European Journal of Control (2010)5:431–441
© 2010 EUCA

European Journal of Control

Discussion on: "Why Is Resorting to Fate Wise? A Critical Look at Randomized Algorithms in Systems and Control"

Sanjoy K. Mitter

MIT, Cambridge, Massachusetts, USA

This is a timely article on the use of randomness and randomized algorithms in Systems and Control. Contrary to the situation in computer science, there is no systematic discussion of this issue and Campi is to be complimented for bringing these ideas to the attention of the Systems and Control community.

One of the earliest uses of randomization is in the theory of relaxed controls due to L. C. Young (cf. L. C. Young: *Lectures on the Calculus of Variations and Optimal Control Theory*, Chelsea, 1960) where one can show that an optimal control exists when the class of controls is enlarged to the space whose controls take their values in a probability space from say the space of continuous functions. One then proves an approximation theorem showing that a relaxed control can be approximated by a continuous control (cf. E. B. Lee and L. Markus: *Foundations of Optimal Control Theory*, Wiley, 1967). This lifting to the space of probability measures is intimately connected to convexification and has been elegantly exploited in the important work of Richard Vinter where Optimal Control problems using the methodology of Dynamic Programming can be viewed as Convex Programming in an appropiate space of probability measures.

Perhaps one of the deepest uses of randomization is Shannon's use of a random coding argument to prove the direct and converse part of the Noisy Channel Coding Theorem which provides the fundamental limitation to reliable communication over a noisy channel. This is a deep theorem and one of the surprises is that even though a random coding argument is used it leads to a conclusion that of the existence or non-existence of a code (deterministic codes included) for achieving reliable transmission depending on whether transmission is taking place at a rate below or above capacity (cf. C. E. Shannon: A Mathematical Theory of Communication, *Bell System Technical Journal*, 27, pp. 379–423, 623–656, 1948). There is increasing interest in control over networks where it would be important to understand the tradeoffs between control and communication performance and randomized algorithms are likely to play an important role there.

Campi points out and cites several results from the Computer Science literature, namely the work on PAC Learning (here there is an important reference missing, L. Valiant : A Theory of the Learnable, *C ACM*, 27 (11), pp. 1134–1142, 1984) and Yao's work on communication and computational complexity. I am not entirely convinced that it is appropriate to transport the PAC Learning methodology to studying issues of synthesis and learning in control problems. However, the examples of control and set membership identification that Campi discusses illustrate the role of randomness in algorithms appropriately and convincingly. In a sense, success of randomized algorithms (in the absence of a successful deterministic algorithm) takes place precisely when the matched pair of algorithm and adversary lies in a "smaller" dimensional space.

Perhaps it would be worthwhile to bring to the attention of the Systems and Control community the subject of Probabilistically Checkable Proofs (PCP), where randomness plays an important role. It sheds light on complexity questions arising in Combinatorial Optimization. It may well have implications for verification of correctness of control algorithms in embedded systems. PCP is motivated by the question: "how many bits of queries are really essential to give some confidence into the correctness of a theorem?" It is easy to argue that the verifier needs to

E-mail: mitter@mit.edu

be probabilistic and also allow it to make mistakes with low probability. A precise definition of PCP can then be given which requires the verifier to have polynomial complexity. The notion originates in the idea of an interactive proof, first proposed in the seminal paper of Goldwasser, Micali and Rackoff (The knowledge complexity of interactive proof systems, *SIAM J. on Computing*, 18 (1), pp. 186–208, 1989). The importance of PCP theory is that it shows that many combinatorial problems that have exponential complexity for finding an exact solution also have exponential problem for solving them approximately. It would be interesting to see if these ideas shed light on approximation questions for control problems.

In summary, Campi's article hopefully will stimulate research on the boundaries of control, communication and computation where probabilistic algorithms will have a central role.

Discussion on: "Why Is Resorting to Fate Wise? A Critical Look at Randomized Algorithms in Systems and Control"

Arkadi Nemirovski

Georgia Institute of Technology, Atlanta, Georgia, USA

Let me start with noting that I am highly impressed with the arguments of M. Campi in favour of randomized algorithms. I would also advise a reader to read an excellent, extremely informative, perfectly well-written and (last but not least) non-technical, thus understandable by a non-expert, and "purely theoretical" paper of Widgerson [9]. Sections 5 and 6 of this article present striking illustrations of the tremendous power of randomization in computation and discuss fundamental limitations of this power. Widgerson's examples deal with fundamental but somehow "academic" problems. My intention in this essay is to present a couple of "practical" examples of how randomization can accelerate solving large-scale well-structured convex optimization problems of applied origin, so that what follows can be considered as an addendum to Section 5 of Campi's paper.

When illustrating the role of randomization in reducing computational complexity, M. Campi presents, essentially, a single example: Monte-Carlo evaluation of the expectation of a random variable. While the Monte-Carlo simulation is a technique of paramount importance in a huge spectrum of applications, using randomization when computing an integral seems, at least in hindsight, not surprising — we are looking for an "average" property of an object (a function), that is, for its literal average with respect to some measure. Usefulness of randomization when solving an optimization problem (and thus looking for an "extreme" property of the object at hand) at first glance seems to be much more questionable. One could argue that randomization plays a vital role in numerous *heuristic* techniques for solving difficult optimization problems (genetic algorithms, simulated annealing, etc.). In my appreciation, this cannot be considered as a serious argument in favour of randomization in the optimization context, since the very fact that the techniques in question are indeed "optimization algorithms" is questionable — they do not provide us with any quality guarantees on the resulting solutions and, strictly speaking, are illusive "last resort" tools for doing *something* with problems we actually do not know how to solve.[1] But why should we use randomization when we do know how to solve an optimization problem ("how to find something extreme and thus rare")? The latter is the case in (deterministic) convex optimization, where we have at our disposal theoretically (and to some extent, also practically) efficient deterministic algorithms allowing to approximate, within any prescribed accuracy, the *global* solution to a problem, and can do this at a known in advance (and reasonable) computational price (e.g., in polynomial time fashion). Surprisingly, the randomization can help in this latter case as well. To illustrate this point, consider an important ℓ_1 *minimization problem* in the form of

$$\mathrm{Opt} = \min_x \left\{ \|Ax - b\|_p : \|x\|_1 \le 1 \right\} \quad [A \in \mathbf{R}^{m\times n}], \quad (1)$$

where either $p = \infty$ ("uniform fit"), or $p = 2$ ("quadratic fit"). This problem plays a crucial role in various sparsity-oriented techniques of Signal Processing, e.g., in

E-mail: nemirovs@isye.gatech.edu

[1] A desperate person in a complicated situation can stick to advices of an astrologer and succeed, and augurs played important role in ancient Rome; this, however, does not make neither astrology nor following omens solid forecasting techniques.

