Mid-West Spring & Stamping Finds Innovative Ways to use Technology in Designing Nested Springs

Mid-West Spring & Stamping takes the guesswork out of designing nested springs by using new technology to generate hundreds of comparative iterations and save time in the process.

Romeoville, Illinois (PRWEB) February 14, 2009 -- For years compression spring companies have had the task of designing springs that fit inside or over each other. Nested compression springs are nothing new. Designing them, on the other hand, can be a bit time consuming unless you have the tools (i.e.) software. Fortunately, these are easy to come by nowadays.

Programming languages have become more flexible and user friendly. Microsoft's Visual Basic, which made its debut in 1992, has evolved into an intense graphic powerhouse allowing all kinds of entry formats, buttons, boxes, sliders and gages as well as audiovisual capacity. For non-programmers there's Excel. While Excel lacks the bells and whistles it does provide a powerful way to create and adapt math models, which are the very structure of compression spring design.

Compression spring calculations, being linear for the most part, make it easy to set a condition and then compare them to one another. What do I mean by that, you ask?

Let's take the case of a valve spring- an outer/inner situation. Most assumption or guesswork comes from not knowing what wire size to use for the first spring- the outer. The very first step is to define what we need, beginning with what we know. The following list should be typical:

1. Material type (defines modulus)
2. First load at the first height
3. Second load at the second height (defines total rate).
4. Percent of the load for the outer spring to carry (defines outer spring rate).
5. Stress limit
6. The outside diameter (O.D.)
7. The maximum solid height (defines total coils limit, based on whether the ends are ground or ungrounded)

It should be obvious at this point that we have a lot of initial information. How do we use it? Let's start by using an example and defining each variable with a typical value:

1. Material: Chrome Silicone Valve per ASTM A877
2. First load at the first height: 450 lb. At 3.500 in.
4. Percent of load for the outer spring to carry: 65%
5. 140,000 psi stress limit
6. O.D.: 2,200 in
7. Maximum solid height: 2,400 in

At this point it's important to mention that we would need to have a table of the materials for the torsional modulus of that material. That, however, is a simple list with very little programming time invested.

Now, we need one more parameter before we start calculating: the wire size. I am aware that we don't know
what it is. So, we will work backwards. Not knowing what the wire size should be we put in a ridiculously large size just to have a place to start and let the computer do a ton of work in a second. If manual calculation were the method of choice this idea would not be so user-friendly. But with the computer compression spring design is a cinch. We're going to do a calculation, then have the program remove a wire size increment until we meet certain conditions. We would also need to view all this calculated output (text boxes, cells or whatever format you used to see your resulting parameters). Ready? Here we go:


At this point if you're not wondering what gibberish this is then you're not with the program. A compression spring engineer looks at the resulting data and knows immediately that something is wrong. Actually, we're right on track! What we've done is give the program a wire size. Now the program calculates all the needed parameters based on that wire size. Here's an analogy: think of a tree trunk (all possible wire sizes) being whittled into a toothpick (the desired wire size) one stroke at a time using a chainsaw (the program) that can do the job in a few thousand nanoseconds. Now we are ready to whittle. First we have to state the condition. That condition is the calculated corrected stress. We know what stress level we want to reach - 140,000 psi.

Computers are extremely beneficial for nested compression spring design because they possess an enormous efficiency to compare conditions. Programmers know how to use a DO/LOOPWHILE loop to make this happen. However, for the rest of us, here's what happens. Follow these instructions:
1. Remove 0.001 in. (or any increment you choose) from the current wire size.
2. Do all those calculations again!
3. Is the calculated stress LESS than the stress level I want?
   a. If the calculated stress is LESS, then go back to step 1 and do again.
   b. If the calculated stress is MORE, than go to step 4.
4. Show all data, including the resulting wire size, because we're done with this design.

If you were to actually do the example, the resulting wire size would be 0.296 in. because, at that size, the stress will be just over 140,000 psi. True or not? Let's test it.


This yields a spring with very good parameters, as far as compression spring designs go. You could experiment by raising the stress level and do some "what ifs". Most importantly you get a compression spring design instantly. Now that you have the wire size you can check to see if you have a stock size close to the calculated size and do a detailed compression spring design with all the data you need.

Here's the absolute best part: once you get the outside compression spring nailed you can have another design using the same calculations for the second spring…or a third…or a fourth. You get the idea!

With nested compression springs you can define the interference limit. For instance, if you need to be sure on this particular design that there is a .050 in. space between the two compression springs, then subtract that amount from the outside spring I.D., and let the program instantly calculate your next spring. Or, if you design valve springs, which often must have interference so the inner spring is actually a bit bigger, you can define a negative interference, and the program will add that amount to the outer spring I.D., and the rest is history.

"Iterative" simply means the program goes back and recalculates as many times as needed to get a result. The
"logic" part is merely the path that it takes, one parameter depending on the next for a result. If the calculations aren't in the proper logical order, you're pretty much stopped cold. The many variations of this type of program is beyond this article. However, the concept is completely open to creative changes. For example, it would be an easy task for a programmer to include a check that stops the program when the maximum solid height is reached…or feedback a myriad of information from the already calculated parameters. For valve spring designers, natural frequency would be nice to have. On the other hand, what about having the computer search out only stock wire sizes and skip the "close to" result?

The bottom line is that computers eat math for lunch, which is excellent for nested compression spring design and a perfect example how they can be put to work when hundreds, or even thousands, of comparative iterations are required. You can get a compression spring design result in seconds and remove the guesswork!

Randy DeFord is engineering manager at Mid-West Spring and Stamping in Mentone, IN. Readers may contact him by phone at (574) 353-7611 ext 231.

###
Contact Information
Randy DeFord
Mid-West Spring and Stamping
http://www.mwspring.com/
1-800-424-0244

Randy DeFord
Mid-West Spring and Stamping
574-353-7611 ext 231

Online Web 2.0 Version
You can read the online version of this press release here.