COMPUTER-AIDED DESIGN FOR QUALITY (CADQ)

Lihsin Liu, William A. Nazaret, and Richard G. Beale

This paper describes computer-aided design for quality—a new, cost-effective methodology for designing high-quality, manufacturable products using computer-aided design (CAD) systems. Traditionally, CAD systems have been used to simulate a product's characteristics for a given set of values for the design parameters. Thus, designers could study the effects of design choices on a product's quality before production started. Recent advances in computer hardware technology have given CAD systems increasingly complex and accurate simulation capabilities, permitting simultaneous examination of the effects of changes in many design parameters. But CADQ goes a step further. It provides powerful, statistical, model-fitting methods and optimization techniques that extend the traditional use of CAD systems. Given a set of quality-performance measures, a designer now can interactively pursue a product's optimal design, while considering manufacturing constraints and characteristics of the customers' environment. Through a case study, we show the application of CADQ to integrated-circuit design.

Richard G. Beale *is a supervisor in the Analog Design Department at AT&T Bell Laboratories, Reading, Pennsylvania.* ***Lihsin Liu*** *is a member of technical staff in the Quality Theory and Technology Department at AT&T Bell Laboratories, Holmdel, New Jersey, and* ***William A. Nazaret*** *was a member of technical staff in that department. Mr. Beale is responsible for the design and development of mixed-signal CMOS integrated circuits for digital transmission systems. He joined the company in 1980 and has a B.S. from the University of Virginia and an M.S. from Stanford University in electrical engineering, and an M.B.A. from the Wharton School of the University of Pennsylvania. Mr. Liu manages the CADQ project. He joined the company in 1986, and has a B.S. from the Tatung Institute of Technology (Taiwan, Republic of* (continued on page 60)

Design for Quality and Manufacturing

Many important considerations simultaneously come into play when a component is designed for use in an AT&T product. By far the most important is quality, which has several aspects such as cost, timeliness, and performance. But in the end, the components of a product to be designed and manufactured under a global quality approach—i.e., designed for reliability, manufacturability, serviceability, etc.—must fulfill the performance requirements dictated by customers.

Consider, for example, the task of designing a voltage-level-shifter integrated circuit (IC) for use in an AT&T lightwave repeater

Panel 1. Acronyms and Terms

ADVICE	a computer-aided design simulator for verifying integrated-circuit designs
B	bandwidth
CAD	computer-aided design
CADQ	computer-aided design for quality
CMOS	complementary metal-oxide semiconductor
F	closed-form function, where $Y \approx F(x_d, x_e)$
G	gain
GaAs	gallium arsenide
H	actual physical function
K	covariance kernel
L	a polynomial of degree no greater than 2
OA	orthogonal array
range	set of possible values for a component in x_c
S/N ratio	signal-to-noise ratio; here, a function of the ratio between the mean of a response and its variability
$\sigma_{G \times B}$	variance of gain by bandwidth
$\sigma_{V_{shift}}$	variance of voltage shift V_{shift}
$\mathbf{x}$	14-dimensional vector of circuit parameters
x_c	nominal value (or center) of a design variable (control factor)
x_d	design variables
x_e	external variables
x_n	union of all noise variables (noise factors)
x_v	perturbation of a design variable around its nominal value caused by manufacturing variability
variability	interval on which the x_n are allowed to vary
VLSI	very-large-scale integration
Y	responses or performance characteristics of the product
Z	a random function

unit. The primary performance requirement is that this circuit shift the input voltage, V_{in}, down by 5V (volts) to the output voltage, V_{out}. The IC's gain must stay as close to unity as possible and its bandwidth must be high, because the circuit is part of a high-frequency transmission system. Finally, the circuit's power consumption must not exceed a given limit. (Panel 1 defines acronyms and terms used in this paper.)

Using a computer-aided design simulator (like AT&T's ADVICE system), the designer can produce a circuit design that complies with these requirements. (The ADVICE system, which was developed by AT&T Bell Laboratories, is a design-verification aid for IC engineering.) The design consists of a layout for the circuit, together with nominal values for the circuit's elements (transistors, resistors, capacitors).

CAD systems, like the ADVICE system, simplify the designer's task. They allow a designer to simulate the circuit's voltage level shift, gain, and bandwidth for any set of nominal values for the circuit elements. Thus, design tradeoffs can be examined before the circuit goes into production. A design produced this way is called a *nominal design*.

