Interview with Robert Gallager
June 5, 2013
Cambridge, MA

MK: *Thank you for talking to us today. The goal of this discussion is to get insights on how to do research well in information theory. More particularly, we are hoping that you can give some advice to young researchers. We will discuss five or six topics that correspond to different aspects of the research process. The first topic is motivation. We want to know if your research was motivated more by seeking elegant theory or solving practical problems. Did it change based on the time in your career and the questions that you asked or not?*

RG: I don't think it's ever been motivated by either one. The way I've always done research is by first getting very confused about something. As an academic or as an engineer, one is constantly looking for problems; we spend some of our time reading the literature. When I read the literature, I get very confused. I don't know what the person is trying to say or why. I try to sort that out and think about it, trying to come to grips with it. Or I think about some real problem when writing a proposal, trying to figure out what had to be done. There was always a state of confusion of not being able to figure out what came first, what went second. And this process of trying to sort out what was important and what wasn't. Which isn't so much the question of what is practical or what's not practical, but rather the question of what's based on what. The kind of things you learn when you read a book, when you read an article, when you listen to a lecture, somebody is trying to explain something. And when you're thinking, you're also trying to explain something to yourself. My motivation for research has always been this confusion, which is led to thinking about something and some cases starting to understand it and then delving more deeply. It's not a process of saying I'm going to solve some big practical problem. It's not how I build a communication system of future. It's more the question of how do I understand the general way the communication system should be built. When I was consulting, sometimes I became involved in thinking about how to actually build these things. More often it's just the question of what is the right way of looking at a problem. And it doesn't start from any place; when you're thinking about what is the right way of solving a problem, it's a feedback system. You do a little bit of thinking, then you start asking questions, talk to other people, they ask questions, and you go back and forth. That has no beginning and no end. At a certain point in time, you might want to write something down or you might want to give a talk, and in a sense that is an end. But then you keep on thinking about it until something else comes along and takes over.

MK: *So it has been argued that truly great work should be shocking and controversial, in some sense revolutionary. Do you feel that the field has developed such that these views are outdated or are researchers not taking enough risks these days?*

RG: I guess I never felt that research should be shocking. Sometimes it has to be shocking. If your research leads to something that is unexpected, then yes, it's shocking. But to me, research leads where it leads. You don't do research with some presupposition of what you want to have come out of it. And if that's something that is very different from the way other people are thinking, fine, that's the way it is. If it is something which is what are people are thinking, but you can find a better way of organizing or presenting it, then that's what you want to do. Research should go where it's supposed to go. You shouldn't try to force it.

MK: *So this shouldn't be our criteria for defining the motivations of our problems or finding our problems?*

RG: I don't think people should wonder whether they are taking a big enough risk. To me, it's not the right question to ask. The right question to ask is that I want to understand something. The thing that I want to understand is what my mind tells me is important and good researchers have a good way of understanding what is worth understanding and they cultivate that. When somebody else tells them what they should work on, typically what they do is not worthwhile. Even your faculty advisor doesn't always know what you ought to work on!

MK: *Our next question is about crowded problems. Do you think it's a good idea to work on problems that many people are working on and for which there exists a huge crowd? Or do you think that the existence of a crowd for some problem means that the problem itself is an important one? If you're working in a field that is crowded, how can you distinguish yourself? How is it important to bring out something new or different from other people?*

RG: Well, you can! I think asking whether other people are thinking those questions is not important. I think the burning question of research that we should always have is, "Is this something I feel it is important?" If other people are studying it, if too many other people are studying it, then yes, the chances are somebody else will do something more interesting than what you will do. It would be better if you work on something where the field is a little less crowded. But sometimes the most important questions are those that everybody is working on. Back when I was student, everybody was working on coding theory. The big question was, "How do we build codes that achieve Shannon's promised results?" What we found after a while was that good codes couldn't be realized with the hardware of the day. So the questions that we were asking weren't really the right questions. As soon as I got my degree and started consulting a little bit, one of the questions was, "What kind of coding systems can we actually use in various applications?" The only possible ones were the most expensive things, space probes and things like this. And even there, the things that most people were doing coding research on were far too expensive. Not in terms of dollars, but in terms of weight; you couldn't put these things on a space probe. One had to come up with something which was very simple and practical rather than something which was very elegant. You shouldn't worry about whether you are working in a crowded field or not. You should try to understand the question. As you understand something you start going off in directions that other people are not taking. And then you start doing something which is just different. Almost every problem I've worked on in my life, I've started off when a problem would seem many other people were asking the question also. By time I got to the point of writing a paper, people looked at it and say, "Oh! That was a very original thing you did there!" There was nothing original. It was just I was just thinking about it in a different way than other people were thinking about it. And that's what students ought to work on, trying to think about things in novel and creative ways rather than worrying too much about what they're thinking about.

