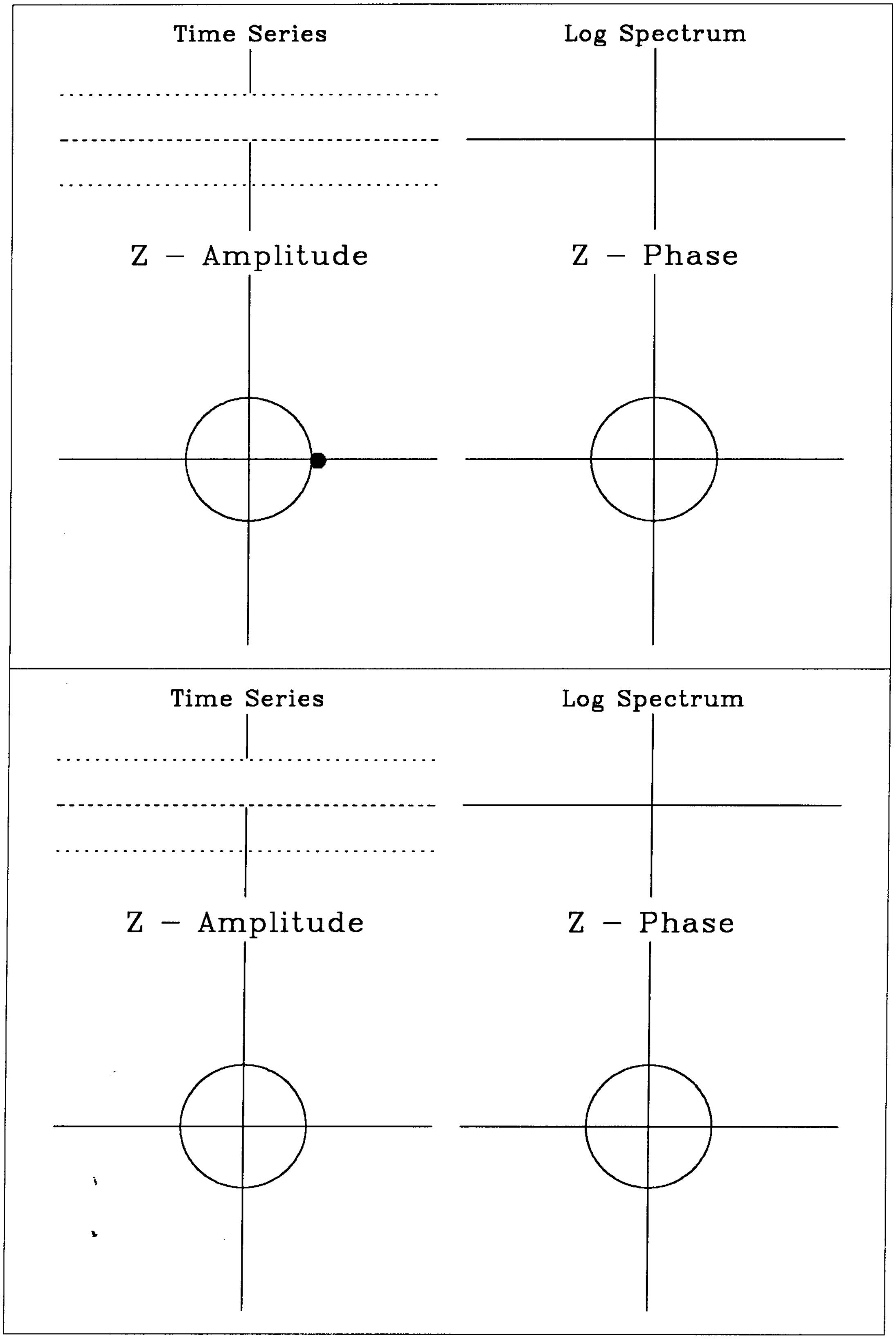
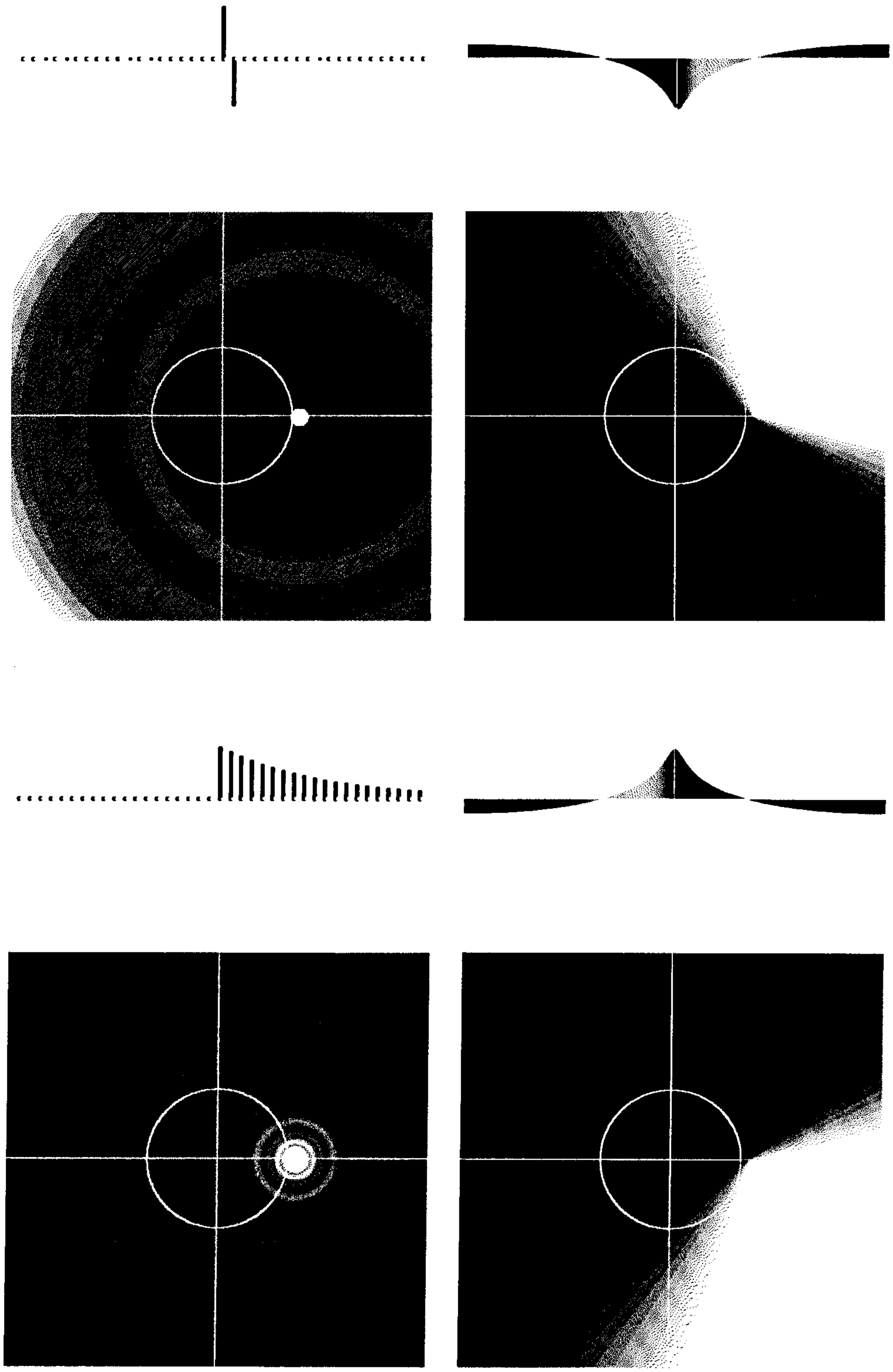
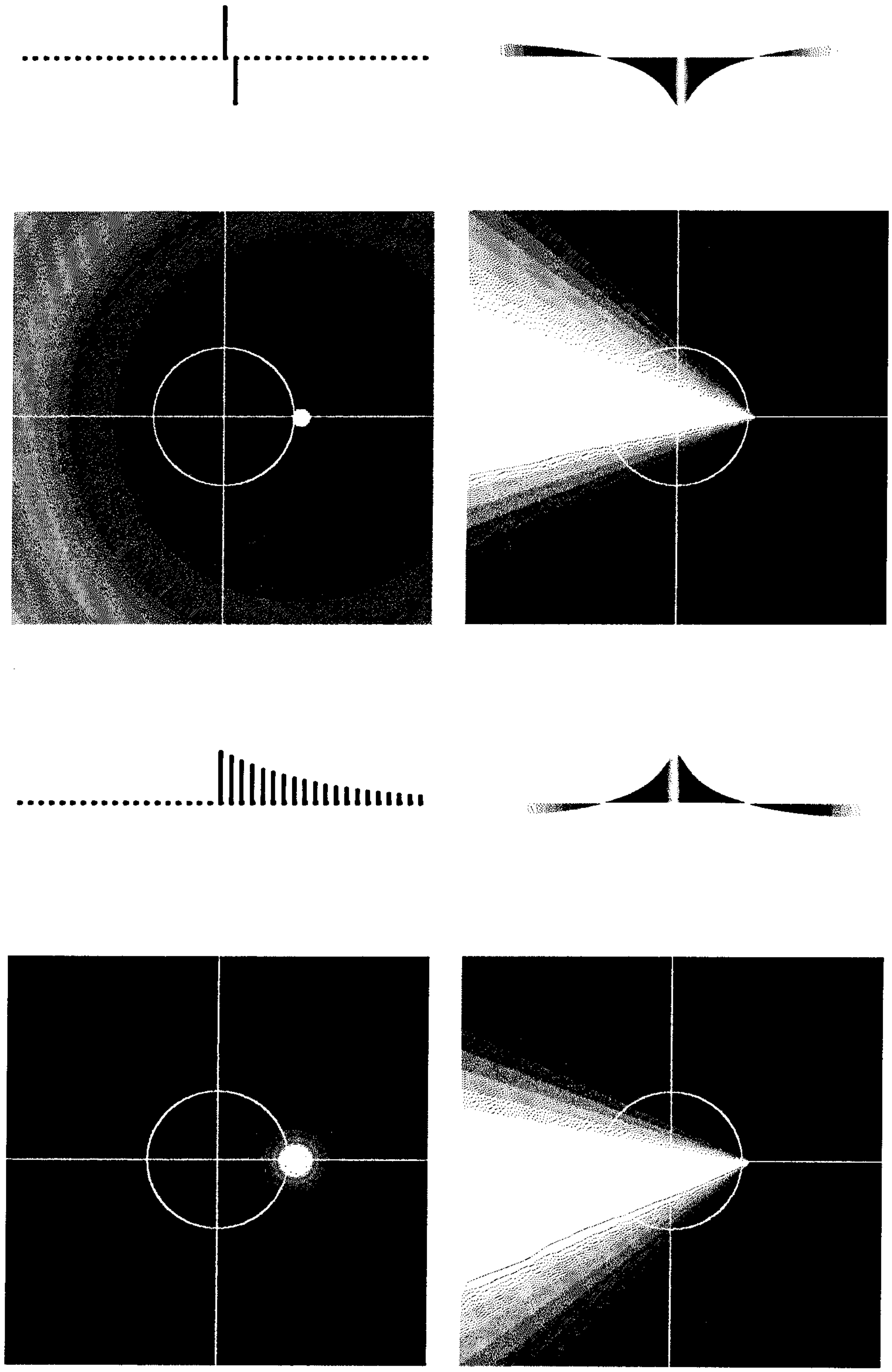
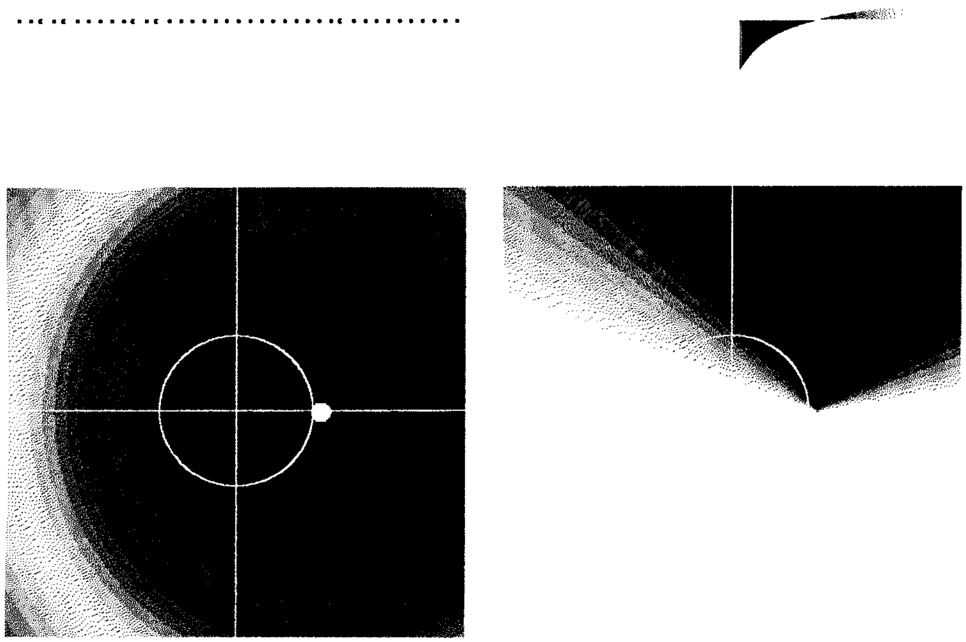
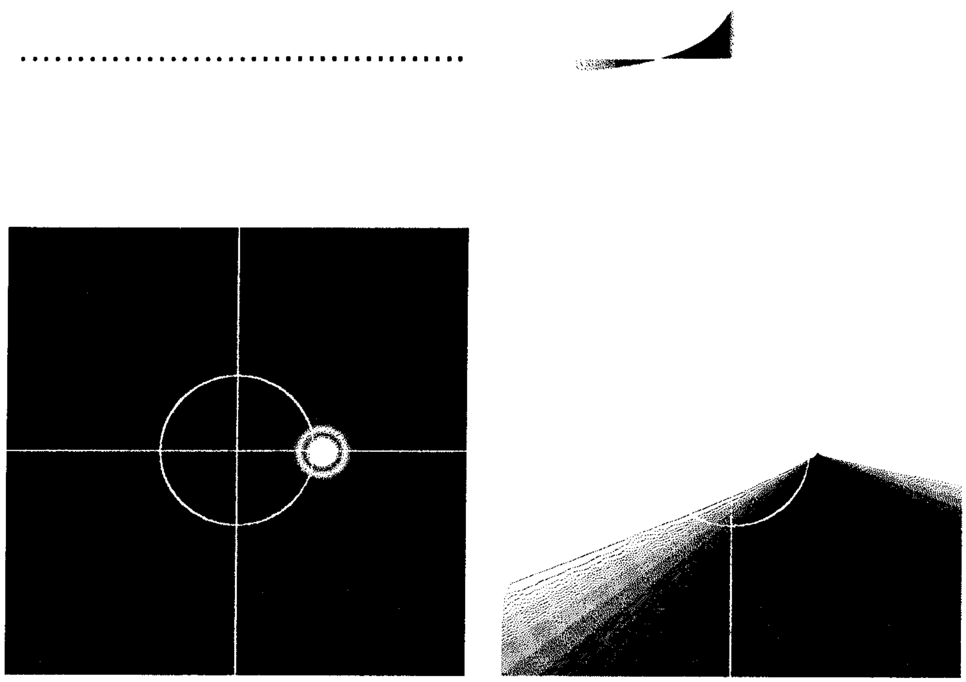


