PSD AND SRS IN SIMPLE TERMS

Presented at ISTA Con '98, Orlando, FL
William I. Kipp
W. I. Kipp Company LLC
[The author was with Lansmont Corporation when this paper was written]

ABSTRACT

Power Spectral Density (PSD) and Shock Response Spectrum (SRS) analyses have been used in packaging laboratories for a number of years, but often with limited understanding of the underlying concepts. This paper will attempt to explain those concepts in terms which make sense to packaging professionals, with discussions of proper and useful application.

PRETTY SIMPLE DESCRIPTION (PSD)

Power Spectral Density analysis is a frequency domain representation of random vibration data. Whoa! — this is supposed to be a *simple* description, so we'll have to back up and discuss "random vibration", and define "the frequency domain". Along the way, we'll also have to talk about "vibration", "sinusoidal vibration", and the "time domain". But it'll be *simple*, I promise…

VIBRATION

Vibration is defined as "oscillatory motion about a reference position". For example, the bed of a truck which is driving down a level stretch of highway may move up and down, but there's no *net* change in vertical position. So it's "oscillating" (moving up and down), about a "reference position" (its static distance above the level road). Transport vibrations are generally considered to be present for relatively long periods of time (whenever the transport vehicles are in motion), and typically cause fatigue-type damage, scuffing of labels, loosening of closures and fasteners, etc.

Vibration is described by its *amplitude*, *frequency*, and *type* (either sine or random). Transport vibrations are usually less than 1G average, 10G maximum peak amplitudes. Frequencies of interest are usually up to 100-300 Hz. at most.

SINUSOIDAL VIBRATION

The simplest type of vibration is "sine" or "sinusoidal", so-called because of its shape: it looks like a graph of the value of the trigonometric sine function plotted as the angle changes. Figure 1 shows sine vibration data: acceleration amplitude in G on the vertical axis, time on the horizontal axis.

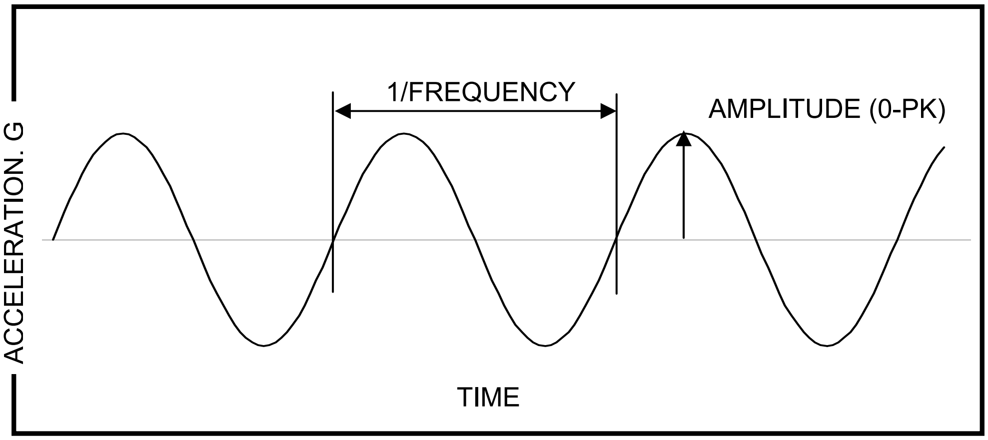

Figure 1: Sinusoidal Vibration

Notice the predictability of this vibration — it's a series of *cycles* (start at zero, up, down through zero, up, back to zero), each having the same height and taking the same amount of time. The amplitude in G (from the zero line to the peak of the wave) is shown. The time for one cycle (seconds per cycle) may be inverted to get frequency (cycles per second, called Hz.). If we are able to see just *one* cycle (or even part of a cycle) and we know that the waveform is a steady sinusoid, we can accurately predict the *next* cycles.

THE TIME DOMAIN

Figure 1, by the way, is a *time domain* plot. That simply means that the information is plotted with *time* on the horizontal axis. For our purposes, this is sufficient definition.

RANDOM VIBRATION

Random vibration, plotted in the same way (time domain) is shown in Figure 2.

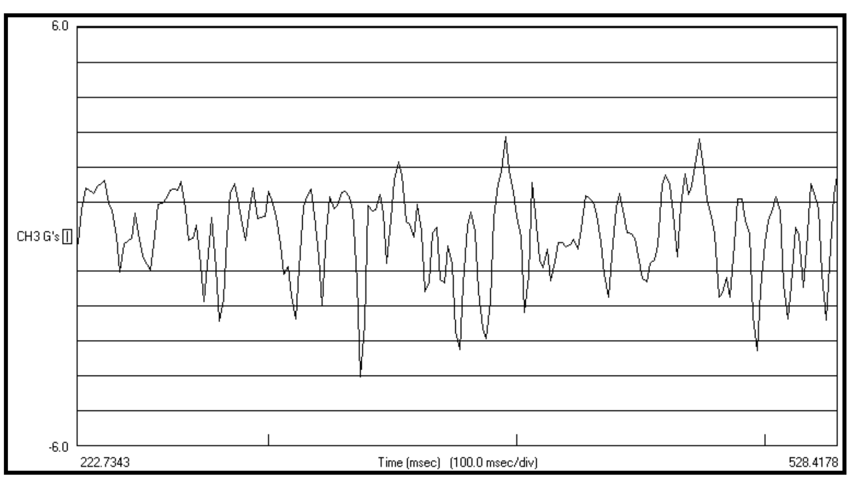

Figure 2. Random Vibration

This is much harder to describe than sine. A single arrow can't be drawn to indicate the amplitude, because it's always changing. A single time per cycle can't be defined, because it's always changing. Also, notice that there is *no* predictability to this vibration — examining a small portion of it gives us little or no clue as to what will happen next.

