A Not-So-Random Perspective on Probability and Random Processes

Scott Brookhart


Probability and Random Processes for Electrical and Computer Engineers is a textbook and reference on probability and discrete and continuous random variable processes for electrical and computer engineers. Author John Gubner offers examples, problems, and solutions to further enhance readers’ understanding—which is one way that I learn material well.

Audience
The book is for readers with an advanced undergraduate- or graduate-level understanding of math. Some background in signals and systems, as well as linear algebra, is a prerequisite. Readers can expect to gain a solid understanding of probability and random processes related to signals and systems such as network traffic analysis and communications (the book’s examples concern problems that computer and electrical engineers would encounter).

Structure
The book’s material could be taught over two semesters, and each chapter includes an exam preparation and notes section. The material is logically presented, building on former chapters as the book progresses. Gubner has effectively achieved his objective of making this material accessible and workable to the intended readers.

The book begins by providing a good basis for probability, discrete methods, and continuous methods. In the two-semester course scenario, this would comprise most of what Gubner advocates for the first semester of advanced undergraduate or graduate coursework. (Calculus is assumed for this work.) Each chapter provides many samples and solutions with varying difficulties. Gubner includes Matlab problems to assist with computing some of the samples. He also addresses central limit theorem and Gaussian curve.

Random processes compose much of the second-semester scenario, which includes aspects of random vectors and stationary and random processes such as the Poisson process, Weiner Process, and renewal process, for example. Additional chapters include topics such as Markov Chains, convergence, and stochastic processes such as self-similarity and long-range dependence. Readers can expect to achieve a thorough understanding of the material; it includes many examples and is well organized, two important characteristics for a book on these topics.

Final thoughts
Engineers with weak math backgrounds might find some of the material difficult to grasp because Gubner uses notation throughout. Nonetheless, the book has many additional benefits, such as the chapter dependencies chart, which indicates the order you
should read or teach the material if you want to move outside the front-to-back discovery. Gubner also includes a solid reference that’s up-to-date with current technologies and concepts. Including Matlab in this reference is also useful, as is including explanations of Matlab functions to provide better understanding of the underlying computations. If readers desire additional Matlab work, an accompanying text might be necessary. You can also access instructor-only solutions at www.cambridge.org/catalogue/catalogue.asp?isbn=9780521864701.

Gubner provides good explanations and brings readers to definite conclusions on such random topics. Probability and Random Processes for Electrical and Computer Engineers stands alone as a textbook that encourages readers to work through and obtain working knowledge of probability and random processes.

Scott Brookhart is a software engineer at the Texas Education Agency and a senior member of the IEEE. Contact him at sbrookhart@computer.org.

A Little Strange and Definitely Loopy

Todd Schultz


Don’t approach this book or even this review if strange loops—self-referential snakes-eating-their-tails situations—are intrinsically at odds with your soul. In I Am a Strange Loop, the reassertion of ideas first presented in Gödel, Escher, Bach: An Eternal Golden Braid (Basic Books, 1999), Douglas Hofstadter describes how human consciousness and the concept of self are intimately tied with strange loops and abstract high-level patterns built from concrete low-level components.

Unlike many other philosophy books, Hofstadter doesn’t build his thesis with Aristotelian logic from first principles. Instead, he takes readers through thought experiments that you might call “analegories,” combining aspects of both analogies and allegories. At the root of this approach is a first principle: the mind (or self or consciousness or soul or whatever you call it) can’t be completely understood by itself, so analogies are all we have to describe it.

All readers need to get on board is to accept two other first principles: that the mind is fundamentally based on concrete atoms, physics, and chemistry; and that the mind (mine at least) transcends (to me at least) any direct explanation based on concrete atoms, physics, and chemistry. From there, Hofstadter launches into historical and hypothetical examples of how surprising epiphenomena arise from more mundane roots.

A key example is Gödel’s incompleteness theorem, a strange loop Hofstadter treats at length in the book. The mundane roots behind this theorem were the attempts by Bertrand Russell and Alfred Whitehead to provide a complete, consistent foundation for arithmetic. These attempts were hardly trivial, misguided, or ignoble, but it took volumes to establish that 1 + 1 = 2. Gödel’s theorem proved that by necessity, mathematics of any power could not be complete (that is, able to identify all truth statements). Surprising high-level phenomena used in Gödel’s proof were huge numbers that were coded theorems; these numbers weren’t only self-referential but also carried the message that they couldn’t be proven. Some might argue about this interpretation of the message or even that it is a message (for instance, who sent it?). But this demonstrates Hofstadter’s point: high-level system behaviors acting as messages are part of what we call the mind, and they’re built on low-level activities.

