

Speed Matching DCC Engines

By Bill Beranek



On my layout I like to run multi unit motive power. I've tried speed matching by running engines back and forth (side by side) on a 15 foot stretch of double track mainline, adjusting CV settings and hoping to get both engines to travel the prescribed distance staying within one car length of each other. Not a very sophisticated way to speed match.

During a recent tour of NCMRC member **John Parker's "BNSF Fall River Division"** layout, I was very impressed with how smooth **John's** multi unit consists ran. **Mike**, a crew member, explained to me how they programmed just three or four CV's to get the engines speed matched.

After trying **John's** system, with great success, I decided to write this article and describe, in detail, how I speed match on my layout. There is more than one way to speed match. Probably the most popular way is using the full range of speed steps built into the better decoders. This requires individually programming 26 to 28 separate CV's in each decoder. Others will program the Back-EMF CV, which will help at slow speeds. While these more sophisticated programming methods work well, I've found by using this relatively simple system you can get amazing results. The system is easy to implement and you'll need only modest DCC skills.

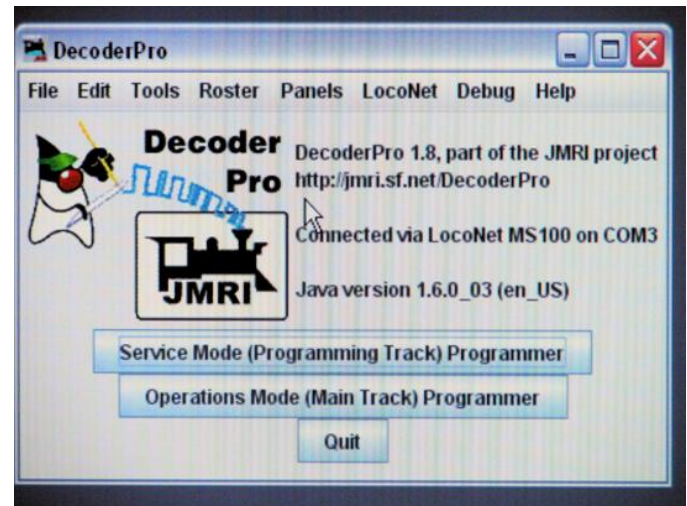
Requirements:

1. A DCC equipped layout or a long test track. My layout has a **Digitrax** DCC system.

2. Decoders that allow you to program CV2 (Vstart), CV6 (Vmid) and CV5 (Vhigh). I primarily use **Digitrax FX** decoders. You can use decoders from other manufacturers; just make sure CV5 & CV6 are programmable. Some manufacturers will let you program only two of the CV's (CV2 & CV5), while others only one (CV2). **WHY?** The **NMRA** has set "standards" for DCC Control Variables (CV's). Some are **NMRA** "Mandatory" (CV1, CV7, CV8 and CV29), while others are **NMRA** "Recommended" and still others are **NMRA** "Optional". Unfortunately, some manufacturers have decided that CV5 and CV6 are "Optional". After reading this article, I think you'll agree with me, **NMRA** made a mistake not making CV5 & CV6 "Mandatory".

3. DCC ready engine(s). The engine can have a factory installed decoder or you can install your own decoder, just make sure the decoder adheres to number 2 above.

4. A programmable DCC throttle or a computer interface program like **Decoder Pro**. I use a **Digitrax DT400** throttle to test my engines, but all CV settings are programmed through the computer using **Decoder Pro**.



(Decoder Pro opening screen)

5. A way to **accurately** measure scale speed. On the advice of **John Parker**, I purchased the "**Accutrack**" model railroad speedometer made by Model Railroad Technologies, Model #210-HO. It's a "speedometer tunnel" that works with both HO & N scales. It operates on two AAA batteries. It is infrared beam based. The digital readout displays in increments of .1 mph up to 10 mph and 1 mph above 10 mph. The unit's accuracy is derived from the 1 Megahertz clock of the microcontroller computer chip. The engine travel time between the two infrared sensors is measured and then used to calculate the scale miles per hour speed. You can purchase the "**Accutrack**" from Caboose Hobbies.

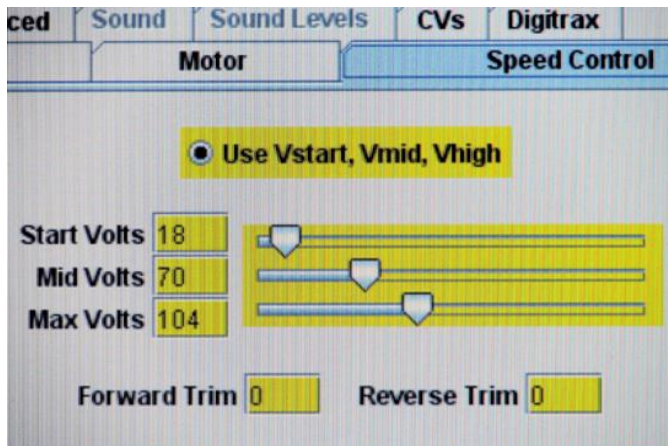


Programming:

First, check your DCC system, is it or can be set to **128** speed steps? You can program in **14** & **28** speed steps but **128** will give you even smoother operation.