MK: *Many researchers fear the influence of funding and citations over their work. What recommendations do you have for young researchers who want to solve questions that excite them but also consider issues such as citations, funding and tenure track?*

RG: Those are always very difficult problems. They have always been very difficult. To what extent do you work on problems because there is funding for them? I think one of the best answers is, you get to know the people who are with the funding agencies, you try to explain to them which ideas you think are interesting, why you think they are interesting, and you try to get them to try to find money for doing them. To a certain extent, they are frustrated by trying to get the research people to do things which are immediately applicable. In a sense that's not what research is about. There is applied research, which should be done by industry primarily. University research should be something which is not a pie in the sky but which is looking at those problems which are fundamental and which will become important at some point in the future. The time to work on a problem is when the theory leads to it as opposed to when practice requires it.

MK: *Thank you. The second topic we want to talk about is problem formulation. As an engineer, when we start formulating research problem, we need to design a system model. Could you describe the criteria for a good system model design?*

RG: I can tell you a story. The first time I talked to Claude Shannon, I was a young researcher. I wanted to talk to him about what I thought was a very interesting problem. His first reaction when listening to it was to start stripping all the pieces from it until the problem became trivial. When it became trivial, we both saw the answer, and then we started to put back the pieces. So the answer to your question is to always start with something simple enough so that you can understand it. If it isn't applicable to your problem after you understand it, then try to change the model where you use the understanding you have. And you change the model in such a way that at that point you can bring it a little closer to what's a real life problem… but not too close. You do something else simple. And then, you get some insight from that. You go back again and back and forth, working your way closer and closer to something that might be more or less real. But you solve the problems which are ready to be solved from what you've already done and from what other people have done.

MK: *Some young researchers often make the mistake of overcomplicating a model or solving simple but mismatched models. Do you think which one is more dangerous?*

RG: Definitely overcomplicated models. If you solve a problem which is mismatched, you will at least solve the problem and can use the insight to get a little closer to what you or other people may be interested in. The nice thing about theory and the reason we do theory is so it can be applied to many different things. Researchers work in a systematic way to try to develop the field which are natural, they answer questions which are posed by the questions that came before, and the field gets built up. More people come in to use it and they delve in different places. I've never seen good research… well I shouldn't say never… I've seen some very good research being done on very real problems, but that research is usually done by people who have done enough similar research that they have this background of theory that they can operate from, taking their experience and picking things which are useful for the problem at hand. One should always separate the question of what do we need to solve today from the questions of how do we enrich this overall field and how do we understand this overall field. The purpose of research is to understand the overall field. The problem of practice is to use what is understood to make a better system today.

MK: *When you make the first model, do you care if it is solvable or not? Or do you just try to make a reasonable model?*

RG: I think the only real tool that you have is your insight. And therefore you have to use your insight. You carry your insight as far as you can and then start to make equations, calculate things or simulate models. And then you get more insight. If you give up with the insight part of it, you're dead. Really! In a very real way you're dead. You can't do research, you can't do anything.

