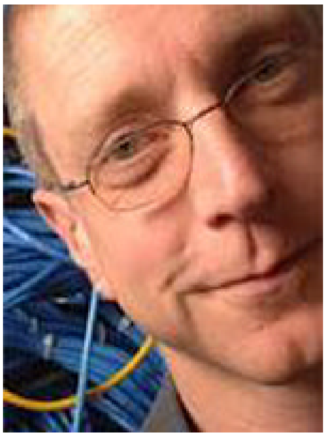
HFT Review
High Frequency & Algorithmic Trading

Man versus Machine, Part Two

The second part of a two-part interview with Professor Dave Cliff
By Mike O'Hara, 11th April 2012



In this second part of a two-part interview for the High Frequency Trading Review, Mike O'Hara concludes his conversation with **Professor Dave Cliff** of the University of Bristol. Professor Cliff is Director of the UK Large-Scale Complex IT Systems Research Initiative, and a member of the Lead Expert Group of the UK Government's Foresight Project on The Future of Computer Trading in Financial Markets.

HFTR: *Dave, welcome back. In the first part of this interview, we discussed the background of computer-based trading and the use of artificial intelligence and machine learning in financial trading systems and we looked at how humans and machines might interact in the financial markets of the future. Now I'd like to get your insight on some of the emerging technologies that are starting to become prevalent, starting with cloud computing.*

DC: Cloud has gone from being a phrase that nobody knew, to a kind of bandwagon buzz word, to something that is now obviously here to stay.

If you look at the history of commercial provisioning of ICT, roughly once every ten or twelve years there's a major shift. Way back in the dawn of the industry, if you wanted a computer, you had to buy what was essentially a single user mainframe from IBM. The dominance of the mainframe lasted roughly a decade and then IBM's incumbent position was threatened by new arrivals, new companies that saw new opportunities. In particular there was this small start-up company called Digital Equipment Corporation (DEC), who realized you could take everything a mainframe was doing, but using new technology in both hardware and software, you could deliver that for much less cost and in a much more manageable form. DEC did very well with the dominance of mini computing for about a decade, but then along came micro computing, i.e. single board computers, based on single chip microprocessors, one per desk. That was very disruptive and threatened both IBM and Digital, when companies like Compaq (who eventually bought Digital) rose to prominence. Then people realized that you could economically network the PCs together to get local area networks, client-server architectures, that kind of thing, and that lasted for about a decade.

HFTR: *Then along came the Internet…*

DC: Actually, the really major evolution was the Web, not so much the Internet. The Internet had been around for 15 or more years before the Web took off - the Web was the recognition that if you have all these local area networks and they can talk to each other, then in principle anyone can talk to anyone, any computer can talk to any computer anywhere in the world, so long as you agree the protocols: http and html.

So we're due one of those big revolutions in ICT and I'm happy to bet a couple of bottles of champagne that cloud is the next one because it offers the opportunity to dramatically reduce the cost of computing. Interestingly it means that what was previously a capital expenditure item becomes an operational expense. You only pay for that which you need, so you don't have to worry about dormancy.

HFTR*: How relevant is cloud to high frequency trading?*

DC: With HFT, one of the issues that is probably never going to go away (unless they legislate it away) is the need for low latency. And there is always going to be a latency issue in cloud because by most reasonable definitions, cloud computing involves a service that is remotely provisioned and accessed over some transmission network. Which means that actually having HFT algorithms running in the cloud, in a huge server farm at some indeterminate position on the planet, is unlikely to ever happen.

We have seen the move to co-location and proximity servers and people starting to talk about the prospect of exchanges actually hosting the algorithms in virtual machines on the exchange server rather than servers connected by a standard length of cable because of that need for low latency. In that sense, superficially at least, cloud is not relevant to HFT.

However, the question you have to ask is where do these HFT algorithms come from and on what analysis of data are they operating? That's where cloud comes in.

HFTR: *In the development and testing of algorithms?*

DC: Yes, because with cloud computing it becomes possible to do mammoth-scale data mining or data analysis, to take pretty much all the data that you can get your hands on and crunch through it. The cost of doing all that crunching is greatly reduced by cloud computing. For example, if you're running statistical arbitrage, then essentially what's going on is that you have some sort of model of what statistically counts for normal, you detect when events are abnormal and expected to return to normal over some timeframe, and you try to trade on that basis. That model of normality is going to come from data analysis that can be very compute-intensive, which is something that can be done in the cloud at very low cost.

Similarly, there's a lot of interest in automated optimization of trading algorithms. This often involves what is essentially a Monte Carlo approach, where in order to do the evaluations, you need to generate lots and lots of examples of either different trading strategies and/or different scenarios that your trading strategy might be confronted with. That's inherently very compute intensive but it is also inherently parallelizable, i.e. if it takes a hundred days on one computer, it will take one day on a hundred computers or a tenth of a day on a thousand computers. With the economics of cloud computing, you can hire one computer for 100 days, or you can hire 100 computers for one day and it costs the same.