However, the results of manufacturing a nominally designed IC are likely to be disappointing, because many of the manufactured circuits will exhibit characteristics different from those predicted by the CAD system. Even some circuits that pass final test may fail to perform properly under real operating conditions.

What went wrong with the nominal design? Nothing. But nominal designs assume that the manufacturing processes and operating conditions are completely stable. In reality, even a well-controlled manufacturing process exhibits fluctuations, and most operating environments are variable.

Because of this, most designers are not satisfied with checking the performance of a single nominal design. Instead, they often study the performance of other designs that consider the effects of extreme or *worst-case* manufacturing and environmental conditions on the nominal design. Although reasonable, this approach cannot tell a designer about the overall effect of these variations on the design's performance.

What is needed is a form of *statistical design*,

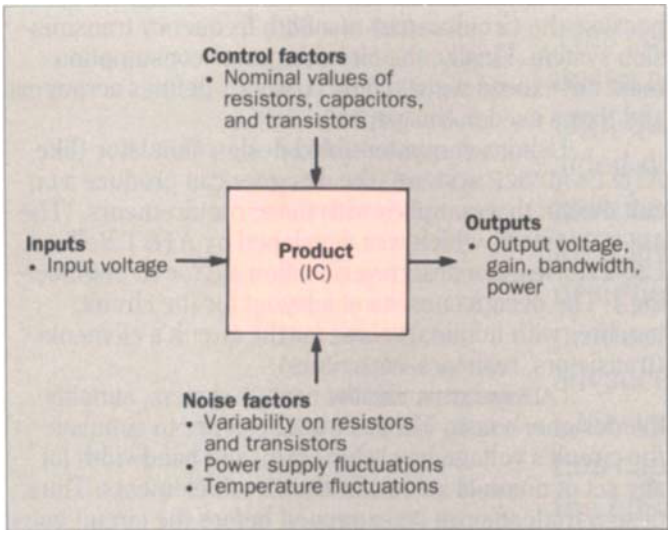

Figure 1. The elements of robust design. Control factors are under a designer's control and can be set and adjusted at will. Noise factors are physically or economically beyond a designer's control.

which explicitly brings into the design phase the random fluctuations that will affect the product's performance during and after manufacturing. Unfortunately, all this new information tremendously complicates matters. For one thing, the difficulty of simulating a product's behavior increases as the number of design parameters increases. Some designs, very-large-scale integration (VLSI) for instance, start out with an already large number of nominal design parameters. When we add all the fluctuation-related statistical parameters, the design problem becomes seemingly intractable.

In its current form, CAD technology is only of relative help for two reasons:

- CAD systems do not have special provisions to help a designer study high-dimensional design problems. (We assume that enough computational power and time will exist to deal with the problem's complexity.)
- CAD systems are nominal-design oriented. That is, there are no facilities to help the designer simulate and study a product's behavior under random perturbations to its nominal design specifications.

In this paper, we propose an approach to product design that addresses this problem. We start by reviewing a method called *robust design*[1,2] that has been used successfully in AT&T[1] and other companies to design high-quality, manufacturable products. The robust design method provides a structured way—a statistical approach—to overcome the problems associated with nominal designs. Next, we describe computer-aided design for quality (CADQ), an implementation of robust design specifically tailored for CAD environments. We discuss how CADQ uses statistical sampling techniques to help a designer deal with the complexity of studying high-dimensional design problems. We also explain how to use statistical models with raw outputs from the CAD system to reduce the computational expense of optimizing a product's performance. Finally, we describe our first implementation of CADQ, which is geared toward IC design and uses AT&T's ADVICE CAD system. We also show the results of using this implementation to design a gallium arsenide (GaAs) voltage-level-shifter integrated circuit.

Robust Design

The diagram in Figure 1 shows the elements involved in the design of a product. The center box represents the product (e.g., an IC) as a physical system. To the right, we see some of the product's outputs; here, output voltage, gain, bandwidth, and power consumption. These outputs are determined by the inputs (identified to the left of the box) and two classes of factors:

- *Control factors*—Design factors that a designer controls and can set and adjust at will. For an IC, control factors are the nominal values of resistors, capacitors, and transistor sizes, among others.
- *Noise factors*—Factors that a designer cannot control for either physical or economic reasons. For an IC, the noise factors include: manufacturing-variability

departures from the nominal values of resistors, fluctuations in input power, and variations in operating temperature or humidity. Noise factors can be internal or external to the design.

