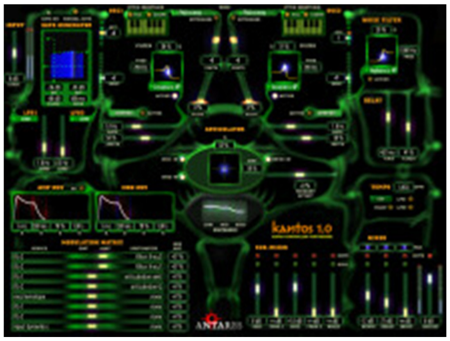


Review: Antares kantos 1.0

by Rick Paul ***- 31st May 2003 -***

Every April Fool's Day or so a "product announcement" tends to rear its head announcing new technology that allows songwriters and music arrangers to get exactly what they hear in their heads out into the world of musical sounds. There's typically something about how it connects to the brain, and how you don't even need to know how to play an instrument -- "just imagine what you want to hear and <insert fictional product name> will do the rest!" Of course, any product that could really do that would, at least for now, be revolutionary technology, and would probably sell in huge numbers to people with active imaginations but limited instrumental technique.

Short of that direct brain-to-music hookup, I imagine the next best thing would be if the composer could just sing a part. It would be nice if he didn't have to worry too much about whether he could sing perfectly in tune. Further, the sounds that emanated from this "instrument", if you will, should sound just like whatever the composer imagined playing that part, complete with any phrasing nuances the composer sang. In particular, there should be none of the limitations that typically affect a keyboard player trying to play, say, a sampled bass part, where the slides and other nuances a real bass player might make either weren't achievable at all or required the keyboard player to jump through hoops, thereby consuming any crumbs of creativity.

Does that still sound too futuristic? Perhaps, but I recently came across a product blurb, on the web site of a highly respected pro audio software company, that read:

From the company that revolutionized vocal intonation processing comes kantos 1.0, a software-based synthesizer that finally liberates you from the tyranny of MIDI, keyboards, controllers or, in fact, anything that stands between you and the music you hear in your mind.

Intrigued? I was.

The company is, of course, Antares, who probably made their biggest waves in the audio industry with AutoTune. AutoTune has been used, and sometimes abused, on a huge number of pop, country, R&B, and other genre recordings. Used as it was originally intended, AutoTune is transparent and, for better or worse, has made vocal intonation issues on major label records largely a thing of the past. One might maintain this is a form of aesthetic hearing protection in this age of singers who look great but who aren't always signed on the basis of their singing prowess. Used to an extreme, though, AutoTune can make human voices sound like they were played on keyboards, with abrupt, stair-stepped shifts between pitches where smooth glides were originally sung.

Whatever your take on the merits of those two scenarios, though, AutoTune can also be a lifesaver when working with a top notch singer who captured a vocal that is dripping with emotion, but had a minor pitch issue on a note or two. True, the singer could probably get those notes spot on next time around, but it would be at the risk of diluting that once-in-a-lifetime emotion, and I'd go for the emotion any day if it weren't for today's standard's demanding technical perfection, too. AutoTune allows getting the best of both worlds, fixing those slightly off-key notes to rescue that emotional take-of-a-lifetime.

Antares is also the developer of Mic Modeler, a product that attempts to make one mic sound like another, and Tube, which adds a variety of analog tube-like effects to digital tracks. Golden eared audio engineers may debate whether one arbitrary mic can ever truly sound like another simply by using software to subtract off the unique traits of one and add on the unique traits of the other, especially if you are trying to make an inexpensive mic sound like a megabucks high-end vintage model. Nevertheless, I've heard some well respected professionals say Mic Modeler has saved the day when a mic used on an earlier session was no longer available for an overdub session, but another vendor's model, which was at least comparable in quality, could be used with Mic Modeler to get more seamless punch-ins. Moreover, even those who aren't limited in the mic closet department have praised Mic Modeler for creative sound sculpting uses where simply using EQ wasn't quite what the doctor ordered. And Tube, which is Antares' newest product, and which is based on technology initially developed for Mic Modeler's tube preamp simulator section, has also gotten high grades from users in its limited time on the market.

As a user of all three of the above-mentioned Antares products, and many other plug-ins from a number of vendors, I have come to view Antares as a company that specializes in creating "horizontal niche" products. That is, their products fill very specific needs, generally ones which either haven't already been filled by other companies or which heretofore haven't been filled adequately -- thus the "niche" part. Yet those needs are common enough that makers of all types of music will need, or at least could use, the products -- thus the "horizontal" aspect. This notion made the claim about what kantos could do all the more intriguing. Could kantos be the next evolutionary step in moving toward that pie in the sky brain-to-music connection? Let's take a look.

The Basics

So, what exactly is kantos 1.0? Is it an instrument? A vocoder? An effect? Some kind of magic? All of the above? None of the above?

Antares describes kantos as an audio-controlled synthesizer. Instead of using MIDI to tell kantos what notes to play, you use audio. It plugs in like an audio effect -- i.e. not like a softsynth --

supporting DirectX, VST, and RTAS on Windows, as well as a few Mac plug-in formats. Yet, it isn't really an effect because the sounds that emanate from it, other than to the degree which you mix the input sound with the output sound, are synthesized within kantos. The input sound is controlling the output sound, but may have little in common with the actual output sound.