MK: *As an information theorist and an educator, what do you think is the best way to understand math?*

RG: Well, I give you my own personal experience, which is to start out with almost no math. It was healthy in a way because I needed to use my insight more, because I didn't have all this machinery sitting there. So that machinery didn't confuse me and so it was simpler in a way. As I go through my career, I needed more mathematics. As I needed to learn it, I find it simplifies my thinking in many ways. It gives me better ways to think about things. That's a matter of balance, which is different for different students. Some students love mathematics, they love formalism, they love the kinds of systems where you start with axioms, learn from axioms, work from the axioms. Other people like to be more practical and see the applications of ideas they learn. They develop enough mathematics such that they can understand the theory but then they try to look at it in a more physical kind of way. Some people like to draw pictures. Other people like to do simulations. So everybody uses mathematics in a different way. First you take courses. Certainly what information theorists need is a lot of analysis, a certain amount of linear algebra, certainly probability theory as a mathematical theory, at some point measure theory. But what order those things are learned, I don't think makes any real difference. I think you learn these underlying theories, and then how to use them. And for everyone it's done in a different way, because we all think differently. The worst thing to do is to try to fight yourself because, as I was trying to say before, the only thing you really have going for you is your insight. And if your insight tells you to look at a problem one way and mathematics tells you to look at in a different way, then you have to resolve the different views, but resolving them by saying mathematics is the right way and insight is the wrong way is not going to work.

MK: *So you believe that mathematical tools are useful in information theory for solving the problems rather than of independent interest? This is related to my other question. How do you differentiate an information theorist from a mathematician?*

RG: If you read the information theory literature, you can quickly decide who the mathematicians are and who the more engineering kind of information theorists are. Many information theorists, and many of the best ones, are mathematicians and they have no idea of any of the engineering aspects of the field. If you ask them why we call information information, the mathematicians don't have a clue. They have no idea what this theory has to do with the idea of how actually one communicates what one wants to communicate. And that's fine because you need some people who can strengthen the mathematical theory, other people to understand what the mathematical theory is for, and others who are going to build systems. All of this has to work together. All of us have to learn to work with other people who are a little different than we are to get things done. None of us do the whole thing ourselves but all of us should bring something to the table… and we have to decide what we want to bring to the table ourselves. We do this by experimenting and talking to other people about what we are doing. It just evolves that way. Sorry about being so soft and fuzzy about it. But in fact it is soft and fuzzy.

MK: *What different ways would an information theorist attack a problem as compared to a mathematician?*

RG: There is a whole spectrum of differences, and part of it leads to some bad feeling when a mathematician feels that the problem has not been stated as a proper theorem and has not been proven as a proper theorem proof. Meanwhile, the engineer feels that what has been stated is irrelevant and the fact that it has been proven is just as irrelevant. For example, a very famous mathematician by the name of Doob always felt that Shannon was a bit of a fraud because Shannon did not state theorems in the way that Doob would have liked. But Shannon, when he was stating his theorems, absolutely understood those theorems for a wide variety of cases. He knew that they could be extended to a very much wider variety of cases and he didn't much care exactly what the limits were. And he was right in a sense because it wasn't important what the limits were. It was important later for people to think about the cases it didn't apply to, but the most important thing was sorting out what this theorem was to start with. The details were in figuring out exactly where it applied where it didn't.

MK: *My other question is about finding one's personal style of for the research. From your experience, how long does it take for a researcher to discover their most comfortable style of problem solving? Do you have any advice on accelerating this process since it usually takes a long time for a young researcher to discover his or her style?*

RG: I think they just have to do that. The only advice I can give them is to have faith in yourself when you're doing it and do not feel that you must be doing the wrong thing and look at somebody else and say they must be doing the right thing. When you're doing research, there's a risk in almost anything you do. Some things work out, some things don't work out, and you have to get used to the idea of having some things work out and some things not work out. There's this question of whether we take enough risk or not. To me, this isn't really a choice. I have to take that risk because I do get confused by things and my nature is to try to figure out those things that I am confused by. For example, when I was a student, I worked very hard to sort out what information theory and quantum theory had to do with each other. That was total failure. I decided that it was not my thing and I will let other people do quantum theory and I do the things that I do. Part of the answer to this question is when you're working on a problem and it's not getting anywhere, the most difficult thing is deciding when to get rid of it and start something else. Shannon's approach there was he was always interested in four or five different questions and would work on whichever one seem to be the most interesting. If he wasn't getting anywhere with one, he wouldn't work on it. Period. End of story. But you see, Shannon could do this, I think, because he had enough faith in himself to know that some of these things would work out. He didn't have to feel that he had to do one particular thing. I think that getting a Ph.D. degree is almost a bad way to start doing research because one feels when one is committed a year or two already, one has to finish it. One is not able enough to twist and turn the way one should do when doing research, so one keeps plugging ahead at it. Even when solving a thesis problem, you still have to move back and forth, try different things, and go in different directions as the problem and your interest dictate.