Hofstadter suggests that I Am a Strange Loop might be a more descriptive if clumsier title for his book, and I initially agreed. He’s after what makes anyone’s mind—anyone’s sense of “I”—exist (or seem to exist), not just his own. His point is that this sense is fundamentally a strange loop. However, he includes a lot of autobiographical material, personal observations, and anecdotes, and these border on tedious. These reflections are usually somewhat relevant, but there’s enough specifically about Hofstadter—his likes, dislikes, tragedies, and small loopy thrills—to make the actual title more accurate.

I Am a Strange Loop makes extensive use of analogies, and it takes some effort to gear up for each new thought experiment. I tried to ensure that I was seeing the connections and references in each new situation, but part of me wondered whether this mental effort would pay off. When I read Gödel, Escher, Bach, it nearly always did. Perhaps my soul isn’t as close to Hofstadter’s soul as it was in the past (I refer to the I Am a Strange Loop anecdote where Hofstadter describes his disappointment with Béla Bartók’s music, concluding that his soul and Bartók’s must be different).

Hofstadter’s analogies can get whimsical, even slightly irritating, when he demonstrates what he sees as a clever perspective. In a thought experiment based on ideal gas laws with the addition that molecules might clump together, he describes how the random action of individual molecules would cause the clumps to move slowly. If we stepped out a bit in perspective and sped up, the clumps would not only appear to have their own behaviors, they would react to outside stimuli (such as

Gubner’s book stands alone as a textbook that encourages readers to obtain working knowledge of probability and random processes.
squeezing the container holding the gas would affect the clumps). Hofstadter admits it’sorny, but he calls these “simmballs in a careenium” (that is, symbols in a cranium). Readers will need some tolerance for this sort of impulsiveness as well as for his chronic use of rhetorical questions (but aren’t questions the important thing?).

*I Am a Strange Loop* isn’t a sequel or revisit to *Gödel, Escher, Bach*, but readers will fare better if they’re familiar with the earlier work. Hofstadter says *I Am a Strange Loop* clarifies what *Gödel, Escher, Bach* was actually about (that is, “I” is built from strange looped low-level components). For me, *Gödel, Escher, Bach* coherently and interestingly demonstrated incompleteness, computation, and formal systems along with how these phenomena lead to thinking-like behaviors such as DNA processing and ant colonies. *I Am a Strange Loop* is less down-to-earth and more introspective. *Gödel, Escher, Bach* built my understanding; *I Am a Strange Loop* provides glimpses into Hofstadter’s understanding.

*I Am a Strange Loop* is relevant for anyone who’s concerned with software’s limits and potential and naturally for those interested in artificial intelligence and how the mind works. For software engineers and developers, it can provoke a little left-brained thinking about the meta-aspects of what we do. At the end of the development day, it’s about working code. Despite my occasional frustrations with Hofstadter’s approach, he always provides food for thought.

**Todd Schultz** is a professor and associate dean of the James M. Hull College of Business at Augusta State University. Contact him at tschultz@aug.edu.

### Holstader’s “I Am a Strange Loop” Is Relevant for Anyone Who’s Concerned with Software’s Limits and Potential.


Are you interested in “chasing down the next big thing?” If so, *Coolhunting* might just be for you. Authors Peter Gloor and Scott Cooper begin with a few definitions:

*Things that are cool make the world a better place, in some way … We think of cool as having an altruistic component; it is cool to act for the benefit of others … Selfless behavior propels great ideas forward.*

Now, doesn’t that sound like open source software development? I’m sorry to say, they mention the topic only in passing (IBM putting patents into the public domain) and in reference to Linux. The authors are much more interested in spotting cool as it takes shape:

*Coolhunting [of course we needed a new word!] involves making observations and predictions as part of the search for cutting-edge trends. It is a way of capturing what the collective mind is thinking, and using what is captured to advantage.*

After defining cool and coolhunting, the authors introduce and describe a Collaborative Innovation Network (COIN). An important quality of COIN is selflessness, and to hammer home what’s needed, the authors look to bees and how they organize themselves. The fact that the bees I know are simply looking to live another day, rather than innovating, seems lost on the authors and weakens their analogy. But then again, their definition of “innovation” is a bit peculiar:

*The innovation process will be served best when businesses realize that people motivated to collaborate and innovate because of shared interests, without specific regard for personal gain, create compelling new opportunities.*

And although they cite companies such as IBM, Cisco, Google, and even a storied Silicon Valley venture capitalist, the emphasis is very much on giving in order to receive—that is, if you’re looking to others for help, a profit motive complicates and hinders collaborative innovation instead of simplifying and promoting it. And here the authors describe Benjamin Franklin’s efforts to promote public libraries byselflessly “leaning on” his wide-ranging circle of friends and acquaintances. (Too bad he was born a few centuries too early for MySpace!)

