

Melodic Inverter

Version 1.1.0

Operation Manual



ENLIGHTENSPEED

Table of Contents

Introduction	4
Melodic/Chromatic Inversion – The Concept	6
The Melodic Inversion Algorithm	6
Notes on context of the intervals	7
The Double Interval Algorithm	8
The Spread Algorithm	9
Random	9
Reset	9
The Combination Algorithms	10
Notes on monophonic and polyphonic input	10
The Sequence and Edit Panels	12
Sequencer, Step and Global Modes	14
Rate and Steps	15
The Randomize Button	15
The Sequencer Panel	16
The Algorithm Channels	16
Passthroughs	17
Mutes	17
Octave Shifts	17
The Step Trigger	17
The Edit Panel	18
Spreads	18

Global Algorithm	18
Randomizer Settings	19
The included patches	19
Using with other Players	20
Players stack routing order	20
Using Direct Record	21
Acknowledgements	22

Introduction

Hello and welcome to Melodic Inverter by Enlightenspeed.

We were looking for ways to do something a bit different and had some ideas about doing a type of “Note Alter-esque” plugin but with more predictable results that were therefore better suited for melodic content. A Reason user posted up a video mentioning the practice of melodic inversion (otherwise known as chromatic inversion), and we were impressed.

Although the technique wasn’t first explicitly documented until the early 16th century, examples might date back nearly a thousand years; earlier composers may have stumbled upon it accidentally. Much later, the brilliant Sergei Rachmaninoff composed arguably the most famous example “Rhapsody on a theme of Paganini”, which was based on the 24th and last of Niccolò Paganini’s “Caprices for Solo Violin”.

Times have changed, of course, and whereas this technique was typically very mechanical and difficult to do for non-score composers at the best of times, the Melodic Inverter brings this technique and much more into the modern era. This is the first and only Rack Extension which gives this type of capability, and we are not aware of any VST/AAX/AU etc which can do it; we did however find some sparsely scattered blog and forum posts about having built similar basic versions using CSound.

This manual will explore the basics of operation for our device, but before we continue, we must give a warm thanks to all our friends at ReasonTalk.com who gave us so many helpful ideas in the development phase. We originally released this as a bare bones algorithm without any real interface, partially due to time constraints, but also because this type of plugin hasn’t to our knowledge ever been publicly available before and we were a bit fuzzy about what the best feature set would really be. So, we asked the Reason using public what they thought, and made it explicit that we didn’t want “just another Loopcracker device” with our now standard note and velocity range parameters. In particular, we have to thank Dj and Producer Rikk Show for helping out with the GUI design. Many others

contributed cool ideas, and of course we wouldn't have got to the finish line without our wonderful testing team. In addition, we need to say a big thanks to Juhani Ekstrand at Reason Studios for saving our bacon near the end when there were some basic necessities that we overlooked which needed to be addressed.

Melodic/Chromatic Inversion – The Concept

The intention of the Melodic Inverter Rack extension is to alter incoming notes in a way that may seem random but is mathematically structured and entirely reproducible. The project started with a single algorithm, Melodic Inversion, and then grew from there, to the extent that it is now a little bit inaccurate to name the whole project Melodic Inverter – but that's the name in the Reason Studios Shop, so we're stuck with it. :D

There are now 5 basic functions that the device performs, including melodic/chromatic inversion, Double Interval, Spreads, a newer gentler random algorithm, and the reset algorithm. Combine these with the extra features such as step triggering, multiple play rates and step counts, and the very powerful device randomizer, and we get an exceptionally robust composition and re-composition tool.