If that sounds hard to grok, welcome to the world of kantos. Or to put it another way, kantos is a seriously deep instrument, almost guaranteed to confound the first-time user. If you're expecting instant gratification from any new plug-in, kantos probably won't be for you. On the other hand, if you don't mind some real work, as long as it is can lead to some truly unique, and useful, results -- hey, remember when years of practice used to be necessary to become adept at playing a musical instrument? -- then kantos may just be the most novel possibility to come along in some time.

Since this is CakewalkNet.com we will, as usual, be looking at kantos 1.0 from the perspective of SONAR, and other Cakewalk products (e.g. Project5), users. As such, we will steer clear of considerations such as Mac versions, RTAS versions, and so on, which are not applicable to these users. However, users of other products may also benefit from much of the discussion below since most of what kantos can do is common between its various platforms. The goal here is not so much to be exclusive, but rather to include enough specific information to help SONAR users get a feel for how kantos might, or might not, fit into their chosen environment.

It should be noted that, though the native plug-in environment for SONAR is DirectX, the DirectX version of kantos 1.0 does not support automation. Thus, if automating kantos' various controls is important to you, you will need to run the VST version. The VST version of kantos does work in SONAR with an appropriate VST Adapter, including the Cakewalk VST Adapter 4 (which was originally developed and marketed by FXpansion). Most of my testing was done with the DirectX version, though I did try the VST version briefly to check out the automation.

We'll dive into the details in a minute. First, though, let's cover some basic considerations.

Software Protection and Installation

Like other Antares products, kantos 1.0 features software protection via PACE InterLok, a software-only system that authorizes the plug-in to a specific hard disk using a driver-level component for added anti-piracy protection. Even installing the kantos demo software on your system will install the InterLok drivers, and "PACE" tends to be a four-letter word within the SONAR users' community due to a number of, mostly historical, incidents where the InterLok drivers have caused serious system stability problems.

There, that's out of the way, and we've lost some readers who won't even consider a product protected by PACE.

While I'm no fan of PACE, and have been frustrated on a few occasions by specific issues with InterLok software protection in other Antares products, the reality is that I use my Antares products too much to just walk away from them due to software protection concerns. This has caused me some grief over time. For example, during the course of writing this review, I upgraded my system hard disk, and this required reauthorizing all my InterLok-protected plug-ins (four plug-ins from Antares plus a plug-in bundle from Waves). It ended up taking over two full business days to get all my Antares products reauthorized!

In the case of kantos, aside from the already-mentioned reauthorization need after replacing my disk, there were no InterLok-related issues at installation time. I did not even have to reboot my system after installing the DirectX version of kantos. However, I did have to reboot later, after installing the VST version. The VST version of kantos bears a slightly later version number than the DirectX version, and the need for rebooting tells me it must have slightly newer InterLok drivers.

Antares currently provides a 10-day evaluation period for kantos. (Note: Antares indicates they will soon be providing a 30-day demo period as part of a "kantos komposition kontest" sponsored by *Keyboard* magazine.) This serves the dual purpose of allowing potential kantos users to test drive kantos' full functionality before deciding if they want to buy it while also allowing new purchasers of kantos to get up and running with the software before authorizing their copy. The latter can be helpful if, for example, they are unable to use the web-based authorization system and there is a delay in authorizing the software via alternate authorization mechanisms (e.g. e-mail, fax). The only minor catch is that a user who starts out by evaluating the software for the full ten days will not get another 10 days upon purchasing the software, so there could be a delay in getting to use the software upon purchase while waiting for, say, a faxed authorization request to be processed.

Because kantos is so unique, and what it can do is so dependent on the nature of the audio that is used to control it, I can't recommend strongly enough that those considering kantos take advantage of the 10-day evaluation period, rather than simply going by the feedback of other users and product reviews such as this one. For those worried about InterLok's being installed by the evaluation, I would suggest that the stability-oriented problems with the InterLok drivers are, as far as I can tell anyway, relatively ancient history by now (they were mainly an issue when Windows XP was relatively new and the versions of InterLok in various plug-ins at that time had not caught up with respect to Windows XP support), so there should be minimal risk.

Documentation

As mentioned above, kantos is a seriously deep product, so it's a good thing that Antares provides seriously thorough documentation in the form of a 60 page manual. Not only is there the requisite reference material, but there is also a lengthy tutorial. Antares suggests the tutorial takes about an hour to complete, but, unless I'm just slow, I'd suggest it is more like half a day. However long it takes, I consider the tutorial to be mandatory for anyone who wants to try and come up to speed relatively quickly on understanding just what kantos can do.

