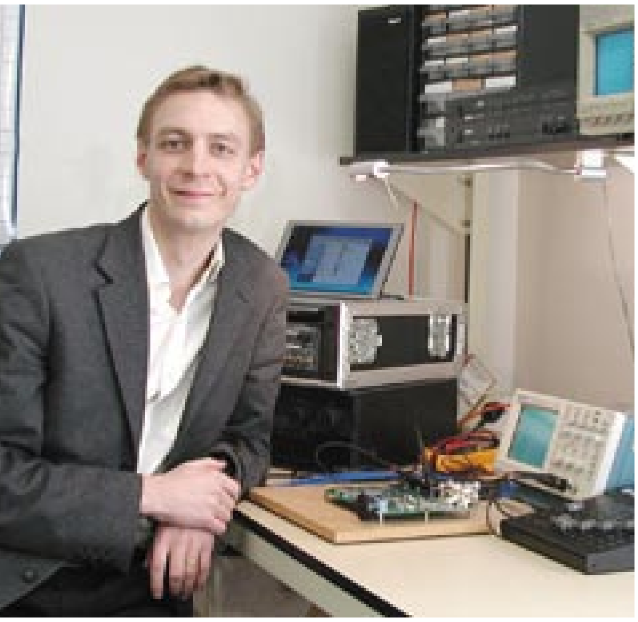
Christopher Hicks

CEDAR Audio's senior engineer talks about the technology behind the company's products, the algorithm refinement process, processing power hikes, and why he likes tea without sugar.

ZENON SCHOEPE



CHRISTOPHER HICKS HAS been associated with CEDAR since 1992 and has worked full-time there since 1996. Prior to this he was at Cambridge University doing a BA and then a PhD in the Engineering Department and maintains his Cambridge University connection by teaching mathematics and electronics to engineering undergraduates at Churchill College for a few hours a week. As a senior engineer at the company his main responsibility has been the hardware products — the Series X and X+ rackmount units, the DNS1000, DNS2000, the Duo declickle and autodehiss boxes, and also the timecode unit that forms part of CEDAR Cambridge.

He was a committee member of the British AES section for many years, and was its chair for 1999-2000. Together with CEDAR's Dave Betts he received a Technical Achievement Award from the Academy of Motion Picture Arts and Sciences in 2005 for their co-design of the DNS1000.

In his spare time he sings with a local chamber choir, admits to playing the violin badly as well as doing a fair amount of classical music recording — 'it's nice to use the kit once in a while, not just design it!' he says.

What is special about CEDAR products?

I hope that many things make them special, but I will concentrate on just one. At CEDAR we have, from the outset, been faced with the problem of making cutting-edge digital signal processing technology accessible to people who are not experts in DSP. This has implications both for how the algorithms are designed, and for how they are embodied as products. For example, on reading the academic literature one soon discovers that 'robust' algorithms are considered a good thing. This really means that the same algorithm will work well in a wide range of circumstances, is not particularly sensitive to changes in the audio, and does not require frequent parameter adjustments.

The practical upshot of robustness is that the exact setting of each control is not particularly critical and sometimes, as an algorithm is refined, it becomes possible to remove a control that one previously thought was needed. So, by designing our algorithms to be robust, our products become quick and straightforward to use, despite containing complex DSP.

From the days of the original DC1 declicker, which we released in 1992, it was a standing joke within CEDAR that the default settings (Medium 10) would work for all but the most demanding applications of the product.

How do years of producing algorithms manifest themselves in your products — are we still listening to the same original declick algorithm?

Absolutely not. The research that grew into CEDAR occurred in the late 1980s. It used FORTRAN code, running on 80286-based PCs, and would take several hours to process a single side of a 78 — the principal focus of that initial research. Since then we have gone through several generations of DSP chips, several programming languages, and several generations of PCs and the like, so CEDAR has been implemented on numerous different platforms.

The easy way to revise a product for a new platform is to re-code existing algorithms and move on. At CEDAR, however, we have used each of these occasions as an opportunity to re-examine the algorithms and refine them. Furthermore, each platform generally represents an increase in computational power over the previous one, and this can allow us to extend the sophistication of algorithms we had previously been forced to simplify for lack of memory or processor speed.