Compressed Sensing.[2] Eq. (1) is a well-structured convex program (a Linear Programming one when $p = \infty$, and a conic quadratic one when $p = 2$), and as such it is well suited for processing by powerful Interior Point methods (IPMs) capable of solving the problem to high accuracy in a small (two-three tens) of iterations, reducing each to a straightforward assembling and then solving a system of linear equations of essentially the same size as those of A. At first glance, all looks nice; however, when A is large scale and dense (which is typical for Signal Processing applications), the matrices of linear systems to be solved at the IPM iterations also turn out to be dense. As a result, problems Eq. (1) with dense matrices of sizes m, n in the range of tens of thousands and more are, practically speaking, beyond the grasp of IPM's — an attempt to solve Eq. (1) by an IPM leads to memory overflow or to the very first iteration "lasting forever."

As far as deterministic algorithms are concerned, the best, at the present level of our knowledge, way to handle large-scale problems Eq. (1) is to note that $\|a\|_p = \max_y\{y^T a : \|y\|_q \leq 1\}$, where $q = 1$ when $p = \infty$ and $q = 2$ when $p = 2$, which allows to rewrite Eq. (1) in the saddle point form

$$\min_{x\in X}\max_{y\in Y}\phi(x,y),\ \phi(x,y) = y^T(Ax-b),$$
$$X = \{x \in \mathbf{R}^n : \|x\|_1 \leq 1\},\ Y = \{y \in \mathbf{R}^m : \|y\|_q \leq 1\} \tag{2}$$

and then to solve the resulting problem by computationally cheap *first order* methods. The computational effort at an iteration of such a method reduces to computing, given a point $(x, y) \in X \times Y$, the vectors $\frac{\partial\phi(x,y)}{\partial x} = A^T y$ and $\frac{\partial\phi(x,y)}{\partial y} = Ax - b$ (that is, to computing two matrix-vector multiplications), plus additional $O(1)(m + n)$-a.o. (arithmetic operations) "computational overhead." With the state-of-the-art methods of this type, e.g., smoothing algorithm of Nesterov [6][3] or Mirror Prox algorithm [4], the iteration count $N(\epsilon)$ to generate an ϵ solution to Eq. (1) (i.e., a feasible solution x_ϵ with $\|Ax_\epsilon - b\|_p \leq \text{Opt} + \epsilon$) is nearly dimension-independent and is inversely proportional to ϵ; specifically, $N(\epsilon) = O(1)\sqrt{\ln(n)\ln(m)}\|A\|_{1,\infty}/\epsilon$ when $p = \infty$ and $N(\epsilon) = O(1)\sqrt{\ln(n)}\|A\|_{1,2}/\epsilon$, where $\|A\|_{1,p}$ is the maximum of $\|\cdot\|_p$-norms of the columns in A. While this count is $O(1/\epsilon)$ instead of the IPM count $O(\ln(1/\epsilon))$, the arithmetic cost of an iteration in a first order method ($O(mn)$), for large m, n, is by orders of magnitude less than in IPM's ($O(n^3)$ for $m = O(n)$). As a result, good first order methods become *the* methods of choice in the large-scale case, provided, as it is the case in typical applications, that medium-accuracy solutions are sought.

Now, what to do if the sizes of Eq. (1) are so large that even matrix-vector multiplications become prohibitively time consuming? This is the point where randomization comes into play. Note that it is easy to randomize computing a matrix-vector multiplication Pu, $P \in \mathbf{R}^{m\times n}$: given u, treat $|u_j|/\|u\|_1$ as the probability to pick j from the index set $\{1, \ldots, n\}$, draw an index $\jmath$ from this set according to this probability distribution, and build the vector $\|u\|_1 \text{sign}(u_\jmath)P_\jmath$, where P_j is j-th column of P. This construction defines a random vector ξ which is an unbiased estimate of Pu: $\mathbf{E}\{\xi\} = Pu$. Note that generating a realization of ξ costs just $O(n)$ a.o. plus the effort to extract from P a single column, given its index (this effort typically is just $O(m)$ a.o.). Thus, for typical ways to store P, the complexity of generating ξ is just $O(m + n)$ a.o., vs. $O(mn)$ a.o. required to compute Pu in a precise deterministic fashion. Besides this, we understand "how noisy" is the resulting unbiased estimate of Pu: for every norm $\|\cdot\|$, $\|\xi\| \leq \|u\|_1 \max_j \|P_j\|$. Now, there is a long-standing line of research in convex (and not only convex) optimization, going back to classical Stochastic Approximation of Robbins and Monro [7], where one is interested in how to solve convex minimization problems in the case when the precise first order information (the values and the gradients of the objective and the constraints) is not available, and one should use instead unbiased random estimates of these values and gradients; for a comprehensive presentation of this topic, see [5]. Along with many other developments, Stochastic Approximation was extended to convex-concave saddle point problems like Eq. (2). Combining one of its versions — the *Robust Mirror Descent Stochastic Approximation* (RMDSA) proposed in [5] — with the outlined "randomized matrix-vector multiplications," one arrives at the following results:

Theorem 1: [5] *Consider the ℓ_1-minimization problem* (1) *with uniform fit ($p = \infty$) and assume that the magnitudes of all entries in A and b do not exceed some $R < \infty$. One can point out a simple randomized algorithm which, for every $\epsilon > 0$, $\beta \ll 1$, with probability $\geq 1 - \beta$ generates an ϵ solution to* (1) *in $O(1)\ln(mn/\beta)\,(R/\epsilon)^2$ iterations, with an iteration reducing to extracting from A a single row and a single column, given their indices, plus $O(m + n)$-a.o. "computational overhead."*

With typical ways of representing A, the overall effort to find, with probability at least $1 - \beta$, an ϵ solution

[2] In fact, the problems of actual interest in the indicated applications are of the form $\min_x\{\|x\|_1 : \|Ax - b\| \leq \epsilon\}$; solving such a problem reduces to a "small series" of problems (1).