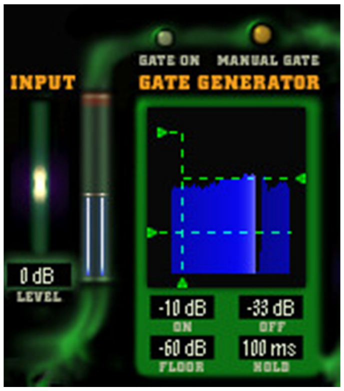
Mind you, just going through the tutorial, won't make you a virtuoso kantos "player". That will take some "practice". However, to use an analogy, trying to use kantos without at least coming to grips with some basics is likely to be as frustrating as looking at a clarinet for the first time and trying to figure out why blowing into the mouthpiece doesn't result in any sound. (For those who are wondering, you need to put the reed on first, and, even then, a reasonably proper embouchure -- er, the way you use your mouth and lips for the non-wind instrument players who are still clueless on the analogy -- will be required to get anything other than some obnoxious-sounding squeaks happening.)

One of the most common things I've heard from people who have tried kantos and did not like it was that they couldn't figure out how to make it do anything even close to useful. I have to wonder what portion of those potential users actually took the time to go through the tutorial before coming to those conclusions. On the other hand, to continue my analogy above, just because you've read the beginning clarinet book doesn't make you a virtuoso clarinetist -- it will take a lot of practice and experience to make that transition. Still, reading the book, and doing the exercises in it, is one step in the right direction (for both the clarinet and kantos). Besides, Antares spices up their documentation with plenty of humor, so it will be an enjoyable read.

Getting Started - What Goes In

Unlike most software-based instrument plug-ins, which are generally inserted in SONAR's Synth Rack and driven by one or more MIDI tracks, kantos is inserted in an audio track's FX bin (i.e. track insert). That's right, I said "audio track" -- remember, kantos is controlled by audio, not MIDI. (Note to SONAR 1.x users: The Synth Rack is a new development in SONAR 2. While you can still insert instruments in individual audio track FX bins in SONAR 2, doing so won't allow you to take advantage of certain advanced softsynth features like multiple audio outputs.) Project5 users would insert kantos into the audio FX slot of the track they wish to use to drive kantos.

Using the clarinet analogy mentioned above, you can think of the audio track that drives kantos as being like the breath that flows through a clarinet. The sound coming out the other end won't necessarily sound like the breath going into it, but you won't get any sound without that breath, either.



0 dB LEVEL

-10 dB ON

-33 dB OFF

-60 dB FLOOR

100 ms HOLD

Audio enters kantos through kantos' audio input and gate generator section. An input level control can be used to raise or lower the level of the incoming audio signal. A hot signal is preferable for optimal pitch detection, so the input level control will most frequently be used to raise the level of a weak signal (i.e. since a too hot signal in the digital world would likely already have been clipped).

As a side note, I might mention that the input level control in kantos can either be dragged, fader-style, with a mouse or entered numerically with the keyboard. This is true for all numerical controls in kantos, so I won't be redundant below. Suffice it to say that this dual control capability makes it possible to choose the precision of entering direct numeric values or intuitive "programming by ear", depending on personal preference for the situation at hand. And all the controls are sitting there right in front of you -- there are no menu pages to scroll through as in some unintuitive hardware synth programming interfaces.

In short, while there is no question that kantos is deep, its interface makes programming it, or at least trying to learn to program it, a pleasure. If I have any complaints about the user interface, it is that some of the numbers and text get pretty small on my 19-inch monitor running at 1280x1024 resolution.

After being raised or lowered by the input level control, the incoming audio goes through the gate generator, which is responsible for basic pitch detection, as well as determination of when to start and stop notes. This is controlled in a manner much like setting a noise gate, and thus the "gate generator" term. The "on" level is the level at which a new "note on", or, more accurately, new note articulation, will be generated. The "off" level determines when a note will be stopped, although a "hold" time provides a minimum note length. A "floor" control tells the gate generator to ignore any background noise below a certain level.

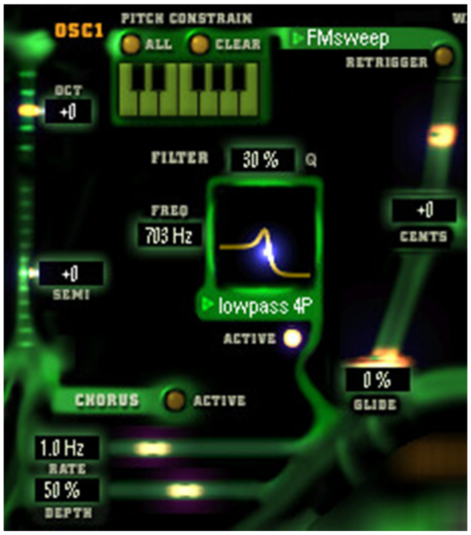
There is also a manual gate that continuously holds the gate open, for example to make it easier to program sounds.

The basic idea of the input and gate generator section is to control what kantos "hears", thus trying to optimize its ability to determine your intentions with respect to, for example, pitch and phrasing. If the audio going into kantos is the breath in our clarinet analogy, and the techniques used to generate that audio are the embouchure, then kantos' input and gate generator section is a tunable mouthpiece and your ability to select reed hardness and materials. While the mouthpiece and reeds don't necessarily affect the breath you put into the clarinet, they do affect the tone and phrasing produced. Perhaps the biggest difference is that kantos is far more tunable in this area. You can tune the "mouthpiece", if you will, to suit the "breath control", whereas with the clarinet, there is only so much that can be done at the instrument tweaking level.