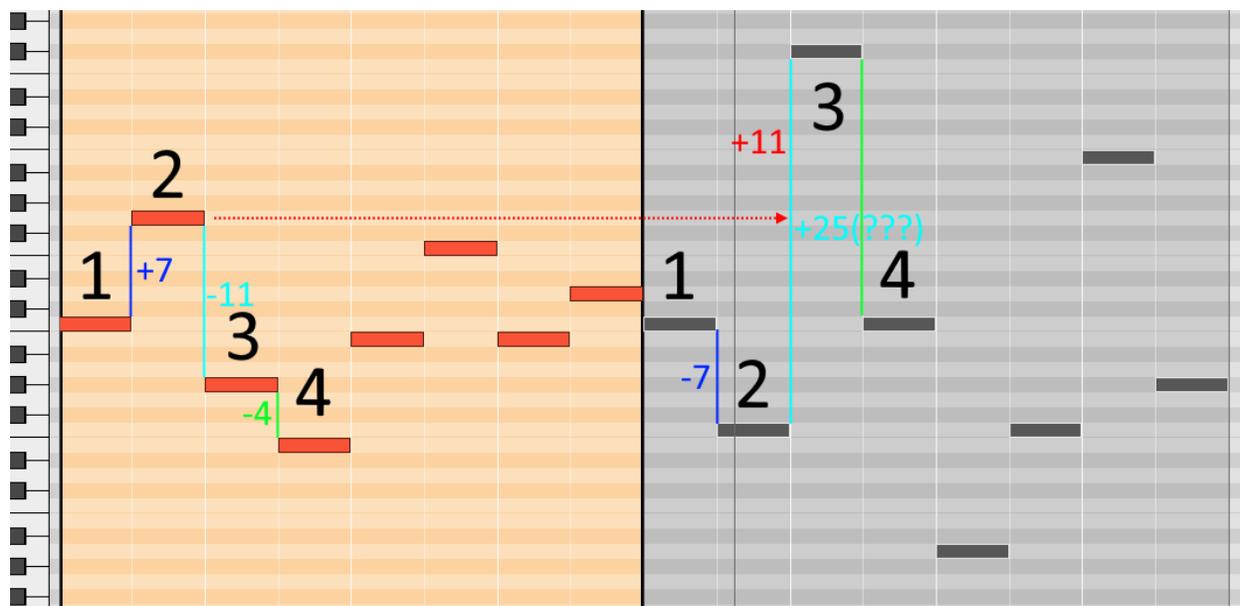
Let's get to grip with the underlying concepts of the device first, though. The algorithms are relatively easy to get your head around, but it is necessary to understand them before starting to work with the plugin.

The Melodic Inversion Algorithm

The idea that kicked it all off.

Melodic inversion is incredibly simple, yet stunningly powerful. At it's simplest it is a mathematical process applied to existing note data, but the result is profound. For the following examples we will only use monophonic material in order to demonstrate the effects more clearly, then at the end of this section we shall discuss the effects on polyphonics.

First up, let's look at a simple monophonic MIDI segment.



The notes above, taken from the Reason sequencer show the differences between the input in the orange section on the left, and the Melodic Inverter output on the right in grey.

The change between the 1st and 2nd notes shows a nice easy to understand process. Note 2 is originally on G, 7 steps up from Note 1 on C, and on the output it is now on the F, 7 steps down from the original C. Simple.

The change between the 2nd and 3rd notes is far more radical. In the original version we see that the interval is 11 steps down, but the output shows a gap of +25 steps! What is happening is that 3rd note is actually +11 steps from the original placement of the 2nd note. The 2nd note has been shifted down by 7 though instead of up, so the resulting interval is thus:
 $7+7+11=25$.

As an exercise, do the maths in your head for the interval between the 3rd and 4th notes, and if necessary, continue until the end of the sequence.

Notes on context of the intervals

Before moving on we briefly need to discuss some other points of interest. Firstly, the opening note. Many users of the original GUI-less version of Melodic Inverter have asked about the first note being no different on the 1st pass, and then different on the 2nd pass if looped. The inversion algorithm, and indeed any algorithm, needs an initial state to start from and then go to work. Because the first note of any sequence has no predecessor, there is no interval for it to be inverted from. In a looped scenario, because the device works in realtime, this interval is generated by the distance between the **final** note of the loop on its previous pass and the first note, which is now no longer the first note. This means that it's usually a good idea to have a listen to both 1st and 2nd pass versions, to determine whether you prefer one or the other, or both.

Next, it shouldn't really come as any surprise when looking at the 2nd to 3rd note interval, that larger original intervals can really move things about a bit. In this case we decided that any notes which are thrown right out of the standard MIDI range of 0-127 are cancelled. Unless you are using a sampler patch where the samples bear no relation to the MIDI pitch, then this should probably come as a pleasant surprise. We toyed with the notion of a foldback function, but realised that there was little musical value to this, as in the extreme high and low ranges things become less discernible as musical pitches anyway.

Lastly, running consecutively increasing or decreasing pitches will yield considerably less radical results. The output will "level out" faster, and the result will be mostly the same intervals just pitched up or down a bit. Mathematically this is exactly what should happen. If you want to increase the effect, just go into the sequencer and grab a few notes at random and move them up and down.

The Double Interval Algorithm

Now that we have some basics, let's look at an even more effective method for altering the original that doesn't even need a diagram! Simply using the basic concepts above, instead of inverting an interval it doubles it. So instead of an interval being +5, it becomes +10.