[3] This is the landmark paper where the approach in question — to use the structure of a nonsmooth convex program in order to convert it into a "nice" (with simple and very smooth cost function ϕ) convex-concave saddle point problem and then to apply first order methods to this reformulation instead of applying them to the problem of interest directly — was discovered.

to (1) is $O(1)(m+n)\ln(mn/\beta)\,(R/\epsilon)^2$ a.o. When R/ϵ and β are fixed and m,n are large, this complexity bound is by orders of magnitude less than the complexity $(O(1)mn\ln(mn)R/\epsilon$ a.o.) of finding an ϵ solution by the best-known deterministic algorithms.

Here is a numerical illustration. We are interested in solving an ℓ_1 minimization problem with a dense analytically given $m\times n$ matrix A which does not allow for fast matrix-vector multiplications. To generate b, we generate at random a sparse (just 16 nonzero entries) vector u_* with $\|u_*\|_1=1$, compute Au_* and add to this vector "observation noise" ξ drawn at random from the uniform distribution on the box $-1.\mathrm{e}{-3}\le\xi_i\le 1.\mathrm{e}{-3}, 1\le i\le m$, thus getting b. What we are interested in is how the computed approximate solution $\tilde u$ to the resulting problem Eq. (1) with the uniform fit approximates the "true signal" u_* (the Compressed Sensing setting). Below we report on two experiments. In the first, $m=2048$ and $n=4096$; the corresponding matrix A can be stored in the 2GB RAM of the computer used in the experiments. In the second experiment, $m=8192$ and $n=32768$, and A is too large to be stored in the RAM, so that exact matrix-vector multiplications, same as extracting rows and columns of A, are based on generating the rows/columns according to their analytical representations.

We compare the performance of the state of the art deterministic first order method — the Deterministic Mirror Prox (DMP) algorithm from [4] — with the one of randomized algorithm RMDSA underlying Theorem 1. The results of the experiments are displayed in Table 1 and Fig. 1. In the table, "Mult" is the equivalent number of matrix-vector multiplications (that is, extracting a single row/column amounts to $1/m$, respectively, $1/n$ of such a multiplication). All experiments were carried out in MATLAB. We see that in the "small" experiment, the deterministic algorithm significantly outperforms its randomized competitor, both in terms of the quality of the solution and the running time. In the second experiment, where both algorithms were terminated after approximately 3,000 sec, the situation is completely different. Now an exact matrix-vector multiplication takes about 300 sec; as a result, in 3,000 sec DMP makes just 5 iterations (since each takes two matrix-vector multiplications — one by A and one by A^T), which is by far not enough to get a meaningful approximate solution, even a poor-quality one. In contrast to this, RMDSA in the same 3,000 sec does find a meaningful solution.

Note that when R/ϵ and β are fixed, *the total number of entries of A inspected in the course of running the algorithm underlying Theorem 1 is nearly linear in $(m+n)$ and thus, for large $m=O(n)$, becomes a negligible fraction of the total number mn of entries in A* (this is called a *sublinear time* behaviour). This striking fact is not new; the "uniform fit" version of Eq. (1) can be reduced to a matrix game $\min\limits_{u\in\Delta_n}\max\limits_{v\in\Delta_m} v^TBu$, where $\Delta_n=\{x\in\mathbf{R}^n: x\ge 0, \sum_j x_j=1\}$ is the standard simplex, and the possibility to solve games in a sublinear time fashion was discovered as early as in 1995 [1]. Note that the complexity of the algorithm in [1] is identical to the one for RMDSA, and the algorithm itself, in hindsight, is pretty close, although not completely identical, to RMDSA. The advantage of the RMDSA and its "relatives," like the Stochastic Mirror Prox algorithm [2] is that they yield the above result "in a systematic way" and allow to get similar results in other situations. Here are a couple of examples taken from the ongoing, yet unpublished, research of the author:

- *Consider the ℓ_1-minimization problem in Eq.* (1) *with quadratic fit ($p=2$) and assume that the $\|\cdot\|_2$-norms of b and of all columns of A do not exceed a certain $R<\infty$. One can point out a simple randomized algorithm which, given $\epsilon>0$, $\beta\ll 1$, with probability $\ge 1-\beta$ generates an ϵ solution to Eq.* (1) *at the cost of $O(1)\left[mn\ln(m)+\ln^2(mn/\beta)(m+n)\,(R/\epsilon)^2\right]$ a.o.*
  The above complexity is *as if* we were computing $O(1)\ln(m)$ matrix-vector multiplications deterministically and then were switching to computing $O(1)\ln^2(mn/\beta)\,(R/\epsilon)^2$ matrix-vector multiplications in a randomized fashion (and this is indeed how the algorithm works). The best-known deterministic algorithms in the situation in question generate an ϵ optimal solution at the cost of $O(1)mn\sqrt{\ln n}R/\epsilon$ operations; when $R/\epsilon\gg 1$ and β are fixed and m,n are large, the latter bound can be by orders of magnitude worse than the one for the randomized algorithm.
- *Consider the problem of minimizing $\max\limits_{1\le i\le m} p_i(x)$ over the standard simplex Δ_n; here p_i are convex polynomials of degree not exceeding d: $p_i(x)=\sum_{\ell=0}^d P_{i\ell}[x,\dots,x]$, where $P_{i\ell}[x^1,\dots,x^\ell]$ are ℓ-linear symmetric forms. Assume that the magnitudes of the coefficients of all the forms $P_{i\ell}$ do not exceed a certain $R<\infty$. One can point out a simple randomized algorithm which, for every $\epsilon>0$ and $\beta\ll 1$, with probability $\ge 1-\beta$ generates an ϵ solution to the problem in $O(1)\ln(mn/\beta)d^2(R/\epsilon)^2$ iterations, with an iteration reducing to extracting $O(1)d(m+n)$ coefficients of the forms $P_{i\ell}[\cdot,\dots,\cdot]$ given the "addresses" $i,\ell,\jmath_1,\dots,\jmath_\ell$ of the coefficients, plus $O(m+dn)$-a.o. "computational overhead."*
  What is instrumental in this result is a simple procedure for building unbiased random estimates of the values and gradients of polynomials at a given point x from the standard simplex Δ_n. Let us illustrate this procedure in the situation when the polynomial is a quadratic form: $p(x)=a+\sum_i b_ix_i+\sum_{i,j}c_{ij}x_ix_j$. We pick at random, independently of each other, two indices $\imath,\jmath$, the

probability to pick $i \in \{1, \ldots, n\}$ being x_i and take as the unbiased estimates of $p(x)$ and $\nabla p(x)$ the real $a+b_\iota+c_{\iota\jmath}$ and the vector $b + 2[c_{1\jmath}; c_{2\jmath}; \ldots; c_{n\jmath}]$, respectively.

