Interview with David Forney
June 5, 2013
Cambridge, MA

HW: *Thank you for talking to us. The goal of the interview is to get your views on how to do research well, with the hope that it inspires young information theorists. We will discuss five to six topics that correspond to different aspects of the research process in information theory. The first topic is motivation. Could you share a particular example of your work that was initially motivated from practical needs or problems that resulted in good theory, and vice versa?*

DF: That is pretty easy for me. As you know, in my thesis I worked on concatenated codes, which at the time was purely an issue of performance versus complexity, trying to get exponentially decreasing probability of error with only polynomially increasing complexity of decoding. I certainly remember interviewing at Bell Labs after doing this work and putting up codes that were thousands of digits long, and people were rolling their eyes and thinking "This guy isn't very practical." But to my surprise, ten years later people were actually implementing concatenated codes in space communications. This is one example. The Viterbi algorithm is another excellent example. Andy Viterbi had no idea that it was even optimal, much less practical, since he invented it as a proof technique. My contribution was to say, "Andy, this is actually optimum," but I had no idea it was practical either. It took other people at Linkabit, Jerry Heller in particular, to say "You know, I think we could actually build this." This became the main business at Linkabit Corporation, to both my and Andy's surprise. As far as practice to theory, my career was in industry so I have plenty of examples of that. My whole work on convolutional codes was motivated by the fact that we were building these things at Codex. They were clearly practical and more implementable than block codes but there was no satisfactory theory to them, so that's what motivated me to look for theory. Another example was trellis codes, which was what really got me back into information theory after being in management for ten years. I was involved in the early work in that and observed much of the development. These then led to coset codes, lattices, and so forth. It is immensely productive to go both ways.

HW: *Great! Let's continue to the next question. It has been argued that truly great work should be shocking and perhaps even controversial. Do you think that the field has developed such that these views are outdated? Or are researchers not taking enough risks?*

DF: I don't agree with your choice of words of "shocking" or "controversial" since I can think of very few IT papers that have ever been shocking or controversial. What would you call Shannon's paper? It was remarkable and immediately attracted a lot of attention. It was extremely well written. It formulated some novel ideas, and proved some great theorems, at least to the satisfaction of the engineering community. I think the great papers, and there haven't been many as great as Shannon's, have attracted a lot of attention not because they were controversial but because they have a certain rightness to them. As you read the paper, you find yourself nodding your head saying, "That's right! That's an insight that I may have had a corner of, a piece of the puzzle, but the author has nailed it." So that is what is characteristic of the great papers of our field. Many of them are well-written as well.

HW: *So for this question, what we thought was that really new ideas are shocking because they provide insights that others have not thought about before. Do you still see that kind of work in this field?*

DF: I think we have very few revolutionary papers in that sense. We have papers that use new tools or find new concepts, and these are the papers that tend to win prizes. But certainly there has to be an element of novelty and rightness. It has to be the right tool or concept for the problem, and furthermore lead to some nice results and insights that we did not know previously. Perhaps you use shocking in a different way than I do. Maybe to a graduate student who has learned to do things one way, seeing a totally different perspective is perhaps shocking.

HW: *It was "shocking" when I read Shannon's paper because young engineers are so used to thinking about achievability. The concept of a useful converse was very different and perhaps a little controversial because I am sure people complained about the model being so simple, that the abstraction can't lead to the best engineering principles, but we now see that it does.*

DF: I didn't have that feeling at all. I read it as a graduate student, and had no idea whether it was a realistic model at all. It turns out it was a realistic model for communication channels, but not for communication sources, which are usually not memoryless. So turns out it was more applicable to channel coding than to source coding. In the paper you cannot tell that since there were equally nice theorems on each. There was controversy about whether the mathematics really proved these theorems or was just hand waving. I think it's long behind us at this point. There was also controversy about his definition of information so to exclude all semantic content, but I think a lot of people said that that was very helpful, and in fact cuts through a lot of stuff that we as engineers don't really want to deal with. That's the right way.