FIG. 1. Upper plot: $1 - .9Z$; Lower plot: $1/(1 - .9Z)$









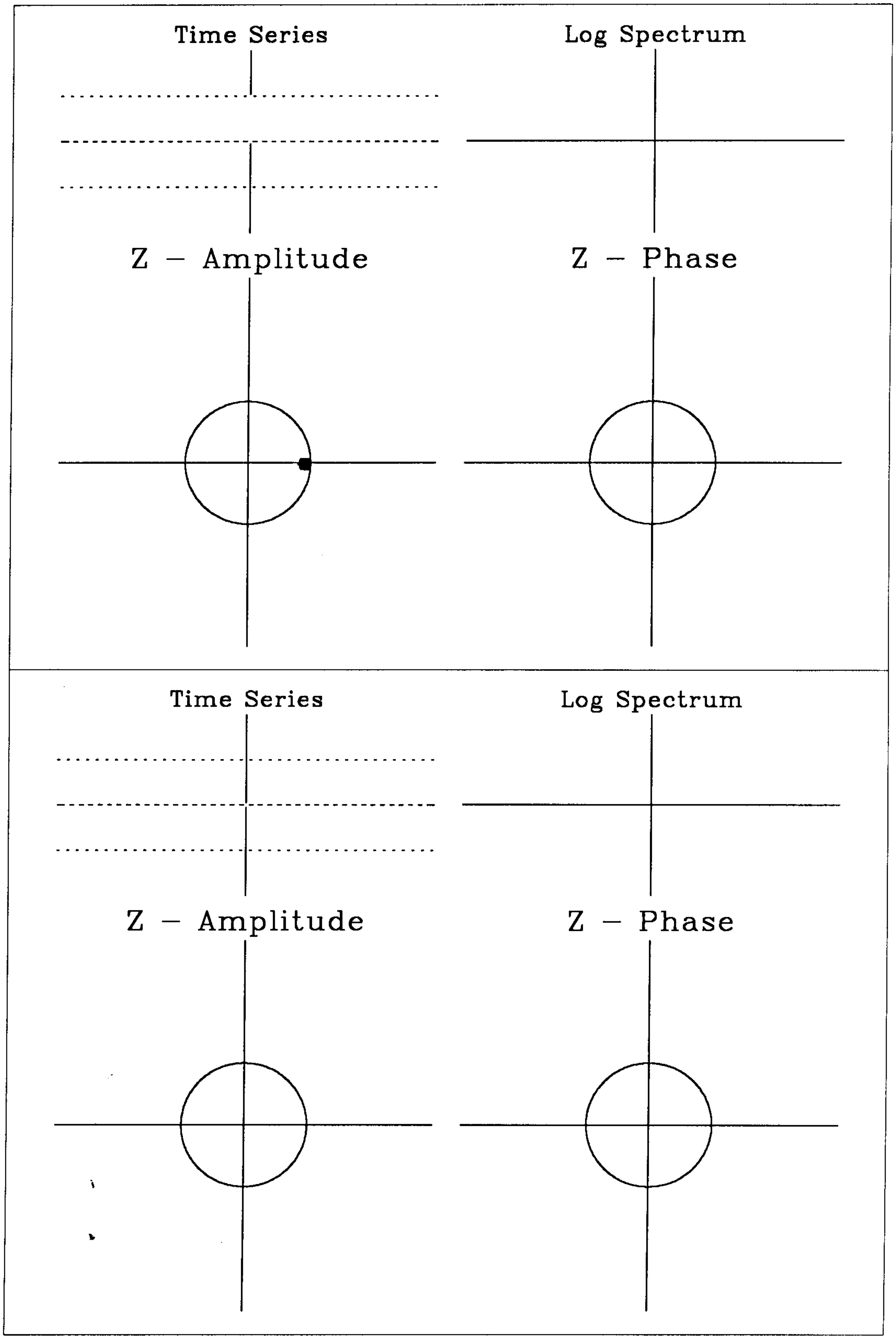
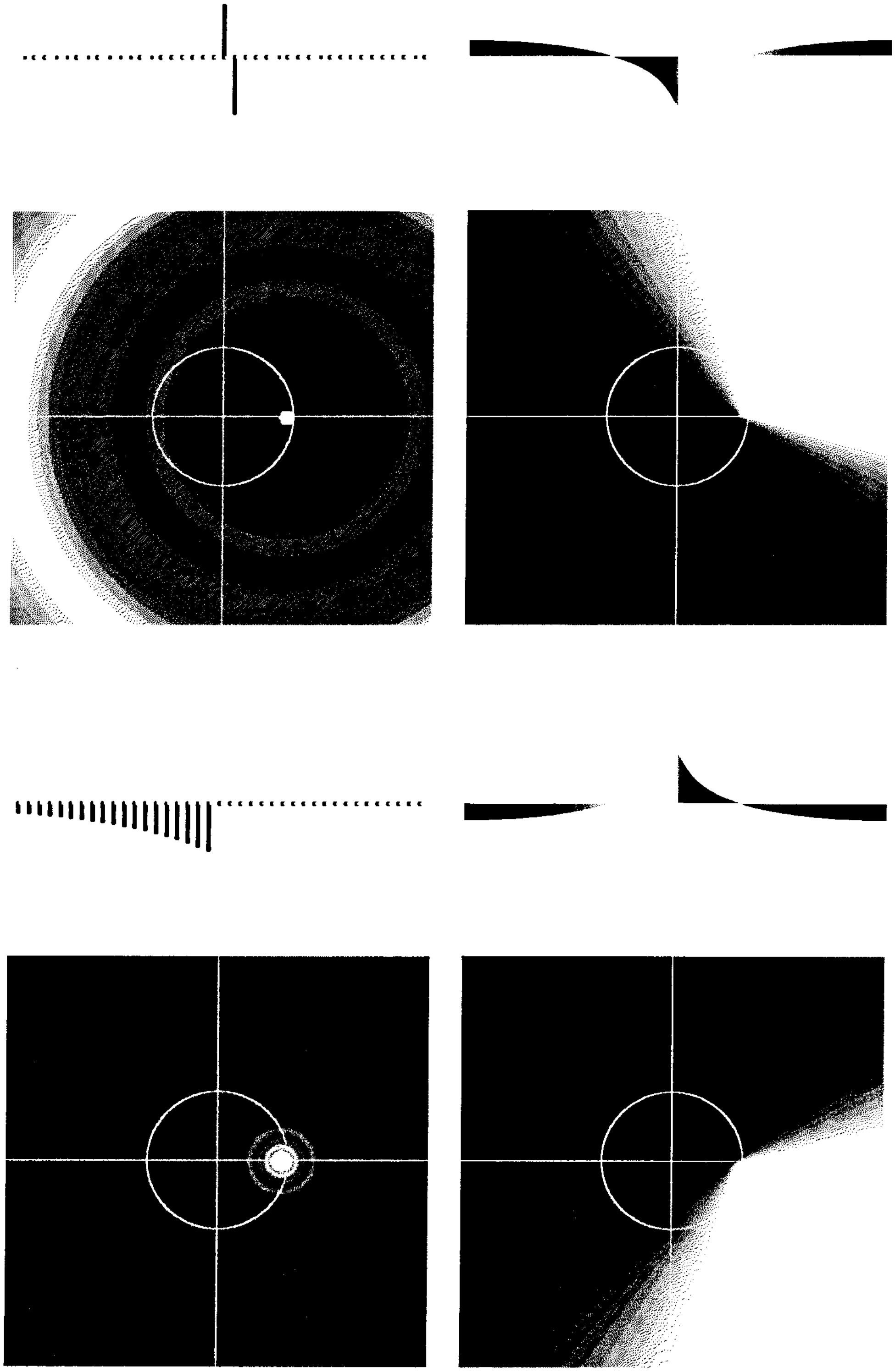
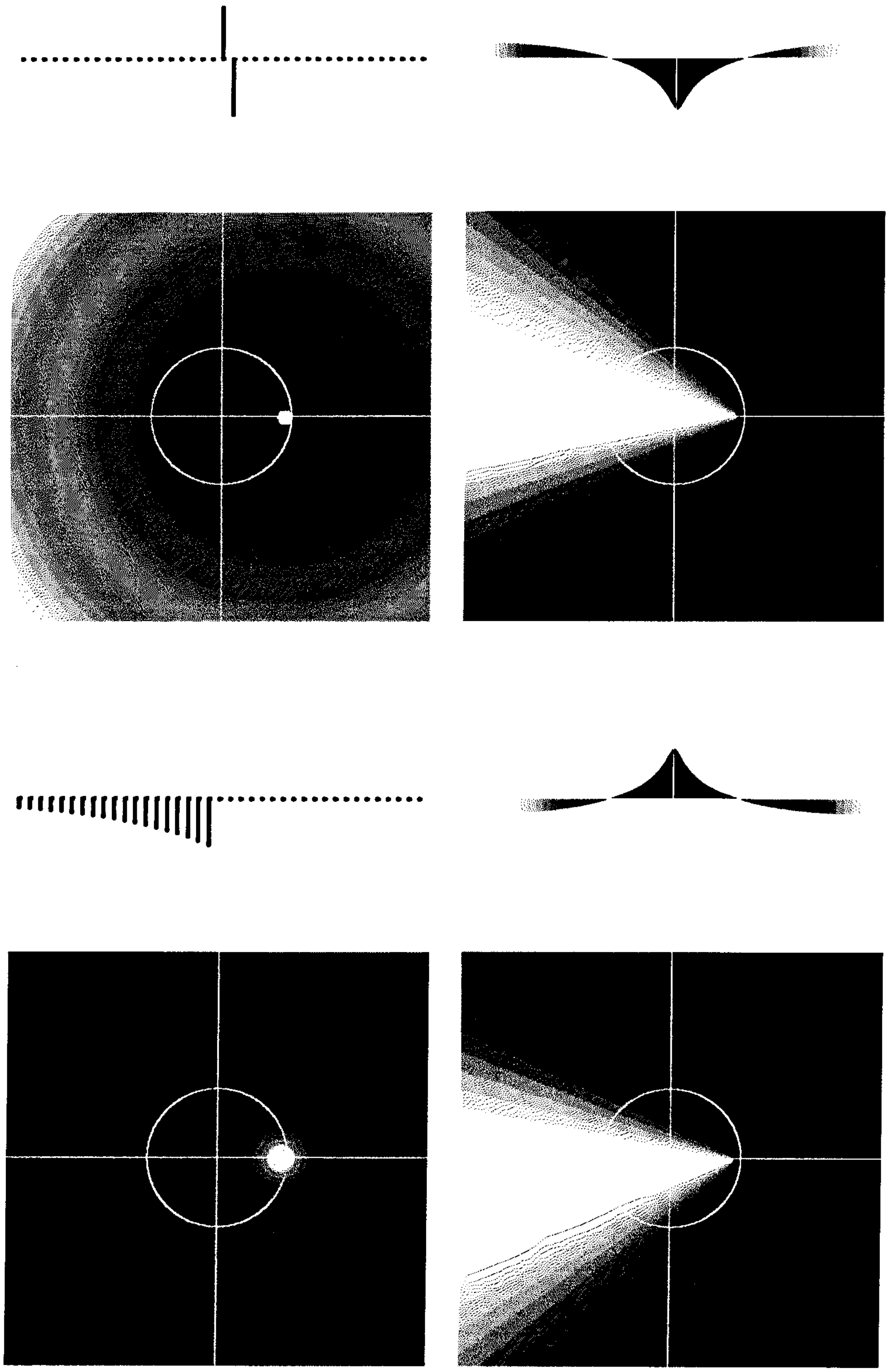


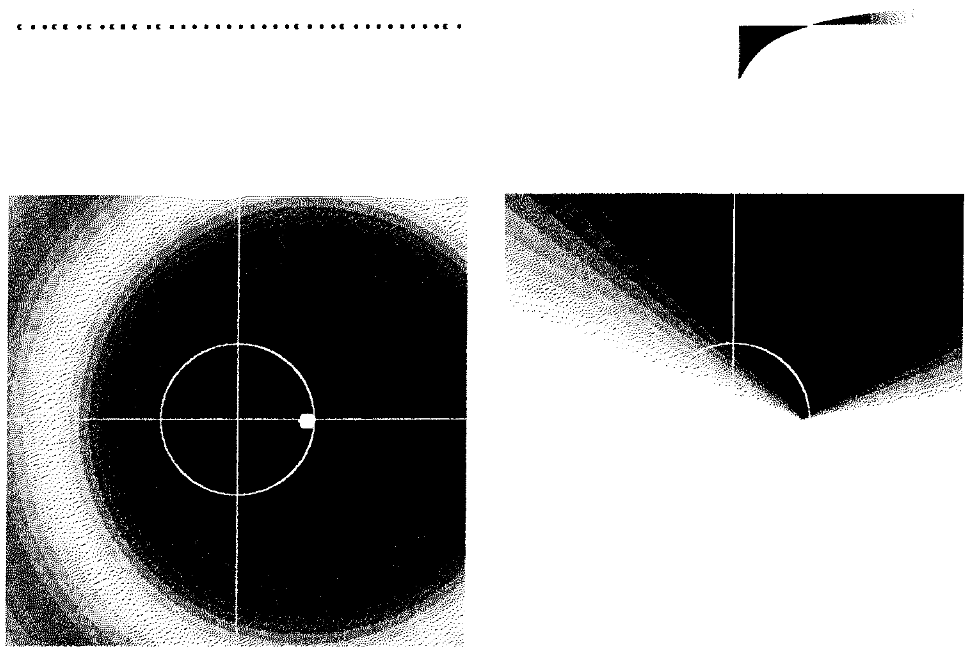
FIG. 2. Upper plot: $1 - 1.1Z$; Lower plot: $1/(1 - 1.1Z)$