As we mentioned earlier, it is relatively easy to choose values for control factors that, in the absence of noise factors, produce a circuit design that meets a given set of performance requirements. That is what we called a nominal design. The real problem is to account for the effect that noise factors will have on the performance of a given product design. Circuit designers refer to this as the *statistical-design problem.*

One can attack the statistical-design problem by using the so-called *worst-case-design* approach.[3] Under this approach, we first determine worst-case directions for the noise factors. These are the directions along which changes in these factors cause the product's performance to deteriorate. Once we have those directions, the statistical-design problem becomes a deterministic one. That is, we consider the product's performance for each set of worst-case conditions or extreme values of the noise factors along each worst-case direction.

However, a solution to the worst-case-design problem might not exist because the control-factor values often affect the product's performance in conflicting ways under different worst-case conditions. In other words, a design that works for a given worst-case condition might not work for another worst-case condition. Even if no conflicts arise, worst-case analysis often produces conservative, overly expensive designs.

A more serious problem is: Worst-case design assumes that the noise factors have a monotonic influence on the product's performance. Verifying this assumption in practice is often as difficult as the design problem itself!

A successful approach to solving the statistical-design problem must be able to find control-factor values that minimize the overall effect of the noise factors on product performance. The so-called robust-design approach, introduced by Dr. Genichi Taguchi,[4] is based on this principle and has become popular in the last five years. In robust design, the objective is to find those control-factor values that minimize the variability of product performance attributed to noise factors. A product designed this way is called *robust* for obvious reasons.

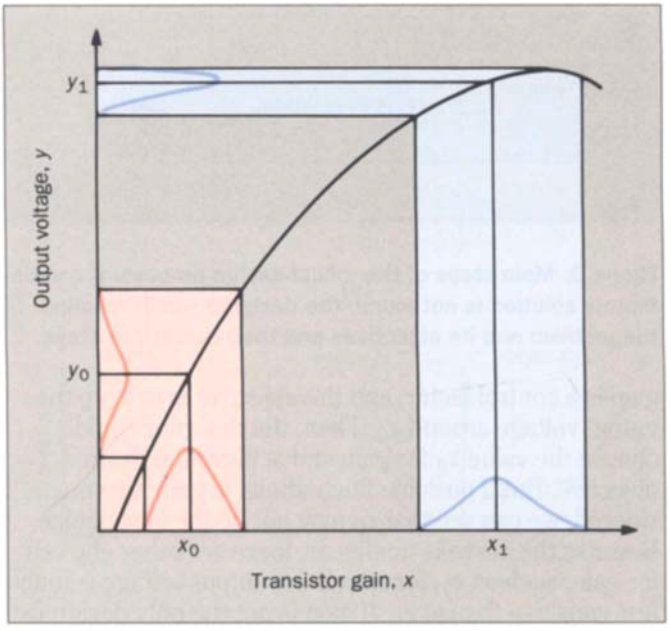

Figure 2. Exploiting nonlinearity to reduce variability. If gain is a control factor in the IC's design, then a designer who wants to keep the output voltage near y_0 would choose x_0 for gain. But other choices may exist if fluctuations in gain must be considered. For example, the output voltage is less sensitive to gain at x_1.

Robust design exploits the existing nonlinear relationships between control factors and the product's outputs to find a point in the design space where the variability that the noise factors introduce does not degrade product performance. Figure 2 illustrates this concept.

In this diagram, we see the output voltage of an IC represented as a function of transistor gain. Suppose

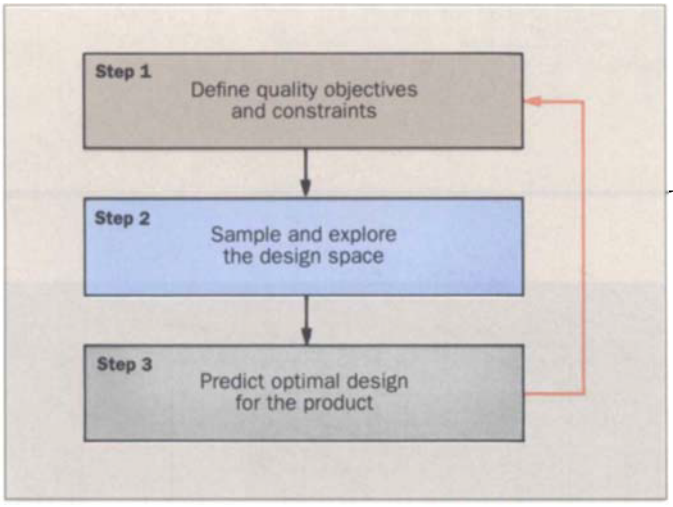

Figure 3. Main steps of the robust-design process. If a satisfactory solution is not found, the designer would redefine the problem and its objectives and then repeat the steps.