Note that the ℓ_1 minimization Eq. (1) with the uniform fit reduces to minimizing the maximum of $2m$ linear forms over $2n$-dimensional simplex (represent a vector $x \in \mathbf{R}^n$, $\|x\|_1 \leq 1$, as the difference $u - v$ of the "halves" of a vector $[u; v]$ from Δ_{2n}), so that the outlined result on minimizing the maximum of convex polynomials is an extension of Theorem 1.

Let us remark that the idea to use randomization in order to solve deterministic (in particular, convex) optimization problems is very old. For example, the random search algorithms proposed in [8] solve an unconstrained minimization problem $\min_x f(x)$ by generating iterates x_i according to $x_{i+1} = x_i \pm \gamma_i \xi_i$, where $\gamma_i > 0$ are somehow tuned stepsizes, d_i are random directions uniformly distributed on the unit sphere, and the sign in $\pm$ is chosen in such a way that the method is monotone. Specifically, after d_i is generated, we compute f at the two points $x_i \pm \gamma_i d_i$ and choose as x_{i+1} the point of this pair, if any, where the value of f is less than at x_i. If both points of the pair are "worse" than x_i, we reduce the stepsize and try again. The rationale behind this procedure and its more sophisticated modifications is to save on computing gradients of f (and thus it is completely similar to the rationale behind the constructions we have presented). To the best of the author's knowledge, these "naive" randomized routines are completely forgotten now, being by far outperformed by deterministic algorithms. The reason, in our opinion, is twofold. First, the very idea that computing the value of a function f of n variables is much cheaper than computing the value *and* the gradient is wrong: a provable fact is that the arithmetic complexity of computing a gradient is just $O(1)$ times the complexity of computing the value (this is called "automatic differentiation"). Second, the "naive" random search algorithms pretended to be "universal" — they did not utilize the problem's structure and were "black-box oriented" — relied solely on an "oracle" which, given as input a point x, returns $f(x)$. There are solid theoretical reasons [3] to claim that *randomization cannot help much when solving optimization problems from wide families by black-box-oriented algorithms.* In this respect, note that the above "success stories" deal with very special optimization problems possessing a lot of structure, and they heavily utilize this structure. In addition, the outlined results are highly sensitive to the "small" details of the problem's setting; e.g., we have no idea whether

Table 1. Deterministic algorithm vs. the randomized one

		Errors			CPU	
$m \times n$	Method	$\|u_* - \tilde{u}\|_1$	$\|u_* - \tilde{u}\|_2$	$\|u_* - \tilde{u}\|_\infty$	sec	Mult
2048×4096	DMP	0.0014	0.00052	0.00036	122.8	3540
	RMDSA	0.039	0.0079	0.0030	325.4	58.6
8192×32768	DMP	1.006	0.319	0.184	3141.9	10
	RMDSA	0.120	0.0196	0.00634	3000.5	9.4

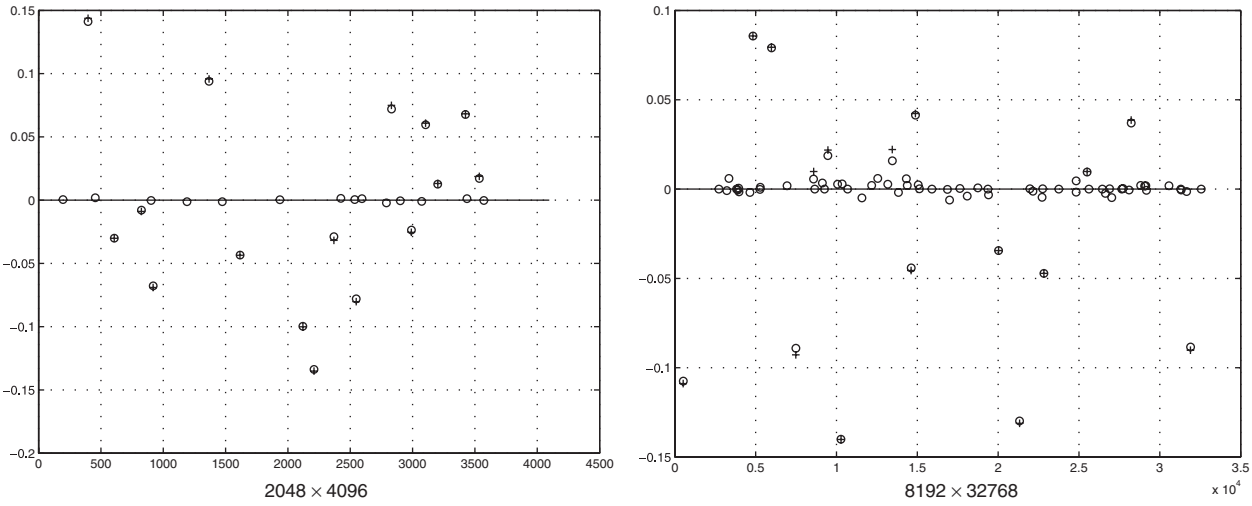


Fig. 1. ℓ_1-recovery: +: true signal; o: RMDSA recovery.

randomization can help when minimizing the objective in Eq. (1) over the unit box rather than over an ℓ_1-ball.

To summarize my opinion, I do not think that randomization can be considered as a universal, or even as a dominating, tool in convex optimization, and in this respect I would slightly disagree with the title of Campi's article : "Why resorting to fate *is* wise?" What I believe is true is that resorting to fate *can be* wise, and that we should look systematically for the cases where we do gain from randomization, and that the latter, being a well-established powerful tool in theoretical computer science, is about to become a powerful tool in applications.