If random vibration is difficult to describe and understand, why do we even want to fool around with it? Because it's the type of motion that *occurs in the real world*. Trucks, railcars, etc. don't vibrate in steady-state sinusoidal fashion, it's random. So if we're going to make meaningful field measurements, and design realistic laboratory simulations, we're going to have to deal with random vibration.

A PSD PLOT, THE FREQUENCY DOMAIN

Random vibration *can* be described — by a PSD plot as shown in Figure 3. A PSD plot is a presentation of the "average intensity" of random vibration, expressed as a function of frequency. The fact that the graph has *frequency* on the horizontal axis makes it a *frequency domain* plot.

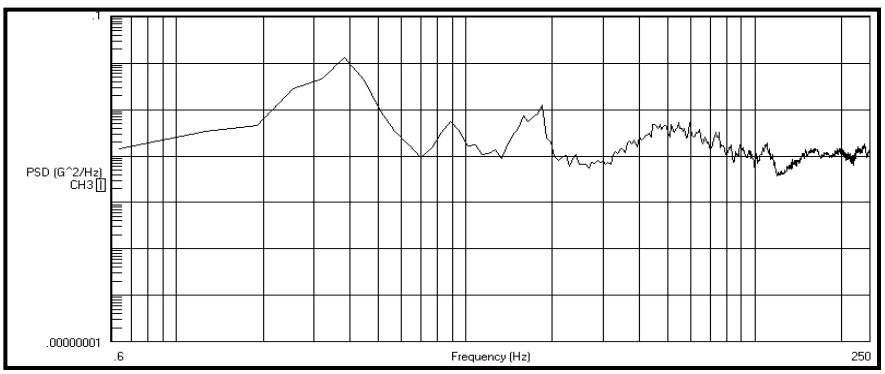

Figure 3. PSD Plot

Notice the peculiar units on the vertical axis — G^2/Hz. For simplicity, we're not going to explain those units here; it's sufficient for our purposes to think of the vertical scale as "average intensity" of the vibration (for those wanting greater detail, visit www.lansmont.com, select "Newsletters", and read the Random Vibration articles in the Feb. 1996, June 1996, and May 1997 issues).

HOW TO GET FROM THE TIME DOMAIN TO THE FREQUENCY DOMAIN

Random vibration "happens" in the time domain, so how do we get to the frequency domain of a PSD plot? The key is what's known as Fourier Theory, named after the French mathematician Jean Baptiste Joseph Fourier (1768-1830). In essence, Fourier theory says that any waveform, no matter how complex, can be analyzed as a combination of *sine waves*. The waveform can be decomposed into its sine wave components (analysis), or various sinusoids can be summed to create the complex wave (synthesis).

Proof of the Fourier Theory is mathematical, of course, and well beyond the scope of this paper. However, a simple graphical demonstration may be helpful.

Fourier Theory Demonstration

Figure 4 looks pretty "random", doesn't it? Actually, it's made from just three sine wave components. Each component has a different (but fixed, not changing) frequency, each component has a different phase (time relationship), and each component's amplitude is changing randomly. But they're all sine waves. When added together, they comprise the waveform of Figure 4.