MK: *As the community grows, it seems to become both easier and harder to foster good collaborations. I think it is safe to say that young researchers always fear we are not collaborating enough. Do you think collaboration is essential for strong research or is it overvalued?*

RG: Again, different people have different needs for collaborations. I know that Muriel Medard collaborates with thousands of people, as an exaggeration. I collaborate with a very few people. We are both very successful in what we do. People should look in their own natural sense of how many people they want to collaborate with. You should always be looking for people you can collaborate with. You should always be trying to explain what you are doing to other people. You should always try to understand what other people are doing. When you find common interest, you should try to collaborate with them. When people are too worried about other people stealing their ideas, I think that's a very damaging thing. It's better trust everyone. When you trust everyone, people usually don't steal your ideas. I've never felt like anybody would ever steal any of my ideas. Maybe my ideas aren't that good but I don't think that people would steal them. And because of that, I've never had that problem. So I collaborate with people that I collaborate with. I should make one more comment about that though. As a matter of being successful in the field, there is a gain in collaborating. If you have ten collaborators, you can write ten times as many papers. If you collaborate with these ten people, then all of them will reference those papers when they write papers. So in citation indices, instead of having ten times as many papers, you have hundred times more citations. Now if that's worthwhile, that's one advantage to collaborate.

MK: *One point of view is that one should just work on a problem deeply to gain the necessary insights and that collaboration can be too much of a shortcut. Another perspective is that no one can get all the insight to a very hard problem and that collaboration is essential. Do you have any experience with that either side of that argument?*

RG: Well my experience is that sometimes collaboration is valuable unless I collaborate to the point that I don't have any time to think on my own. I should always be my own best collaborator. Aside from that, I don't think that there is any norm there. When you really want to think about something, you want to walk and think about it. When you don't quite know what to think about, it's good to talk to somebody else and see what they are thinking about.

MK: *Do you value information theoretic bounds which can be easily understood from insight but still can be improved a lot from mathematical analysis, like Fano's Inequality, for example?*

RG: Fano's Inequality is the best example that I know of something which could be done purely from insight. Mathematical analysis has never added anything to it. There have been people who didn't understand the insight and therefore didn't state it correctly but I don't think analysis has really helped on that issue at all. It's purely an issue that should be handled by insight and very simple mathematics. For other problems such as certain kinds of strong law proofs to the coding theorem, deep mathematics is more helpful. And the mathematics working with insight, back and forth, is very powerful. My own feeling is that we have gone a little overboard in the information theory community about using too much mathematics and not enough interpretations of what the mathematics means. That's a matter of opinion and many people would disagree with that.

MK: *So what do you think is the main damage that it's causing, apart from the fact that we are losing the beauty? Do you think that we are going to lose the ability to solve specific problems in the future if we give up this insight to gain more mathematical tools? In long run, if you stick to this method of using heavier and more complicated mathematical tools at the expense giving up insight, do you think that there's going to be some problems that we will not be able to solve?*

RG: It's a complicated question. To me, the more deep mathematics you know, the more insight it should give you. When you start to use the mathematics that you don't have insight about, it becomes much harder for your work to have much significance. All of the best mathematicians that I know, when you get them aside and start talking to them about something, they will not say, "I can prove this." They will say, "Well, this is the way it ought to be." When they say this is the way it ought to be, it means they have some insight about what it is that they are doing and that's what they rely on to prove things. You don't prove things by using mathematics. You prove things by using insights. So you shouldn't learn more mathematics than what you have time to understand in an insightful way.