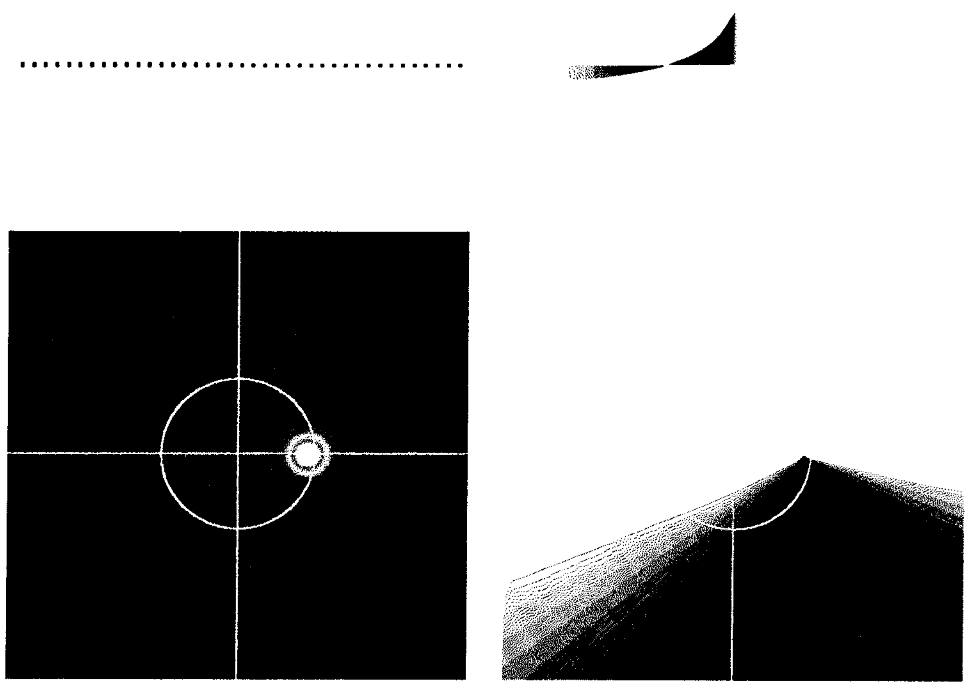


C



M





Y

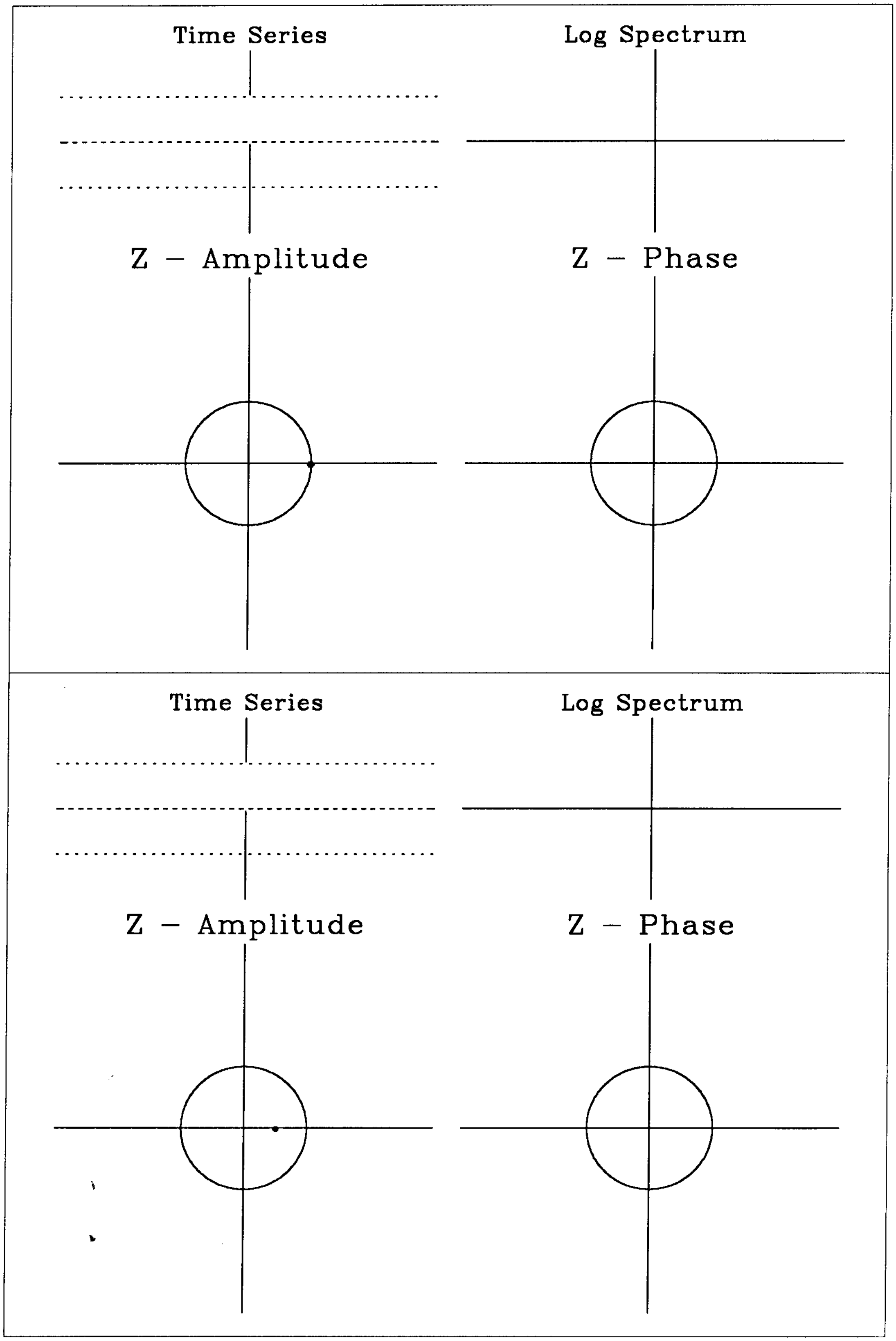
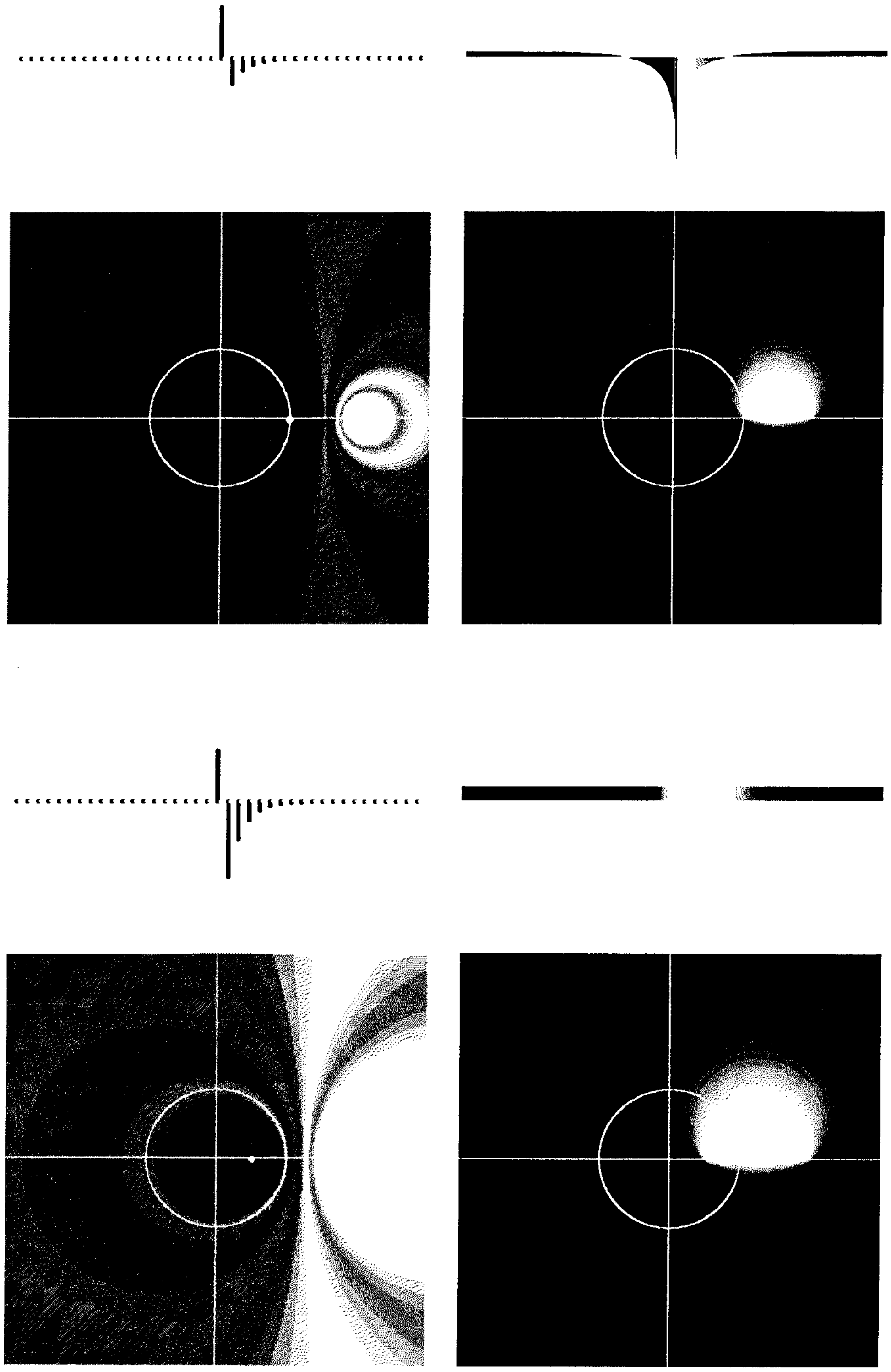
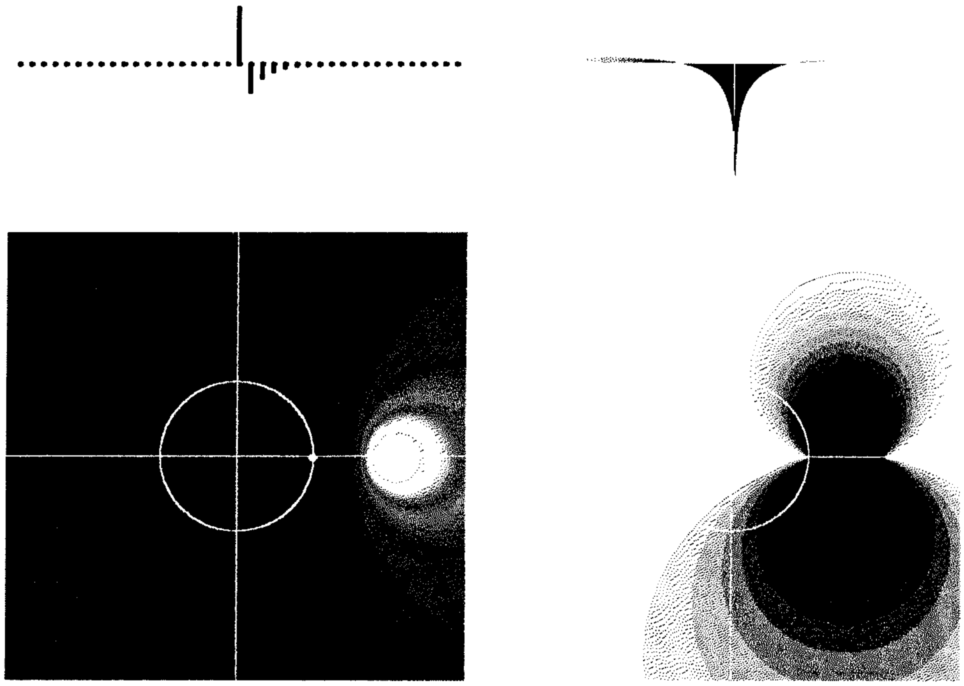
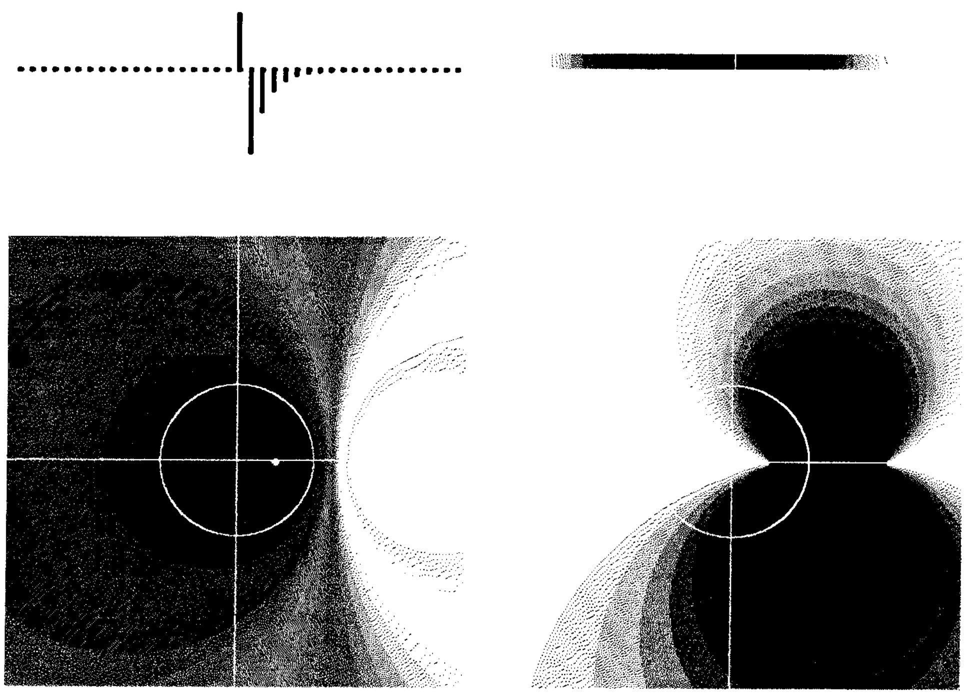