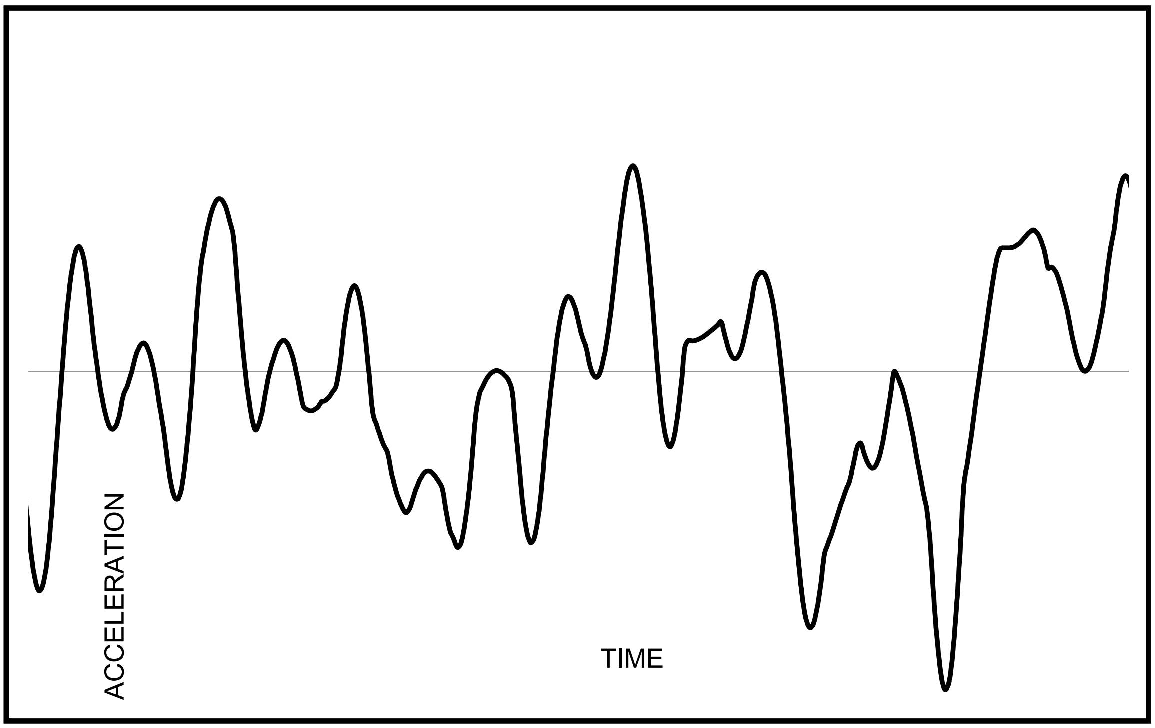

Figure 4: "Random" Waveform Synthesized From Three Sine Waves

Figure 5 on the next page shows how this is done. The sine wave components are shown as a, b, and c. The resultant from Figure 4 is shown again as d. Notice that the frequency of each wave is constant (i.e. the time for each cycle is always the same), but that the amplitudes are varying (randomly). To synthesize the result, the waves are summed on a point-by-point basis for each discrete time on the horizontal axis. The figures would get too crowded if we tried to show every time point, but we've shown several to give you the idea.

At each point in time, we sum the amplitudes from waves a, b, and c to get the resultant amplitude shown on d. The summing is done algebraically, of course — amplitudes above the zero line are added, amplitudes below the zero line are subtracted.

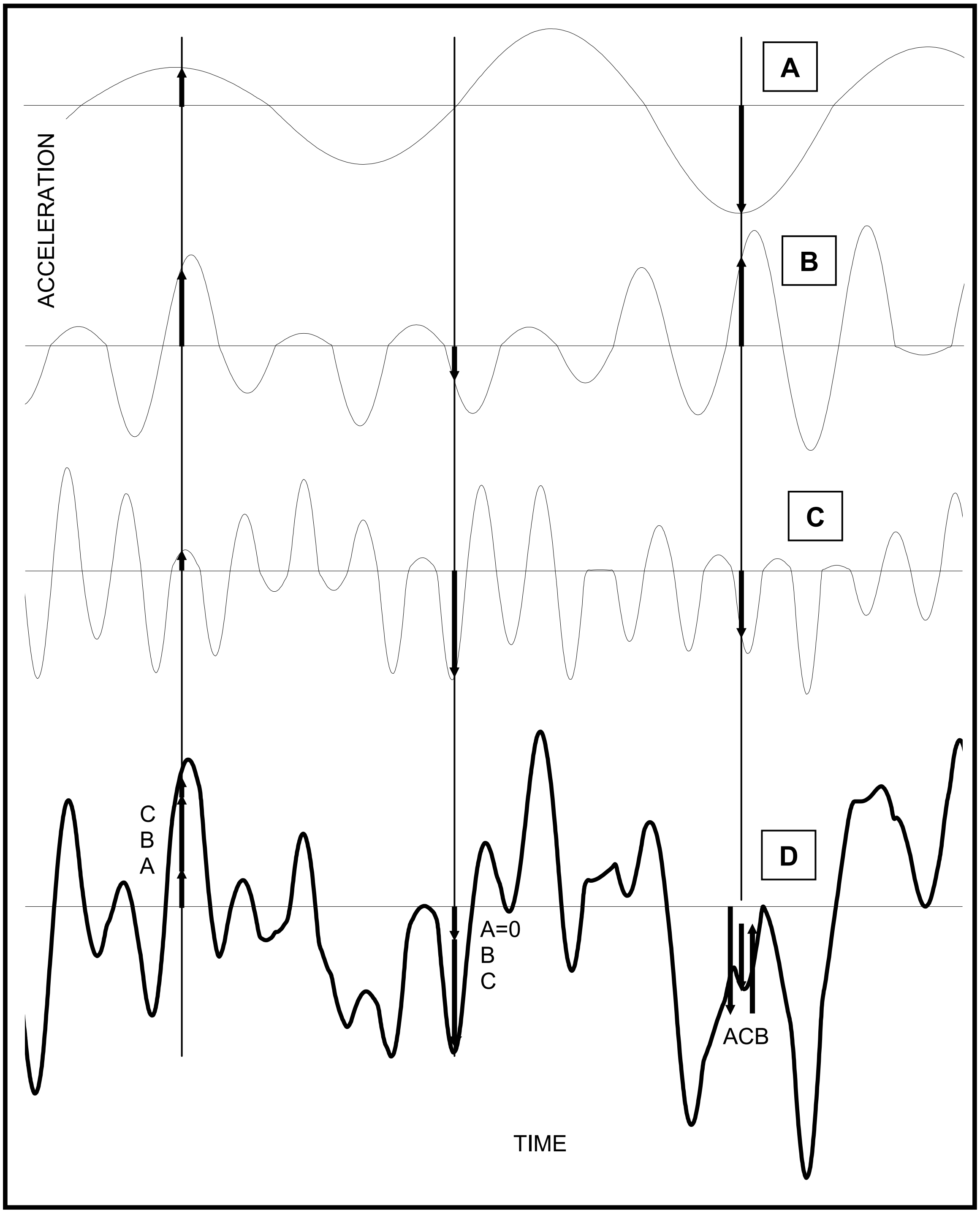

Figure 5: Details of “Random” Waveform Synthesized From 3 Sine Waves

Try more of this on your own if you like: draw a vertical time line on the above figure, measure the amplitudes of waves a, b, and c, add/subtract them as appropriate, and verify that you get the amplitude of wave d.

GETTING TO THE PSD PLOT

The fact that random vibration can be analyzed as the summation of a number of discrete frequency sine waves (you believe it now, don't you?) gives us the means to translate time domain information into the frequency domain. This is done by looking at the time-domain random signal through a bank of *filters*, each of which pass only a single frequency. The output of each filter looks like Figure 5a, 5b, or 5c — a single-frequency sine wave with a randomly varying amplitude. This is shown schematically in Figure 6. Only three filters are shown, but there are typically hundreds (indicated by the dots) — not actual hardware filters, but the equivalent mathematical calculations performed by a computer, each with a different sine wave output.