I'm assuming your DCC system has some type of digital readout on the programmable throttle letting you know what speed step you are currently on. The **DT400** throttle reads out from **1** to **99**. On the **Digitrax** system the **128** speed steps are divided equally between step **1** and step **99**.

We'll be working with just three speed steps during the programming; on my **DT400** that's Step **1** (Start Volts), Step **49** (Mid volts) and Step **99** (Max Volts).



(Screen shot is from **Decoder Pro**)

The screen shot above shows the final settings for a **Proto 2000 GP38-2** with a **DH163LO (Digitrax)** decoder installed. At step **1** (Start Volts) **CV2** is set at **18**, at step **49** (Mid volts) **CV6** is set at **70** and at step **99** (Max Volts) **CV5** is set at **104**. At step **1** (Start volts) this engine averages **3.8** mph, at step **49** (Mid Volts) it averages **21** mph, and at step **99** (Max Volts) the average is **40** mph.

1. Prior to the start of programming, run the engine for approximately 10-15 minutes. A cold motor will run slightly slower until it warms up, giving you a false reading through the speed tunnel.

2. Run the engine through the speed tunnel at full throttle, readout **99** (Max volts) on the **DT400**. Write down the scale mph speed displayed on the tunnels digital readout. Do the same thing running the engine in reverse. In most cases you'll see a slight difference between forward and reverse speed. Reverse may be a little faster. Don't worry about that now.



3. Next, run the engine through the speed tunnel again, this time with the throttle set to **49** (Mid Volts). Write down the scale mph speed.

4. Lastly, turn the throttle to **1** (Start Volts), if the engine starts and runs at **1** (Start Volts) without stalling, congratulations! I've programmed dozens of engines and maybe one or two has started and ran at **1** (Start Volts). Don't worry if yours doesn't, we'll solve that below.

5. At this point, I'll assume you know how to program CV's using either a programmable throttle or **Decoder Pro**. Within the scope of this article, I can't cover how to do this for every DCC system. If you're not sure, it's time to read the manual.



6. I'll assume your engine did not start and stay running at step **1**. Start by programming **CV2** to **15**. Turn the throttle to **1** (Start Volts), does the engine move and stay running? If not, reprogram **CV2** to a higher number, say **25**. Continue to do this until the engine runs without stalling at **1** (Start Volts). Write down not only the scale mph but also the number you programmed into **CV2**.

The highest I've ever had to set **CV2** was **55**. If your engine still will not start and run at step **1** (Start Volts) try setting the "kick start" CV to **1**, **2** or **3**. Be careful not to set the "kick start" too high or your engine will jackrabbit when it does start moving. I've had to set the "kick start" on five of my engines and most of those are set at **1** or **2**. If your engine still will not move at step **1** (Start Volts), you'll probably need to do some maintenance. I've never had a decent running engine not start and keep running, once I found the right number.

7. Once you've got the engine moving at step **1** (Start Volts), the next thing to do is fine tune **CV2** so the engine will run at a constant slow speed. I try to have all my engines run at a constant **3.5** mph on step **1** (Start Volts). I allow a range of between **3.3** and **3.8**. *It doesn't*

have to be exactly 3.5 mph. I have gotten some of my engines to start and keep running at a scale 1 mph. An engine running at 1 mph might be fun to watch, for a short while, but for operations it's not necessary.

Fine tune by adjusting CV2's number up or down from the initial number. (*You did write it down, didn't you?*) When you finally get the engine to start and run on step 1 (Start Volts), at the scale mph you've decided on, write down the CV2 setting and the scale mph. Numbers programmed into CV2 on my decoders ranges from 3 to 55. All the engines start on step 1 (Start Volts) and run within the 3.3 to 3.8 mph range.

8. Now, repeat step 7 to program CV5 (top speed) and CV6 (mid speed). I usually program CV5 next. I've decided the top mph speed (on my layout) should be 40. My decoders have CV5 (top speed) set between 60 and 140. Why such a wide range? Differences in engine manufacturers, differences in decoders, how the engines are geared, all play a part in determining what the final programmed number will be. You will have to decide what scale speeds you want your engines to run at and adjust the CV numbers accordingly.



9. Prior to setting CV6 (mid speed) I add the numbers programmed in CV2 & CV5 together and divide by 2. *Example: CV2=42 & CV5=114 or $156/2=78$.* I then program 78 into CV6. Run the engine through the tunnel again and write down the speed. Adjust CV6 up or down until the engine will run on step 49 (Mid Volts) at a scale speed 22 mph (the scale speed I've decided on for my layout). Again, if it's 21 or 23 mph, close enough.