FIG. 3. Upper plot: Notch filter; Lower plot: All-pass filter

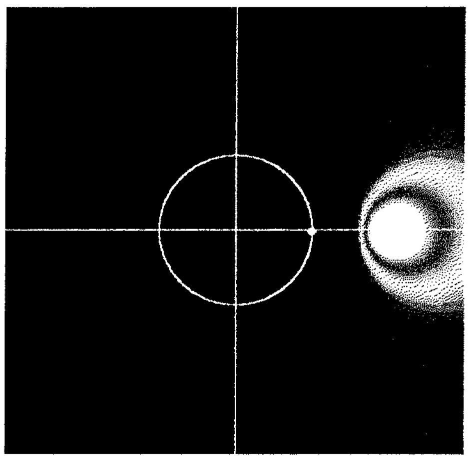


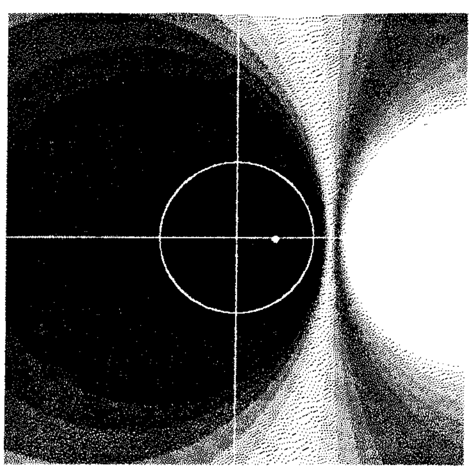
C

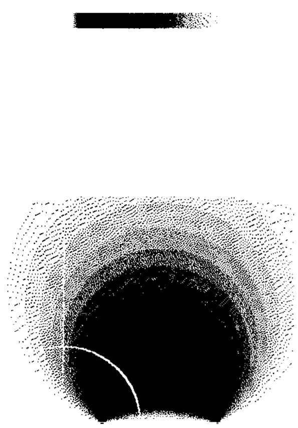




M







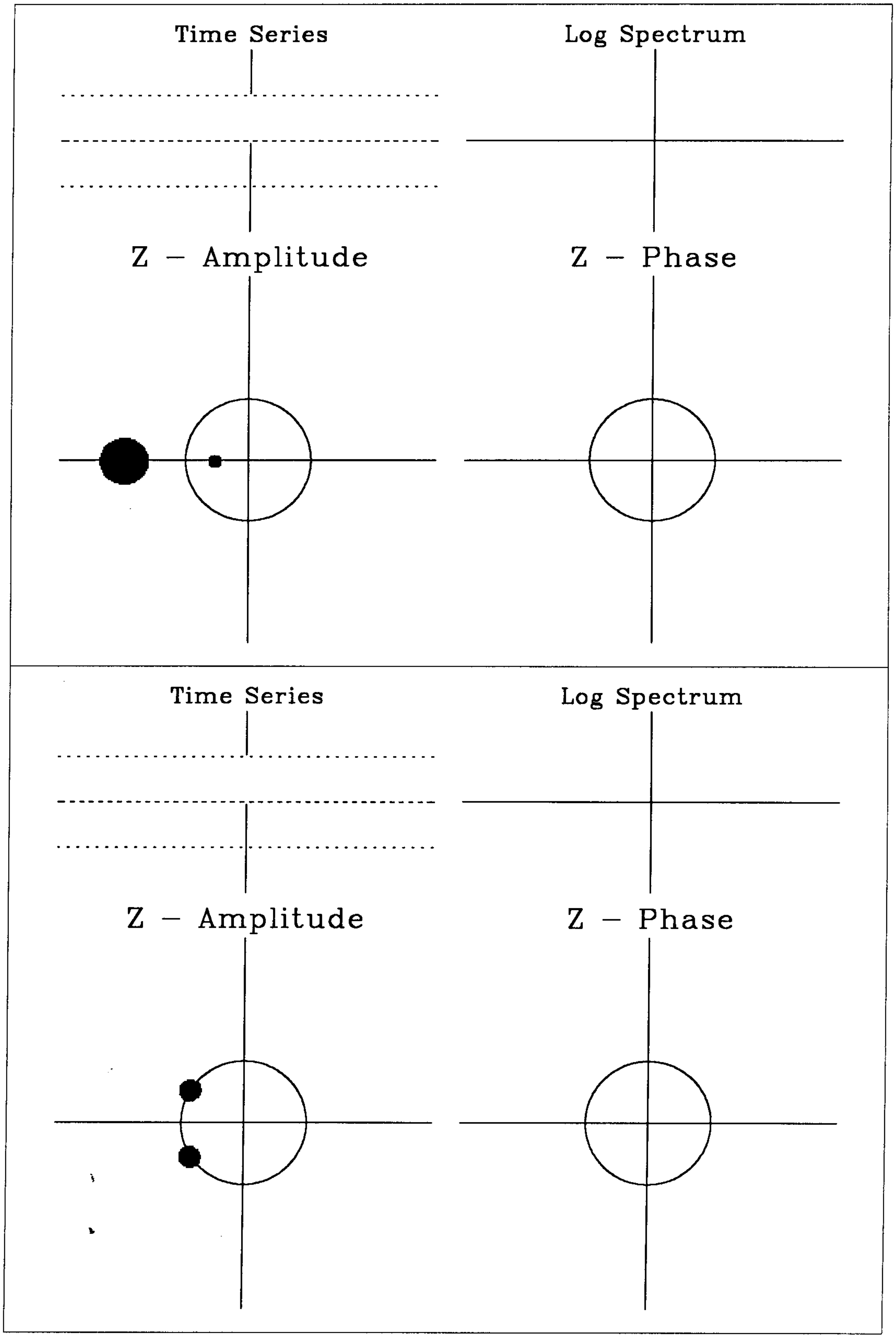
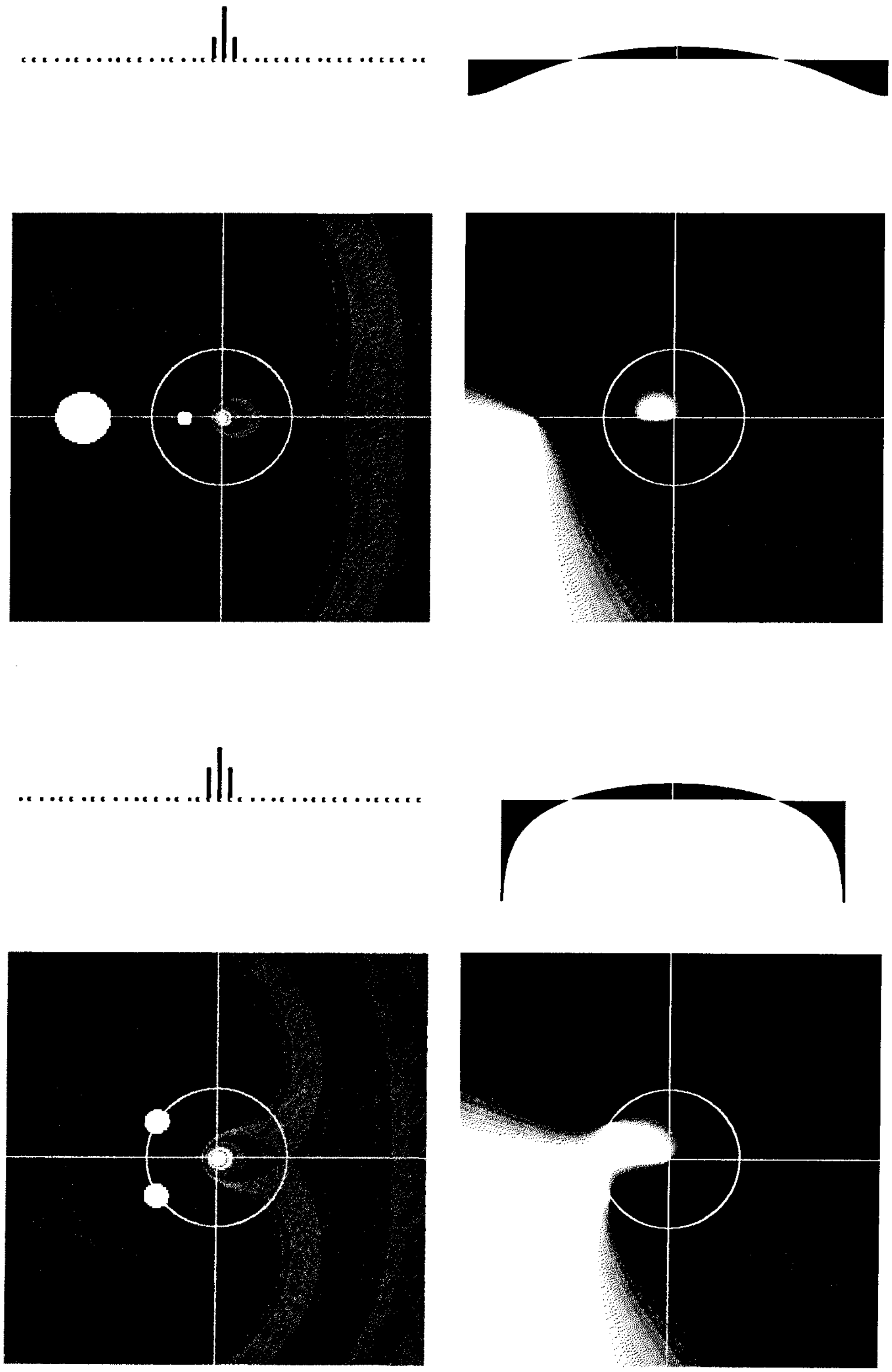
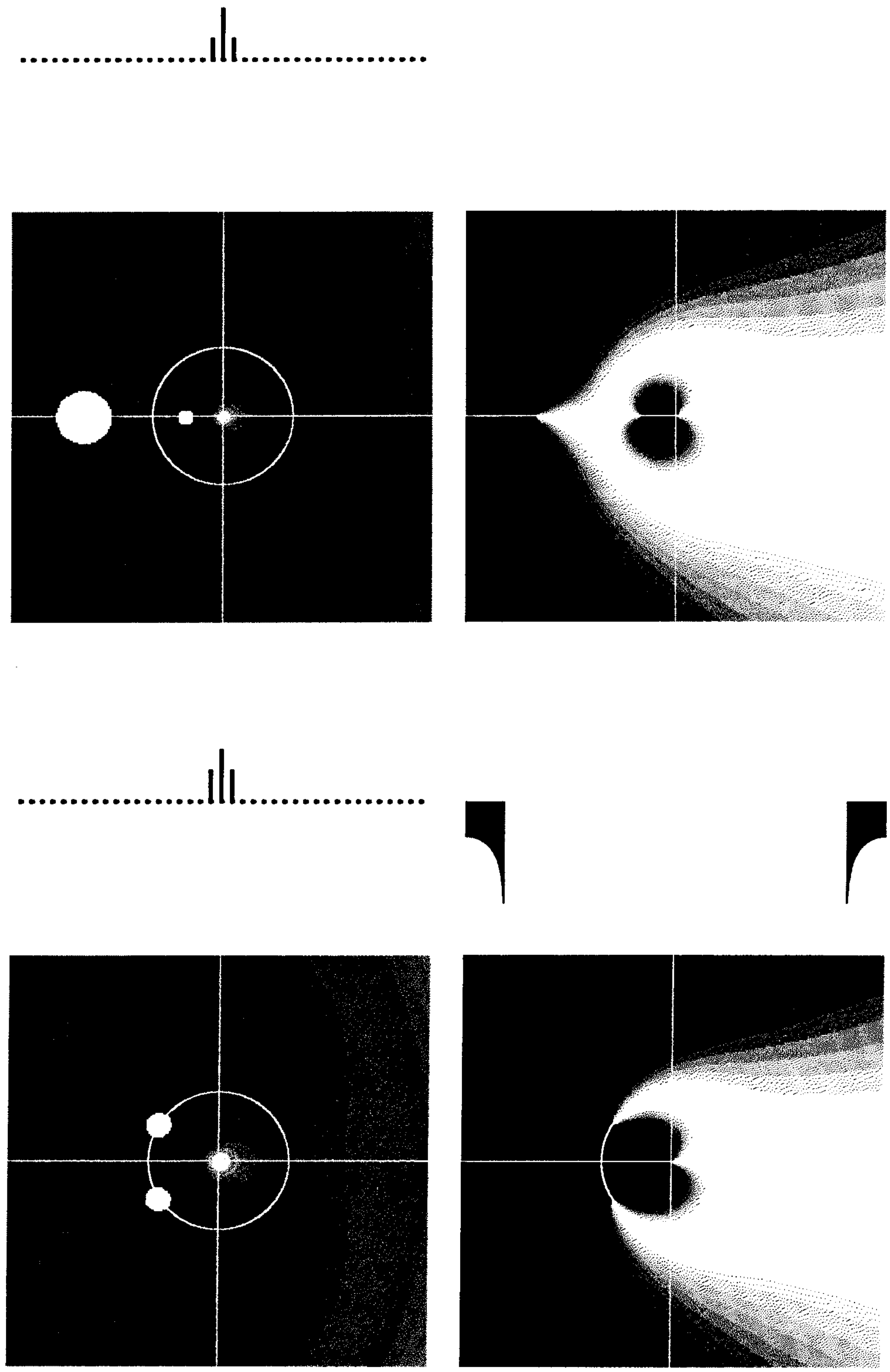


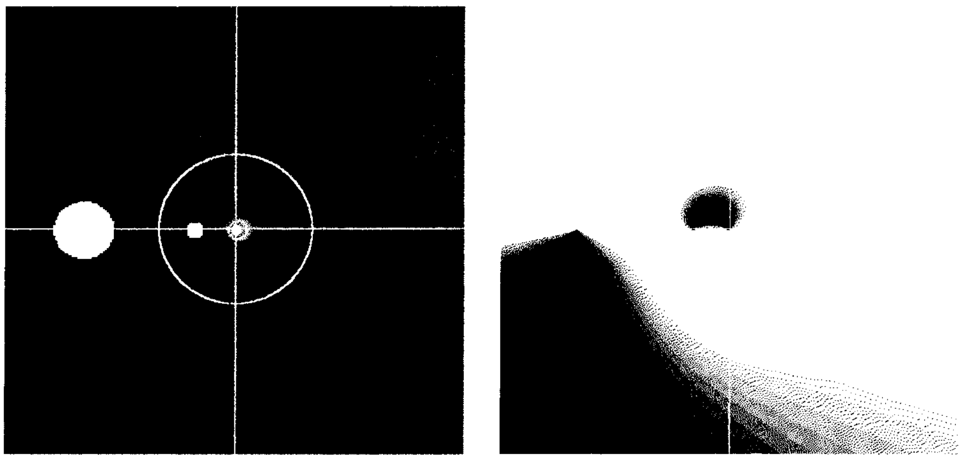
FIG. 4. Upper plot: $.4Z^{-1} + 1 + .4Z$; Lower plot: $.6Z^{-1} + 1 + .6Z$

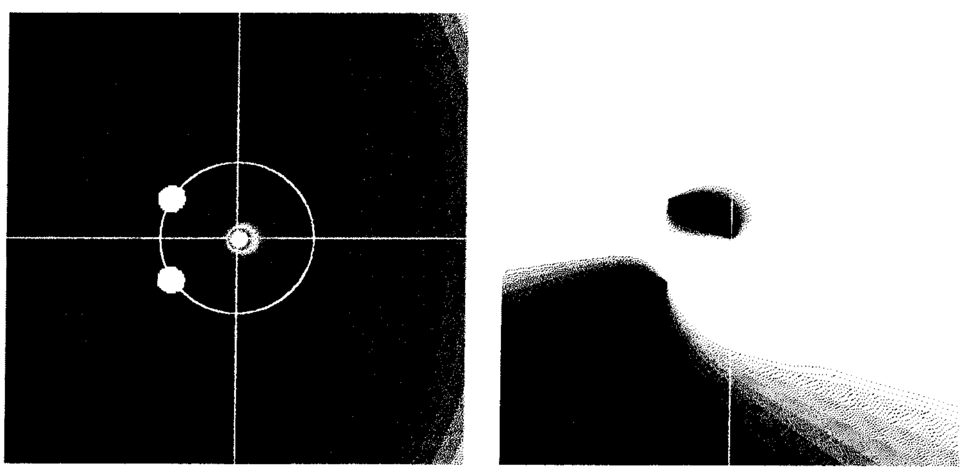


C



M





Y

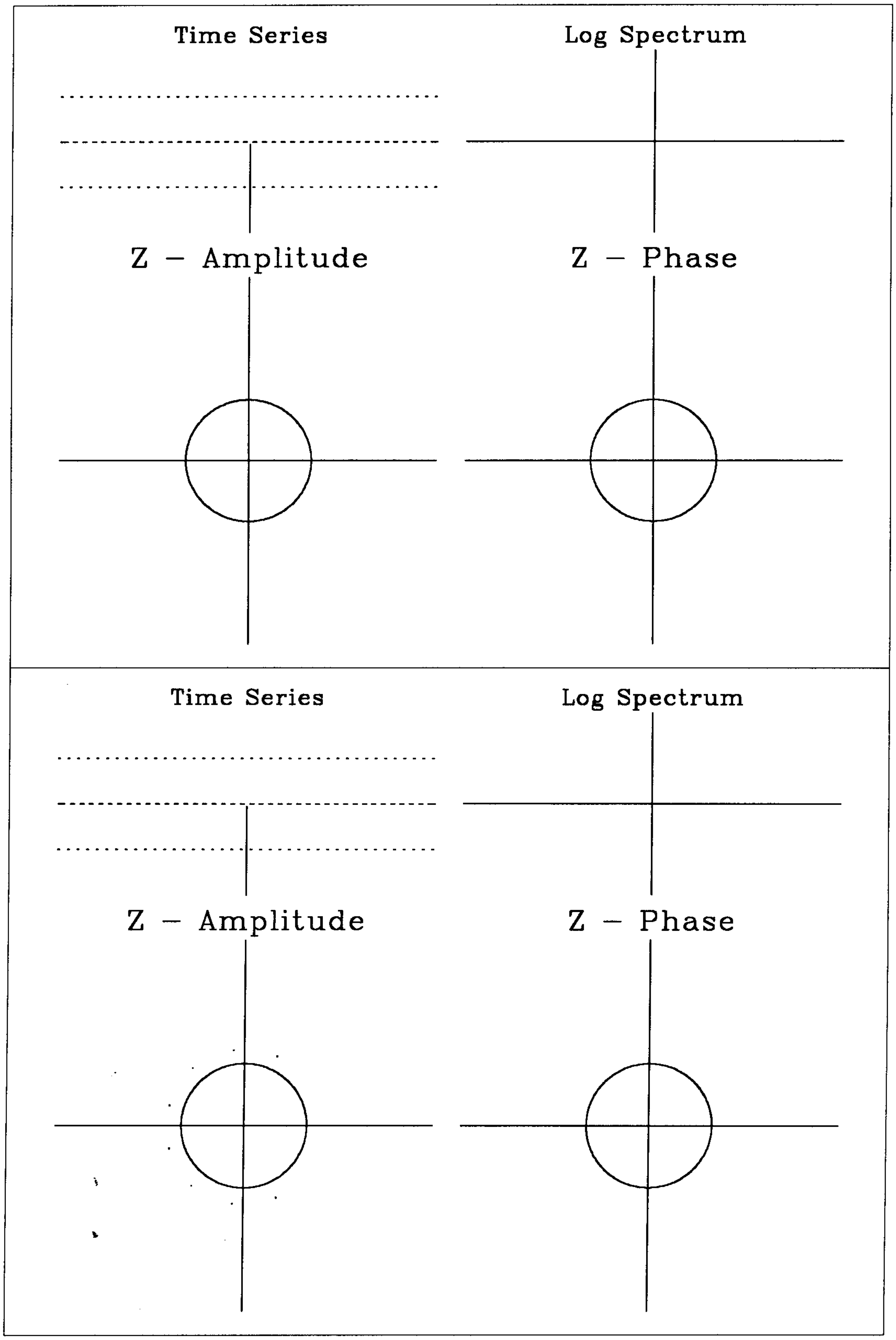
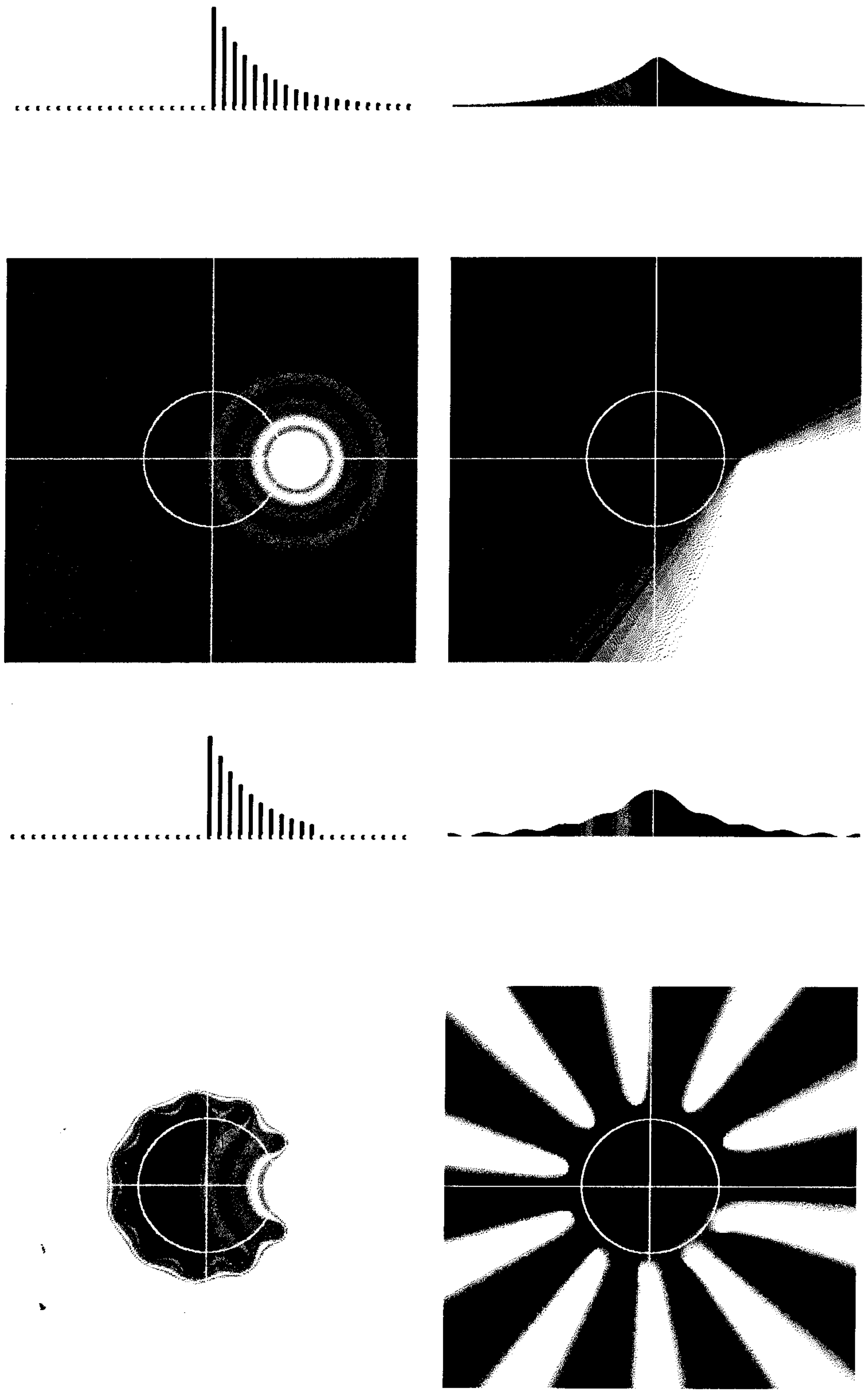
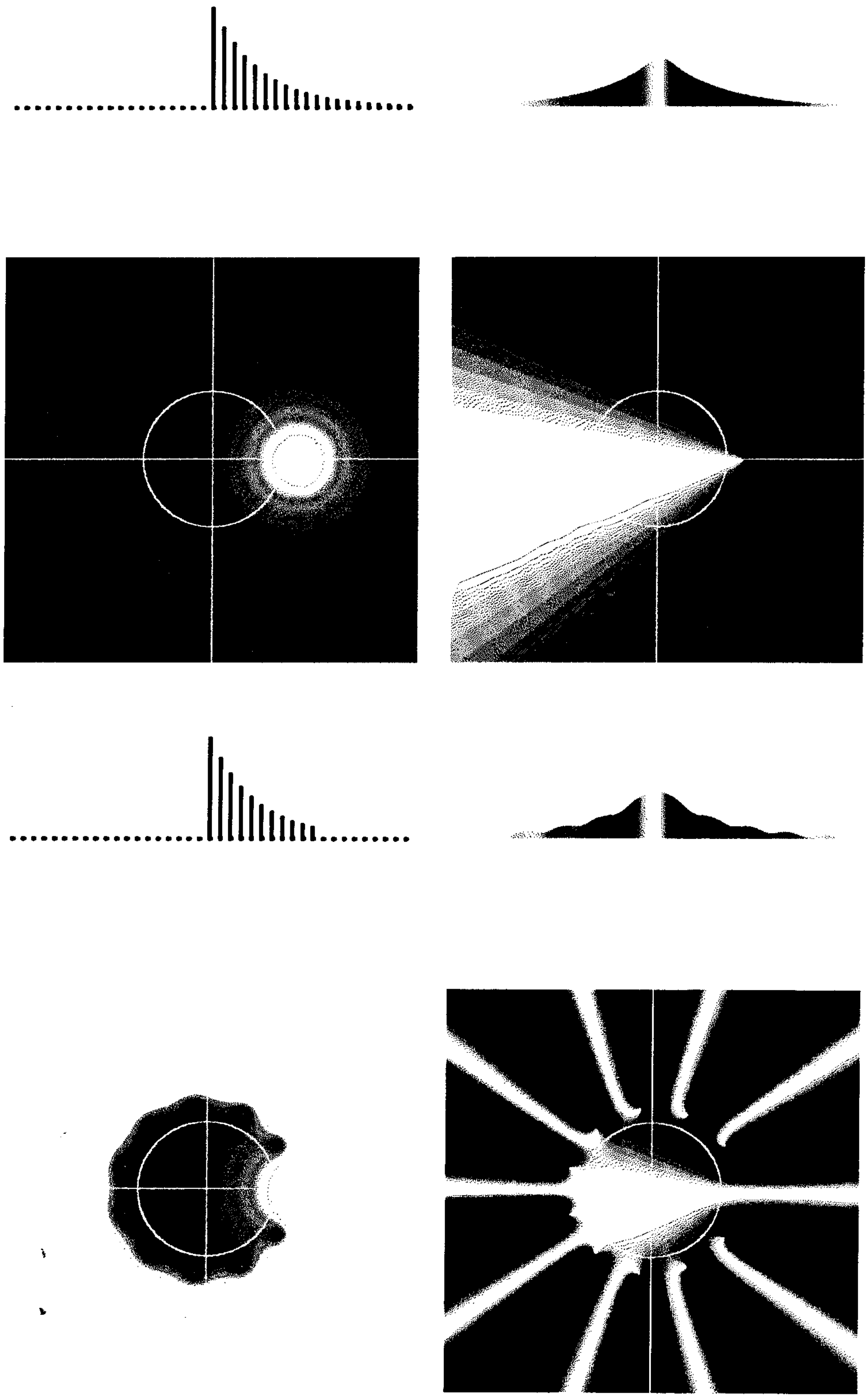