HW: *We know there is a huge crowd for network information theory. Do you think that huge crowds mean that the problem is important? Do you have any advice for young engineers who work in a crowded space?*

DF: My own personal instinct when I see the crowd rushing to one side of the boat is to tend to stay on the other side of the boat, but I am at a very different stage in my career. To some extent you certainly have to participate in the field and do what other people are doing to achieve recognition. If you can cut off a piece of a well-trodden field and do something interesting with it, then you have really achieved something. I tend to think this is the hard way to do it. It is much easier to take something that the crowd is not working on, and that has been characteristic of all things that I've done. I have never had the feeling when writing a paper that there are ten other people who would come up with the solution if I didn't write it down quickly. I always felt that I have infinite time to write papers. Get it right, spend a year, whatever it takes, and then put it out. But that was in part because I wasn't a professor; I was in industry and had all the time in the world. I had my own problems that interested me, and that turned out to be a great way for me to go. I can't say I would recommend it to everybody.

HW: *Your answer reminded me of a question I was going to ask in a later section. If there isn't a huge crowd that has interest in the problem that you are solving, how do you sell the new and big ideas to the community?*

DF: That's a good question. Looking back, most of what I have done, which I'm not sure is applicable to everyone, were recognized immediately as interesting. I spoke about the Viterbi algorithm. To prove that it is optimum is obviously interesting. To show it is optimal for intersymbol interference, people immediately saw that. I would say with concatenated codes, people did not see it immediately. The issue of performance versus complexity was not something that a lot of people were working on then. There was actually more interest in Russia than in the US. In fact, it never became a big research topic;

it was just incorporated into practice. I would say that a lot of my more recent work on system theory of groups has turned out to be of interest to very few people. I think it's respected as respectable work, but it certainly hasn't taken off. But, at this stage of my life, that's OK. I can work on what is interesting to me. It's of course more satisfying that everyone reads it and says it's great but sometimes that happens and sometimes it doesn't.

HW: *Now that we have discussed motivation, we will now to move to problem formulation. As an engineer, we need to devise a system model to work with, so could you describe good criteria for a system model?*

DF: Most of my models came directly out of practice. The deep space communication channel is a pristine AWGN model, which is very nice. The telephone channel is almost as nicely modeled. It's a linear time-invariant channel and you can start out assuming it is Gaussian noise. You run into minor perturbations but you can engineer through them. So picking good models has not been a big issue for me. I think that it's more of an issue in the network setting, where as I said I don't feel I have as good a sense for it. As Bob Gallager always said, the essence of finding a good model is to abstract the important elements of the problem into a form that is simple enough where you can think clearly about it and hopefully get some nice results. That's easy to say, sometimes not so easy to do. But that's what you are aiming for. You don't want to abstract the problem so much that the results you get lose their relation to reality, but you want to abstract it enough so that it's a tractable problem. In information theory, we don't so much like these papers that say "Well, I set up the following system and I modeled all the different aspects and here is the probability of error versus signal-to-noise ratio, and it's really hard to tell what relevance that has to any other problem that is even slightly different from them." So there's an art to this. It's one of the harder things for students to learn. That's why students have professors and mentors. It helps a lot to have had some practical experience to understand the difference for practice, particularly if complexity is supposed to be part of your model. What is complex and what isn't is sometimes very hard to tell as a student until you have gone out and coded some software. Finding a relevant performance metric can be a problem as well.

HW: *You mentioned overcomplicating models and developing simple but mismatched models. Which one is more dangerous for a young researcher?*

DF: I don't think either is really dangerous, but the work won't have much impact either way. If the model is too simple, then no one will pay attention because they won't think it's relevant to their problem. If it's too detailed, no one will pay attention either. There's obviously a happy medium.

HW: *At the stage of problem formulation, do you have some insight on how to solve the problem? Or should the problem formulation be independent of whether the problem is solvable or not?*

DF: It's often a recursive process. So often in my life, the problem formulation came out of working with real systems, understanding something about them, and then asking what more fundamentally is going on here. For instance, I remember Gottfried Ungerboeck announced trellis-coded modulation and immediately the whole modem community became terribly interested in this. A thousand flowers bloomed and many people suggested different kinds of trellis modulation. After a while, particularly while working with Lee-Fang Wei at Codex, I said "Aren't we always talking about lattices and cosets here," and Lee-Fang said "Yes, we are." This led to some of the papers that I wrote on coset codes and the fundamentals of trellis-coded modulation. You had something that you saw worked well and obviously had some mathematical structure. What is that structure? It turned out to be more group structure than linear structure. That certainly motivated the work I and others did on codes over groups, system theory of group codes, that sort of thing. It comes out of looking at real systems, how they work, and asking what is going on fundamentally here.