What are the common technical approaches involved when designing restoration algorithms and at what point do they start to differ?

Engineering is all about analysing problems, and solving them through innovation. A key component of that process is scientific method, where progress is based upon evidence gleaned from experiment. However, engineering is differentiated from pure science by the existence of the customer. A perfect technical solution to a problem is unacceptable if it is impossible or impractical for the customer to adopt it, perhaps because of the final cost or the nature of the product that would result.

Furthermore, defining the problem can be a challenge in itself. Faced with a customer who says, 'I have an analogue thingy, but my studio is going all-digital, so can you make me a digital thingy?' one could, of course, comply by creating a digital thingy. A deeper approach, and one that we regularly adopt, is to identify the problem the analogue thingy is being used to solve, and then to develop a solution to that problem; a solution that also happens to be digital.

This occurred in the early stages of the DNS1000 project, which was triggered by a request to produce a digital version of an existing piece of analogue equipment. By adopting the deeper design process, we improved substantially on the performance of the 'analogue thingy' and the DNS1000 consequently had much broader appeal than it otherwise would have had.

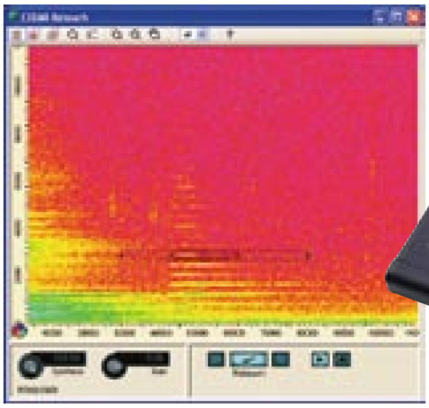
Which restoration algorithm is the most difficult for you to implement?

Broadband hiss removal. Let me explain.

I like my tea without sugar. Declicking is a bit like trying to fish the fragments of a badly-dunked Rich Tea biscuit out of a cup of sugary PG Tips; the individual bits remain reasonably intact, so if you're quick and careful you can get most of them out and the tea remains drinkable. Removing hiss is like trying to get the sugar out; a chemist could possibly devise a sequence of chemical reactions with nasty reagents that would do it, but would you want to drink the result? The cure may be worse than the problem. Now to get back to signal processing...

It is a well-established tenet of Information Theory that information is lost when two random signals (for example, music and noise) are mixed together. It is therefore impossible to separate them perfectly, so 'perfect' dehissing is not possible. (Nor, as it happens, is it necessarily desirable, but that leads us into a discussion about psychoacoustics, which is probably inappropriate here.)

To make a useful attempt at separating the wanted



signal and the noise, we rely on at least one of the signals having some inherent degree of predictability. This predictability might be embodied in a 'fingerprint' of the noise alone, or a statistical description of the wanted signal based on some assumption such as 'music is composed of a multitude of sinewaves'. Any deviation of the actual signal and/or noise from the chosen model (which there will inevitably be, the signal and noise both being random) could reveal itself as an undesirable artefact in the processed result. Choosing the models and refining them such that the artefacts are minimised is the complicated (and interesting) bit; finding a chemical reaction that removes the sugar without ruining the tea!

How reliant is the development of your restoration products on progress in processing power?

