Plugged In

E-Drum Workshop

by Norman Weinberg

Tips to Make Your Sound Move

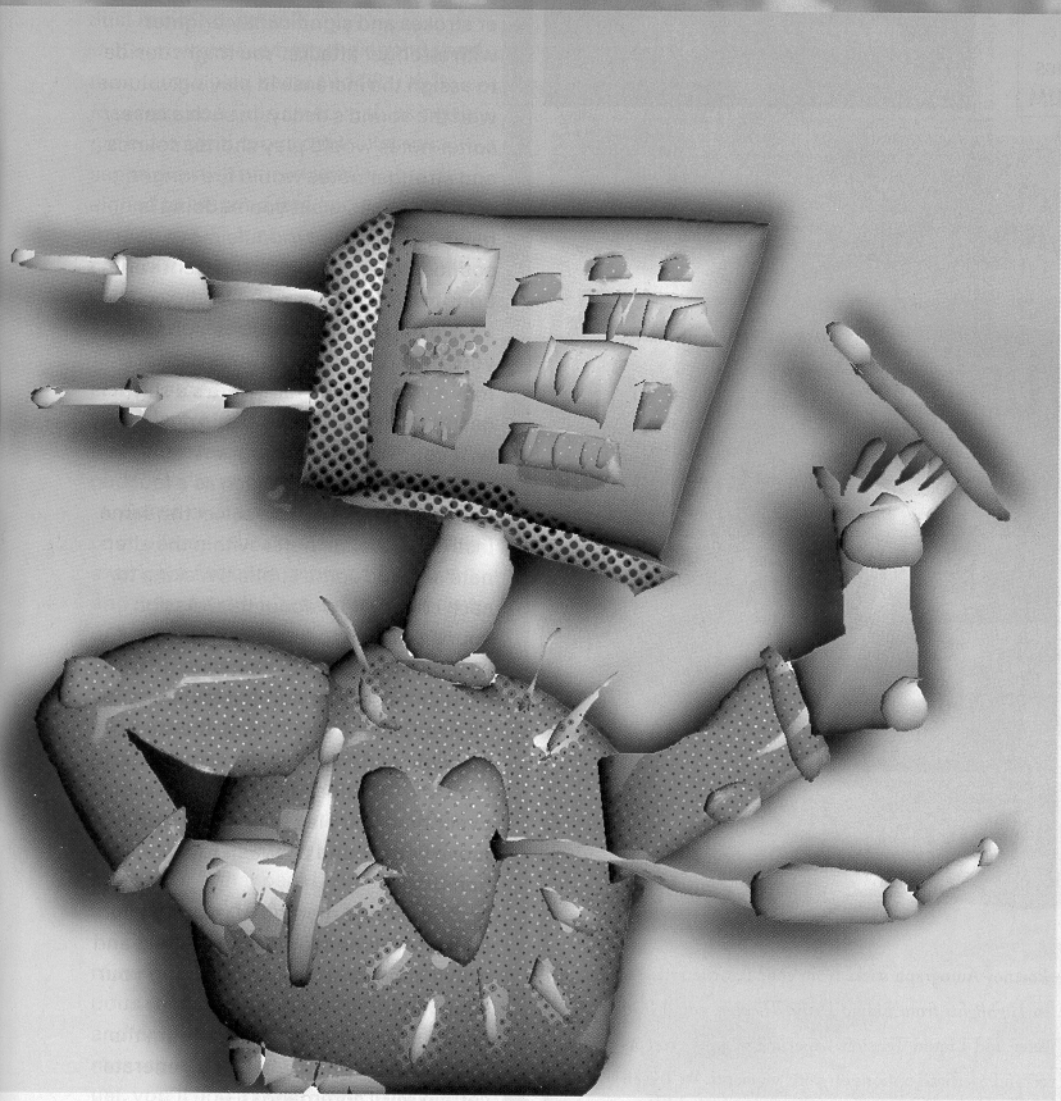


Illustration: Scot Halpin

CURRENT TECHNOLOGY IS AMAZING! Drum machines, electronic drum kits, workstations, and samplers have gotten so good that it's often difficult to tell the difference between acoustic and electronic generated performances. One telltale sign of electronic origin is when you hear the exact same sound repeated over and over for each note.

An acoustic instrument will exhibit very minor and subtle differences from stroke to stroke. Your left hand sounds just a little different than your right hand, each crash cymbal attack has a slightly different attack and decay, each hi-hat opening happens with somewhat different dynamics and character. The purpose of this article is to offer some suggestions to help you make your sound move and change through time. And even if you aren't trying to emulate acoustic sounds, giving movement to your drum sounds can generate some totally unique and unreal textures.

Before we get into some specific ideas, it might be a good time to offer a couple of cautions. Some options will work better than others, depending on the sound. For example, adjusting the low-pass filter on a cymbal sample is going to be much more effective than the same technique assigned to a bass drum sound. Also keep in mind that the more flexible your controller and sound module, the more you'll be able to tap your creative potential.