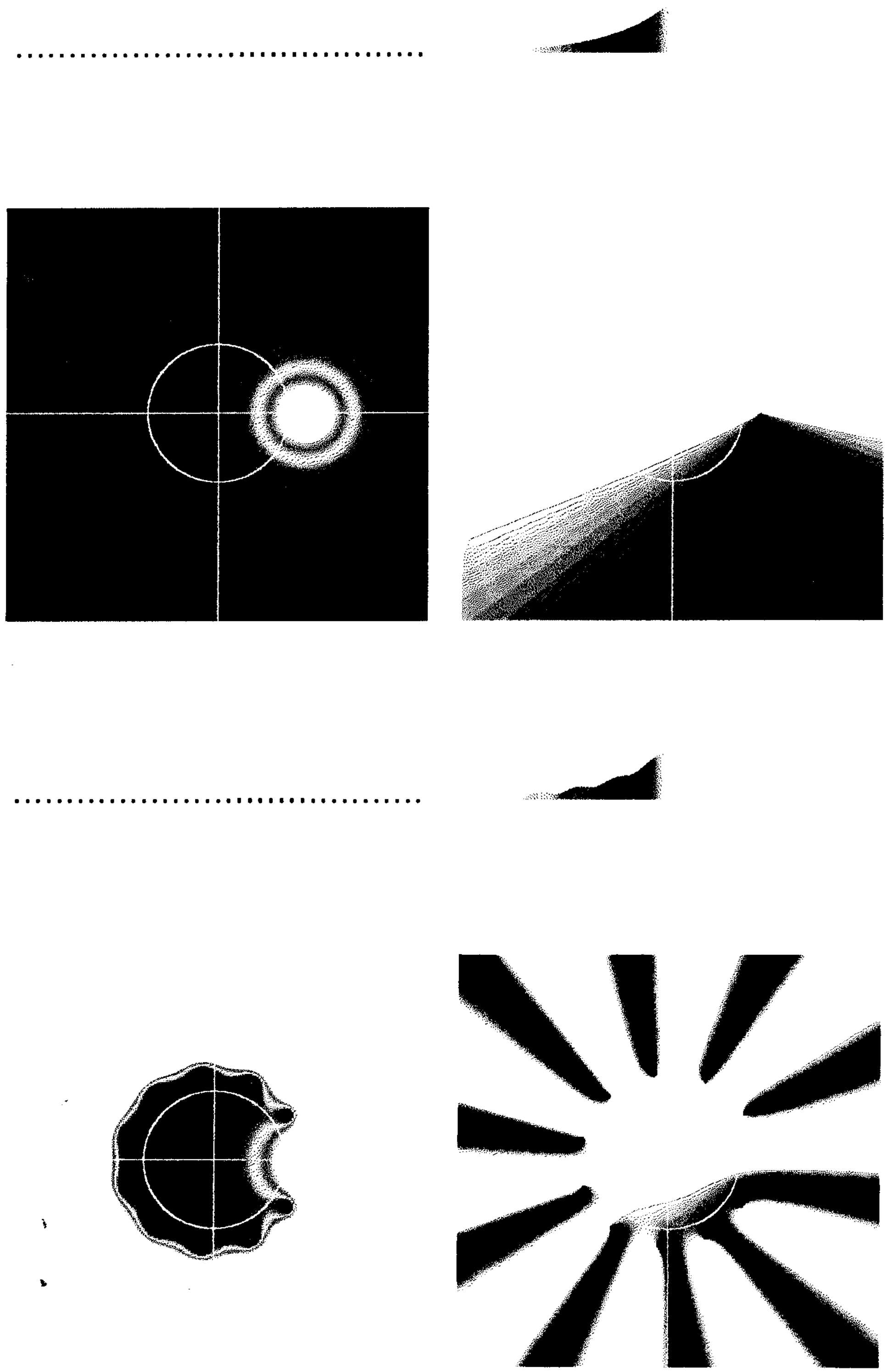
FIG. 5. Upper plot: $2/(1-.8Z)$; Lower plot: $2(1+.8Z+.8^2Z^2+\ldots+.8^{10}Z^{10})$



C



M



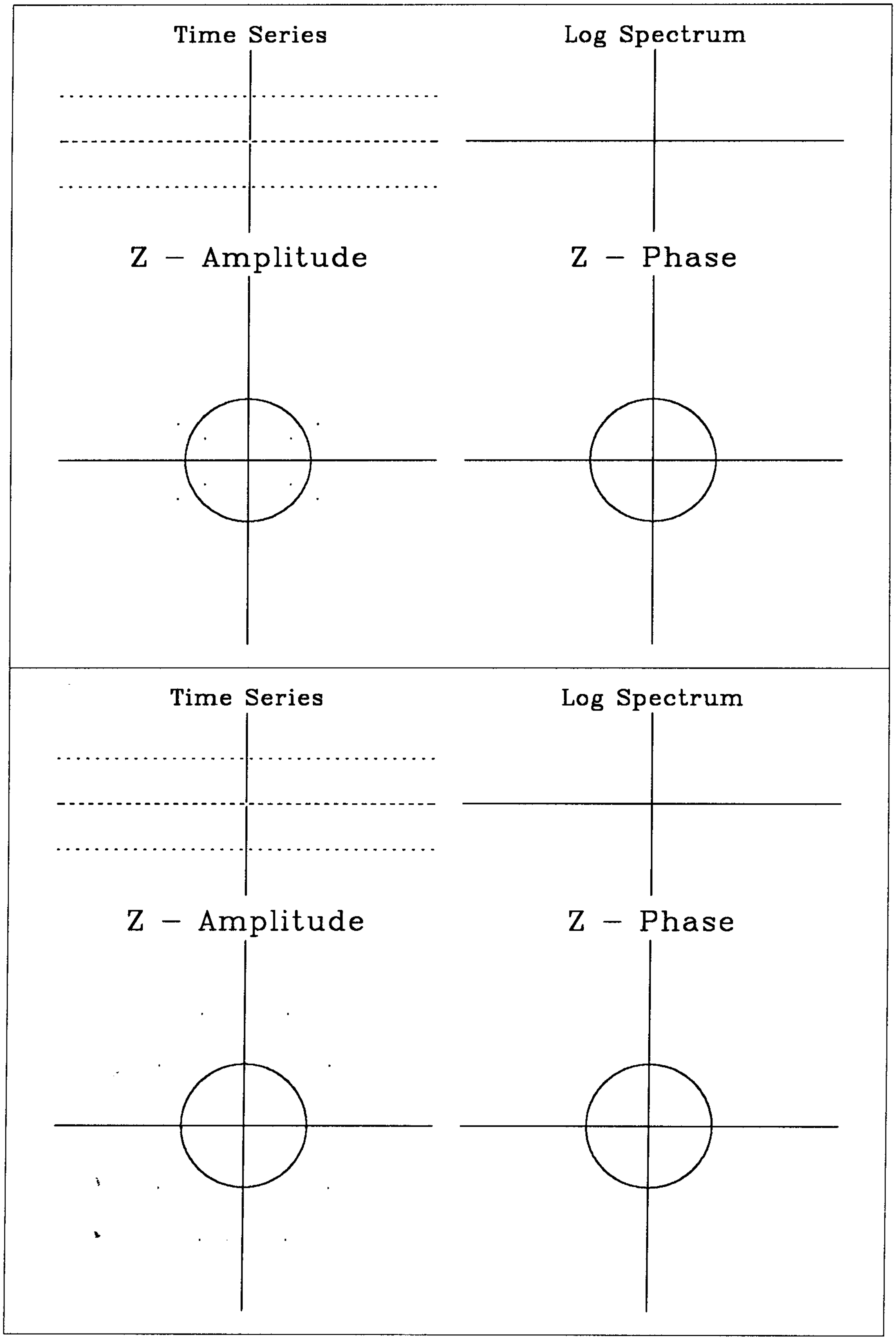
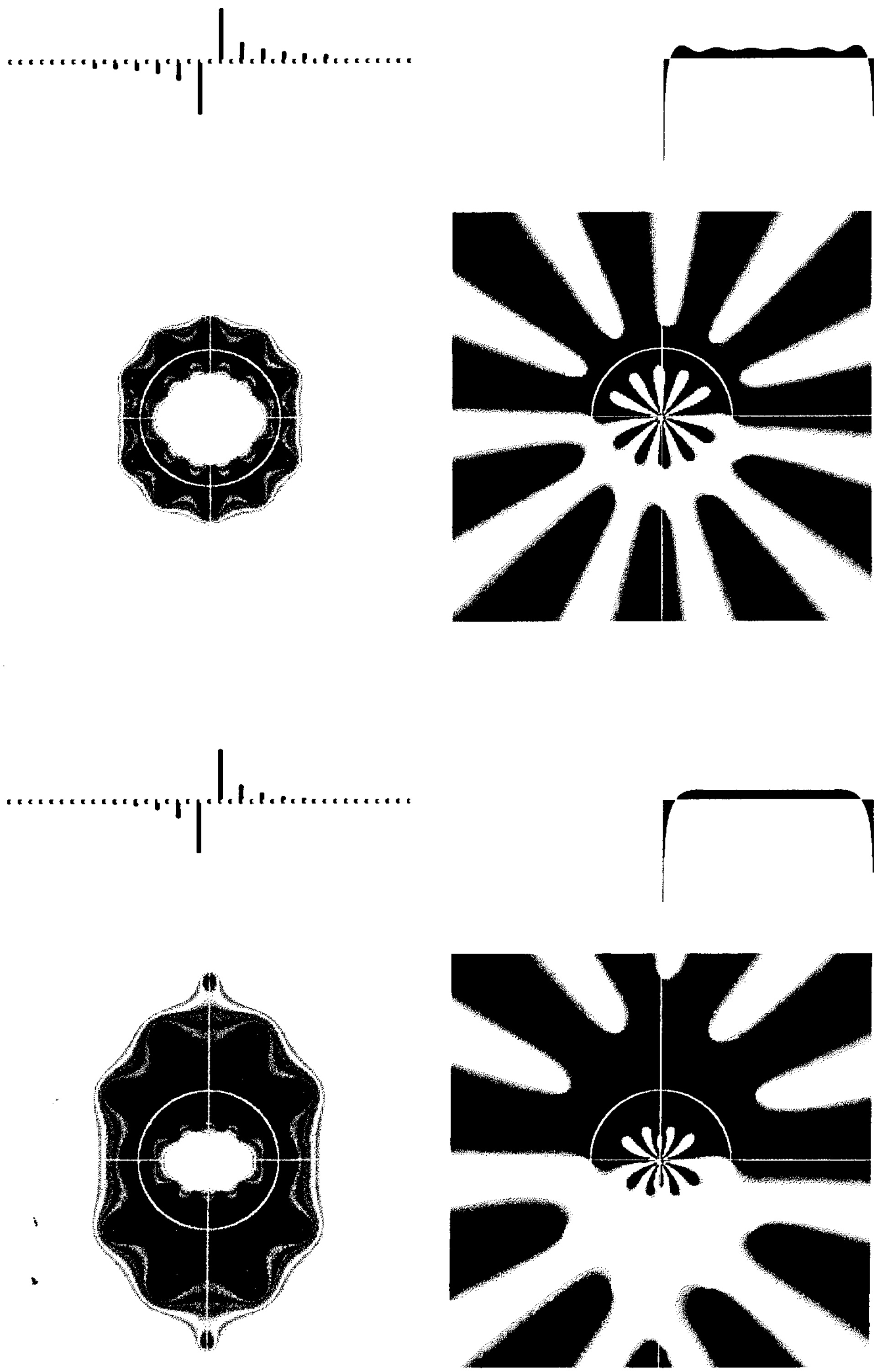
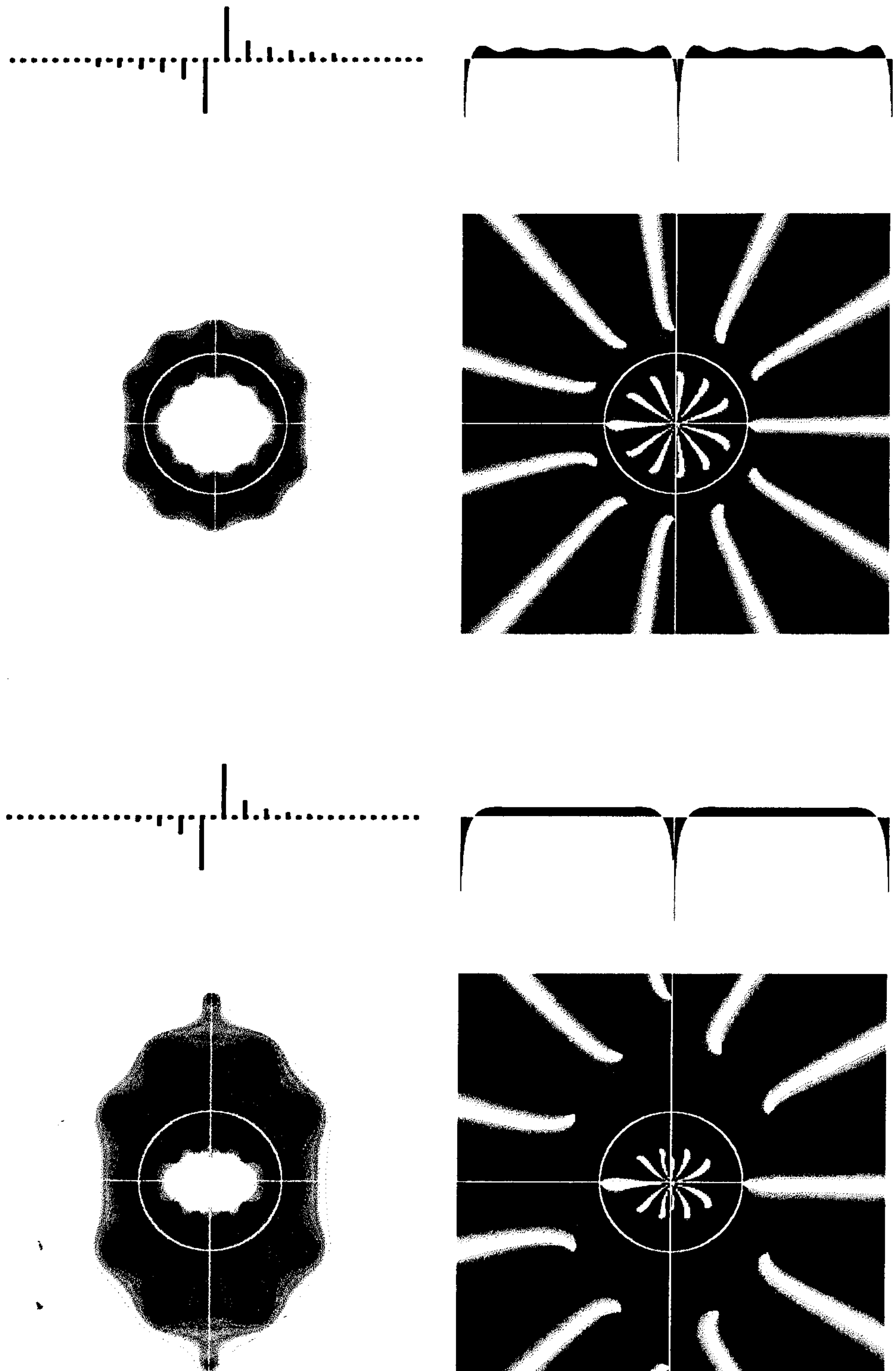
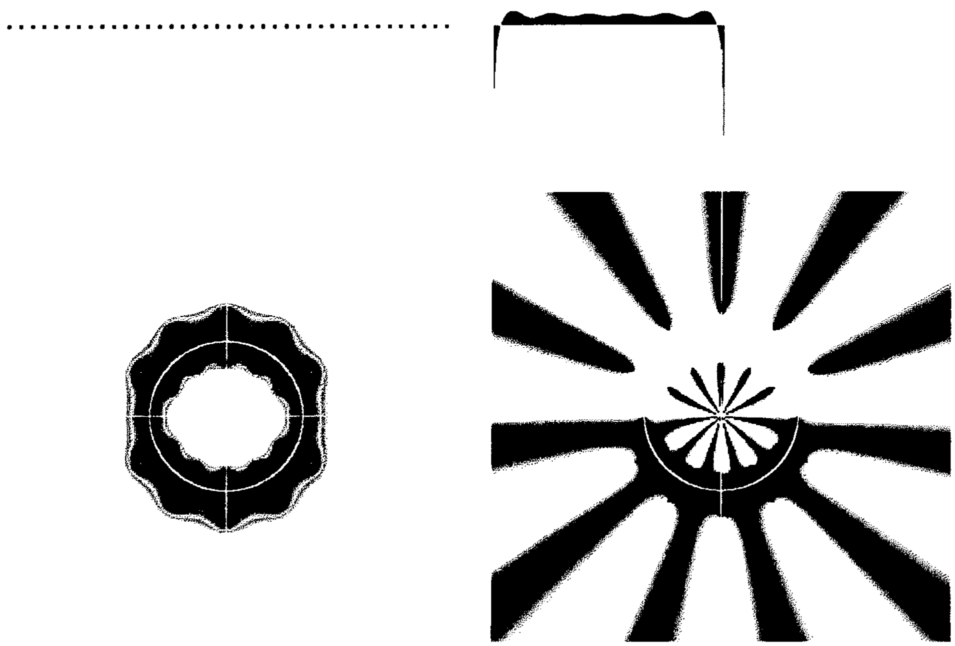