In conclusion, I want to remark on a completely different issue. Exponential bounds on large deviations, like Hoeffding's Inequality cited in Campi's article, imply that in many cases "reliability of a randomized algorithm is costless" — we can make the failure probability to be $\leq 10^{-20}$, or $\leq 10^{-100}$ by a quite moderate increase in computational complexity (cf. how probability of failure β enters the complexity bounds we have presented). I wonder how the statements of this type should be interpreted. In the context of randomized algorithms, a conclusion "the probability of failure is $\leq 10^{-20}$" would be solid if the algorithm were working with an ideal generator of random numbers, which is not the case in reality. With all due respect to the tremendous progress in the area of designing generators of pseudo-random numbers over the last decades[4], it is difficult to believe that the quality of generators we use indeed justifies conclusions involving probabilities like 10^{-20}. I wonder whether we can just believe that "for all practical purposes," the existing generators can be considered as ideal ones, or some level of caution is needed here.

References

1. Grigoriadis M, Khachiyan L. A sublinear-time randomized approximation algorithm for matrix games. *Oper. Res. Lett.*, 1995; 18: 53–58.
2. Judistky A, Nemirovski A, Tauvel C. *Solving variational inequalities with stochastic mirror prox algorithm.* http://www.isye.gatech.edu/″nemirovs/SMP.pdf
3. Nemirovski A, Yudin D. *Problem complexity and method efficiency in optimization*. Nauka Publishers, Moscow, 1979 (in Russian); English translation: John Wiley & Sons, 1983.
4. Nemirovski A. Prox-method with rate of convergence $O(1 = t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM J. Optim.,* 2004; 15: 229–251.
5. Nemirovski A, Juditsky A, Lan G, Shapiro A. Stochastic approximation approach to stochastic programming. *SIAM J. Optim,* 2009; 19(4): 1574–1609.
6. Nesterov Y. Smooth minimization of non-smooth functions. *Math. Program.*, 2005; 103(1): 127–152.
7. Robbins H, Monro S. A stochastic approximation method. *Ann. Math. Stat*., 1951; 22: 400–407.
8. Rastrigin LA. Random search in optimization of multi-parametric systems, Publishing House "Zinantne," Riga, 1965 (in Russian).
9. Widgerson A. *P*, *NP* and mathematics—A computational complexity perspective. In M Sanz-Sóle, J Soria, JL Varona, J Verdera (Eds.), *Proceedings of the International Congress of Mathematicians, Madrid, August 22–30, Spain, Volume 1: Plenary Lectures and Ceremonies*, European Mathematical Society, 2007.

Discussion on: "Why Is Resorting to Fate Wise? A Critical Look at Randomized Algorithms in Systems and Control"

Jan C. Willems*

ESAT/SCD (SISTA), K.U. Leuven, Kasteelpark Arenberg 10, 3001 Leuven, Belgium

1. Introduction

Probability is one of the success stories of applied mathematics. It is universally used, from statistical physics to quantum mechanics, from econometrics to financial mathematics, from information theory to control, from psychology and social sciences to medicine. Unfortunately, in many applications of probability, very little attention is paid to the modeling aspect. That is, the interpretation of the probability used in the model is seldom discussed, and it is rarely explained how one comes to the numerical values of the distributions of the random variables used in the model. The aim of this communication is to put forward some remarks related to the use of probability in Systems and Control.

*E-mail: Jan.Willems@esat.kuleuven.be, http://www.esat.kuleuven.be/∼jwillems

[4] This progress indeed is tremendous: I remember times where people working with Monte-Carlo simulation used tapes where "physical white noise" of some kind was recorded, and a good tape of this type was a key to success.

2. Interpretations of Probability

One of the main difficulties both in using probabilistic models and in criticizing their use is that there are widely diverging interpretations of what probability means. Libraries full of books have been written on the topic on interpretation of probability, starting at the time of Pascal and continuing to the present day. See [2] for a comprehensive treatise, and [5] for some remarks and references.

Two main views have emerged among an uncountable number of intermediate nuances.

1. Probability as a subjective notion, as *degree of belief*.
2. Probability as an objective notion, as *relative frequency*.

The distinction between these two interpretations can be illustrated by considering coin tossing. To the question "*What is the probability of heads?*" the subjectivist answers $\frac{1}{2}$ because there is no reason to assume that tails are more likely than heads, or vice-versa. The subjectivist does not claim to predict what will happen when the coin is actually flipped. The answer quantifies the person's individual belief. The objectivist, however, argues that the probability of heads is $\frac{1}{2}$ because it is claimed to be a physical law that in a repeated experiment with the number of tosses going to ∞, the average number of heads will be $\frac{1}{2}$.

This example is perhaps atypical because it could be argued that the subjectivist argues $\frac{1}{2}$ because he or she believes that the objectivist's relative frequency of a repeated toss will turn out to be $\frac{1}{2}$. For repeatable experiments, there is some agreement between both views. However, in other situations, the distinction is more striking. If a sports commentator states that the probability of the Dutch team winning the World Cup in South Africa is 0.1, then it is difficult to interpret this statement as anything but the commentator's subjective belief. However, if an official of the registry of motor vehicles with knowledge of the prices of the automobiles sold in a country last year states that the probability of the price being below P is 0.5, then this is obviously a relative frequency.

For ease of exposition, we left out a third interpretation, namely

3. Probability as *propensity*.

The propensity interpretation, due to Karl Popper, brought a logical foundation to the single-event probability required in the physical probabilistic interpretation of the wave function in Quantum Mechanics.

3. Probability in Systems and Control

At least three main areas of Systems and Control are dominated by probability: filtering, system identification and stochastic control. It would take us too far to analyze the use of probability in each of these areas. We therefore limit our remarks to filtering.