MK: *Now to a different topic. How important is implementation in different subfields of information theory? Do you feel researchers are overvaluing or undervaluing the ability to affect real systems directly?*

RG: I don't think they're doing either. I think that a lot of researchers have fairly good balance there. More mathematical researchers are not interested in implementation, which is fine because other people who understand the research and are interested in implementation can do it. My Ph.D. thesis was something that was not implemented for forty years. I was not the one who implemented it. At the time it got to the point where technology was ready for it, I was not really interested in it any more. It was not that I though implementing it was not important; it just was that I was doing other things at the time. I've been telling you to follow your insights. For me, I always followed my insights but also try to be consistent about things. So when I worked on data networks, I concentrated on that. If I'm working on stochastic processes, I spent my time there. If I try to do too many things, I get confused and then don't do anything alright. Yes, implementation is important. You don't always have to implement the things that you work on. You can get ideas or take other people's ideas and implement them. And in a good career, you do a little bit of each. If you have a faculty career, you are going to consult also. And when you consult, that's a wonderful opportunity to try to implement things. That's a wonderful opportunity to explain the theory to people who are implementing things. You can do this in many different ways. Part of it is to give the overall system and architectural view, and let other people implement it. Some people like to become involved with the actual chip set design or the things at that level. I don't think that's a problem. Most large systems are done by quite a large number of people and I think that's fine. You have to do the research before you do the implementation. Many systems where people do them experimentally, particularly large software systems where they try to change the software to correct the errors, get worse and worse. I think there are many examples of that.

MK: *I was wondering if you think implementation or real world constraints can affect the insights that we have toward the problem we are trying to solve. The example that I have in mind is LDPC codes, where practical considerations such as convergence and message passing algorithms gave new insights into the problem. Do you think that real world constraints can give you a better view into the problem?*

RG: Of course! And the more experience you have with the way they work, the better the chance you get more insights. Those insights will let you implement it better.

MK: *Have you ever encountered a situation where your theoretical work had the potential to impact practice but did not yet. What do you think is the remedy for that?*

RG: The only advice I can give is to become involved with the people who are actually building the systems. That's a tough way to operate. I know examples of people who even join standards committees as a way of changing what the standards are so that new ideas get implemented. It's a hard way to do it and requires a great deal of patience. It also requires a great deal of determination to see these things get implemented. I think that it's a very worthwhile thing. I have never been pushed to go that far myself, but I know people who do.

MK: *You didn't want to do that or it just didn't happen?*

RG: Some people enjoy that, some people don't. I'm probably at the lower end of people who really dislike sitting in committee meetings.

MK: *Do you remember any significant theoretical work that could be implemented and change system but didn't happen?*

RG: If doesn't happen, very often it's because there are very good reasons why. Some things may look very good on paper but, when I talk to people who are much closer to system design, I often find out that there is some real reason why it's not being used. That's happened to me a number of times. A lot of very good engineers are out there designing systems, designing architecture for the systems who have wonderful insights about why particular ideas don't get used. I get very upset when I read theoretical articles that say here is the new way of doing something and it will have a 20 dB advantage over the way things are now. Now, things usually don't have a 20 dB advantage because usually the ideas that went into that research are partly understood as far as the community who builds things. If there wasn't some reason why it didn't work, people would have been doing something like that. Research results can add something of course and often over time they add a great deal because you start building something in different way. You start building a CDMA system instead of some other kind of system. It has little advantage at first. As time goes on, the advantages get bigger and bigger. Digital systems were not immediately far better than analog systems. Over time they become better and better because when you build digital systems, then the people who build hardware will build digital hardware. And suddenly the advantages get bigger and bigger over time.

MK: *In your opinion, what aspects of information theory are closest to practical system design?*

RG: Certainly, coding theory has been very close to practical design for a long time. Many people who work on coding also work on real systems, and I think the connections there are very tight. People who work on wireless, much of that is very close. Not all of it, but a lot of it is quite close. People who work on multi-access information theory, a little of that is close, but most are not. That's the field where the problems are very hard, people are groping around to try to find better ways to look at things, but it's a very hard problem. So yes, some places the connections are close, some places they're not. I don't find that to be a reasonable reason to work or something or not. If it's not close, then that just means there's more work to be done before it becomes close.