The great thing about cloud is that offers all these high performance computing optimization, data analysis and data mining facilities that previously, just a decade or so ago, would only have been available to really major institutions such as very large fund management companies, hedge funds or investment banks. In that sense, what cloud is doing is lowering the barriers to entry. The kind of compute power that you would only have found in the biggest institution, you can now essentially rent from a bookshop called Amazon or a web advertising firm called Google. That means in principle at least, three or four people in a garage as a startup can perform the kind of advanced data mining and advanced machine optimization that previously only major institutions could.

So my take on cloud is that at the coalface, at the sharp end of HFT, you're unlikely to see it, but the things that are running the HFT algorithms and the data models that they're based on are quite likely to have been optimized in cloud systems.

HFTR: *How do you see that changing the participation of different players in the industry? Would you see a shift away from the big banks to smaller prop trading firms and hedge funds for example?*

DC: It's not necessarily going to cause a shift but I think it levels the playing field. A decade ago, the larger the institution and the more money it had available to spend on IT (or indeed the more PCs it had in its office building that could be shackled together as a kind of desktop grid for running overnight optimization programs), the better. If you had just five people in a garage with five PCs, you couldn't really hope to compete. But now, as long as you have a credit card that Amazon recognizes, you can hire very large amounts of computing power, just as much as you need. In that way, it levels the playing field; it opens up opportunities for smaller and potentially more agile companies to enter the market.

HFTR: *What about the issue of control? If you are doing everything yourself on your own servers in your own data center, then you have full control over the environment you're working with. But on the cloud that's not necessarily the case because the applications are being run – and the data is being processed and managed - by an Amazon or a Google or whoever it might be. How much of an issue is that?*

DC: Certainly, a quick way of giving a compliance officer a heart attack is to tell him that you stuck everything on the cloud!

Obviously there can be jurisdictional issues. If you put data in the cloud and you don't know where the server is, then you might find yourself liable to obligations under jurisdictions that are different to the one you currently operate in. For example, there are various bits of American legislation that essentially say, if data is held on a server situated on American soil, then that data is searchable by American law enforcement agencies. Now while it may well be that a good law-abiding company has nothing to worry about if American law enforcement agencies see their data, that company may have contractual obligations to its clients to ensure that doesn't happen.

However, the major cloud providers woke up to that some time ago. Certainly Microsoft, Amazon and Google offer you assurances on the geography of where your data will be stored. If you happen to be a European company and you want your data only to be subject to European law, then you can strike a contract with a cloud provider where they promise that the data will only be stored on servers based in Europe; or perhaps the promise is that the data will never find its way to North American servers.

So yes, there are jurisdictional and disaster recovery and availability issues but in principle they're no different to the kind of issues that you find in medical records systems or indeed in traditional non-cloud banking systems, they don't necessarily present technical or legal challenges that are unique to cloud computing.

HFTR: *The next topic I'd like to ask you about is hardware-accelerated trading. It seems as if every event I go to these days, people are talking about FPGAs (Field Programmable Gate Arrays). There seems to be a real buzz around that topic in the HFT space, particularly in areas like market data feed handling and pre-trade risk checking. But I'd be very interested to hear your thoughts on how you see hardware-accelerated trading evolving.*

DC: Well, I've had a wry smile on my face at the transition to FPGA that seems to have happened in the last 18 months, purely because when I stood up at TradeTech about five years ago and said that everyone should be doing this stuff on FPGAs, I was met with a room full of blank faces! But I guess the big payoff is in predicting what is needed tomorrow, not predicting what is needed in five years' time.

There has been this realization that if you use cheap commodity hardware, it means you are basically buying a bunch of circuit boards and logic gates that don't actually need to be there, so in principle you could strip those out. And one way of stripping them out is to do it via FPGA programming.

Now, field programmable gate arrays have been around for the best part of 20 years and the technology is well understood. Nevertheless, it's a pretty tricky task to go from the specification of some sort of trading procedure or algorithm to a description of how you wire that into an FPGA. If you think of an FPGA as a big bag of logic gates and a bag of wires, and all you do is connect the logic gates together with the wires, the nice thing is if you get it wrong you can re-program the connections. Nevertheless, going from a high level description as an algorithm or as a set of equations to what is essentially a circuit diagram for logic gates is a very skilled task and the number of people who have that skill is actually quite small. In fact, I'd say probably the primary limiting factor in the uptake of FPGAs is that they are not at all easy to program.