In continuous time, and over an infinite interval, the filtering problem may be formulated as follows. Given two (vector-valued) signals, $z : \mathbb{R} \to \mathbb{R}^{\mathrm{z}}$, the to-be-estimated signal, and $y : \mathbb{R} \to \mathbb{R}^{\mathrm{y}}$, the observed signal, construct a filter F that takes y into $\hat{z} : \mathbb{R} \to \mathbb{R}^{\mathrm{z}}$, the estimate of z. F is a map that takes (a suitable subset of) $\mathbb{R}^{\mathrm{y}}$-valued functions on $\mathbb{R}$ into $\mathbb{R}^{\mathrm{z}}$-valued functions on $\mathbb{R}$. A basic restriction is that F should be non-anticipating, that is $y_1(t) = y_2(t)$ for $t \leq T$ should imply $F(y_1)(t) = F(y_2)(t)$ for $t \leq T$. Filtering is a very well-motivated problem, and numerous applications of it can immediately be seen. However, in order to set up an algorithm to construct F, we must clarify how z and y are related, so that y contains information allowing to obtain a reasonable estimate $\hat{z}$.

3.1. Wiener and Kalman Filtering

Wiener masterly solved this problem by assuming that (z, y) is a realization of a stochastic vector process and taking for $\hat{z}(t) = F(y)(t) = \mathcal{E}[z(t)|y(t')$ for $t' \leq t]$. The problem thus became a precise mathematical one, formulas involving spectral factorization were derived for the filter F in the stationary Gaussian case, and a research field was born that is very succesful up to the present day. The Kalman filter addresses in essence the same problem as Wiener did, but, by taking a very convenient representation of the stochastic process (z, y), a filter algorithm was obtained that is far superior and much more easily generalizable.

The problem with the stochastic formulation of the filtering problem is that it requires to model (z, y) probabilistically. Presumably, in the case that we use a frequentist interpretation, this should be done (in the zero-mean Gaussian stationary case) by obtaining the autocorrelation function of (z, y) using statistical methods. However, statistics typically assumes a probabilistic framework to begin with, and therefore it assumes the implied existence and persistence of the limits required to define relative frequencies. But where would, for a physical signal, this statistical regularity of (z, y) come from? Surely, there are some applications where (z, y) can indeed be modeled well as a stochastic process, but these are, in my opinion, few and far between. What physical laws ensure that outcomes of a real physical signal is a realization of a stochastic process in the frequentist sense? If, however, we use a subjective interpretation of probability, then it should be explained where the detailed numerical values of the degree of belief required for the autocorrelation function of (z, y) come from.

3.2. Least Squares Filtering

There is an interpretation of the filtering problem that avoids probability. This is most easily explained in the state space setting of the Kalman filter. Assume that (z, y) is modeled as

$$\frac{d}{dt}x = Ax + Bw,\ y = Cx + Dw,\ z = Hx. \tag{1}$$

In words, the relationship between z and y stems from the fact that they are both outputs of a linear system that is driven by an input $w : \mathbb{R} \to \mathbb{R}^{\mathrm{w}}$. Think of w as an underlying latent variable that serves to model (z, y).

Classically, w is assumed to be a stochastic process with known distribution, for example white noise. However, we could also proceed on a purely deterministic setting as follows. In order not to get into limit arguments that are not germane to our purposes, assume that the filtering interval is $[0, \infty)$, instead of $\mathbb{R}$ as in the previous section. The signal $z : [0, \infty) \to \mathbb{R}^{\mathrm{z}}$ needs to be estimated from the observations $y : [0, \infty) \to \mathbb{R}^{\mathrm{y}}$, and the estimate $\hat{z}(t)$ can only depend on $y(t')$ for $0 \le t' \le t$. The observed trajectory y and the to-be-estimated trajectory z are completely determined by the corresponding input $w : [0, \infty) \to \mathbb{R}^{\mathrm{w}}$ and initial state $x(0)$. Now, in order to compute $\hat{z}(t)$, use the input w and initial state $x(0)$ that minimize

$$J(w, x(0)) = \int_0^t ||w(t')||^2\, dt' + ||x(0)||_Q \tag{2}$$

over all $(w, x(0))$ that generate the observations $y(t)$ for $0 \le t' \le t$. Here $Q \ge 0$ is a suitable weighting matrix.The estimate $\hat{z}(t)$ defined this way obviously depends on the observations $y(t)$ for $0 \le t' \le t$ and on the model parameters A, B, C, D, H, Q. It can be shown that this minimization leads exactly to the Kalman filter formulas. This result is more or less obvious from a maximum likelihood interpretation of the Kalman filter, and is derived very clearly in the textbook [3] (see also [4], where more references can be found).

This least squares interpretation of the Kalman filter uses the model (1) in an essential way. The use of such representations was new when the Kalman filter was derived. The least squares interpretation of the filtering algorithm readily extends to the Wiener filtering formulas, using a similar representation as Eq. (1). However, such models were not available when the Wiener filter was first derived, whereas models with (z, y) a stochastic process were very much part of mathematics and probability. The adoption of a stochastic process framework allowed Wiener to pose the filtering problem in precise mathematical terms. This fact, and not the underlying physics, was undoubtedly the main motivation for introducing stochastics in the filtering problem.

Using the least squares interpretation of the Wiener and Kalman filter shifts the burden of justifying the methodology and algorithms from the *descriptive* to the *prescriptive*. The stochastic process interpretation makes claims regarding the model of reality, in other words, stochastic assumptions are part of the physics, of the descriptive part of the problem. However, using the minimization of 2 specifies the performance. It is up to the designer to choose this as the prescriptive part of the problem. The choice of 2 as the functional to be minimized does not impose anything on the system model, it merely articulates the designer's preference.

In a sense, it is possible to interpret the least squares algorithm in terms of a subjective "degree of belief." However, this interpretation only requires one to state that the $(w, x(0))$ which minimizes 2 is the most believable explanation of the observations. This interpretation is much more parsimonious than the subjective probabilistic interpretation of $(w, x(0))$, which requires giving numerical values of the degree of belief of many more events.

Similar least squares interpretations of many of the algorithms used in system identification and in stochastic control are readily given.There is also the $\mathscr{H}_2$ interpretation of the Wiener and Kalman filter and its extension to $\mathscr{H}_\infty$ filtering. When the optimal LQG controller is interpreted as minimizing the $\mathscr{H}_2$ norm of the closed loop system, we also shift again the burden of the descriptive to the prescriptive, while opening up the generalization to $\mathscr{H}_\infty$ control. These deterministic methods of designing filters and controllers are, in my opinion, very much to be preferred above the stochastic formulations, precisely because they shift the problem justification from the descriptive part of the model to the prescriptive part of the design. Often the argument is put forward that since the stochastic interpretation and the least squares interpretation of filtering and system identification algorithms lead to the same formulas, it is pointless to argue that one interpretation is to be preferred to another one. Of course, this is a two-edged sword and cannot be used as a defense of the stochastic interpretation. However, the fact that the same formulas are obtained does not mean that it is wise or valid to suggest an unverified structure on a model.