HW: *You had good industry colleagues that understand the practicalities of the system you are developing. Do you try to get advice from practical engineers to understand the systems model better? When is the right time to communicate with these engineers?*

DF: In my experience, as I said earlier, we were dealing with in general clean enough channels that we did not require a great deal of guidance from practicing engineers. I guess one counterexample to that is in the very early days of modems. The first 9600 bit-per-second modem was a single sideband modem and, when it got out on the telephone network, we found it was having unanticipated problems which were tracked down to something called phase jitter. The carrier phase rotates by a few degrees on the telephone network and this is very troublesome for single sideband carrier. Maybe Bob Gallager talked to you about this, but he realized that if you used quadrature-carrier double sideband, the phase jitter would come through coherently in fact, so perhaps you could deal with it a lot better. Those insights are what led to my putting together a QAM modem as our second-generation modem, with a signal structure that was engineered to be very resistant to phase jitter. It didn't even need a phase tracker. That modem eventually became the international standard. So that's certainly one case where feedback from the field was very important. Similarly, in trellis-coded modems we got some feedback about how they were working, but basically by then we had a good enough model of the channel that there were no surprises about how they worked.

HW: *Now we are moving to the problem solving section. From your experience, how long does it take for a researcher to discover his most comfortable style of problem solving?*

DF: I don't have a good answer for that question. I always said that I just followed my nose. I didn't conceive of myself as having any particular style of problem solving. You have a problem, you throw whatever you can at the problem. This may involve talking to people. It may involve going to the library and reading a book. It may involve just lying on your back and thinking about what is important here. To this day, I don't think I can identify my style. I don't have a cookbook to solving problems. It just involves persistence, sometimes reading, sometimes discussion, but most times just thinking.

HW: *It seems like many subfields of information theory, you see systematic thinking. For example, network information theorists often begin by using a powerful lemma to develop new tools or insights to solve their problems. Do you employ this kind of systematic thinking?*

DF: In many fields of science, you have certain known techniques that you tend to apply. I remember that once in my paper on exponential error bounds for convolutional codes, I tried to systematize how you would develop such bounds by writing a cookbook. To the best of my knowledge, no one has ever used the cookbook, including me. Systematic thinking hasn't been important in my life. I'm generally interested in the front end of a field, where you don't even know it's a field. It's simply some mess out there that you try to make sense of. However, it's certainly true that a developed field consists of an established array of knowledge, techniques and tools that you can use to attack problems, and so network communication theory has benefited a lot from this.

HW: *We now discuss collaboration. Especially these days, young researchers try to collaborate a lot to learn about new fields and impact each other's research. Do you think collaboration is essential for strong research or is it overvalued?*

DF: I think it comes down to the person. I've always enjoyed my colleagues enormously, but in fact have not collaborated very much. Many of my papers are single author. In many of the joint-authored papers, it was helpful to work with other people, but I tend to do my parts of the paper myself. So I don't think I'm a very easy person to work with. Sometimes, I felt badly about this because I tend to take over a paper because I have a very individualistic system of working. Now, there are tremendous insights that you can get from other people, especially people who have strengths in different areas than yours. For a young person, you do want to collaborate with as many people as you can. But my own experience is that I did my PhD and all of my early papers on my own, and that worked out just fine.

HW: *How do you study math?*

DF: I think I mentioned before that I have often found it very helpful just to pick up a mathematics textbook. A lot of mathematicians write really well and show the connections between ideas very insightfully. For instance, when I realized that the theory of convolutional codes have something to do with principal ideal domains, I read the appropriate mathematical textbooks and realized that there wasn't much to it, just one theorem. With that insight, I wrote a paper using that one theorem again and again. Similarly, when I realized that trellis-coded modulation was really coding over groups in many cases, I picked up a very nice book on elementary group theory and learned it, and that's been helpful to me ever since. The book is not something I can say should be part of the electrical engineering curriculum, but a lot of people should go out and learn group theory when it becomes relevant, because it's not that hard and is a really powerful way of thinking. I had a very poor mathematical education in college and in graduate school took very few courses, but I found it's very possible to learn math as you need it.