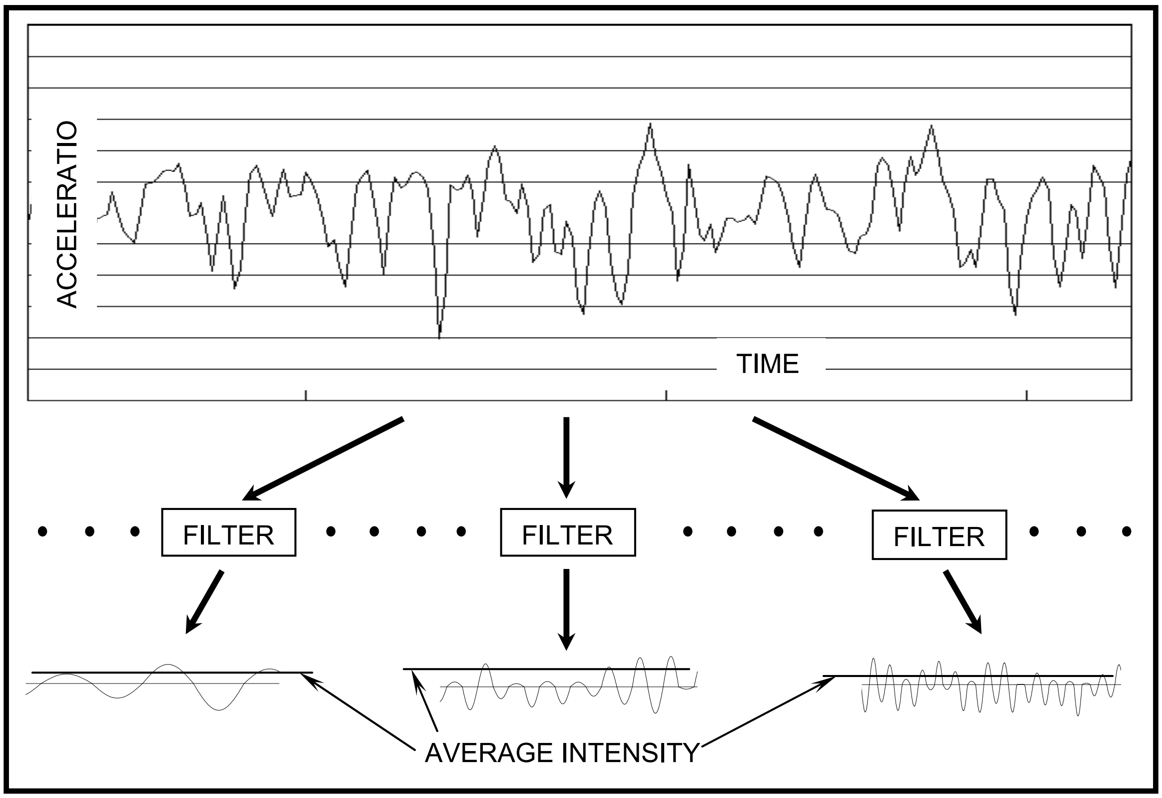

Figure 6: Discrete-Frequency Filters Outputting Sine Components

After the sine components have been extracted, the next step is to determine the time-average amplitude of each. This is easily done by the same computer which implements the filters. Actually, what's determined is a special kind of "average" called "root-mean-square" (rms). But for purposes of our explanation, let's just call it "average intensity".

So now, for the output of each filter, we have two numbers: a frequency (the filter's frequency) and an average intensity. Hey — didn't we say earlier that a PSD plot presents the "average intensity" of random vibration, expressed as a function of frequency? We're almost there…

Figure 7 shows how the two numbers, frequency and intensity, are plotted in the frequency domain. When all the points are placed, from the hundreds of filters, and all the dots are connected, the result is a PSD plot. You can identify frequencies at which the vibration is more intense by looking at peaks in the PSD plot (something you can't do in the time domain). Simple, isn't it?