This is very simple of course, but the important thing to note here is just how quickly things can move out of the MIDI range, or indeed just out of the realistic range for a particular instrument or musical arrangement. For cases where there is problematic results you can either move the resulting notes by hand, or alternatively use note filtering to remove anything above a particular threshold – there's a few tools for this including, of course, our own Note Filter plugin which is part of the Loopcracker Suite.

The Spread Algorithm

The 4 Spread algorithms give a user definable version of the Double Interval whereby if the user inputs notes with intervals of +4, +6 and -4, then the output, when the Spread control is set to +3, will be +7, +9, and -7.

This gives a more controlled and easier managed approach.

Random

We decided that with the random algorithm on this product, to make it a little tamer than it is on, for example, Note Alter. In this case the Random function is set to choose a random note within one octave up or down from the original note.

Reset

The reset algorithm is there simply to allow the user to take back a little control of sequences which are getting messier. When a reset function is called it will pass the incoming notes through untouched, and also reset the holding variable which tells the other algorithms the interval of the previous note. This is particularly of use in the Sequencer and step modes.

The Combination Algorithms

There are two types of combination algorithms:

Melodic Inverter & Spread

Double Interval & Spread

These work mostly as you would expect, with the Spread values being added to the melodic inversion or double interval outputs, respectively. As there are four different Spread controllers, you get 4 of each one, linked to the appropriate Spread parameter value.

So, in total, from a single monophonic input segment, you can get an extra 12 arrangements of the same source which are all reproducible without bouncing down the MIDI and manually adjusting things. If you desire you can either abandon the original key and scale or quantize them all to the same scale; but we're sure you will agree that despite all the weird maths stuff going on above, the output results are almost always magnificent.

Notes on monophonic and polyphonic input

A final word on this before we talk about setting up the device itself.

The classical techniques employed by various baroque and romantic era composers were almost exclusively monophonic. This is just common sense really because monophonic sources, at least would not require a

whole new set of complex rules regarding how they were to be inverted or otherwise altered.

We decided that we wanted polyphonic anyway because we're letting a machine do all the heavy mathematical lifting. :D

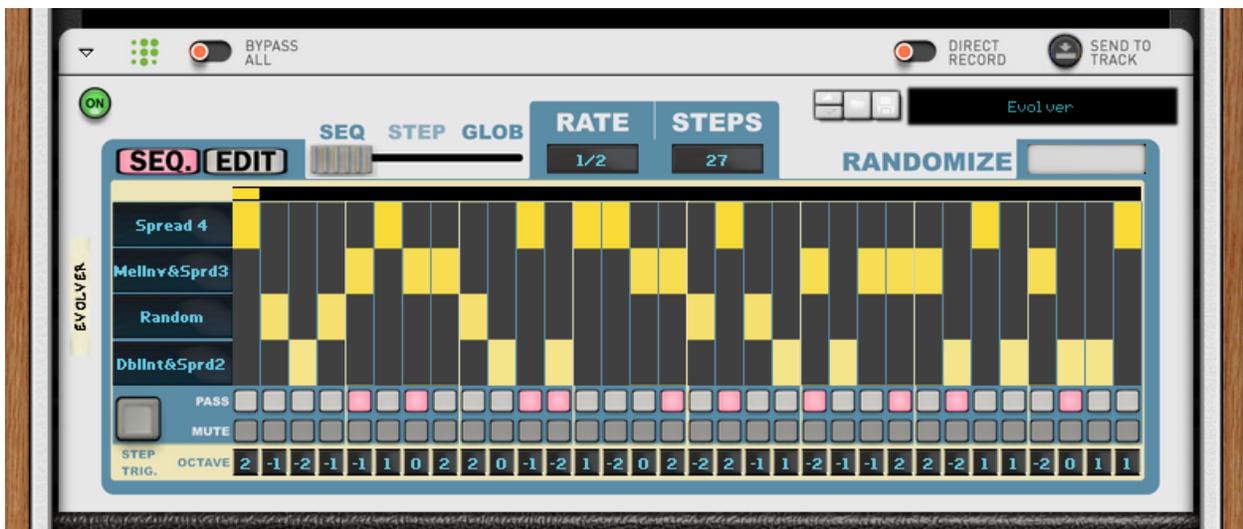
The results are considerably less predictable, and far more likely to move around all over the place, more so when using "open chords".

Again, note filtering may be the best solution here if things are getting out of hand.