HW: *How do you differentiate information theorists from mathematicians?*

DF: I think there's a huge difference in terms of motivation. A mathematician is much less concerned about whether the problems that he or she is solving are relevant to anybody in the real world. Information theorists for historical reasons are in electrical engineering departments, but I think that's appropriate. As engineers, we are always interested in the real world implications of our theorem, and I personally lose interest in a paper if it seems to me that it's purely theory for theory's sake, even if it's beautiful. Mathematicians tend to judge work on degree of difficulty; it's like diving. For information theorists, I think, the first criterion is whether it solves a real problem. Of course we can admire the degree of difficulty and beauty of execution as anyone can.

HW: *Now we're moving to the implementation part of the research process. In your opinion, how important is implementation in different subfields of information theory?*

DF: You mean the implementability of our solutions?

HW: *Yes. For example, in coding theory it is important to apply the theory to practice. But for calculating capacity bounds of channels, direct application is often not as valued.*

DF: If that is true, then I think it is too bad. If what is admired is simply getting good bounds whether or not they have any relation to practical networking problems, then to me that somewhat devalues the work. As a student, as a young professor, or as a worker in a research lab, you master tools. Mastery of these things and the ability to innovate in these known areas is much prized. You get some points for having achieved these things but the ultimate reason we're doing these things is because we hope it should be relevant. Actually, you mentioned coding theory and how we measure its success by applicability. But for a long time in coding theory, that was not true. Coding theory was basically a field of applied mathematics and a lot of the field was about filling out tables of codes that exist or don't exist. Much of it does not even consider the problem of decoding; it was codes and their capabilities. Implementability didn't enter into it at all. And of course, when I started out, I started with a company that was working on convolutional codes and wondered why isn't everyone working on this? It worked so much better. Well, we really didn't have a theory of convolutional codes. We can't generate papers on convolutional codes. Algebraic coding theory dominated for at least 20 or 30 years, and the residue of this era of algebraic coding theory for applied math's sake still exists. It's been an important field academically. A lot of people have worked in it, but it's had little practical application, and therefore I value it less than the kinds of coding theory that are more applicable to the real world.

HW: *What subfields of is research in information closest to practical system design?*

DF: I'm not sure what spin you intend to put on that question, but the area in which information theory had the most applicability to the real world was space communications, because you really did have an ideal additive white Gaussian noise channel, you had a situation where basically cost was no object, you could implement as complicated a system as you liked---at least on the ground, not in the spacecraft in the early days. That was a case where 1 dB was worth a lot, so all the conditions were absolutely right for applying information theory in practice. I think Jim Massey said once that it was "a marriage made in heaven," because you could take everything that you developed in the laboratory or through analysis, and apply it directly. No bandwidth limitations, et cetera. With everything else, there were practical considerations that you had to take into account. With space communication, I guess there were political considerations and scheduling considerations, but in the scheme of things, they were fairly easily dealt with.

HW: *If your work was not that closely aligned with practice, what effort did you make to get engineers take notice?*

DF: If I had something that was really applicable, I was in a company. We applied it and then would advocate it to standards bodies for everybody to do the same thing. In the modem world, we have had a long history of success. In the things that I have done that have had less direct applicability, I tried to recognize that myself. For instance, trellis shaping is a lovely idea and works very well. It's the natural dual of trellis coding. I wrote one paper on it. In the V.34 standard that was being developed, a lot of people asked why I wasn't advocating trellis shaping. I said, "You know you can only get 1.5 dB. There are simpler techniques that can get you a dB which are perfectly good." It's an elegant idea and certainly could be implemented but for a tenth of a dB, who cares? So in a case like this, you try not to oversell your ideas if they're not that applicable. There is some work I've done, especially in recent years, which I don't think is applicable at all, and I'm not going to sell it as such. In my career I have been surprised at the applicability of some things that I never expected to be practical, but other things have not caught fire, probably for legitimate reasons. By at this stage of my career, I have given myself permission to work on such things for purely intellectual curiosity. I never used to do that earlier. I can't recommend it to you younger people. I think you should work on things that are applicable.