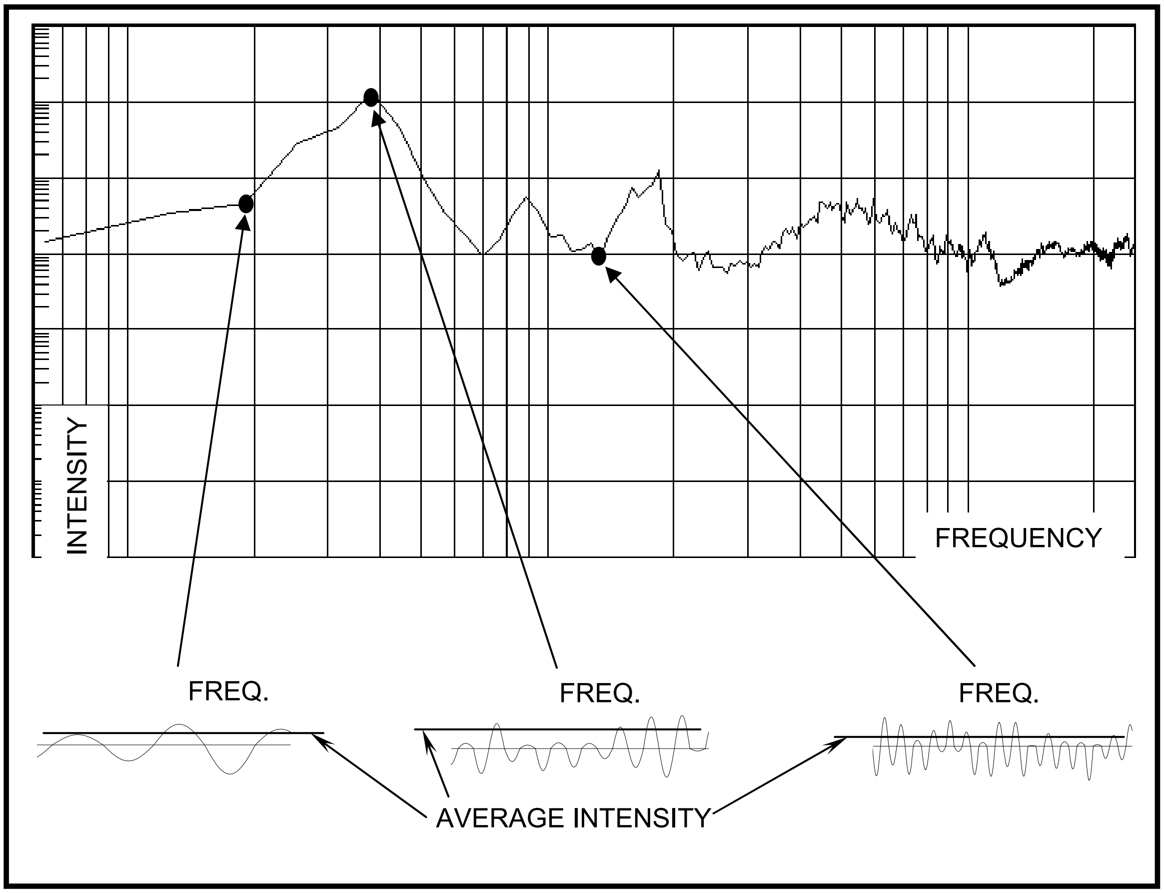

Figure 7: Construction of the PSD Plot

TELL ME AGAIN WHY WE WENT TO ALL THIS TROUBLE?

OK, maybe PSD *isn't* that tough to understand, you say. And you can see why random vibration is important, since it's what actually happens in the real world. But why not just take the time-domain random from field recordings, and reproduce *that* in the laboratory? Wouldn't that be the best simulation of all? Why bother with PSD?

"Real-time" testing, as it's called, is practiced in some industries. In terms of simulating the trip or section that was recorded, it obviously is the best approach. But for packaging/product-design/logistics/distribution purposes, we really don't want to reproduce only *one* trip — we want to simulate *all* trips (within limits). So we want to *compile* data from multiple recordings into a test that has the greatest possible statistical significance.

There's just no practical way to compile random vibration in the time domain. That could be the subject of a whole separate paper, and you'll have to take my word for it at this point — but there's just no practical way. In the frequency domain, however, with PSDs, it's a piece of cake. Since the intensities at the various frequencies are "averages" anyway, it's permissible and proper to "average averages" across multiple trips and recordings. The more data you have and compile, the higher your confidence in the resultant test. And it's easy to add in safety factors, calculate weighted averages, perform accelerated tests — all the things you need to do to address your vibration-related issues.

In addition, "real time" testing is very costly. The equipment and systems to implement it are expensive, and it takes considerable time and effort to prepare each test. So it's not used for our applications. Random vibration with PSD control *is*, and we hope this discussion has helped your understanding.

SIMPLE, REALLY SIMPLE (SRS)

Shock Response Spectrum analysis is intended to provide an estimate of the *response* of an item to input *shock*, with the information presented as a frequency *spectrum*. If that definition doesn't do much for you, then stay tuned — this section will attempt to explain SRS in *simple* terms. Along the way, we'll have to learn about shock, how to describe shock pulses, about spring-mass systems, natural frequencies, and damping. But it'll be *simple*, I promise…

SHOCK

Shock is defined as "a mechanical disturbance characterized by a rise and decay of acceleration in a short period of time. A transient event." For example, when a package drops and hits the floor, a "shock pulse" is produced. If we were measuring acceleration on the product inside that package, we'd see a low acceleration prior to impact, higher accelerations during the impact, and a low acceleration again after it's all over. And the shock would occur quickly — bang! Severe shocks to products can result in immediate breakage, damage, or failure.

Shock pulses are characterized by their *peak acceleration*, *time duration*, and *waveshape*. Transportation/distribution shocks to products are generally in the range of 10-100G, with durations measured in milliseconds (1000^{ths} of seconds).

Typical shock pulses are shown in Figure 8 on the following page. The peak amplitudes and time durations are indicated. Regarding shapes, the top wave is called a "half sine", because its shape is roughly that of half of a sine wave (remember our discussion in the PSD section about "sine" and "sinusoidal"?); the bottom wave is called "square" or "rectangular" because of its steep sides and relatively flat top. Other waveshapes are given descriptive names if possible, but sometimes you have to just give up and say "…I'll fax you a copy…".

Notice that Figure 8 is a *time domain* plot, since it has *time* on the horizontal axis.