The Sequence and Edit Panels

The device has two major modes of display, compressed into hide-able widget panels, similar to Thor's oscillators if you've never used anything else which does this.

We have the Sequencer panel...



... and the Edit panel.



Each of the panels is accessed by the Seq/Edit radio buttons, and hides the other to save space – the Seq/Edit button has no effect on the input and output of the device. We'll discuss the rest of the panel details in their individual sections.

The Sequencer, Step and Global Modes

Control of the general mode of the device is governed by the 3-way switch shown below.



The Sequencer and Step modes are governed by the parameters on the Sequencer panel, and the Global Mode is governed by the Global Algorithm parameter, which is present on the Edit Panel.

Each mode has its own merits, but in general, the Step and Global modes are quite cognitive and deliberate in usage, whilst the Sequencer mode, particularly when randomized is an experimental free for all. That said the Sequencer can be the most deliberate of all if using it as a controller with a slow step rate.

The 3-way switch is always present on the main panel of the device regardless of the display mode.

Rate and Steps

Also present on the main panel at all times, are the Rate and Steps controllers.



These controllers are only applicable to the Sequencer Mode. The Rate controller specifies how often the step will change when the sequencer is running, and it has the following values available:

4/1, 2/1, 1/1, 1/2, 1/4, 1/8, 1/16, 1/32.

The Steps controller determines the number of Steps before the Sequencer loops back to the start. The range is any integer between 1 and 32 inclusive. Any power of two (2, 4, 8, 16, 32) is safe for predictable looping, but the more adventurous settings are the higher odd numbers, which can give a constantly evolving set of outputs.

As with almost all of the controls on this device, the Rate and Steps controllers are Remoteable and automatable, so this allows your patterns to become even more exotic and evolutionary.

The Randomize Button

This last control on the outer panel is simply a trigger button to tell all the selected elements to Randomize. For more details on this see the Randomizer Settings section.

The Sequencer Panel

Parameters on the sequencer panel are governed by the step number, which is indicated by the timeline bar along the top - this is very similar to many other sequencer devices.



When using the device in Step Mode, the step on the timeline is only changed by the Step Trigger, and in Global Mode it is not relevant.

The Algorithm Channels and Step Types

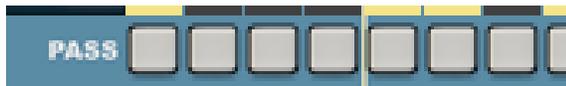


There are four algorithm channels in the Sequencer panel: A, B, C and... I can't remember the last one. :D These are on the left-hand side of the image above and can be assigned to any of the 16 algorithms.

Each step has its own fader, with channels A-D available as steps. So the basis of operation is pretty much the same as any other sequencer, where steps are controlled by the timeline, the Step Type governs which of the four selected algorithms is in use, and when notes come in they are assigned to an algorithm as applicable.

All the Step Types and Channel Algorithms are fully Remoteable and Automatable.

Passthroughs



Each step has its own Pass button, and when it is activated the incoming note will pass through the device unchanged. All of these are Remoteable/automatable.

Mutes



Each Step also has its own Mute button, and as would be expected, this silences any incoming notes while the timeline is on this step. Again, all of these are Remote-able/automatable.

Octave Shifts



Lastly, each step has its own Octave shifting option. These allow you to shift any notes within a step up to 2 octaves up or down. These are all Remoteable/automatable.

The Step Trigger



The last parameter on the Sequencer on the panel is the Step Trigger. This simply controls what Step the sequencer is on when the device is in Step

mode. In Global or Sequencer mode it is ineffective. This is, of course, Remoteable and automatable.

The Edit Panel

Spreads



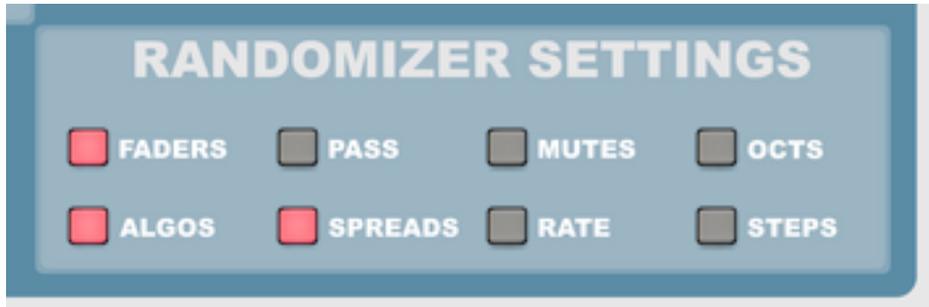
There are 4 Spread controls in the Edit Panel. Each of them ranges from 0-12, where 0 is effectively off and 12 is a full octave. Each value simply adds or subtracts the appropriate amount to the incoming notes, depending on which Algorithm is selected.