Making Sounds

I love analogies. They can make it much simpler to understand complex things. I'm afraid, though, that we are now at the point where my clarinet analogy breaks down. It might be nice to suggest that the next section of kantos was similar to the clarinet's keys and construction materials in determining what notes were played and how the breath coming in affected the actual tone of the instrument. However, that would be only partially true.

Unlike an acoustic instrument, where the sound is generated through a combination of physical interactions of the player with the instrument and the physics of the instrument itself, kantos is relatively unrestricted by physical limitations. While the pitches it plays and the shaping of the tones it makes are affected by the "player" (i.e. whatever is producing the incoming audio), the raw sounds themselves are synthesized within kantos or provided by sampling sounds generated elsewhere (i.e. with no inherent relationship to the incoming audio).



FMsweep
OCT +0
FILTER 30 % Q
FREQ 703 Hz
CENTS +0
SEMI +0
lowpass 4P
GLIDE 0 %
RATE 1.0 Hz
DEPTH 50 %

Raw tones in kantos are generated by two identical wavetable oscillators. The oscillators can use either periodic waveforms (e.g. sawtooth, triangle, etc.) or 16-bit mono digital audio samples (e.g. plucked strings, human voices, etc.). Antares provides an assortment of waveforms and samples as a starting point, but users with a sample editing program that can create 16-bit mono WAV or AIFF files with the appropriate loop point settings can generate their own wavetables, too.

I should note here that we are not talking about key-mapped multi-samples. Thus, while it is possible to use samples of most anything, kantos is not the way to go for getting realistic acoustic instrument sounds. Then again, there are already numerous samplers out there that attempt to do that trick. Still, those of us who mainly make acoustic-oriented music with samplers, and harbored visions of kantos' liberating us from the constraints of a keyboard controller, say, for generating more emotion-laden lead "guitar" parts, this aspect may be a bit disappointing. (Since a fair portion of my music tends toward country and pop/rock, you may already be guessing what is at the top of my personal wish list for kantos 2.0.)

The signal coming into the oscillators from the gate generator first has its pitch detected, no doubt using technology Antares initially developed for AutoTune. The detected pitch is not necessarily the pitch that will come out of kantos, though. First, there is a "pitch constrain" keyboard that can be used to indicate the scale kantos must obey for each oscillator (and, yes, you can set different scales for the two oscillators).

While this can be helpful for making slightly off-pitch notes come out as the intended notes (think AutoTune's similar scale controls), and when set to all notes will constrain the generated notes to a diatonic scale, for example removing vibrato, it can also be used to really mess with the pitch. Constraining pitch to the root and fifth of the song's key results in any notes closer to the root in the input audio's becoming the root coming out of kantos, while any notes closer to the fifth in the input become the fifth emanating from kantos.

In addition, kantos features octave, semi-tone, and fine tuning (cent) controls to modify detected, constrained pitch. There is also a glissando control to set how quickly one note glides up or down to the next note, potentially putting back some of the expression removed by constraining notes to a scale.

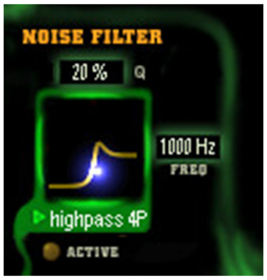
Additional optional components of each oscillator include:

A retrigger control on each oscillator determines whether each sample starts at the beginning of a phrase (i.e. "on" trigger in the gate generator) or on each new note (as determined by kantos' pitch detection).

A resonant filter provides 2 or 4 pole lowpass, highpass, and bandpass filters for shaping the sound. It provides filter frequency and "Q" controls for shaping the filter. Check out the cool "biaxial graphic adjuster" (Antares' term, not mine) in the above screen shot.

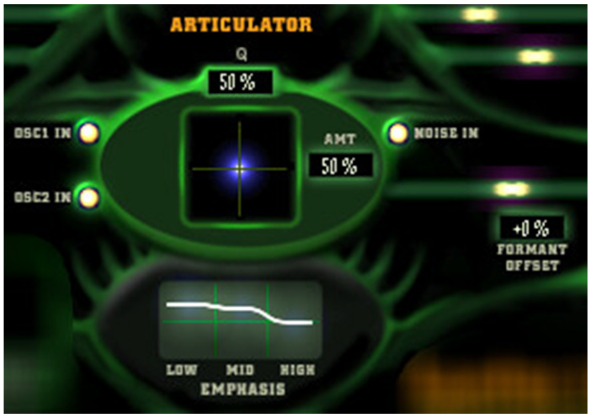
A chorus generator provides the ability to fatten up the sound of the oscillator. It provides the typical depth and rate controls.

The two identical oscillators in kantos are laid out as mirror images of one another. While the mirror image layout makes for a nice visual, I would have preferred identical layouts on each side as I frequently found myself reaching for the wrong control on oscillator 1 after working on oscillator 2, or vice-versa. That's a relatively small niggle, though.