Translating from a high level description of what the thing has to do to the actual hardware level is something that requires skill and judgment on the part of a trained engineer.

HFTR: *So do you expect to see a big growth in tools that enable you to take high level code and translate it into something that the silicon can understand?*

DC: It's a little like the old days of assembly language, it was a pretty opaque thing but then in time people developed high level languages. The move to software-defined silicon is I think comparable to the development of interactive programming languages, or program languages that were intelligible to people rather than custom-built for machines.

There's a British company called XMOS for example, which has developed an approach to reconfigurable or redefinable silicon, where essentially people can write in high level languages and those high level languages are then compiled down on to something that's similar to FPGA but actually much more sophisticated, it's a kind of reconfigurable many-core processing chip, but any one of these chips can be connected to similar chips to the north and south and the east and west of it. Then a collection of chips that are all connected together become like one "super chip" called a systolic array, which you can program at once. And where currently the development time for creating custom silicon with FPGAs is days or perhaps weeks, if you do it with software-defined silicon, then that offers the promise of doing it in hours or perhaps just minutes. At the end of the day you've still got a chip that embodies the thing that you previously had a whole computer doing.

In terms of high frequency trading, the latency on an FPGA is probably going to be – for a while at least - quite a lot better than the latency on a software-defined silicon chip. But the key thing really is that people can reconfigure an XMOS systolic array much more quickly than they can reconfigure an FPGA, and the array can do an awful lot more heavy processing. So from a development and maintenance perspective, they're much more attractive for some applications.

HFTR*: How extensively do you think these technologies will be used in HFT?*

DC: Pretty extensively. FPGA and software-defined silicon are examples of technologies that offers a competitive advantage to the early adopters who are in the minority, because they've got a weapon that no one else has got, but what we can expect (because it's happened with every other technology innovation) is that eventually everyone will have them. At that point, you won't be able to compete without them, but just having them won't actually offer you an advantage, it will just bring you up to the level of everyone else. So then the question is, what's the innovation after that?

HFTR: *How about adaptive, self-learning genetic algorithms? I know you've done a fair amount of work in that space…*

DC: Well, I do have a bit of history in these areas. Immediately after I finished my PhD (over 20 years ago now) I started work with a couple of colleagues at the University of Sussex on applying genetic algorithms and self-learning mechanisms to the design of autonomous self-governing, self-managing mobile robots, and it was that work that caught the attention of Hewlett Packard Labs, who invited me to spend some time as a visiting academic there and that was where I invented the ZIP trading algorithm that we discussed in the first part of the interview.

Shortly after leaving HP, when I went to work as an associate professor at MIT in 1997, in the Artificial Intelligence lab, I applied genetic algorithms to optimizing ZIP traders and I discovered that the genetic algorithm could produce a better ZIP trader than I ever had. So I managed to make myself redundant!

Although that was seventeen years ago, there still is some resistance to the use of self-learning algorithms and automated optimization, but it's important to realize that genetic algorithms are just one form of automated optimization and there are lots of other forms, many of which are more efficient or faster. The key thing is that because you devolve the task of designing the algorithm to a computer process, and because that process is very often dependent on random trial and error and evaluating tens of thousands (or perhaps even tens or hundreds of millions) of different alternatives and then finally spitting out an answer, the problem you face is that although the algorithm it gives you may well be the best algorithm the computer has found, when you ask how it works, the computer doesn't have an answer because it didn't actually go through a deliberative intellectual design exercise, it just flailed around at random until it found something useful. That means that the problem anyone faces if they want to run one of these algorithms live in the market is determining the risk profile and how to do quality assurance, which is really quite a big issue.

Similarly if you make something that learns, something that's adaptive, and put it into the market live, the danger is that it might start by making you lots of money and doing good things, but then it might, in time, learn bad habits. And the time it takes to learn a bad habit in high frequency trading could actually be very short. It could be that the system has been working well for three days and then in the space of 20 minutes, its learning goes wrong and it adapts in such a way that suddenly it's putting you in a very difficult position.

HFTR: *Could you give me an example of what that might look like?*

DC: Well, we all know about the big flash crash event of May 6, 2010, but since then there have been a number of smaller flash crash style events in different securities and in different markets. Off the top of my head I can think of events in gold, silver, oil and natural gas. And only the other week, BATS had a little bit of trouble, not only with the IPO of their own stock, but some other stocks like Apple didn't fare so well on the BATS exchange either.

Now, one of the nice things about human traders is that they have a sense of proportion, a sense of "that number looks sensible" versus "that number is absurd". But it's hard to program that into a robot. If you make a robot adaptive, you're effectively saying to it, "you need to learn from your experience, I can never tell you exactly what numbers you're going to deal with so you just need to look at what's going on in the market and if it's been going on for long enough, then treat that as normal and trade according to that sense of what a normal price is".

