

System Complexity

Some people get bogged down in the complex topic of simplicity. Just what is simplicity, and what is complexity. One would quickly say things that are easy to understand are simple, and things that are difficult to understand are complex. Good enough? Maybe not...

One thing to keep in mind is that what is simple for one person, may be complex to another - and vice versa, of course. Also, the addition of options, features, versatility, and expansion capabilities, naturally breeds a certain amount of complexity. And, efforts to keep the operation of anything simple, usually limits the item's options, features, versatility, and expansion capabilities. This is generally true for everything, but let's get specific about DCC.

If you have a system that provides only 14 speed steps, the system doesn't need to have provisions for any other speed step options. Therefore, users do not have to learn the intricacies of anything else. But if you add 28 speed steps, for example, to what the system can do, there is more to learn. With no options, there's nothing to learn because it can't be changed that's just the way it is. But with options, the user has to learn how to switch from one to the other if he wants to change it from the default.

Now, add a User Loadable Speed Table that just happens to also be 28 speed steps, then add a 128 speed step mode, and top it off with the capability of mixing 128 speed step commands with the 28 speed step User-Loadable table. By now you should be getting the idea that it is the addition of options that causes this complexity, not that the other feature is more advanced.

For example, if a company built a system and decoders that did only 128 speed steps, it would be just as simple to use as the one that did only 14 speed steps. But, that isn't possible if the company wants to market the system as DCC compatible at the outset, the NMRA set up 14 speed steps to be required, even though it could do higher speed steps.

However, sometimes simplicity itself can breed complexity, and can even cost more. For example, the original Lenz system was said to be a simple to learn system. It did only 14 speed steps, and was limited to MU consisting of two locos. But, in the real world of American railroading, it's quite common to have MU consists much greater than that.

On my own railroad, I have a long (about 50-foot) 2% grade that I like to pull long freights up.

To do this with my original Lenz system, I had to MU two consists of two locos together, and then control them independently. It had a button to toggle between two locos (or consists), so that allowed me to use four locos to pull the grade, which allowed up to about 60 freight cars. To do more than that would have required additional programming tracks around the layout to simply program addresses alike. But, when entering the programming mode, you have to shut the layout down. The way out of this is to purchase a second command station to do programming while the layout continues to run. And, even after all that, you still can't run consists any way but elephant style (head to tail) unless you did more programming or wired one or more locos backwards. And you still can't do MUing anywhere on the layout only where you had the foresight to install a programming track.. Read more about it in [Consisting](#).

With any Digitrax system, except for their original Challenger, I could pull an 86 car freight up the 2% grade with four locos on the lead, and two locos midway.

For me, it's much easier to learn how to use the MU consisting options of a more versatile system, than it is to figure out how to get around the limitations of a "simple" system. Further, with the "simple" system, a lot of extra wiring has to be done to accommodate the work arounds, and additional equipment purchased to make it work without having to shut the system down.

One more thought: Most people who complain about complexity are generally trying to use a feature that isn't even available on less complex systems. If they were to use only the features found on "simple" systems, ignoring the additional features, they would find that even the so called complex system is just as easy to use as the simple system. Yet, they'd have the additional features to "grow" into without the frustration of changing systems mid-stream.

For example, if you limit your use of the Chief or Zephyr to 128 speed steps, you don't have the complexity of using the User Loadable 28 speed step speed table or any other options. And, you still get the 128 speed step advantage, the reduced complexity of unlimited MUing, a programming track that can be used while the layout continues to run, and all the other features that you care to learn how to use.

With the Chief or Zephyr, you have the capability of programming with any method you want paged mode, physical register, or direct CV access. You can even program locos anywhere on the layout with OPS Mode programming. This sounds complex. And, it is if you want to be fluent in every one of these programming methods. But, you don't have to use or learn all of them. Learn whichever programming method is needed to program the decoders you use, and ignore the rest. And remember, you don't HAVE to use every feature and mode that is made available just because it's there. But if your system doesn't offer alternative ways of doing things, you absolutely HAVE to use whatever method is provided - whether you understand it or not. Personally, I think Physical Register programming (the one most "simple" systems use) is horrendous when you get above CV #6, and should have been eliminated with the stone age.

So, if you opt for simplicity, you can still run into complexity trying to get around the limitations of the limited system. On the other hand, if you opt for a system with more options and capability, you don't have to use the options (you don't have to get complex) unless (until) you choose to.