In engineering (and prescriptive aspects of economics) one can, it seems to me, also take the following intermediate position as a justification of the use of stochastics. An algorithm-based engineering device, say in signal processing, communication, or control, comes with a set of "certificates," that guarantee that the device or the algorithm will work well under certain specific circumstances. These circumstances need not be the ones under which the device will be used in practice. They may even be circumstances which cannot happen in the real world. These certificates are merely guarantees of good performance under benchmark conditions. Examples of such

performance guarantees may be that an error correcting code corrects an encoded message that is received with on the average not more than a certain percentage of errors, or that a filter generates the conditional expectation of an unobserved signal from an observed one under certain prescribed stochastic assumptions, or that a controller ensures robust stability if the plant is in a certain neighborhood of a nominal one.

4. Why Resorting to Fate Can Be Wise

In [1], it is argued that in many problems in Systems and Control, advantage can be taken of randomization. It is difficult to argue with this. Randomized algorithms are used for secure communication in cryptography, they lead to game theoretic equilibria, they can be effective for evaluating integrals, and so forth. As such, many of the points made in [1] are eminently valid.

However, it becomes more difficult sometimes to follow the thesis in [1] when it comes to control. I find it difficult to sympathize with the discussion surrounding what is called "Bayesian approach." The control problem discussed assumes that we have a situation in which a designer needs to design one single control algorithm for a whole family of plants. The designer knows exactly (from measurements?) the probability (in the sense of degree of belief?) of the various plants but seems to be unable to measure the unknown parameters in the actual plant on which the controller will act. It would have been nice to see a description of such a situation. I struggled to imagine a convincing engineering example where such a problem would come up. Where would the numerical values of this degree of belief probability come from? However, could it be that once again probability is used here as a panacea for uncertainty, as a way to make the problem into a mathematical one, without going back to the physics?

5. Let Us Get the Physics Right

It is my belief that modeling is the most neglected aspect of theoretical engineering in general and, more specifically, of Systems and Control. This is very evident in areas which use probabilistic models. To begin with, the interpretation of probability is seldom explained. This would pose no problem if the interpretation of a particular concept is evident, but in the case of probability with its highly divergent interpretations, this neglect to explain the interpretation is objectionable. Often, it is vaguely implied that a frequentist interpretation is used. But then, why can measurement inaccuracy be modeled as an additive stochastic process? This is perhaps more or less acceptable to model the effect of quantization, but what about all the other sources of measurement uncertainty? Why should an unmeasured nuisance signal in system identification be a stochastic process? Why should a communication channel change a 0 to a 1 and a 1 to a 0 with a fixed relative frequency? Where would this regularity of error generation come from? I do not claim that in many circumstances, these probabilistic methods cannot be rationalized. What I claim is that by and large, we are doing research and teaching without bothering to explain the physics that leads to probabilistic models.

The neglect of the physics in Systems and Control is much more widespread than for probability alone. As I have argued extensively before [6], it applies to the input/output thinking that is universally used for modeling open systems. Physical systems are not signal processors. The methods based on inputs and outputs are especially awkward for system interconnection. Interconnection of physical systems leads to variable sharing, not to output-to-input assignment. This is already evident in simple systems as electrical circuits and mechanical devices. There is no reason why this situation should suddenly be different for complex systems, as those found in biology. Signal flow graphs have their place, but as a description of the functioning of an interconnected physical system, they miss the crucial point of expressing what interconnection entails. The input/output setting poses difficulties in the first and simplest examples.

How can such a situation have occurred? Why is the physics of models not more prominently present in areas as Systems and Control? Why are probability, inputs, outputs and signal flow graphs used without analyzing the physical situations to which they claim to pertain? The explanation, in my opinion, lies in the sociology of science. Normal science uses an established paradigm in which to operate. When a problem is cast in an input/output setting with disturbances modeled as stochastic processes, we are operating in a clear and often sophisticated mathematical framework, with results that may be difficult to obtain and hard to prove and that are verifiable mathematically. The results are judged by their mathematical depth and difficulty. In other words, the explanation lies in the *Lure of Mathematics*. There is no other explanation.

References

1. Campi MC. Why is resorting to fate wise? A critical look at randomized algorithms in systems and control. *Eur J Control* 2010; 16(5): 419–430
2. Jayes ET. *Probability theory: The logic of science*. Cambridge University Press, Cambridge, UK, 2003
3. Sontag ED. *Mathematical control theory: Deterministic finite dimensional systems*. Springer Verlag, New York, 1990
4. Willems JC. Deterministic least squares filtering. *J Econ* 2004; 118: 341–370

5. Willems JC. Thoughts on system identification. *Control of Uncertain Systems: Modelling, Approximation, and Design: Springer Lecture Notes in Control and Information Sciences* 2006; 329: 389–416

6. Willems JC. The behavioral approach to open and interconnected systems. *Control Sys Mag* 2007; 27: 46–99

Final Comments by the Author M.C. Campi

It has been a pleasure receiving and reading the discussion articles three outstanding scientists, namely, S. K. Mitter, A. Nemirovski and J. C. Willems have generously provided. Their comments increase the value of the discussion.

In my article [1], I presented simple examples to get facts and concepts underlying randomized algorithms easily through. Arkadi Nemirovski presents nontrivial examples in optimization which suitably complement those in my article and add concreteness to the role of randomized algorithms in difficult problems.

Sanjoy Mitter's comments open up new directions where randomization is used and broaden the scope of the discussion in a significant manner. I would like here to only add some remarks about the PAC learning methodology Mitter mentions in his comment.