Who's in Charge Here? There are two methods of getting movement into your sounds. The first involves setting up your sound module so that most of the work is done automatically. Several current sound modules have a "patch-cord" interface that can route one musical parameter to another (instruments like the e-Mu Proteus-based modules are well known for this feature). Here's an example: Low-frequency oscillators (LFO) on a piano-style synth are usually assigned to produce a vibrato. But routing a low frequency oscillator to stereo position would result in a vibrato-like change — not with volume, but within the stereo field. The faster the rate of the LFO, the faster the sound will move between speakers. As LFO depth increases, the sound will move harder to the left and right. Once this is programmed on your sound module, your job is over. Any stroke you play will move the sound between your speakers. Depending on your sound module, you might be able to route the LFO to a number of different destinations that can move your sounds in a subtle or not-so-subtle manner.

Normally, playing with more power will result in a louder sound. But there are more options with this patch-cord con-

cept. You could route your playing velocity not only to volume, but also to filter cutoff frequency. Snares, cymbals, and other percussion samples that have a good deal of high-frequency information will sound slightly muted with softer strokes and significantly brighter with stronger attacks. You might decide to assign the increase in playing volume with the sound's decay. In such a case, softer notes would play shorter sounds and stronger notes would fire longer sounds. Again, what you're doing is having your sound module move your sounds between several different variations for you.

Did you know that you can program your drum controller to automatically move your sounds too? Here's a suggestion that will illustrate this possibility. If your kit lets you program multiple notes for a single pad (such as a four-note alternate), you can select the same instrument for all notes within the alternate and add some subtle tweaking to each sound. How about this? Assign your snare pad to fire MIDI notes 38, 39, 40, and 41. Copy your snare sound into each of those note number locations. Then adjust the fine-tuning, the stereo position, and the tone, the sample start time, the duration, and so on for each. Every time you play your snare, you'll move your sound between four variations.

The second method of moving your sound involves taking a more direct and active approach. Instead of having your sound modules programmed to respond automatically, you program the module to respond when you generate certain MIDI commands.

These messages are called "control change messages." Control change messages include a very flexible set of commands that are initiated by "continuous controllers." Continuous controllers are a set of instructions that are commonly associated with sliders, wheels, knobs, and various pedals. Most often, continuous controllers send data that affects the sound in some manner. For example, a keyboard player might move the modulation wheel during a performance. Since most keyboards are built to recognize the movement of this wheel, they will generate a series of messages as continuous controller #1. If the sound module is programmed to respond to modulation by adding some distortion to the sound, then the player can control the amount of distortion added to the sound while playing the note. Want a dirtier sound? Just move the wheel.

Drummers use both their hands and their feet to play beats and fills, so it may require a little creative thinking to generate continuous controller messages. Some drum controllers are designed to respond to a pad's pressure (the drumKAT is one example of a drum controller that will read pressure), and can be controlled by leaving the stick against the head and pressing harder. This is because they use a technology called force-sensing resisters (FSR). If your electronic percussion controller doesn't generate pressure from the pads, you're not totally out of luck.

Some of the more modern hi-hat controllers — the ones that can produce a variety of sounds between fully open and fully closed — are actually continuous controller pedals. Depending on your controller and your sound module, you could assign your hi-hat pedal to send a particular continuous controller MIDI message, and program your sound modules to respond to those messages in any way you wish. Once completed, every time you open or close your hi-hat pedal, you'll be making your sound move.

Check your owner's manual to determine if your drum controller and continuous controllers send information and/or if it will respond to these MIDI messages. Toward the back of the manual, you'll find a page titled "MIDI Implementation Chart." On the left side of this chart, you'll find a heading called "control change." Look for an open circle (an open circle means "yes" while an "X" in this position means no) across from any controller number. The table below lists all of the controller messages that are currently included in the MIDI specification. As you can see from the list, getting your sounds to respond to some of these parameters can offer a great deal of variety: effect level, sound attack time, chorus level, etc. ➔

Plugged In

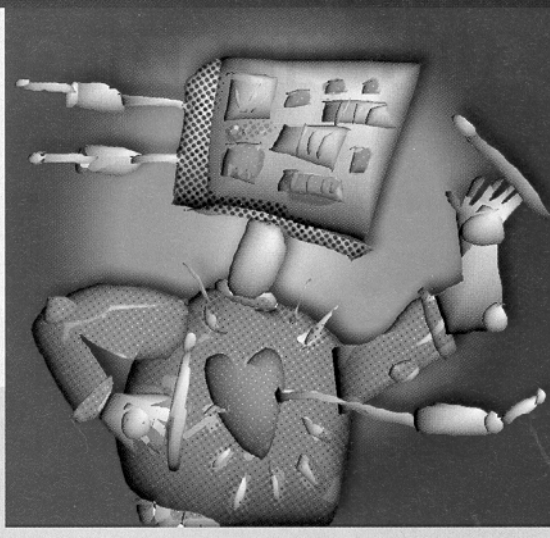
If your hi-hat pedal isn't the type that can send continuous controller information, you might think about adding a pedal or two to your setup. There are many times when you might be able to transfer a foot or two from your drum pedals to your controller pedals.