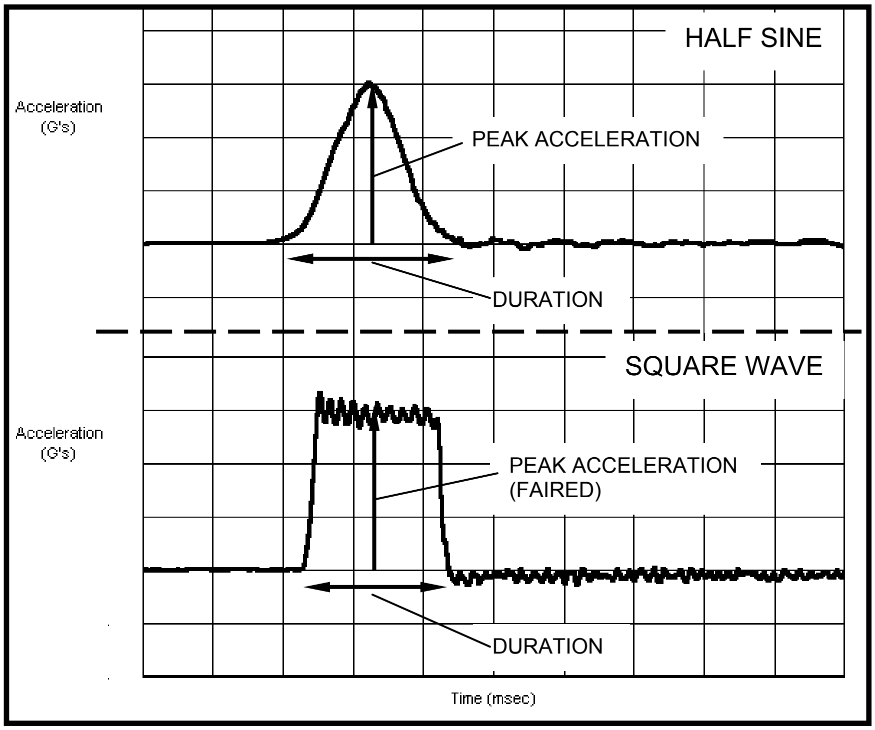

Figure 8: Typical Half Sine and Square (Rectangular) Wave Shock Pulses

SRS PLOT

Figure 9 shows an SRS plot for the half sine shock pulse of Figure 8. Notice that the vertical axis is "Acceleration", the same as Figure 8 — so that's easy. The horizontal axis is frequency in Hz., which makes this a ***frequency domain*** plot (in the strictest engineering sense, this is not truly a frequency domain plot, but for our purposes the differences don't matter).

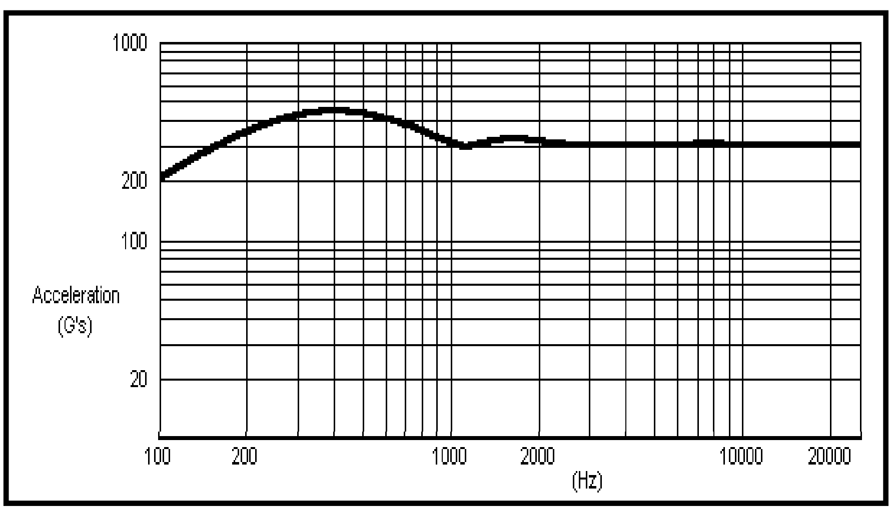

Figure 9: An SRS Plot

So the question becomes, as in the PSD section: how do we get from the time domain to the frequency domain. For PSD, the key was Fourier theory — for SRS, the key lies in the behavior of *spring-mass systems*.

SPRING-MASS SYSTEMS

The "R" in SRS stands for the *response* of "something" to the shock pulse. Of what? Of an idealized model called a *spring-mass system* (sometimes also called a "single degree of freedom" (SDOF) spring-mass system, to indicate that there's only *one* spring and *one* mass involved). A spring-mass system is shown in Figure 10.

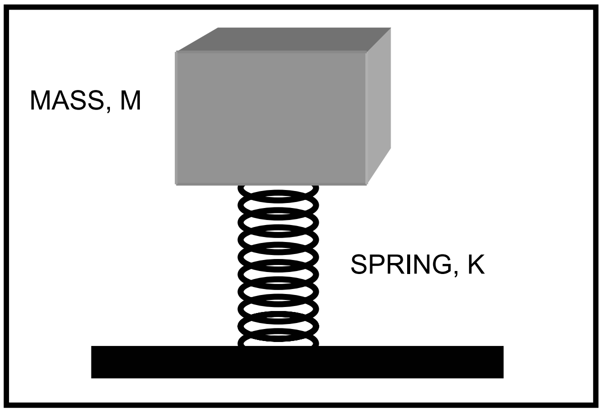

Figure 10: A Spring-Mass System

Every spring-mass system has an associated "natural frequency". This is the frequency at which the mass bounces up and down after the system is disturbed. The natural frequency depends on the weight of the mass and the stiffness of the spring — heavy masses and soft springs result in low natural frequencies, light masses and stiff springs result in high natural frequencies.

Let's say we disturb the spring-mass system by putting a shock pulse into the base of Figure 10, and measure the response of the mass. If the spring is very soft, and/or the mass is very heavy, the response will be very small (visualize that if the spring is so soft as to be essentially *not even there*, the response of the mass would be *zero*). If the spring is very stiff, and/or if the mass is very light, the response will be the same as the input (visualize that if the spring was as stiff as a solid bar of steel, the mass would follow the movement of the base). In between these extremes, the mass will respond to varying degrees. Different-shaped shock pulses will, of course, create different responses.