In addition to the two oscillators, kantos provides a noise generator. Like the oscillators, the noise generator features its own resonant filter, which can be enabled or not as needed.

The blend of the two oscillators and the noise generator, along with any of the original signal passed through, is controlled in a mixer section (see below).



Articulate!

The oscillators may be what produces the sounds kantos makes, but the articulator is what sets kantos apart from other synths. Its job is to apply harmonic content and formant characteristics from the input signal to the sounds generated within kantos (i.e. by the two oscillators and the noise generator).

According to the kantos manual, the articulator's "amount" determines the degree to which the input sounds characteristics are applied, while the "Q" control sets the overall tonal character of the articulator's processing. Given those relatively general descriptions, and my observation on the very different sound qualities at the various extreme positions, I was curious about the individual contributions of the two parameters. Antares indicated that the Q controls the width or sharpness of the peaks and the amount controls the average peak-to-valley range. Looking at some before and after sound wave graphs seemed to bear this out.

Technospeak aside, though, perhaps the best way to describe their interaction is to suggest they work together similarly to how cutoff frequency and resonance controls work together in a traditional subtractive synthesis model. That is to say the effect on the sound comes from the interaction of these two parameters moreso than individual contributions from each of the parameters. While you can set the numeric values individually, Antares recommends simply dragging the "virtual ball" around until it sounds good.

On/off toggles are provided in the articulator to govern which kantos signal generators are affected by the articulator. Note, though, that these toggles are not to set which signals are heard in kantos' output -- that is done in the mixer. For example, if the noise generator is not affected by the articulator, kantos' output will have a pure noise component. On the other hand, say the noise generator is the only signal set to come out of kantos in the mixer, and is being fed by a vocal track, with values of somewhere around 70% for both amount and Q -- the result will be something similar to a whisper track, where the words come through reasonably clearly, but pitch information is lost since the harmonic characteristics of the sung line are being imposed on white noise. In fact, the effect can be enhanced using a highpass filter in the noise generator to take out some of the lower frequency information. Of course, if you'd prefer something more akin to a science fiction, Darth Vader-like "whisper", try a lowpass filter in the noise generator instead.

One potential wish list item for a future kantos release would be to provide separate amount and Q controls for each sound source. As it is, the settings for all three sound sources are identical, and the only option is whether the articulator affects each sound source or not.

The articulator also features a formant offset control, which I tend to think of as a "gender bender", though it has other uses. Technically, the formant offset control shifts the harmonic structure detected in the input signal up or down, which affects the timbre of the "articulated" sound. The "gender bender" notion is that this can be used to keep a sound, such as the human voice, more natural when pitch shifting.

For example, if you raise the pitch an octave (i.e. in the oscillators), without using the formant offset, you can get a chipmunk-type effect, but raise the formant offset percentage a bit, too, and male voice can turn into something more like a female voice. Do a similar thing in the other direction, and a tenor becomes a bass, a female voice gets manly, and so on. Of course, you can also do strange things like bring the pitch down while raising the formant offset control for "Darth Vader chipmunk".

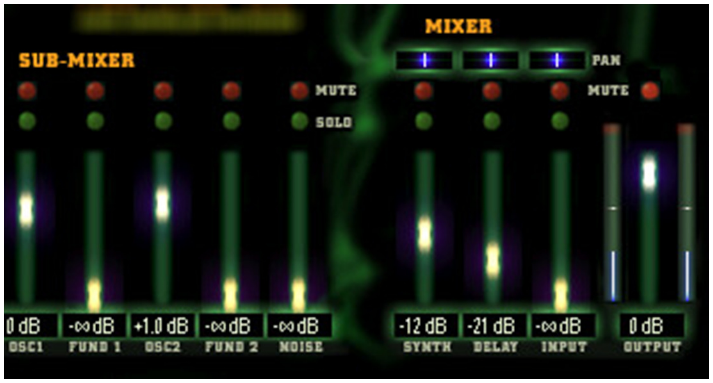
There's nothing saying this has to be use with vocal tones, either. For example, in my testing, I used a viola sound as an input with both oscillators being set to sawtooth waves.

When raising the pitch using the octave control, the output sound lost its bowed string-like, or perhaps "bowed synth-like" in this case, quality. Adding a bit of formant offset to the picture brought that quality back. By contrast, lowering the formant offset resulted in more of a muted sound.

The final component of the articulator is the emphasis control, which has separate settings for high, low, and mid components. For practical purposes, this is essentially a three-band graphic EQ. Due to the relatively extreme width of the bands, the emphasis control provides some fairly dramatic sound shaping abilities.

Mixers and Delay

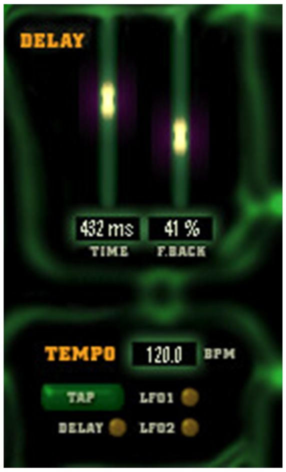
Okay, so now we've got an input audio signal being used not only for governing kantos' pitch and phrasing, but also to tell the articulator how to mutate the character of the synthesized or sampled signals coming from kantos' two oscillators and noise generator. There is still more that can be done to mutate the signals. Before we introduce additional complex possibilities, though, we need to get some sound out. To this end, kantos provides two mixers.