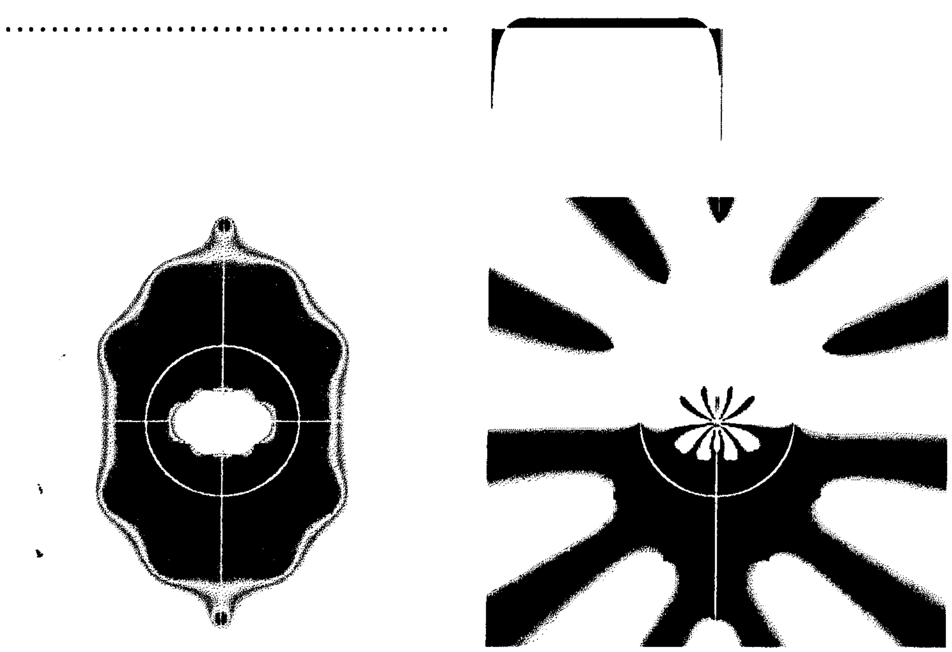
FIG. 6. Upper plot: Hilbert function with boxcar window; Lower plot: Hilbert function with cosine taper.



C







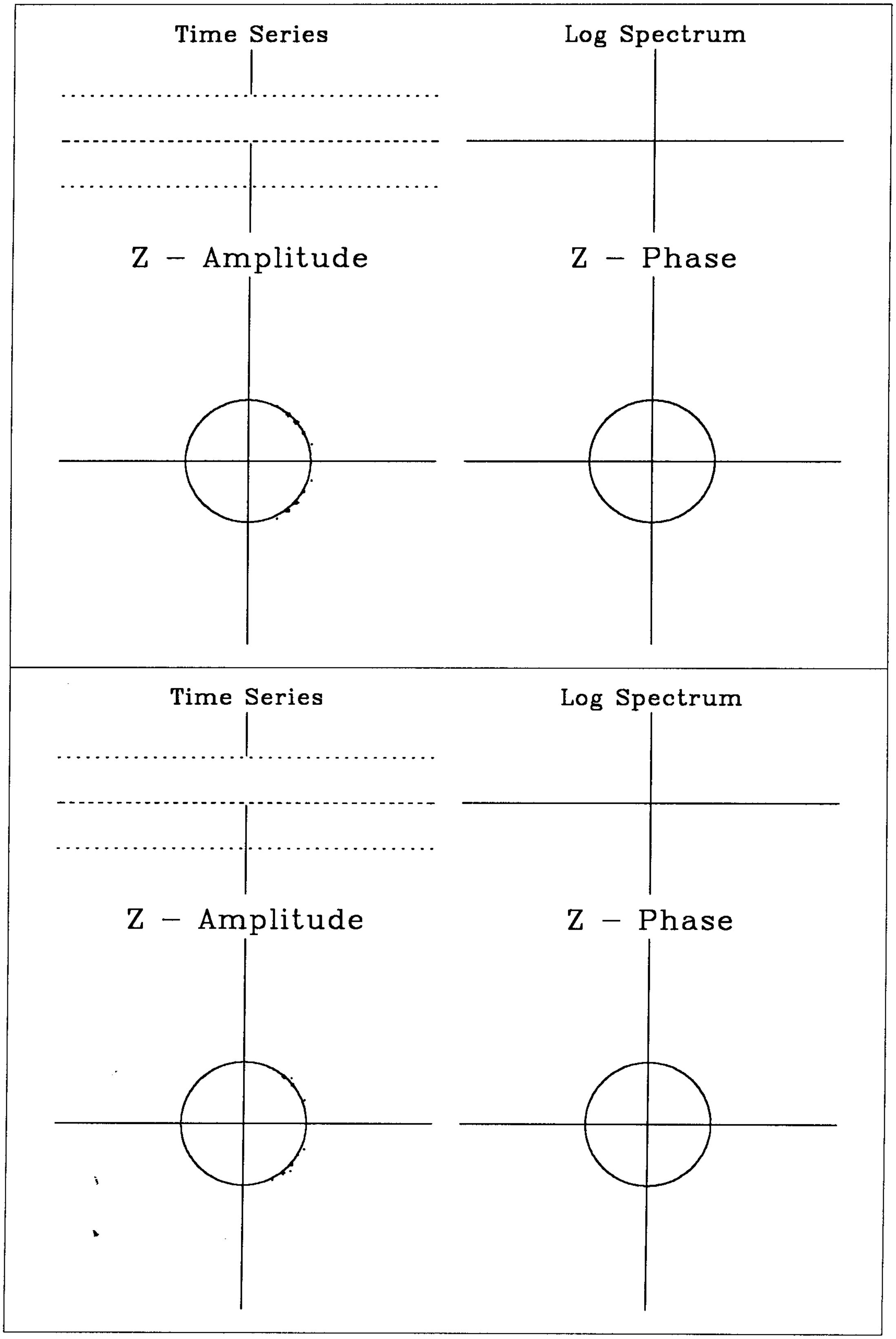
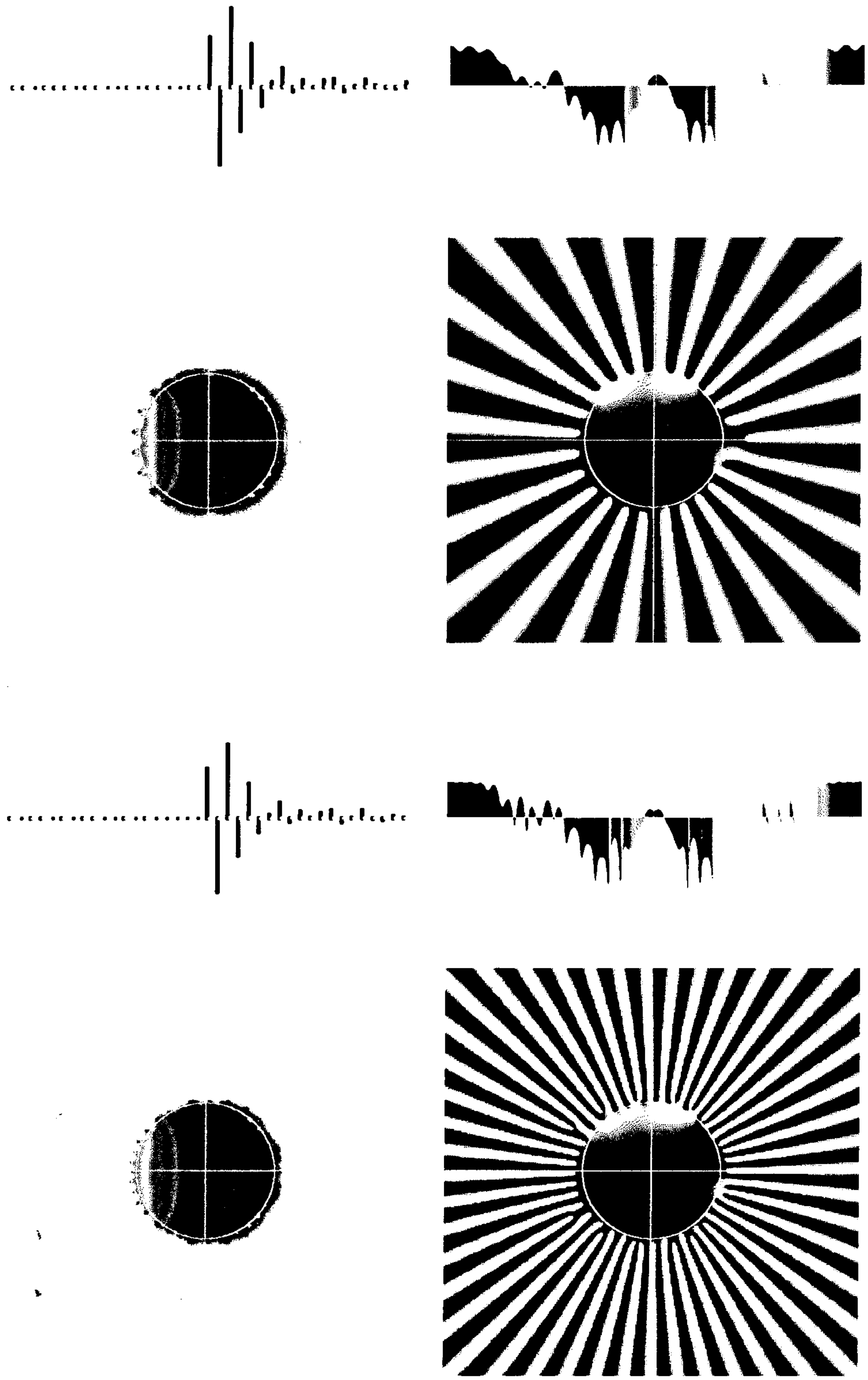
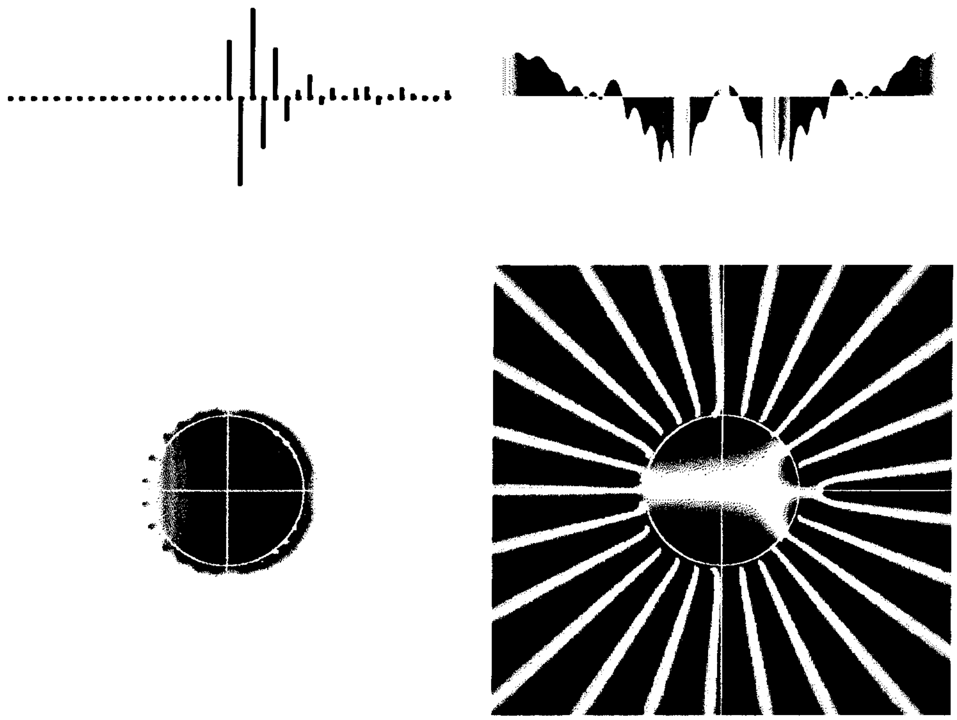
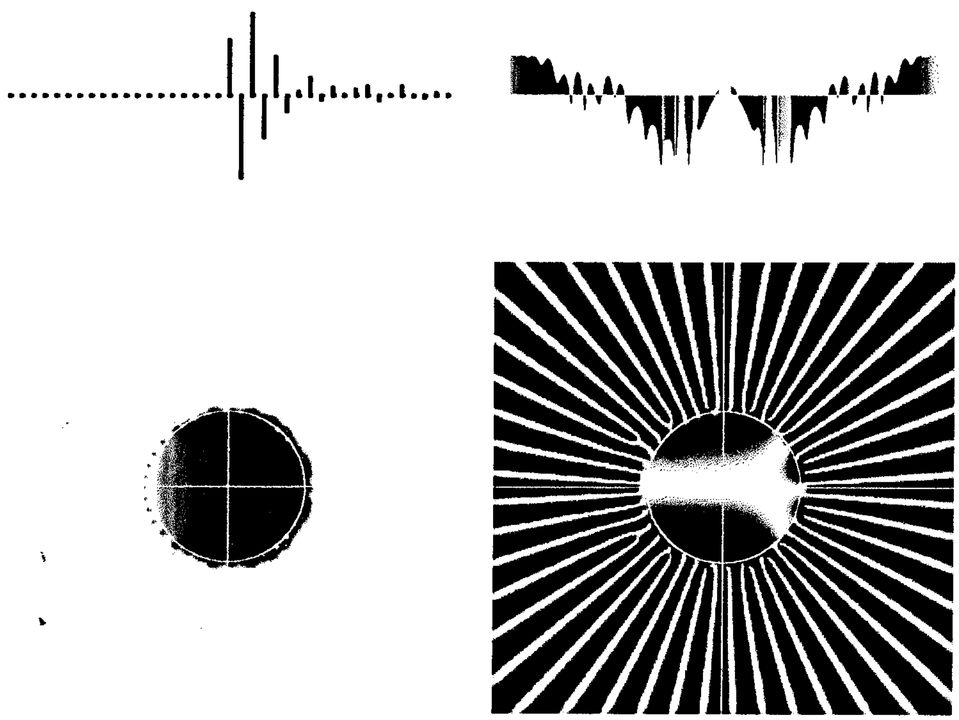