Take a look at Figure 11 on the next page. Here we have three different spring-mass systems (a, b, and c) all disturbed by the same half sine shock pulse. The first system (a), with a very low natural frequency, hardly responds at all. The second system (b), with a middle-of-the-road natural frequency, responds to a higher acceleration level than the input pulse! And the third system (c), with a high natural frequency, follows the input shock pulse quite closely.

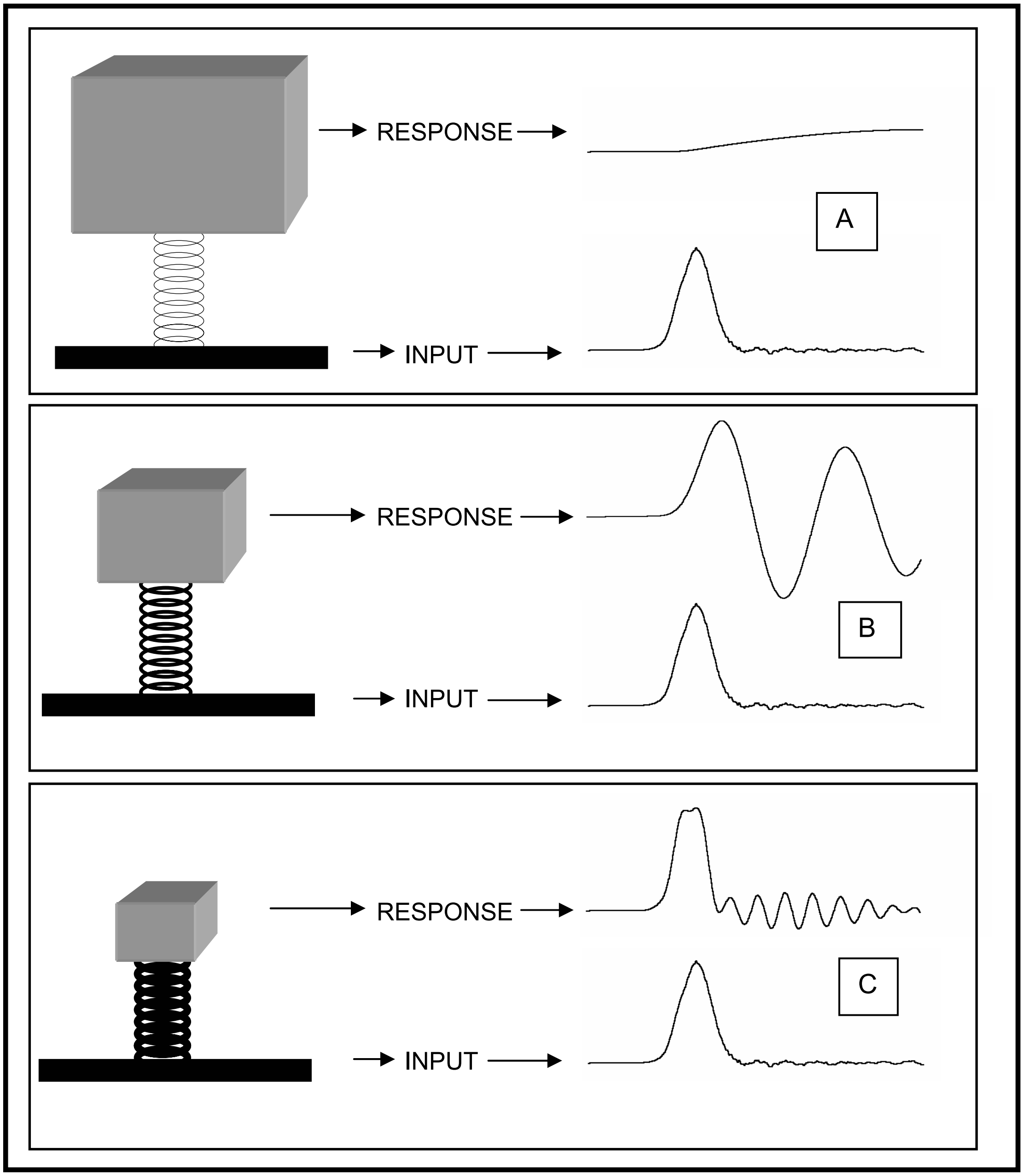

Figure 11: Responses of Spring-Mass Systems
(With Different Natural Frequencies) to the Same Input Shock Pulse

GETTING TO THE SRS PLOT

For each spring-mass system, we know (can calculate) its natural frequency, and we can also determine the *maximum* amplitude of its response. This gives us a frequency number, and an acceleration (G) value. Hey — isn't that just what we need to construct an SRS plot? Bingo…

Figure 12 shows how the two numbers, frequency and maximum response G's, are plotted in the frequency domain. When all the points are placed, and all the points are connected, the result is an SRS plot. Simple, isn't it?