0 dB OSC1
-∞dB FUND 1
+1.0 dB OSC2
-∞dB FUND 2
-∞dB NOISE
-12 dB SYNTH
-21 dB DELAY
-∞dB INPUT
0 dB OUTPUT

A sub-mixer mixes together the basic signal components. In addition to the articulated outputs of the two oscillators and noise generator, kantos provides the option to mix in some of the fundamental signal from each oscillator. If used, this mixes in a pure sine wave of the fundamental frequency detected by kantos pitch detection circuitry, as modified by any tuning and pitch constraint settings, thus reinforcing the basic pitch, for example if the articulated sound isn't providing enough pitch information.

Kantos' mixer allows mixing the synth output (i.e. the output of kantos' sub-mixer) with the output of the delay section (see below) and the original audio input. These sources can also be individually panned (if kantos is inserted in a stereo channel or bus), muted, and soloed. There is also an output trim control available in case kantos' output is too hot. A clip indicator aids in setting the output level.



432 ms TIME
41 % F.BACK
TEMPO 120.0 BPM

The delay section in kantos provides a rudimentary digital delay with delay time and feedback controls. Quarter note delays can also be set by entering the tempo in beats per minute or via a tap tempo feature, which lets you click with your mouse on the beat to allow kantos to calculate the tempo. If you want a delay time other than a quarter note, simply use the tap button at whatever delay rate you want (it won't know you're not tapping quarter notes), or calculate an alternate BPM setting (for example, if the tempo is 120 BPM, but you want an eighth note delay, tell it the tempo is 240 BPM), or get out your calculator and figure out what time you want in milliseconds and enter that directly.

Unfortunately, the delay time does not sync to SONAR project tempo, nor can the DirectX version of kantos be automated to change tempos semi-automatically over the course of a song. If tempo automation is needed, it will be necessary to use the VST version of kantos to manual insert any tempo changes via automation envelopes.

Everything in Modulation

In the MIDI world, just playing sounds from a keyboard is often not very satisfying, at least for many non-keyboard and non-percussive sounds. The reason is that sounds that aren't designed to simply be struck once and let go later just get plain monotonous when they are played that way. MIDI module and keyboard designers, and the programmers who design sounds for MIDI keyboard use, compensate by either designing some evolution of the sound into the sounds themselves, which can tend to get predictable, or by letting the musician use MIDI controllers, such as the mod wheel and aftertouch, to tweak the sounds in real time, which can result in the need for unintuitive playing techniques.

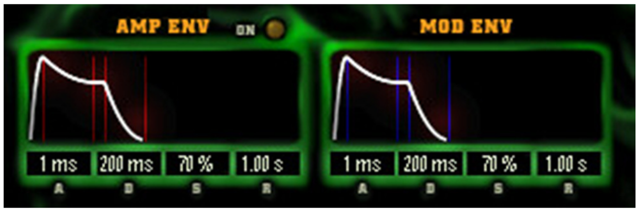
In the acoustic music world, sounds tend to naturally evolve over time, and that evolution is generally already under the control of the accomplished player or singer. Wouldn't it be great if kantos, which takes, or at least can take, its input from the acoustic world, could take advantage of that natural sound evolution to impart similar evolution on the sounds that emanate from it?

The good news is that kantos not only allows modulating the sounds it is making based on information from its audio input, but also provides a several additional modulation sources for those who want the best of the acoustic and synth worlds. The further good news is that kantos allows routing any of its seven modulation sources to one or more of its thirty-five(!) possible modulation destinations. If there is any bad news in this, it is that any given kantos setting or program can include a maximum of 8 modulation routings. We'll get to the modulation routing below, but first let's take a look at kantos' potential sources of modulation.

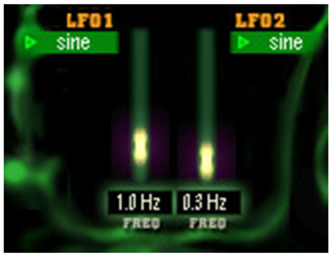
Starting with its audio input, kantos can use the input's pitch, dynamics, and timbre to modulate its sound. Certain potential routings are obvious and set by default. For example, it makes sense that input pitch would generally drive the pitch generated by kantos' two oscillators. But perhaps you would also like input pitch to drive other parameters, for example to serve as a "key tracking" type function such as found on many synths, or even to do something that might seem on the surface to be totally illogical, but which just happens to have creative uses in your particular project.

Care to let input timbre drive pitch instead? I got some interesting bumblebee-like buzzing textures by letting kantos do just that with a recording of a sampled viola part playing "Mary Had a Little Lamb". Okay, so maybe there aren't any constraints that limit choices to what is in good taste, but this is art, right?

Also based on audio input triggering, as described above in the section on the gate generator, are modulation and amplitude envelopes. The amp envelope can optionally be used to override using input dynamics to control the dynamics of the sounds kantos generates. When used this way, it acts as a traditional ADSR-style amplitude envelope as found on analog and other synths. When not engaged for this purpose, it is still available for routing to other modulation destinations.