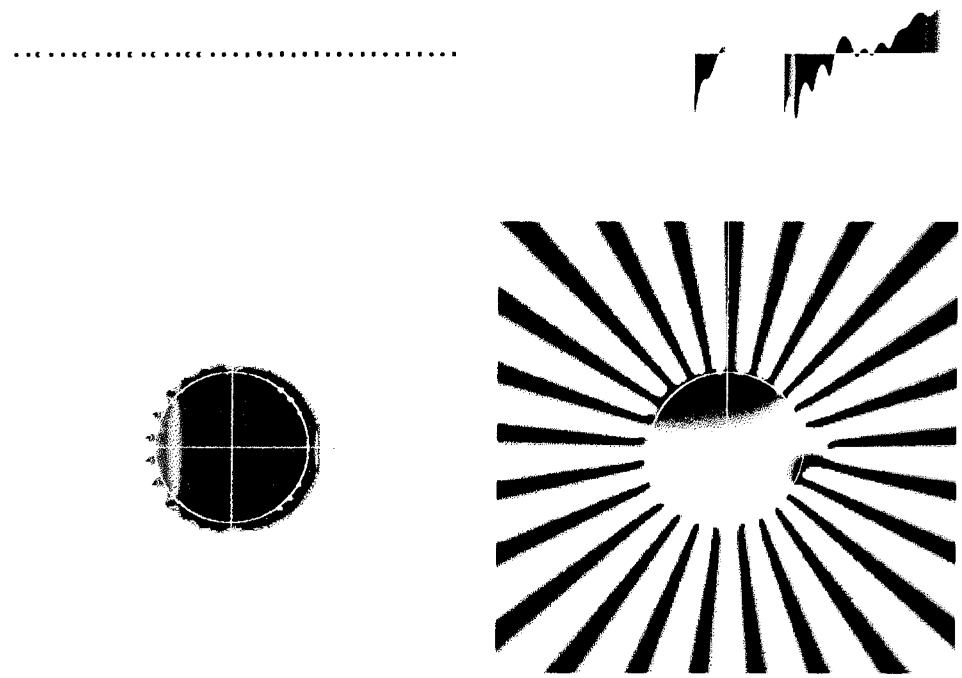
FIG. 7. Upper plot: Filter at the 25th step of the Levinson recursion; Lower plot: Filter, at the final (50th) step.

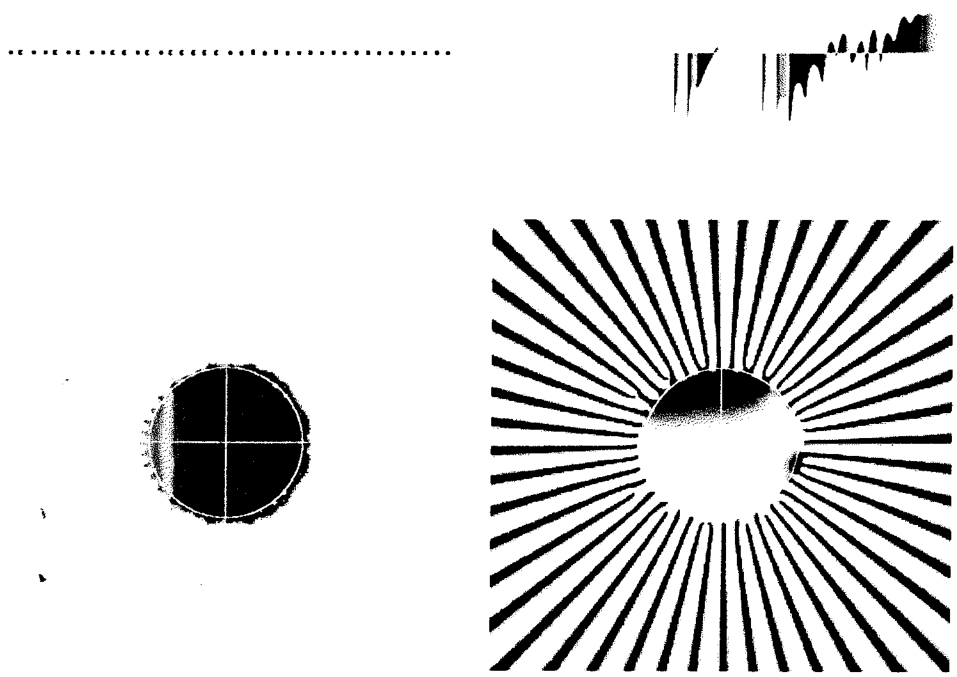


C









Y

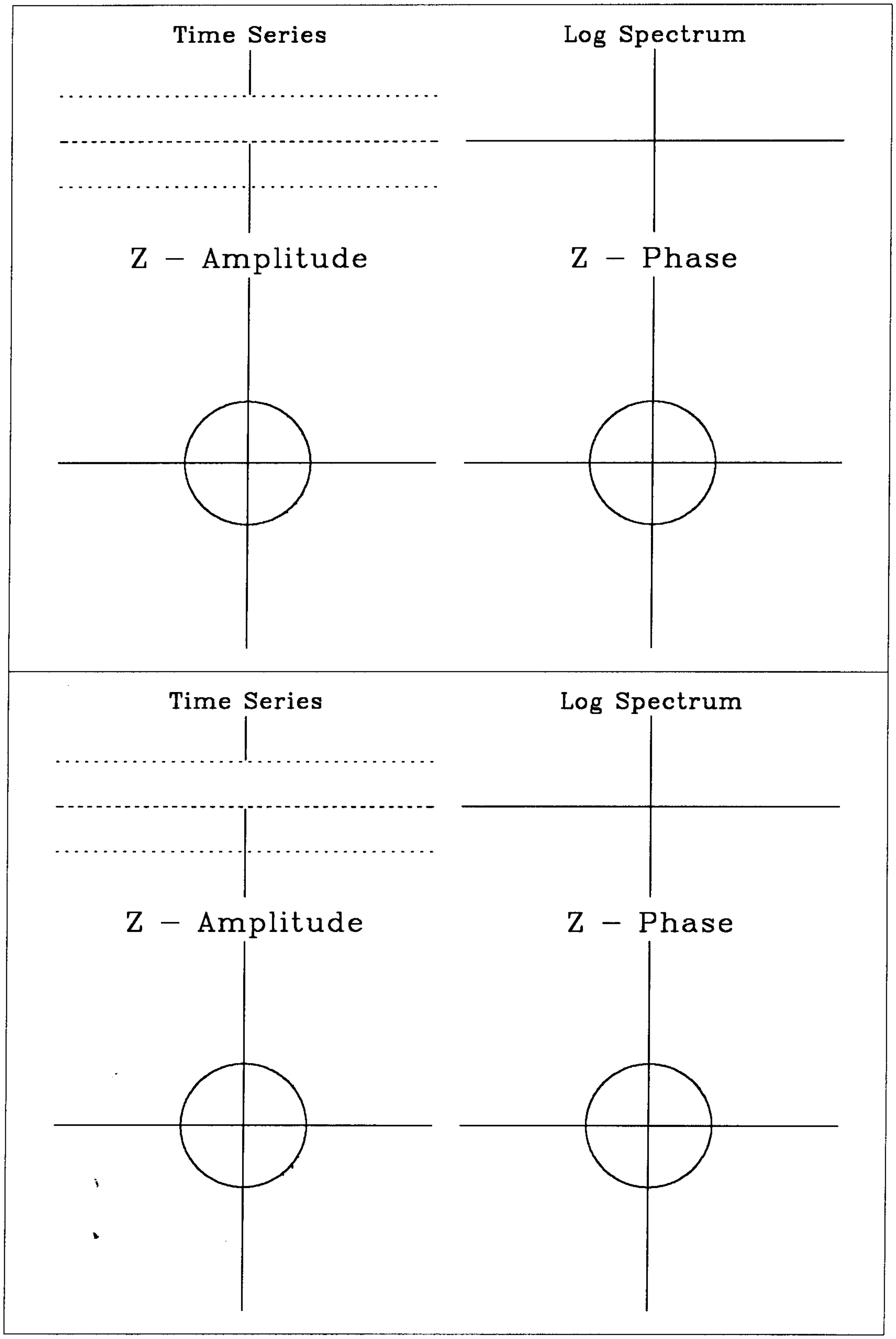
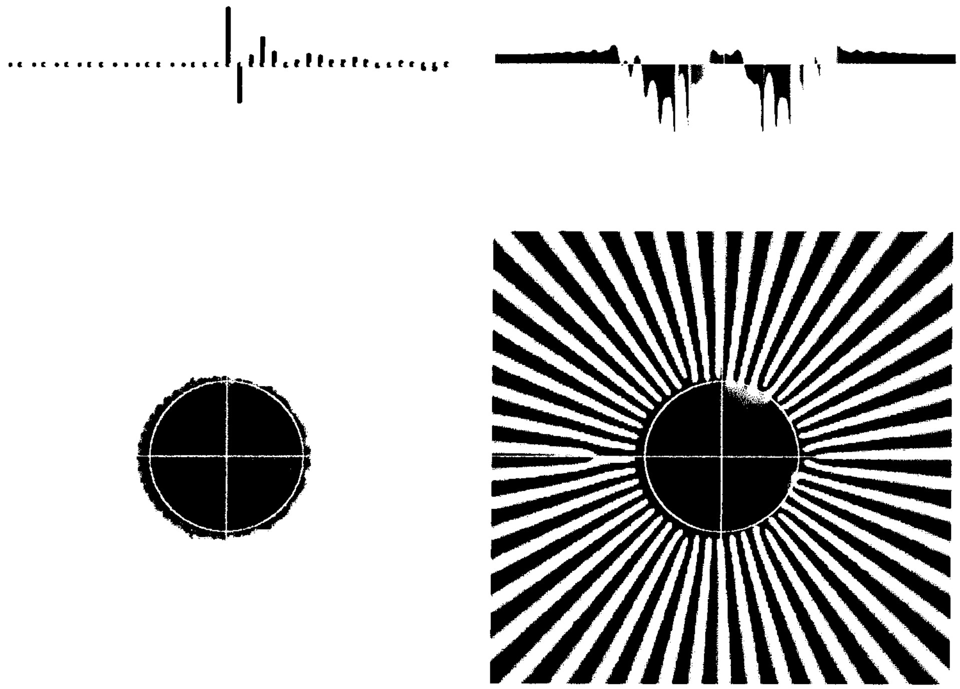
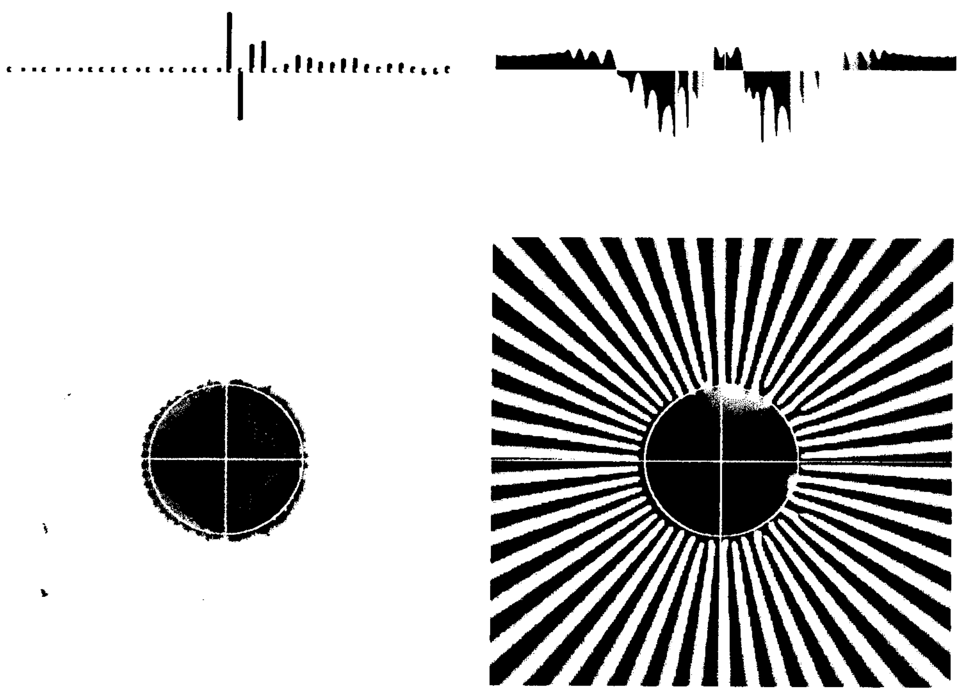
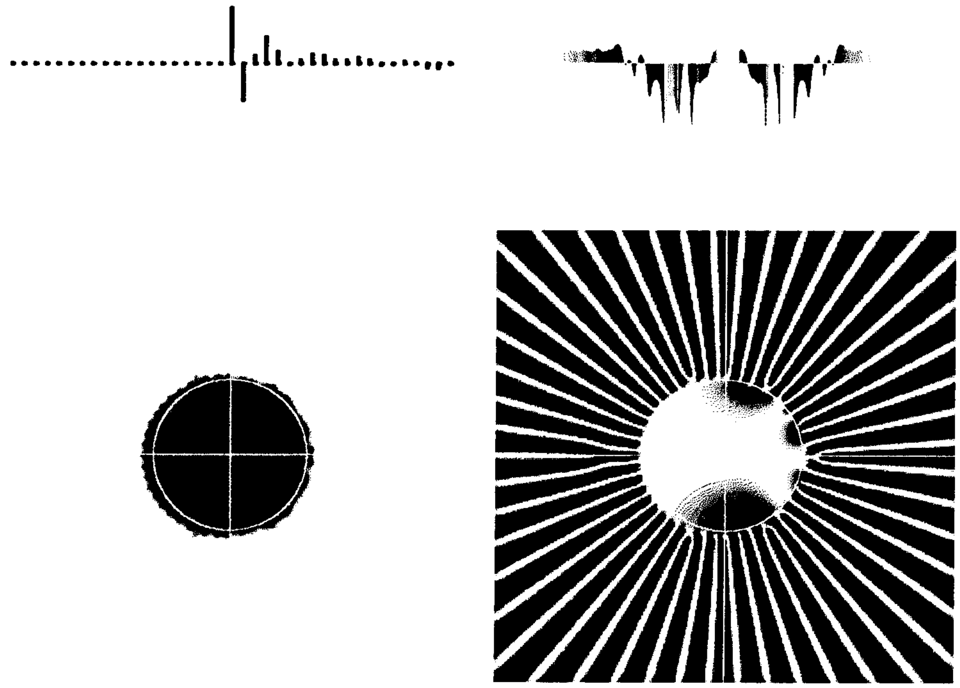


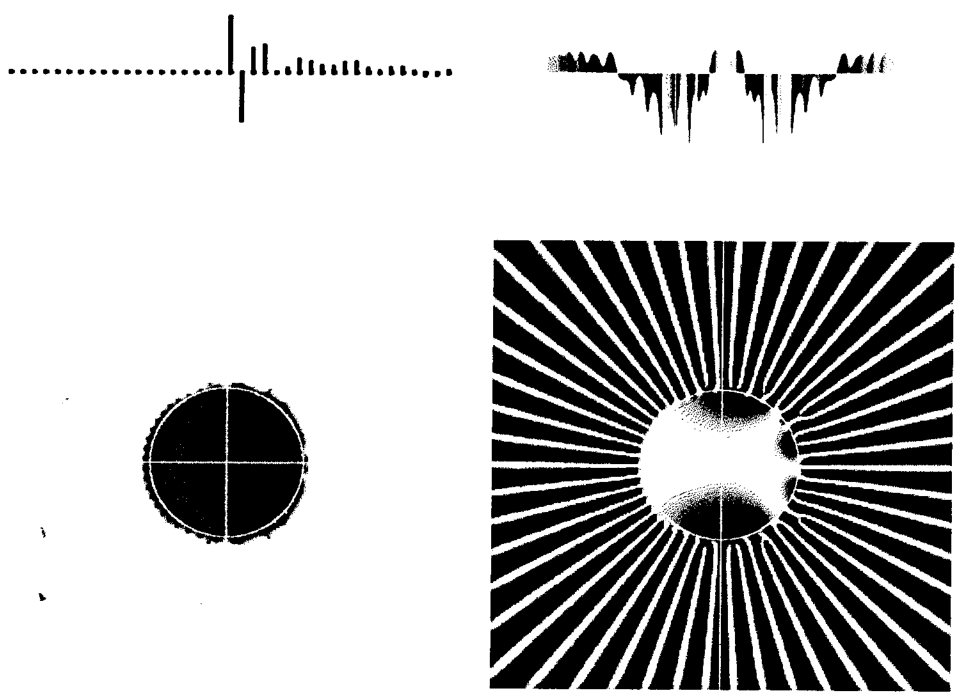
FIG. 8. Upper plot: Filter at the 25th step of the conjugate-gradients algorithm; Lower plot: Filter, at the final (50th) step.