HW: *So now we are moving to communication with peers. I think that many of the classic papers were well written to deliver their ideas. Do you think that papers these days are as well written? What are some distinct differences between how work is presented then and now?*

DF: I think it's easy to write a good paper if you have a great idea. With a powerful train of thought, you just have to explain it. If it's new and seems right, then people will go with you. I think you're right that the early papers were quite well written. If I look at the Transactions today and pick any paper at random, it's very well written. I think the average level of writing is probably better than it was thirty years or forty years ago. But because the ideas are so dense and depend so much on earlier papers, they're hard to read unless you are an expert and are tuned to the particular issues that the paper are addressing. I think it's not really that people don't write well, but rather the papers have been become necessarily more incremental. When you have a big step, a lot of people will go with you. If it's a tiny step, it is just a less interesting paper regardless of how well you present. I don't fault the quality of the writing. We just need bigger ideas.

HW: *Do you think people are writing too many papers these days?*

DF: Yes, there's been a huge change in that. When I was a graduating student, it was considered perfectly fine to write one paper a year provided it was a really good paper. Go back and look at Bob Gallager's bibliography, or mine. I was working in industry and in general, would try to have one paper a year and a half because they had a long paper and short paper distinction in ISIT, and I always wanted to have a long paper. So I was always working on something, but I think I had one paper a year or two during those times, and I think it was a pretty good collection of papers. Nowadays, people are writing a paper a month, or there are joint papers. So people have 20 papers a year. Senior professors have their names on half a dozen or more papers at each ISIT. It seems insane. Now, all bad trends start in the biomedical area, and they have had this concept of the minimum publishable unit for at least the last 25 years now, but I think we're getting into the stage where people try to convert any incremental advance into a paper. And that's not the way to write papers that will be remembered for a long time. Maybe it's what you have to do but I'm sorry to see it and I'm sure this is a common complaint of older people: far too many papers, far too incremental and so forth. I think it's been ever thus and you have to discount the complaints of the older generation to some degree but I would like to see fewer papers, bigger papers.

HW: *So we're going to wrap up the conversation with some general observations on research in information theory. Can you compare and contrast the state of the field when you joined and now?*

DF: I think I have been doing that.

HW: *Let's refine the question. What has surprised you? For example, what ideas are you surprised still play an important role today? What great ideas are no longer applicable today?*

DF: In the realm of ideas, it's hard to say that I have been surprised by too much. When I was a graduate student, it was kind of accepted that sequential decoding was the end of the line as far as coding goes, that it was a good enough solution and got us close enough to channel capacity and we shouldn't expect anything better… and that was the end of the golden age of information theory. People went elsewhere and into other fields. It turned out that sequential decoding was not the end of the road, so I would say I was mildly surprised because I had drunk the Kool-Aid, but the real world has a way of surprising you with things like that. Within 10 years, maybe 5 years, that wisdom was not wise anymore. So that's a small example of how ideas become obsolete, but most of the things I believed 50 years ago I still believe today. They have been elaborated to an incredible degree. In the realm of technology, I would say I have been immensely surprised by what Moore's law has made possible. I just don't think, as someone said, the human mind can comprehend what a three-order increase in performance complexity is going to make possible. Things like Google still seem like a miracle to me. The idea that people are proposing LDPC codes at tens of gigabits in optical networks is mind-blowing to me, even though it's a natural evolution of where we come from. In that sense it's not a surprise, but it's just mind-blowing. In terms of the evolution of the field, the explosion of network information theory is something of a surprise, because for a long time multi-terminal information theory was just considered to be too hard and progress was infinitely slow, so the fact that people have finally accumulated enough ideas, tools and techniques to solve a lot of the problems of the field has been a surprise to me.

HW: *Finally, can you name one problem you think hope will be solved in the near future?*

DF: I'm still tracking the question of whether there is a (72,36,16) binary block code, which is the longest outstanding such problem, and one which so many good people have worked on. There has been a flurry of work lately that has been tending towards proving that the automorphism group of such a code would have to have size 1, which for me has already answered the question. My more mathematically inclined friends argue that it could still exist. I hope to see a resolution to that problem in my lifetime. Besides, I have $100 worth of bets that I would like to collect.