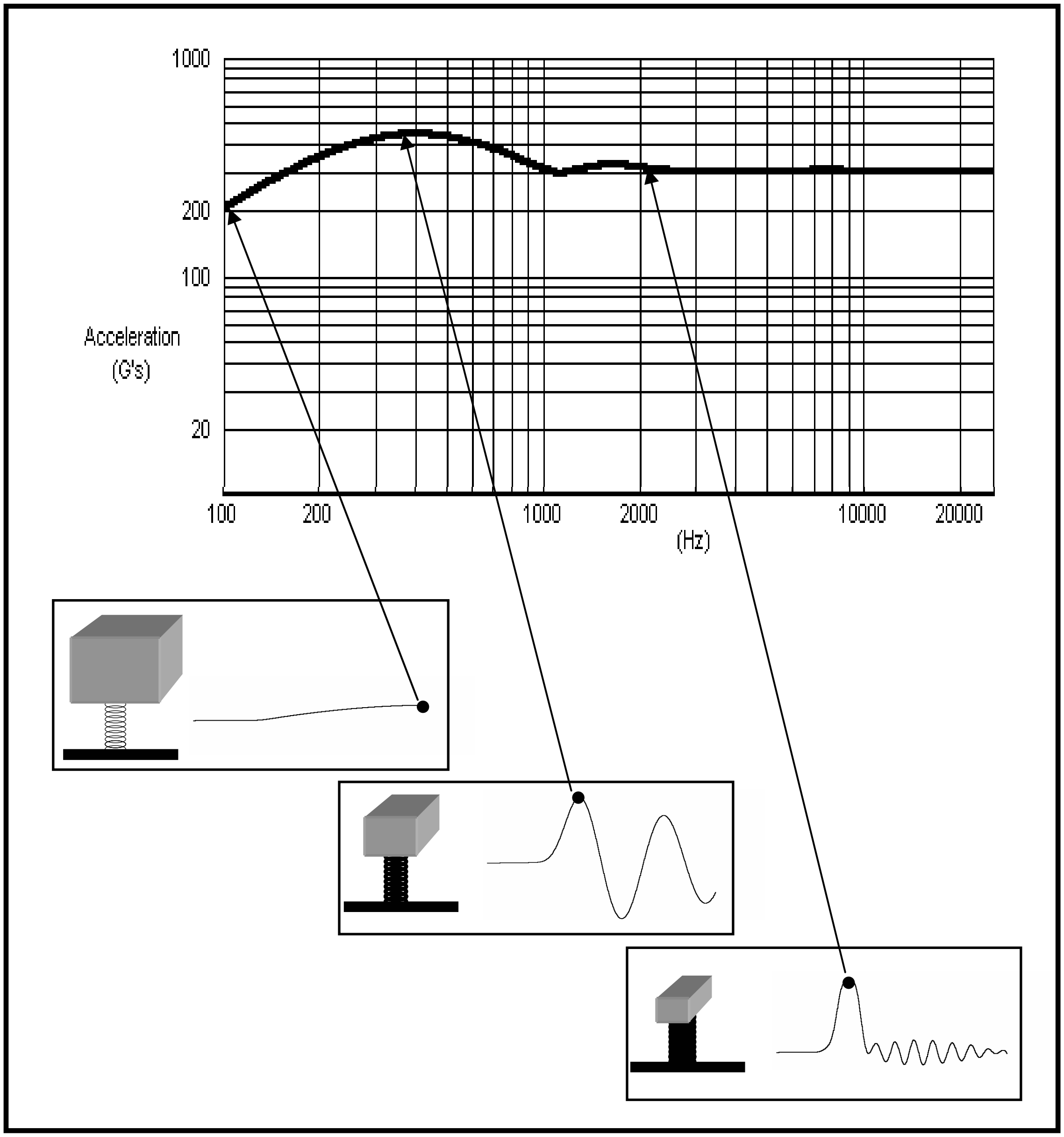

Figure 12: Construction of the SRS Plot

WHY DO WE CARE ABOUT SPRING-MASS SYSTEMS?

It turns out that spring-mass systems are a good way to think about real products and packages. At least in a first-order-approximation kind of way. An electronic instrument sitting on a cushion in its package can be thought of as a spring-mass system. A transformer sitting on a circuit board inside that instrument can be thought of as a spring-mass system. The basic principles apply, and allow us to visualize even fairly complex situations in relatively simple terms.

That's the good news. Beware, however — SRS doesn't necessarily predict the performance of *your* product or package. Remember, SRS calculates the response of a number of idealized spring-mass systems. Period.

SOME WORDS ABOUT DAMPING

Spring-mass models without damping would, once disturbed, oscillate (bounce) forever. Perpetual motion hasn't been invented yet — and all spring-mass systems include *damping*, which eventually brings the disturbed system back to rest. Low damping means the system will respond at greater G-levels, and oscillate for many cycles. High damping reduces the peak response, and stops the oscillations sooner. SRS analyses are usually specified with rather light damping, to be conservative.

TYPES OF SRS

SRS analysis is sometimes subdivided into three types, always categorized by *when* the maximum response of the spring-mass system occurred. The *primary* SRS only considers the time *during* the input shock pulse, and looks for the maximum response during that time. Similarly, the *residual* SRS only looks for the maximum response which occurs *after the shock pulse is over*. The *composite* (often called "maximax") SRS looks for the maximum *anytime*. Obviously, most people are interested in the composite, since splitting hairs on exactly *when* the maximum was reached seems pointless much of the time.

THE USEFULNESS OF SRS

SRS is another tool for analyzing shock. Viewing a shock pulse in the time domain gives you a good picture of the peak acceleration, duration, and shape. But it's difficult to determine the predominant frequencies, and of course gives no prediction of how an item might respond to the pulse. This is where SRS shines.

SRS is also being used to accurately reduce the safety factor built into Damage Boundary testing. A discussion of Damage Boundary theory is beyond the scope of this paper, but the reader is referred to ASTM D3332 for information and test procedures. During testing, when the damage point is determined, an SRS of the damaging pulse is calculated. This then becomes the "target", not to be exceeded, when designing and assessing package performance. If the natural frequency of the damaged component is known, the SRS of the shock pulse transmitted by the cushion need only be below the "target" SRS in that critical frequency area.