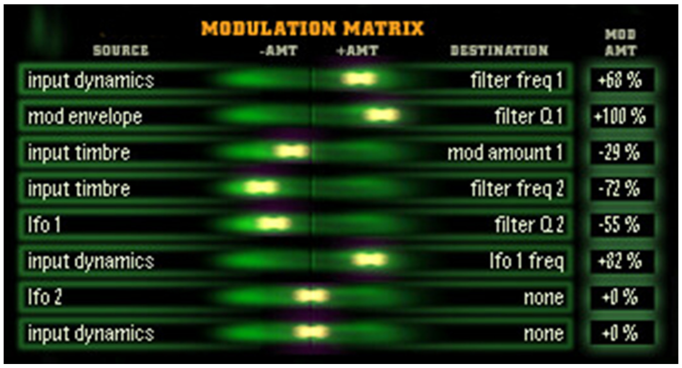


The mod envelope is only available for modulation routing purposes, and will have no effect unless it is routed to some modulation destination.


LFO1 sine 1.0 Hz FREQ

LFO2 sine 0.3 Hz FREQ

Finally, kantos provides two LFOs (low frequency oscillators) for use in modulation routing. The frequency of each LFO can be set independently, and waveform selections include sine, triangle, ramp up, ramp down, square, and random. Each LFO can optionally be synced to tempo, as described in the above section on delay. Note, though, that there are not independent tempo settings for each LFO, nor for the LFOs versus the delays. Thus, for example, if you want both LFO cycles to be every quarter note, but the delay to occur every eighth note, you will only be able to use the tempo sync feature for either the LFOs or the delay, and will have to calculate the frequency for the other.


MODULATION MATRIX

SOURCE	-AMT	+AMT	DESTINATION	MOD AMT
input dynamics			filter freq 1	+68 %
mod envelope			filter Q 1	+100 %
input timbre			mod amount 1	-29 %
input timbre			filter freq 2	-72 %
lfo 1			filter Q 2	-55 %
input dynamics			lfo 1 freq	+82 %
lfo 2			none	+0 %
input dynamics			none	+0 %

Okay, so now we've got seven potential modulation sources -- what can we do with them? Quite a bit, actually. There are thirty-five potential modulation routings. There are too many to list here, but suffice it to say that most, if not all, parameters you might conceive of modulating can be modulated, from levels to filter frequency and Q settings to pitch to articulator settings.

The modulation matrix section of kantos allows mapping any modulation source to any modulation destination, setting the degree to which that modulation source will be used to offset the value of the destination parameter in a positive or negative direction. For example, if the modulation envelope is used to modulate pitch, using a negative amount will modulate the pitch downward while using a positive amount will modulate it upward, while setting an amount of zero will result in no modulation of the pitch.

To provide even more complex possibilities, kantos allows modulating the modulation amounts themselves. If this seems daunting, it is. Unfortunately, the example given in the current kantos manual, doesn't help much either, because, though the scenario sounds logical enough -- using LFO1 to modulate oscillator pitch, then using the modulation envelope to modulate the LFO1 amount, thus slowly bringing the modulation in like a delayed vibrato -- it doesn't actually work. In this case, the LFO modulating pitch has its amount figure specify the depth of the pitch change, but instead of taking the depth from zero to the depth specified in that modulation matrix entry, the modulation matrix value for the modulation envelope is simply added to the depth for the LFO. The closest I could get to delayed vibrato was actually the inverse -- i.e. by specifying the opposite of the depth in the LFO to pitch setting (e.g. 20% in the LFO to pitch amount and -20% in the setting that used the modulation envelope to modify the modulation amount for the LFO to pitch mapping), I was able to achieve vibrato at the start of the note that turned off over the course of the attack time specified in the modulation envelope. Antares has acknowledged the error in the user manual, and intends to fix it. In the meantime, suffice it to say that this area goes pretty deep, and it may take some experimentation to understand what is going on with any particular modulation amount modulation.

The bottom line on kantos' modulation capabilities is that there is a good deal of power there for those who want or need it, though learning how to use that power may take some doing. I consider the modulation matrix to be something akin to a patch panel for modulation routings. That said, the default modulations based on the audio input source characteristics provide a fair amount of capability without having to dive deeply into mod matrix routings. That should let anyone whose eyes glazed over during the previous paragraph can breathe a sigh of relief.

Playing

For optimal predictability, you'll want to feed kantos a monophonic sound source (i.e. as opposed to a polyphonic sound source -- either mono or stereo is fine) with a decently strong fundamental. For example, your singing voice will likely work fine, but whistling may be tougher for some due to the heavy balance of unpitched noise (i.e. from the air flow) to pitched material (i.e. the actual whistle tone).

You can also feed kantos polyphonic material, percussive tracks, pure noise, and just about anything else that makes a sound, but the results will generally be much less predictable since kantos' pitch tracker will more likely be all over the place. Still, this may be just the thing for, say, coming up with an original bass line or rhythmic riff, for example by using the pitch constrain features to limit kantos' note changes to stay within a few strategically chosen notes.