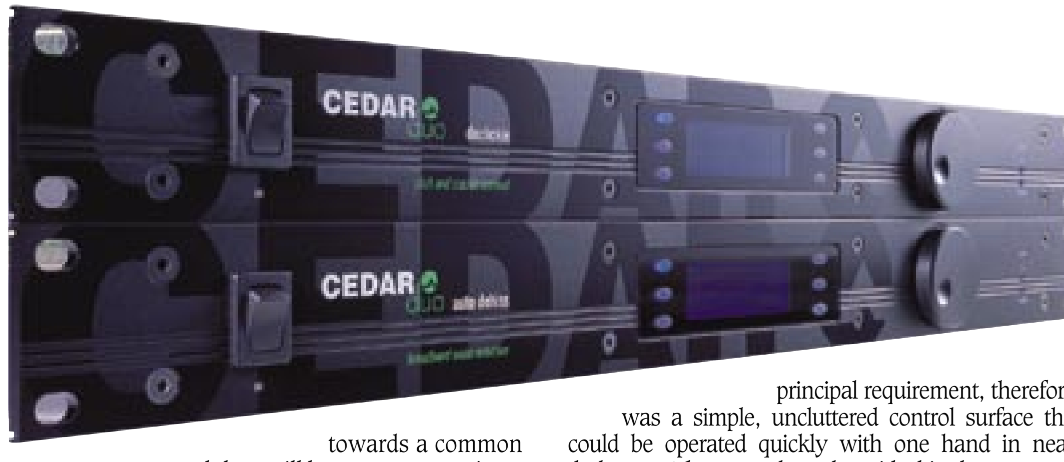
For technological reasons, CEDAR was later than many other audio equipment manufacturers to support the 88.2kHz and 96kHz sample rates. When you double the sample rate, simple algorithms such as EQ, mixing, and compression require approximately double the memory and double the processor power. In contrast, many of CEDAR's algorithms require up to about four times the processor speed to support a doubling of sample rate. We therefore had to wait longer before we could support these higher sample rates.

Having said that, it has only been 20 years since the aforementioned FORTRAN code was written, and it is remarkable to think that my laptop PC could process that 78 to a much higher standard in a matter of seconds!

Where are the limitations and bottlenecks in the technology you currently use and what advances will herald the next step up?

Humans (particularly males, I am told!) are poor at thinking about more than one thing at a time. One consequence of this is that we naturally break down solutions to problems into sequences of steps, one to be executed after the other. In the computer world, multiprocessor and multicore machines are rapidly becoming the norm, and to utilise these effectively requires a different approach in which we think about which steps can be executed in parallel with, and independently from, each other.

If these independent processes are working



towards a common goal they will have to communicate and synchronise with each other at certain points. This creates another set of problems, exemplified by the deadlock; four cars arrive at a mini-roundabout simultaneously, and everybody waits for everybody else.

The situation is exacerbated when one tries to make a real-time audio application co-exist with a general-purpose operating system and other software. If the operating systems in general use had been designed from the ground up with real-time applications in mind, it would make life a whole lot easier for real-time software developers. As it is, we all have to go to extreme lengths to ensure that interruptions to the audio stream do not occur.

The DNS1000 has achieved incredible success in post yet it is unusual in your product range as a standalone product type. Can we expect to see similar 'console-top' type processors?

The reason that the DNS1000 is console-style has nothing to do with the requirements of the signal processing, and everything to do with the way our customers want to use it. The point is that each of our hardware products is a complete design, and not just an algorithm in a box; we take a great deal of account of the environment our customers will be working in, and how our products will best fit into their workflow.

The main market for the DNS1000 is in postproduction, where time is always of the essence. A principal requirement, therefore, was a simple, uncluttered control surface that could be operated quickly with one hand in near-darkness. The console-style with big buttons and faders fits that requirement well. We have other hardware products that are based more upon a 'set-and-forget' approach, and a rackmount format suits those better.

Why did you release CEDAR for Pro Tools on PC rather than on Mac?

At the moment, the PC environment has proved to be better for hosting and supporting CEDAR's algorithms and products. For the future, we are keeping a close eye on the rapid changes occurring in Apple's product lines and operating systems, although I would be loath to speculate where these might lead us.

How long-term is the business of restoration?

The supply of vintage recordings needing restoration is large, but finite, so I suppose an end to that business is inevitable, if a long way off. However, modern recordings also suffer from similar problems; electrical interference, equipment malfunction, operator error and other 'technical hitches' cause all manner of unwanted noises and distortions, quite apart from the background noise that is inevitable on location and in environments such as concert halls. Consequently, the need for audio restoration will never dry up; quite the opposite in fact, as increasing amounts of material are recorded and broadcast under greater pressure and on shrinking budgets.

And then there's the matter of forensic audio investigation… but that's a story for another day.