Global Algorithm



The Global Algorithm is how you change the algorithm when in Global Mode. Of course, you can use this as an on the fly controller for the whole device, but another great usage for it is to help you fine tune the individual algorithms, particularly the Spreads – then once this is done you can go to town with the Randomizer on the Sequencer.

Randomizer Settings



We wanted a really quick and easy way of setting up custom randomizers, and this seemed like the best way to do it. Simply use the toggle switches here to specify which values you want to be randomized.

Once you have the settings you want to randomize just hit the Randomize button on the Main Panel.

The Included Patches

We kept the patches to a minimum for Melodic Inverter. To put it simply, you won't know what works until you've fiddled around a bit, and every source of input material can produce varied results, so although it was necessary to include some "general ideas" patches to show different setups, this device is best used with patches as a cognitive aid rather than a replacement.

Using Melodic Inverter with other Players

All Player devices in the Propellerhead Shop are standards compliant with The Rack Extension SDK version 3.0 and onwards. This means that a certain set of behaviours are enforced, ensuring interoperability between all Player devices and how they receive and transmit data to other devices, including those which may comply with different standards.

As boring as the above paragraph may sound, it is hugely important when it comes to having fun musically using these devices. Effectively, it means that you are safe to assume that any Player is compatible with Melodic Inverter. Good results are easily and safely obtained when using Melodic Inverter alongside Note Echo, Scales and Chords, Dual Arp, and an ever-growing collection of third-party devices, including our own Loopcracker Suite devices, of course.

There are only two details which really need to be given consideration: Routing and Direct Record.

Routing

Players will always work in a top to bottom order as presented on the rack screen. This starts at the main control unit at the top, which hosts the Direct Record switch and the Bypass All, along with the Send to Track button.

The routing of the Players is then in order from top to bottom unless something is done with CV if applicable connections are present on the device, or a special device is used to break the routing; there are currently 2 such devices in the Propellerhead Shop at the time of writing.

Finally, the note events reach the instrument at the bottom of the stack and **ONLY** that instrument. Should you have multiple instruments that you wish to send the Player notes to, then all the instruments will need to be present in a Combinator and the Players need to be external to that Combinator. Should you wish separate FX processing for each of these you then need to send the audio outputs to separate Mix Channels.

Direct Record

The Direct Record switch is arguably the most important standard feature of the Player stack. When it is engaged the Player will transmit the output results of whatever manipulations have been set up to the applicable sequencer track. This is all fairly simple if using a Player or CV device as the sole source of note generation, you just record straight into an empty instrument track.

If using data from the sequencer track itself though, another issue arises; a sequencer track cannot act as both source and target for Player notes, it is one or the other.

The solution to this is again to place the instruments in a Combinator, which means you then have two sequencer tracks to play with. From this point you use one as the source and one as the target – the exact nature of this is dependent on what has been setup, and with what intent.

There are multiple possibilities, of course, but no seasoned user should struggle with it. If you are a novice then it's worth visiting one of the Reason user websites such as ReasonTalk and asking for advice, or one of the many social media platforms such as Facebook. We all start somewhere, and with the exception of one or two individuals who have social issues, you tend to find that the majority of the people in these groups are very friendly and wish to help.

Our own Facebook page will be fairly well monitored so if there is any specifics that you need covered then this is one of the best places to do it. Simply like the page, add yourself to the group and send us a message. Our response time will be "reasonable", and we will endeavour to give you the info you need.

Acknowledgements

The following people made this possible, without them it wouldn't have happened...

All family and close friends

Juhani Ekstrand and everybody at Reason Studios, Stockholm

Kenni Andruszkow and the rest of the fine folks at ReasonTalk

Navi Retlav and all at NaviRetlavSound

Giles Reaves and all at Selig Audio

Bes

Loque

Burningtoken

Diminished

Boombastix

CatBlack

MarkTarlton

MrFigg

Tiker

Nisse

Reason2Dance

Olive

Challism

All other testers and content providers have our massive gratitude.

Thanks to all the good folks on the various Reason-centric Facebook pages as well for giving moments to laugh, cry and feel inspired about.

Any questions? As in, reasonable ones not answered by this document?

We might not answer at all, if you send us a line to
enlightenspeed@gmail.com