Later, the authors refer to something called TeCFlow, which they use to automate statistical clustering analysis for “data mining” large collections of person-to-person communication, with a graphical element for presenting results. In particular, analyzing social networks should reveal “correlations between the positions of people in the social network and identify the characteristics, or properties, of individuals or the entire group.” So, Gloor and Cooper present sample analyses, including one that scrutinizes Enron internal email that became part of the public domain during the senior management’s criminal trials.

Of course, once you come to appreciate coolhunting’s merits, you’ll want to go the extra step and get into “cool-farming,” or promoting your particular kind of “cool” (presumably for profit-seeking business purposes). To help you do this, the authors list four guiding principles:

- promote a sense of shared ownership,
nurtur[e]e the resulting motivation by encouraging the sharing of new knowledge, and
recruit “leaders” to promote communication within the COIN.

These four principles can also be used to describe “diffusion dynamics,” or how early adopters come to embrace innovation before the mainstream does. However, James Utterback’s Mastering the Dynamics of Innovation (Harvard Business School Press, 1994) and Geoffrey Moore’s works, especially Crossing the Chasm (Collins, 2002), provide much better business guidance about how to win over those early adopters and then nurture your growing market to (eventually) hit a mainstream home run.

Later, Gloor and Cooper suggest that the company shareholders’ short-term emphasis works against the company stakeholders’ long-term interests. (Luckily, in privately owned companies, the shareholders and the stakeholders are the same people, so they ought to be able to do better!)

So, who was this book written for? Well, sad to say, it’s not for IEEE Software readers. Perhaps more careful editing (to remove, for example, repetition in content from chapter to chapter) and additional technical detail about TeCFlow could have tipped the scales differently. But honestly, other books provide better introductions to this and related topics. James Surowiecki’s The Wisdom of Crowds (Anchor, 2005) has a better presentation of some of the same examples as Coolhunting. Malcolm Gladwell’s The Tipping Point (Back Bay Books, 2002) features a better description of human “connectivity”—how, in human communication, some people play more important roles in spotting new things and delivering news. Finally, other folks have written more extensively about the benefits, at least for business, of looking outside your company for innovation to include customers, partners, and even strangers. Here, a good reference is Henry Chesbrough’s Open Innovation (Harvard Business School Press, 2003).

TR1 extensions

Becker covers the TR1 extensions step by step. The first half of the book covers TR1's core, including smart pointers, containers, and call wrappers. TR1 has several new smart pointers, such as the Weak and Strong pointers. It also has new containers, such as the array class. Unlike the C-style array, the array class directly supports some of the operations you’d expect from an array class, such as comparisons, iteration, insertion, and size. Starting from array, the TR1 extension adds unordered associative containers, which are hash tables. As you know, hash maps are more efficient than regular maps or multimaps, and they’re long overdue in the C++ standard.

If you’ve ever needed to do event-based programming, you’ll appreciate the TR1 extensions. Function pointers and templates are just some of the additions to TR1’s class wrappers. C++ developers can now create, for example, callbacks, without referring to C-style functions.

The TR1 proposes adding mathematical and numerical functions to C++. It now includes random-function generators and various other numerical functions—and, hopefully, the standard C++ will add them soon as well. Templates make adding these features to the language possible without breaking its object-oriented features.

Each chapter is full of examples that demonstrate usage and “gotchas.” Becker concludes each chapter with several exercises that test readers’ understanding of the topic and demonstrate its practicality. I highly recommend The C++ Standard Library Extension to any C++ programmer.

Art Sedighi

is the founder and CTO of SoftModule. He’s also an editorial board member of IEEE Software and editor of this department. Contact him at sediga@rpi.edu.

Paul Freedman

is president of Simlog. Contact him at paul.freedman@simlog.com; www.simlog.com.

Much-Needed Additions to C


C++ never fails to amaze me—it’s such a well-designed language that, to this day, it can accommodate major enhancements and improvements without affecting the fundamentals that the language designers started with. In The C++ Standard Library Extensions: A Tutorial and Reference, Pete Becker expands on and exploits what’s known in the C++ community as the TR1 or technical report. The TR1 isn’t yet part of the C++ standard, but much of its suggested improvements are expected to be added to the standard in the next couple of years. What does this mean to you as a developer? Nothing! You can still use the TR1 extensions and classes in your code, and once TR1 is approved, you can change the namespace and recompile. Well ... it should be that easy, but no one really knows.

Contents and structure

TR1 takes the best of the “open” C++ frameworks that have been popping up—such as Boost, Metaprogramming, and ACE—and aims to add them to the C++ language’s core. Becker covers different types of memory management classes (Part I), different container types (Part II), and mathematical and numerical functions that use templates for better efficiency (Part V).

I wish that all other programming books were as organized as this one. C++ (or any other programming language that’s been around for a while) is very complex. Texts that try to cover a specific part or library of the language have been most successful, and this book takes the same approach.

Starting in Jan./Feb. 2008, look for Bookshelf online at www.computer.org/software!