For my testing purposes, I didn't have a project on my front burner that fit what kantos does, so I decided to have a little fun. I spent on the order of 5 minutes tracking a four part acapella version of Stephen Foster's classic "Oh, Susannah". It was truly just in fun, and not exactly well sung. You can hear just how non-special it was here (this will download a roughly 350KB MP3 file). I then spent some time playing around with running each part through kantos.

For the bassline, I shifted one oscillator downward two octaves, leaving the other, which called up a more bass-like tone anyway, as it was. I also shifted the formant down a fair amount, and added some fundamental into both oscillators. Pitch constraint was used to keep things within note boundaries, with a little glide added on to compensate. The amplitude envelope was used to go for short notes with a fairly rapid decay. The idea was to go for a tone somewhat reminiscent of an acoustic bass, though still electronic.

Two additional background parts used pitch constraints, as well. One was dropped an octave and shaped with filters to achieve something on the order of a gritty trombone-like function, while the other was raised an octave and given an amplitude envelope to get something akin to a clarinet. Overall, I was going for something of an electronic Dixieland feel, I guess.

As for the lead vocal, which is the only part I actually sang with lyrics, I decided to go for a somewhat robotic feel, letting the lyrics come through, albeit imposed upon synthetic tones vocoder-style. I also dropped this an octave, again using the formant shift to keep things slightly more natural -- er, for a robot singer anyway. This time I didn't use pitch constraints, and all modulation was left at its default values.

The resulting silliness can be found here (another approximately 350KB MP3 file). Okay, so it's not exactly an enduring work of art, but hopefully it will give you some idea of the types of transformations kantos can make with melodic material.

Of course, kantos doesn't have to be melodic. Using the pitch constraints to an extreme, for example constraining it to a single note or just a few notes (e.g. the root and fifth of a key signature), can provide for interesting semi-pitched drones with the right kinds of samples. And tying input pitch or input dynamics to something like filter amount or Q can result in some interesting sound evolutions that could be useful as pads.

One thing I discovered when playing with kantos is that there are two basic ways to use it. The most obvious one is what I've mostly been describing here -- i.e. record a vocal or other track then process it with kantos, trying to achieve some useful result. However, using SONAR's input monitoring, with kantos as a track insert, it is also possible to "play" kantos in real time -- i.e. set whatever sound parameters you like, then hum or sing or play an instrument into a mic through kantos on an input monitored track, so what you're hearing in your headphones is what comes out of kantos, rather than the original, unprocessed sound.

While the latter way takes some getting used to, especially if your headphones let a lot of room sound in, it can be fairly satisfying from a player's perspective. It offers the possibility of learning to control kantos as a musician would control his or her instrument. Once you start doing this, too, you will realize that what sounds good in the acoustic world is not always the best thing for driving kantos. Some very interesting things can be done by making sounds that wouldn't necessarily have much use in the acoustic world. For example, it can be fun to play with kantos' tone simply by changing the shape of your mouth while holding a tone or by varying the dynamics of the sound over time. Try doing that with an ordinary synthesizer!

And in the End...

It's probably reasonably clear by now that kantos is powerful, and provides a good deal of depth for those who are inclined toward synth patch programming. That's all well and good, but does it do anything useful for music making and other potential audio production applications?

The short answer is, "it depends." Remember the claim I mentioned seeing on Antares' web site? It included the phrase, "a software-based synthesizer that finally liberates you from the tyranny of MIDI, keyboards, controllers or, in fact, anything that stands between you and the music you hear in your mind."

That's a pretty powerful claim, and I can tell you now that, if the music you hear in your mind is, say, traditional bluegrass, no matter how expert you get with kantos, you aren't likely to agree with Antares' claim. If, on the other hand, your style is techno-bluegrass (is there any such thing? -- I'm imagining someone plays a banjo or fiddle and out come synth sounds that retain some of the characteristics of the respective acoustic instruments), kantos might be just the ticket to help realize your ambitions. If creative sound design is your thing, or experimental music, or most anything where the unusual is useful, kantos is definitely worth a closer look. For those who fall in between the extremes, kantos is worth checking out, too, though my sense is that kantos will not be as horizontally applicable as Antares' other products.

However you skin it, though, kantos is ground-breaking among synths. In particular, kantos fosters uniqueness by helping you avoid the "synth curse", where anyone can play the same patches, and, in most cases, many do. Sure, you can use the same kantos presets as anyone else. Even in that case, though, kantos helps you inject the one thing no one else has -- yourself, be that in the form of your voice, the way you play an acoustic instrument, or whatever else you may do to make sounds that are uniquely "you" -- into the sounds themselves. As a result, even the same kantos presets won't sound the same "in the hands" of any two "players".

With kantos, the sounds you make may not necessarily be the music you hear in your mind. With practice you will likely come closer on that count, but there is still a long way to go in making that brain-to-music connection a reality. However, kantos may well be the first synth where it is possible to honestly say that the sounds you make will be as unique as you are.

**Rick Paul is a songwriter living in Southern California. You can contact him at http://www.RickPaul.info.*