gain is a control factor, and the objective is to keep the output voltage around y_0. Then, the designer could choose the value x_0 for gain and achieve the desired objective. But if possible fluctuations in gain are considered, we can see that x_0 may not be the best choice. Because the curve is nonlinear, there are other choices for gain, such as x_1, for which the output voltage is much less sensitive than at x_0. If gain is not the only design factor, we can now use some other factor—unrelated to gain—to lower the voltage to target, while avoiding the effects of gain variability.

Summarizing, robust design allows designers to incorporate variations into the design. Among the most important consequences of using this approach are:

- Less rework is needed to get a satisfactory design.
- Designs are more manufacturable, because the noise characteristics of the manufacturing process can be taken into account.
- High yields result from reduced variability in product performance.

Computer-Aided Design for Quality

Here, we describe our implementation of robust design for CAD systems and explain how it is better than traditional implementations in non-CAD environments.

The Robust-Design Process. Figure 3 shows the main steps of the robust-design process:

1. *Define quality objectives and constraints.* First, the designers must clearly specify all elements involved in the design of the product: control factors, noise factors, responses, and performance measures. (We described control and noise factors earlier.) *Responses* are the product's characteristics used for judging its quality, and *performance measures* are the metrics used on the responses to gauge quality. With all these elements in hand, performance targets and constraints on the control factors are then specified.
2. *Sample and explore the design space.* Here, we study the interrelationships among the control factors, noise factors, and responses. The goal is to determine what control factors are important in minimizing the effects of noise factors on the responses. Responses are typically analyzed in statistical terms, such as mean and variance. Statistical sampling methods are used to select a finite number of product designs for analysis. Thus, one can predict the product's behavior under other designs from information contained in this limited design sample.
3. *Predict the optimal design for the product.* Using the information gathered in Step 2, the designer now attempts to find the control-factor settings that minimize the quality lost from the effects of noise factors. To measure quality loss, we use the performance measures defined in Step 1. If no satisfactory solution is found, we may need to redefine the design problem and its objectives. This is represented in Figure 3 by the arrow going back to the first step.

This view of robust design is common to all implementations known to date. However, each of the steps may be implemented in different ways.

Taguchi's Method. Taguchi's original implementation[4] calls for using *signal-to-noise ratios* (S/N ratios) as performance measures in Step 1. S/N ratios are functions of the ratio between the mean of a response and its variability. This is sometimes known in the statistical literature as the *coefficient of variation*, and represents a measure of the relative size of variability in the response. In this sense, maximizing the S/N ratio for a given

response corresponds to minimizing the effect of the noise factors on the response's variability.

In Step 2, Taguchi's method of sampling uses orthogonal arrays (OAs); see Figure 4. As Figure 4a illustrates, a *control array* defines the settings (nominal values) of four control factors to be tried. The first and third factors are the nominal values for resistors, the second is for the capacitor, and the fourth is for transistor size. A *noise array* does the same for two noise factors: voltage and temperature. To construct the *product array*, we combine the noise array with each row of the control array. The result is what is known as an *experimental plan* (that is, a set of different product designs to be tested). The experiment consists of obtaining, either by physical experimentation or simulation, readings of the performance measures for each design.

For Step 3, Taguchi's method exploits the statistical properties of orthogonal arrays to obtain the main effects of each control factor on the S/N ratio. To do this, we average the values of the S/N ratio obtained at each setting of these factors during the experiment. Figure 4b illustrates the process. Then, we use the resulting main effects to predict an optimal design by choosing the setting for each control factor that yields the largest S/N ratio among all the settings.

When multiple performance characteristics are to be optimized, this procedure is done for each characteristic and, if possible, the overall results are reconciled. The reconciliation procedures are subjective.