While not exactly a continuous controller, another popular tool is pitch bend. Sending pitch bend information to your drum module can result in some pretty wild sounds. If you set the pitch bend depth to an octave, you'll be able to tune your drums and cymbals over a huge two-octave range (from full negative to full positive). If you send the pitch bend depth to a half step, you'll be able to apply some subtle changes to your sounds.

Controller Messages Included in MIDI Specification

0. Bank Select
1. Modulation Wheel
2. Breath Control
3. Continuous Controller #3
4. Foot Controller (coarse)
5. Portamento Time (coarse)
6. Data Entry Slider (coarse)
7. Main Volume (coarse)
8. Stereo Balance (coarse)
9. Continuous Controller #9
10. Pan (coarse)
11. Expression (sub-Volume) (coarse)
12. Effect Control 1 (coarse)
13. Effect Control 2 (coarse)
14. Continuous Controller #14
15. Continuous Controller #15
16. General Purpose Slider #1
17. General Purpose Slider #2
18. General Purpose Slider #3
19. General Purpose Slider #4
20. Continuous Controller #20
21. Continuous Controller #21
22. Continuous Controller #22
23. Continuous Controller #23
24. Continuous Controller #24
25. Continuous Controller #25
26. Continuous Controller #26
27. Continuous Controller #27
28. Continuous Controller #28
29. Continuous Controller #29
30. Continuous Controller #30
31. Continuous Controller #31
32. Bank Select (fine)
33. Modulation Wheel (fine)
34. Breath Control (fine)
35. Continuous Controller #3 (fine)
36. Foot Controller (fine)
37. Portamento Time (fine)
38. Data Entry Slider (fine)
39. Main Volume (fine)
40. Stereo Balance (fine)
41. Continuous Controller #9 (fine)
42. Pan (fine)
43. Expression (sub-Volume) (fine)
44. Effect Control #1 (fine)
45. Effect Control #2 (fine)
46. Continuous Controller #14 (fine)
47. Continuous Controller #15 (fine)
48. Continuous Controller #16
49. Continuous Controller #17
50. Continuous Controller #18
51. Continuous Controller #19
52. Continuous Controller #20 (fine)
53. Continuous Controller #21 (fine)
54. Continuous Controller #22 (fine)
55. Continuous Controller #23 (fine)
56. Continuous Controller #24 (fine)
57. Continuous Controller #25 (fine)
58. Continuous Controller #26 (fine)
59. Continuous Controller #27 (fine)
60. Continuous Controller #28 (fine)
61. Continuous Controller #29 (fine)
62. Continuous Controller #30 (fine)
63. Continuous Controller #31 (fine)

So, think about your individual set up and gear. How many ways can you program your sound module to move your sound? How can you generate continuous controller information from your kit, and how can you route it creatively into your sound module? I'm sure you'll be able to come up with some clever ideas and unique sounds.



64. Hold Pedal (Sustain)
65. Portamento on/off
66. Sustenuto Pedal on/off
67. Soft Pedal on/off
68. Legato Pedal on/off
69. Hold Pedal 2 on/off
70. Sound Variation
71. Sound Timbre
72. Sound Release Time 0..127
73. Sound Attack Time 0..127
74. Sound Brighness 0..127
75. Sound Control 6 0..127
76. Sound Control 7 0..127
77. Sound Control 8 0..127
78. Sound Control 9
79. Sound Control 10
80. General Purpose Button
81. General Purpose Button
82. General Purpose Button
83. General Purpose Button
84. Undefined on/off
85. Undefined on/off
86. Undefined on/off
87. Undefined on/off
88. Undefined on/off
89. Undefined on/off
90. Undefined on/off
91. Effects Level
92. Tremulo Level
93. Chorus Level
94. Celeste (Detune) Level
95. Phaser Level
96. Data entry +1
97. Data entry -1
98. Non-Registered Parameter Number
99. Non-Registered Parameter Number
100. Registered Parameter Number
101. Registered Parameter Number
102. Undefined
103. Undefined
104. Undefined
105. Undefined
106. Undefined
107. Undefined
108. Undefined
109. Undefined
110. Undefined
111. Undefined
112. Undefined
113. Undefined
114. Undefined
115. Undefined
116. Undefined
117. Undefined
118. Undefined
119. Undefined
120. All Sound Off
121. All Controllers Off
122. Local Keyboard On/Off
123. All Notes Off
124. Omni Mode Off
125. Omni Mode On
126. Monophonic Mode On
127. Polyphonic Mode On (mono=off)