On some decoders a one number change can affect the scale speed by 2 or 3 mph. That's why I leave a little leeway in the scale mph settings.

Different decoder manufacturers have different limits as to how large of a number you can program into a particular CV. Example: one N scale decoder manufacturer will only let you program CV2 between 0-31. Any higher number, the CV won't recognize it. If

you're decoder is not responding to the numbers your programming into it, check and with the manufacturer and see what the upper limit numbers are for that decoder. With the **Digitrax FX** decoders, I have not experienced this kind of problem.

Consisting together:

After all this work, theoretically we should have engines programmed to run at the same scale mph through the entire range of speed steps. That said, "Can't we just consist of couple of engines together and go?" The quick answer is "Yes".

If I have engines that may get switched around with others, I stop at this point. With few exceptions, I can interchange most of my engines.

However, if I have engines that will never be separated, I take the programming one step further.

As mention earlier, very few engines will run exactly the same speed forward and backward. In most cases the engines will run slightly faster backwards. You can correct this by programming the CV's that control forward trim and reverse trim, assuming your decoder will let you program those CV's. On my layout, I haven't found it necessary to do this extra programming step.

I run my multi unit engines rear end to rear end, I don't run any units rear end to head end (both engines facing the same direction). With engines that will not be separated, I take the "trailing" engine (the one facing backwards) and reprogram it to run at the exact (or as close as I can get it) scale speed of the "lead" engine. I also set the "normal direction of travel" to reverse, this lets the headlight function properly. Once you get the speeds matched it's amazing how well they will run together.

Example: I placed a "trailing" engine on the end of a twelve car train, thus turning the "trailing" unit into a "helper". I was able to run the train up and down my 2.5% curved mainline grade (about 30 ft) without any problems.

Keeping track of the numbers:

When you have twenty five engines programmed, you need some way of keeping track of all those numbers. **Decoder Pro** does that for you, up to a point. **Decoder Pro** does not track the scale mph. I created a spreadsheet that tracks the same information stored in **Decoder Pro** plus the scale mph speeds. This way if I want to interchange engines I can see at a glance which ones would match up the best.

Decoder Pro does make programming the decoders a snap and if you ever have a problem with a decoder, **Decoder Pro** can easily to reset all of the CV numbers for you.

Engine Cards:

When you run multi unit engines, what number do you give the engines? I keep things very simple, using the last two digitals of the lead engine, or if that number has already taken, I use an arbitrary two digit number. I program both engines with the “same” number. When I dial up the consist number; I’m automatically picking up both engines at the same time. Remember when running rear end to rear end the trailing engine needs to have its “direction of travel” switched to reverse. In **Decoder Pro** it’s very easy to do.

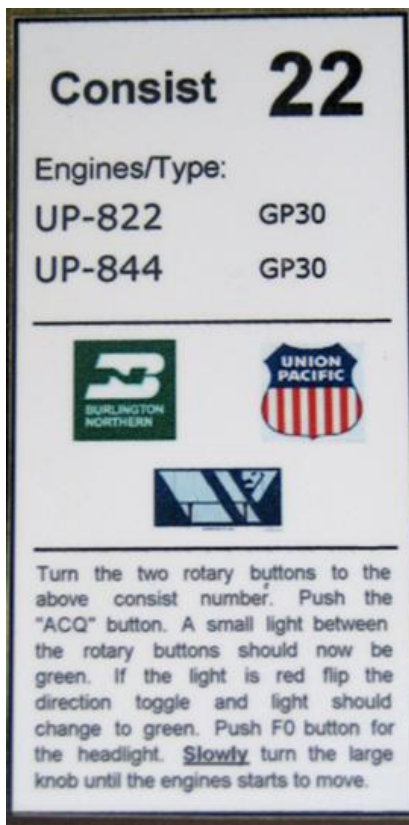
To keep track of what numbers go with what engines, I made up engine cards for each consist, listing the number for that consist (*the number you program into the throttle*) and the engines that make up the consist. On the engine card I also include instructions (for new operators) on how-to “acquire” consists. The engine card stays with the engines regardless of what train those units might be in charge of.

the numbers programmed into it. Yes, they will do that every now and then.

If you take time to study the spreadsheet, you will quickly realize that no two engines or decoders match up exactly the same. It becomes obvious that without the speed tunnels feedback, programming two engines into consists would be significantly harder and much more time consuming.

There may be better or other ways to consist engines together, but this system is working well for me and I think anyone interested in running multi unit engines should give it a try.

If after trying this system you have any questions or concerns, please let me know. If you come up with a better system, I would love to hear about it.



(Sample engine consist cards)

Some final thoughts:

I’d like to thank both **John** and **Mike** for taking the time to explain the system. By setting three CV’s you can get engines that would, under normal circumstances, never run well together, run like they were twins.

Included with this article is a PDF file of the spreadsheet I created to keep track of all my engines and the numbers programmed into their decoders. This is very useful if you have a decoder that decides to forget