1. The Lesson of PAC Learning

In PAC learning, [2], one is given a set of data and is asked to select a hypothesis from a given class of functions. The acronym PAC, probably approximately correct, refers to the fact that, with *high probability*, the selected function must have low generalization error that is it is *approximately correct*. PAC learning offers in a sense a broader set-up than it is usually done in system identification, while it is narrower in another way. It is broader because it assumes little or nothing about the underlying distribution that generates the data, and it is narrower because data are assumed to be independent one of the other. I believe that importing this framework into system identification is most desirable, the main challenge in this process being the treatment of dynamics.

From a more general point of view, in my appreciation the PAC paradigm teaches us an important lesson: meaningful results can be achieved *in the presence of little prior information* provided that we relax our requirements for precision: we should be content with answers which are approximately correct most of the time. Both aspects, *approximate correctness* and *most of the time* as opposed to always are intimately tied to the generality of the approach. Randomized methods are a means to pursue this philosophy, and this is one reason why randomized methods provide powerful tools of synthesis, see e.g. the examples in the discussion paper of A. Nemirovski.

Turning to Jan Willems' discussion article, I would like to say that I have had the opportunity to talk more than once with him on topics related to probability and its interpretation, and I share his view that justifying probabilistic models is most important and yet this issue has at times been underestimated by the control and systems community. This is particularly evident in the field of system identification, where much of my interest lies. Educating ourselves to the use of probability is a priority and, in my opinion, this is even more important than replacing probability with alternative models.

2. The Need for Probability

A central issue in filtering and identification is to provide guarantees in the form of quantitative statements capable to credit an estimation result with reliability. This is relevant to the practice of these methods and it is necessary for their scientific use. A single estimate (point estimation) is unsuitable to the purpose of providing guarantees since a single estimate can hardly be announced to be the true value. Intervals and regions have to be used instead[1]. However, claims like "the true parameter value *certainly* lies in this interval" can only be made under very stringent assumptions on the noise, assumptions that are difficult to justify from a modeling point of view. We therefore see that we have more realistically to seek guarantees valid for most of the noise sequences, not for all of them, and our goal is to look for regions having this property. However, if we really want to be quantitative, we need to provide ourselves with a mathematical tool to measure the "extension" of these regions, and this is measure theory; and when

[1] Interval estimation is a very well-established field in statistics, pioneered by masters such as J. Neyman, [3], A. Wald, [4], and J. W. Tukey, [5].

a measure is interpreted as "chance of happening", this measure is called probability. So, I believe, renouncing to use probability leaves us unarmed to tackle the challenge to be quantitative.

3. A Cautious Use of Probability

While probability is an essential element in the formulation of many problems, I again agree with Jan Willems that it is also true that probability should be used with care and it is preferable to design methods whose probabilistic justification can be made at different levels, that is with different degrees of probabilistic knowledge. Kalman filtering is one such examples: in a highly structured Gaussian framework, a Kalman filter computes the conditional mean, the best *nonlinear* estimation in norm 2. However, only assuming knowledge of the second-order moments still permits one to justify the Kalman filter equations as a recursive method to derive the best *linear* norm 2 estimation.

There are fields where so-called distribution-free results are common. In Probably Approximately Correct (PAC) learning theory, for example, beautiful results have been derived that are valid independently of the underlying probability distribution. Thus, a probabilistic model is needed to justify the approach, but the results hold independently of the probability distribution that generates the data. To a similar result one can arrive in prediction using Interval Predictor Models (IPMs), as done in [6]. Justifying a result, algorithm or method without a full description of the underlying probabilistic model is important simply because this result, algorithm or method becomes more widely applicable and it gains credibility.

Randomized methods have a very special role in this discourse. As I pointed out in my article [1] the probability used in a randomized method is not introduced for the purpose of modeling, it is instead part of the algorithm, we create it for use in the algorithm and therefore we have no reason to doubt its validity. This is a significant positive mark in favor of the use of randomization.

4. Randomization and the Probability of Success

Jan Willems says that it has been difficult for him to follow the thesis in [1] when it comes to control. I take this opportunity to better clarify my view.

When we board a plane we would in principle like the idea that the plane crashes did not exist. However, planes do crash, and it is a relief to know that statistically a plane only crashes once every some 10^6 to 10^7 flights. If a full guarantee is not possible, we seek for a probabilistic guarantee, and this leads to a shift of the notion of robustness.

Further commenting on this point, I would like here to express doubts as to whether a full guarantee has always to be preferred to a probabilistic guarantee. Full guarantee is with respect to an assumed level of uncertainty, but how sure can we be about the level of uncertainty assumed? A robust design is tailored to the prescribed uncertainty level and falls apart if uncertainty is not in the prescribed uncertainty set. At times, it may be convenient to introduce a larger uncertainty set to cover more situations for which a robust design is not possible and allow for a small probability of failure. This approach may possibly lead to a design that safeguards against more situations simply because the algorithm accounts for all the situations in the larger uncertainty set when doing the design. Voltaire said "doubt is not a pleasant condition, but certainty is absurd"; I personally tend to look suspiciously at any theory expressing certainty.

5. About a Bayesian Perspective

In Section 4, Jan Willems comments about the Bayesian perspective that I have in Section 2.3 of my article [1] and says that a situation where "the designer knows exactly the relative frequency of the various plants, but seems to be unable to actually measure the unknown parameters in the actual plant" is unnatural. His point is my point. In Section 2.3, I argue that the big difference between the randomized approach and the Bayesian perspective is that in the latter poor modeling is a constant risk, a thing that cannot happen with the artificial probability introduced for use in a randomized algorithm.

References

1. Campi MC. Why is resorting to fate wise? A critical look at randomized algorithms. *Eur. J. Control*, 2010; this issue.
2. Valiant L. A theory of the learnable. *Commun. ACM*, 1984; 27: 1134–1142.
3. Neyman J. Outline of a theory of statistical estimation based on the classical theory of probability. *Philos. Trans. R. Soc. Lond. A, Math. Phys. Eng. Sci.*, 1937; 236: 333–380.
4. Wald A. An extension of Wilks method for setting tolerance limits. *Ann. Math. Stat.*, 1943; 14: 45–55.
5. Tukey JW. Nonparametric estimation II. Statistically equivalent blocks and tolerance regions—The continuous case. *Ann. Math. Stat.*, 1947; 18: 529–539.
6. Campi MC, Calafiore G, Garatti S. Interval predictor models: Identification and reliability. *Automatica*, 2009; 45: 382–392.