MK: *The fifth topic is communication of your work after the research is done. It seems that one of the underappreciated aspects of the success of information theory is the pioneers in the field were great communicators. Many of the foundational papers are really nicely written. Do you feel that papers are as well-written now? What are some distinct differences between how work is presented then and now?*

RG: My sense is that papers are better written now than they were 50 years ago. The talks are generally better organized now. 50 years ago, people didn't quite realize that your work wasn't worth much unless other people could understand them so there wasn't a lot of attention paid to making it understandable. I think people are more careful about that now than they used to be. Part of why it looks like papers in the past were so well-written is the papers in the past that you still remember are the ones that are still read, and they are still read because they are well-written. You think the best people were writing these papers. They were the best people, but they were the best people because they wrote the best papers, not because they did the best research. So yes, do spend a lot of time writing a paper. That's where you need collaborators. When you write a paper, have collaborators who are willing to tear it apart.

MK: *Since you have many well-written books and papers, do you have some advice on how to sell big ideas?*

RG: Well, I think selling big ideas is basically wrong. I think what one should try to do is to explain ideas, explaining both the weak points and the strong points. If you try to hide the weak points and sell the idea, eventually people stop having a great deal of faith in you. You really want to spend a lot of time with a piece of research, understanding how it connects to other things, being able to explain how it connects to other things. Not how you interpret it after spending six months working on this problem, but how other people would interpret. You have to put yourself in their shoes and explain why this is a worthwhile thing. Why it's something that should be pursued. Why you want them to steal your ideas.

MK: *When you are reading papers, what do you think are the common mistakes that young researchers make in communicating their results?*

RG: I think the main mistakes people make, especially students, is that they write their paper for their faculty supervisor and they don't write their paper for their student friends. I think if you visualize your audience as being a student friend or better still, as a student who is two years younger than you are, then your paper will be much clearer. People who haven't been thinking about the problem that you have been thinking about for a long time are just not going to catch on it very quickly. You catch on to ideas very quickly if they are part of the large picture that you gather in your mind. If it's an unfamiliar area for you, you can get lost very quickly. A good writer will recognize that a broader audience is going to get lost quickly and therefore spends a great deal of time to get the underlying ideas across as clearly as possible. They will spend pages talking about the very simplest aspects of the problem.

MK: *Just to finish the interview with some big picture questions, can you compare and contrast research when you first started and the state of the field today? In particular, what has surprised you in these years? For example, what ideas are you surprised are still are important today and what ideas did you value in the past that are no longer important?*

RG: The big difference is that the field was much simpler when I first started. There was very little known and it was very possible for people to make major discoveries which now can be understood very quickly. The Huffman code the result of a term paper in an information theory class. It is now something that we can understand in perhaps half an hour. At the time, people couldn't understand it. They would study it for weeks because people didn't think algorithmically back then. I have a hunch that there are ways of thinking that are not familiar to us now which could become familiar and very helpful. The people who think of these things will be the famous people of the future. If I knew who they were, I would tell you.

MK: *Future Huffmans. Could you name one problem that you wish would be solved in the near future, one empty spot that is bothering you?*

RG: I can't be very specific about it, but there is this whole field of multi-user information theory, which has been running along for 20 years with two or three underlying ideas. We desperately need another two or three underlying different ideas, things which are new. What are they? I don't know.

MK: *Do you have any hunch on how to attack the problem? Where should we look?*

RG: Right now, my sense is to apply them to martingales, because I'm writing about that in a book.

MK: We have one more question. There is often debate about how useful information theory is in other fields. What fields have the opportunity to be synergetic with information theory in the near future?

RG: It's hard to tell without seeing people start to work on these things. Certainly, the work that David Tse is doing now in terms of DNA sequencing; it's almost information theory but applied to a different problem. That's fascinating! For network information theory, there are traffic problems. On a highway, you find this accordion effect. It seems obvious that somebody could point out using simple dynamics that if all the cars would drive the right way, everyone would get through much faster. I don't know what that right way is but it doesn't sound much more complicated than understanding how a spring works. That's almost an information theory problem; it's a system problem. It seems like there's many problems that people can get interested in and then solve. I've been writing a book on stochastic processes and I can't work on anything else until I finish writing all the solutions to all the problems. So that's what I find fascinating right now.