That robot could then get itself into trouble if there's a flash crash. If it's adapting in a very high frequency timescale, if it's adapting over minutes or seconds, if there's an instrument that normally trades at $50 that suddenly goes to 5 cents but then because of a market failure it's trading at 5 cents for three or four minutes (or perhaps just three or four seconds), you can imagine a learning algorithm thinking, "yep, that's normal", because it's been going on for four seconds. It may then engage in the market even when it's clear that the sensible thing to do is just to stay out and wait for the system to recover.

However, none of this is insurmountable. Fundamentally, what we're talking about are automated control systems for some engineered artifact, and we already have those in other industries. We have autopilots that fly aircrafts full of people around the world. In military applications, a missile is basically an autonomous killing machine, and clearly the engineers who are responsible for building air-to-air or surface-to-air or air-to-ground missiles have to get the control systems right because if they get them wrong, they might end up killing the wrong people.

There is actually a whole mathematics of control theory that can be brought to bear here. In principle, you could take a machine-designed high frequency trading algorithm or system and subject it to some sort of control-theoretic analysis in order to understand the limits of its ability or the nature of its learning and you can then profile that against the market.

One way of doing that is just to run shedloads of simulations - which brings us back to using cloud for Monte Carlo evaluation - but while that has a lot of promise and often is actually the path of least resistance, one of the things that we're working on in Bristol for example, is applying analytic approaches, i.e. formal proofs where you can express your trading algorithm as a set of equations, and using tools from control theory, chaos theory and an approach called Lyaponov Exponent Analysis, identify whether the system composed of your trading agent and other trading agents is going to be stable or not. This work is being done by my PhD student Charlotte Szostek, I'm one of her advisors and her other advisor is a mathematician, Krasi Tsaneva. I'm really excited about what Charlotte is doing.

But until those tools and techniques are better developed and more widely understood and used in the industry, there's going to be a slight suspicion of anything that hasn't been designed by a human being who ultimately can be fired or put in prison. So it's one of those instances where the technology is there but - for all the right reasons - the industry has perhaps not picked up on it yet because you need to make sure that your risk and compliance is lined up with the kind of technologies you are deploying.

***HFTR:** That takes us neatly into a question sent in by one of our readers. He wanted to get your thoughts as to what extent HFT algorithms require steering inputs, either from the trader or from other algos, as they trade. Typically how adaptive are they to movements in a particular trading session for example? (And apologies to the reader if I haven't paraphrased this correctly).*

DC: What I would say in response to that is I know of some institutions that operate what's essentially a kind of "algo of algos", so you have a set of different algorithmic and HFT strategies and then you have a sort of executive, which is often an automated system. So there is a program that basically looks at the market and decides which of its algorithms is going to be playing in the market at any one time.

It's a little bit like a football manager, you've got some algorithms out there in the field playing and others on the bench, and the question is, "what's the best team I could have out there at the moment, given all the options that I have available to me?"

I don't actually know of anyone who dynamically reprograms their systems as the trading day progresses, just because programming involves human heads, which operate at speeds somewhat slower than the market changes nowadays.

HFTR: *Okay, to wrap up, what you can tell me about the work you've been doing on the UK Government's Foresight project (The Future of Computer Trading in Financial Markets)? I know your lips are sealed about much of it, but I'm hoping there is something that you can share…*

DC: The thing to say about Foresight is that when the UK's Government Office for Science runs these Foresight projects (and this is the fourteenth such project that they've run, but the first one that's ever looked at the financial markets), they always operate on roughly a 24 months schedule, where they spend around a year gathering evidence and another year digesting that evidence and coming up with potential policy recommendations.

For this project, the first year of evidence gathering ended last September, when they published all of the evidence that had been gathered thus far, which was seventeen reports - about 450,000 words of evidence - produced and peer-reviewed by leading academics and industry practitioners from around the world. The response to the publication of that evidence base has been almost entirely positive, which has been good to see, but the interesting thing is of course that at the same time, there's the Dodd-Frank Act happening in the States and there's MiFID II, MiFIR and the MAD (Market Abuse Directive) reviews happening in Europe. So in addition to being forward-looking and talking about what might be happening in five or ten years' time, it's been quite interesting to see that the kinds of things coming up in the evidence are relevant to legislative issues being discussed throughout the industry and by politicians around Europe, as we speak. I'd like to think that the Foresight project work has fed into those current discussions to some extent.

HFTR*: When can we expect to see the final report and policy recommendations?*

DC: The full report is due out later this year in autumn, but I don't know that a date has actually been set yet. Speaking personally, I've enjoyed working on it and found it really fascinating and stimulating.

HFTR: *I look forward to reading it. Dave, thank you for two such insightful conversations.*