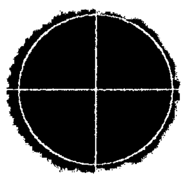


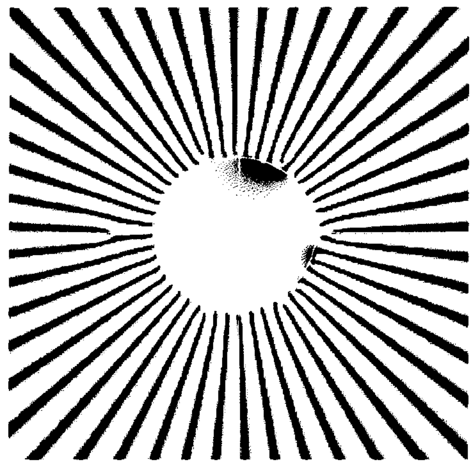
C

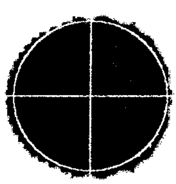


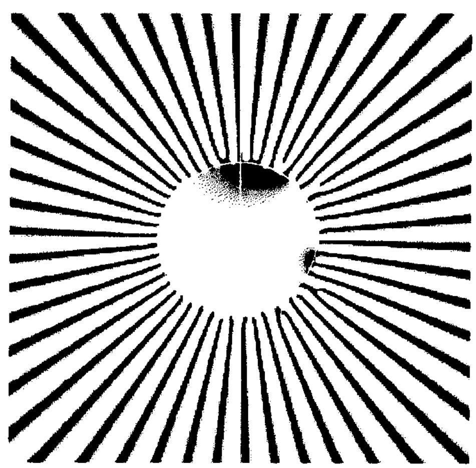


M









Y

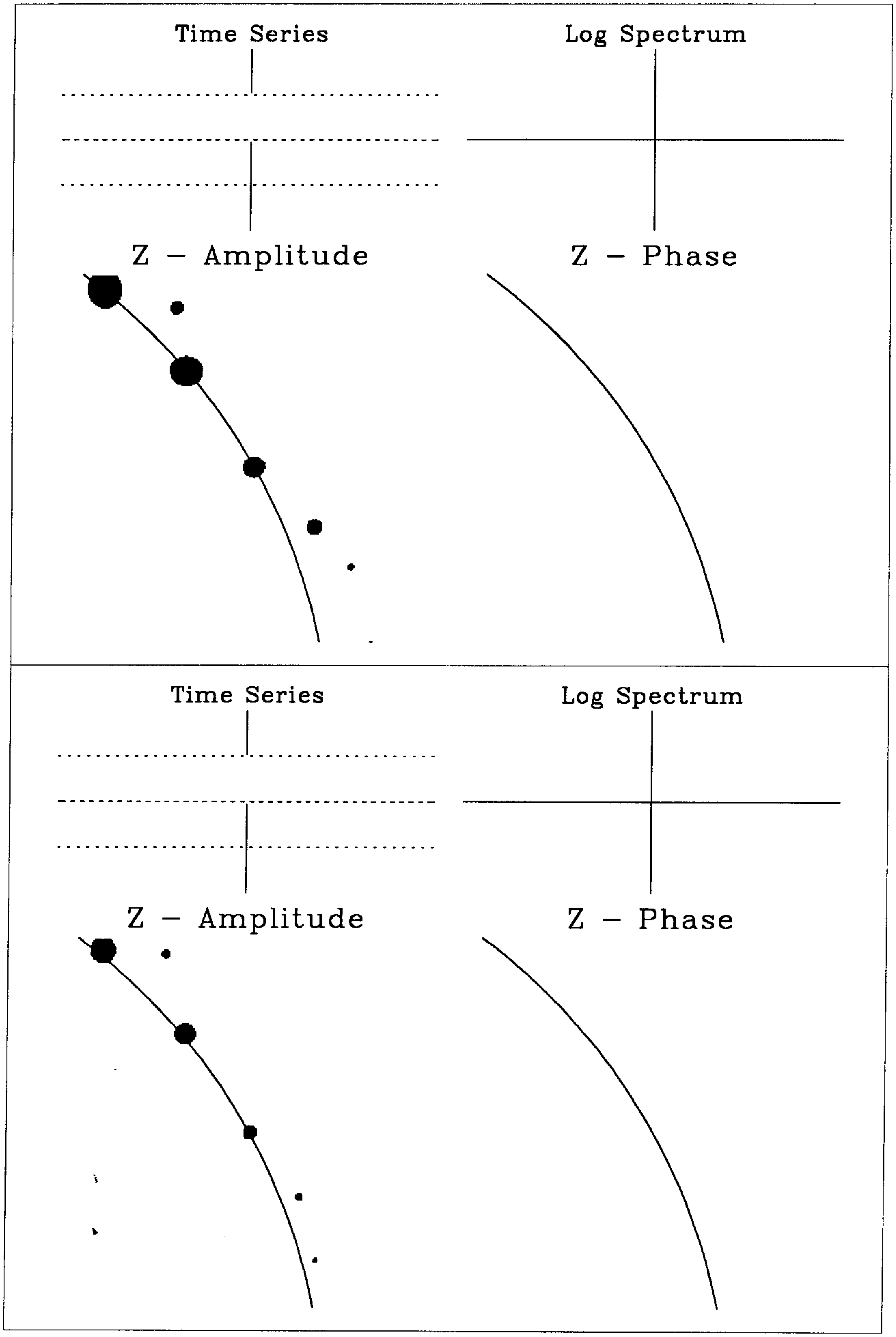
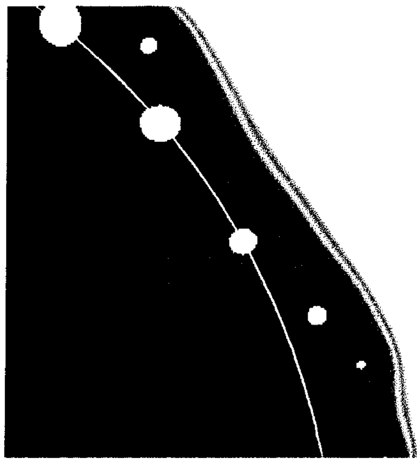
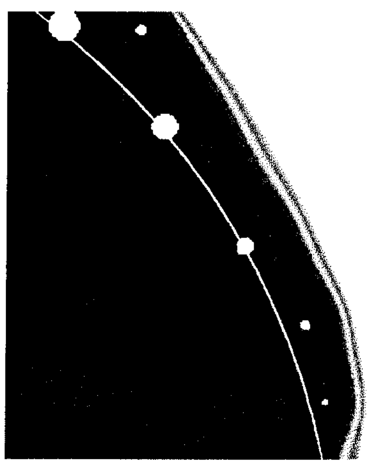
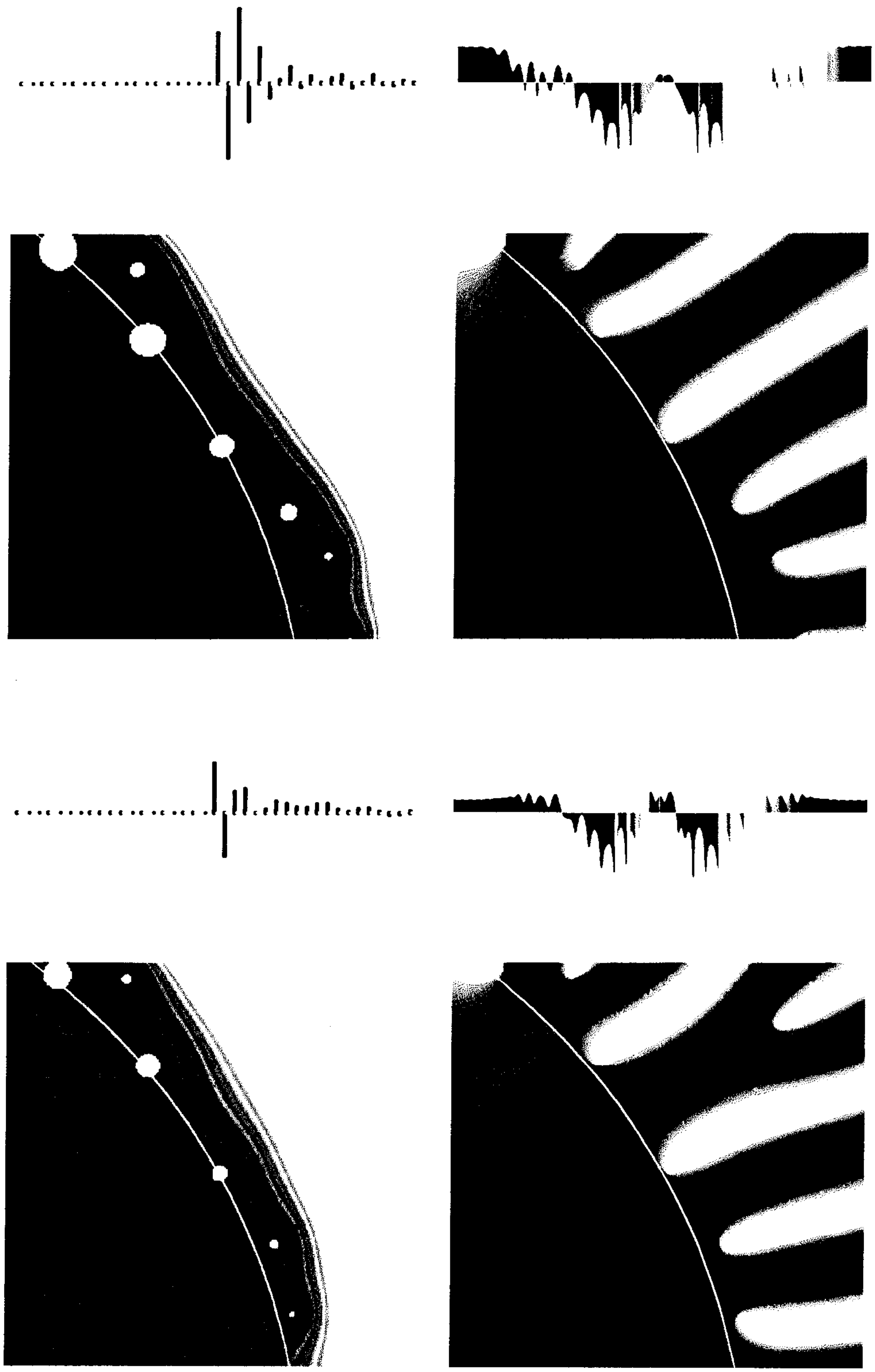


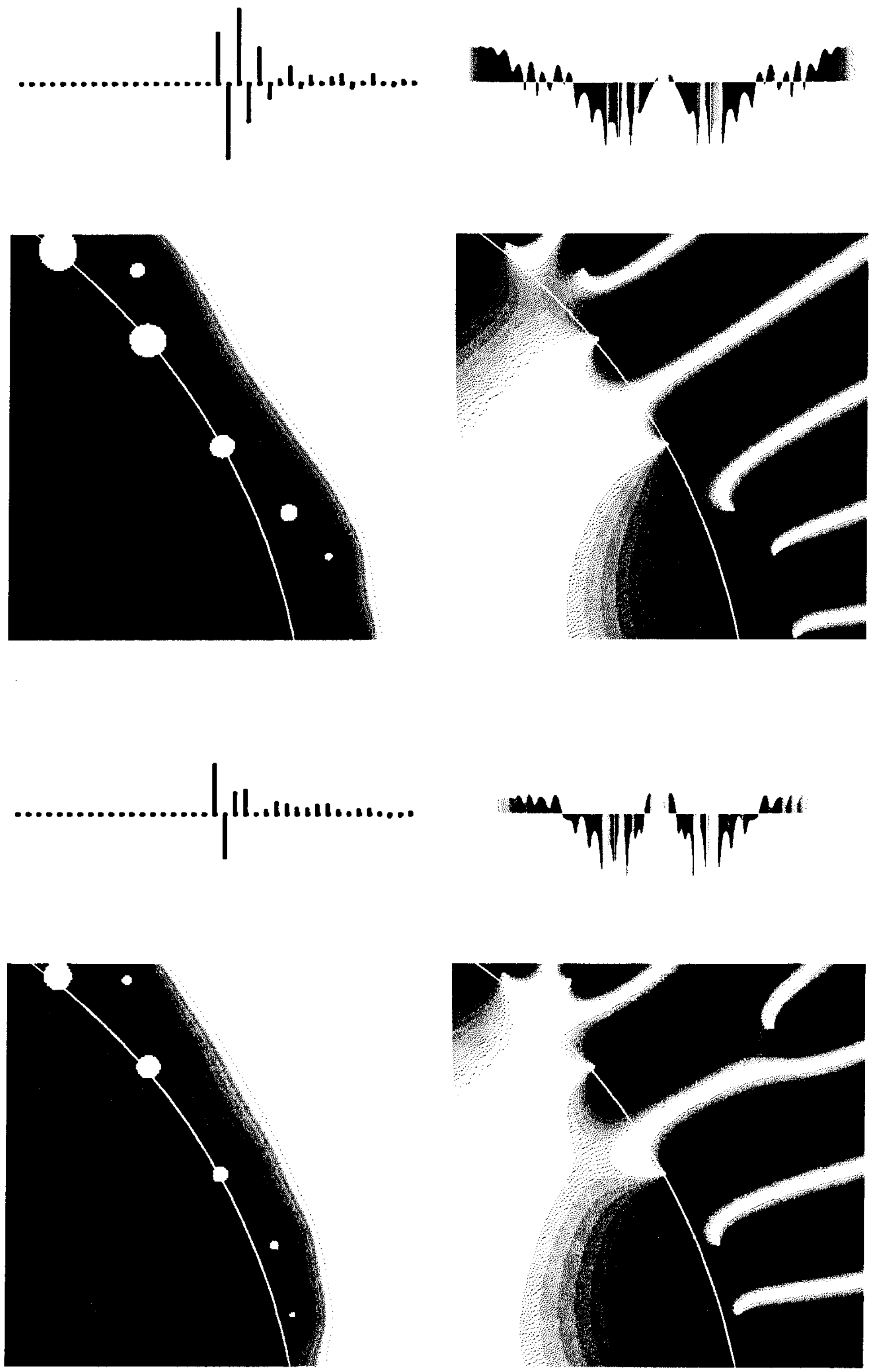
FIG. 9. Upper plot: Enlargment of the lower part of Figure 7; Lower plot: Enlargment of the lower part of Figure 8.







C



M