This implementation of robust design has been widely used in Japan and the U.S. because of its economy, simplicity, and superiority over nominal and other one-factor-at-a-time design methods. However, it was developed when computing power and software sophistication were limited, which is reflected in several important limitations. The most important limitations are:

- The rationale for constructing an S/N ratio is very ad hoc and is difficult to generalize to multiple responses. Also, it is difficult to incorporate manufacturing and operational variables into the problem.

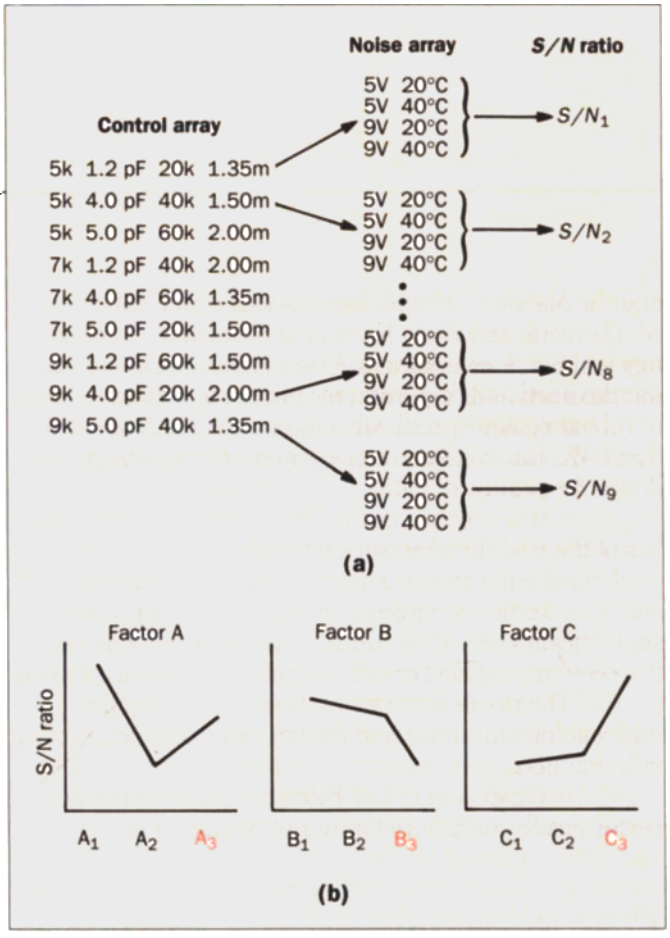

Figure 4. For Step 1 of Figure 3, Taguchi used signal-to-noise (S/N) ratios as performance measures. In step 2, he used (a) orthogonal arrays for sampling and exploited their statistical properties to obtain (b) the main effects of each control factor. (The best setting for each factor is highlighted.)

- When many control and noise factors are involved, the product-array technique can produce large experiments.
- Optimization is not done in a sophisticated way. In particular, a control factor's optimal setting is restricted to being among the discrete set of levels selected in Step 2. Specifically, in Figure 4b, it is impossible to tell if a value of factor B exists between levels B_1 and B_2 that results in a larger S/N ratio.

CAD for Robust Design. At a November 1986 workshop on design for quality, sponsored jointly by AT&T

and the National Science Foundation, J. H. Hooper, M. Hamami, and W. A. Nazaret of the Quality Technology and Services Center at AT&T Bell Laboratories laid out the need and requirements for a new implementation of robust design specifically tailored to the CAD environment. We call this new implementation *computer-aided design for quality*, or CADQ.

CADQ overcomes the limitations mentioned earlier of the traditional approach to robust design. To implement each step of Figure 3, CADQ uses statistical and optimization methods that exploit the power and sophistication of CAD systems. Yet it does not abandon the economy aspects of original robust-design implementations. The result is a state-of-the-art, cost-effective methodology for designing quality and manufacturability into products.

How CADQ works. Let Y denote the responses or performance characteristics of the product. Let:

$$Y = H(x_d, x_e) \quad (1)$$

describe the relationship between the values of the outputs and the values of the design variables x_d and external variables x_e. H is the actual physical function. Design variables, in turn, can be expressed as the sum of two components:

$$x_d = x_c + x_v \quad (2)$$

where the x_c represent nominal values (*centers*), and the x_v represent perturbations around their nominal values caused by manufacturing variability. External variables represent factors that are external to the design and affect the product's performance during operation. For an IC, the x_c in equation (2) could be nominal values of resistors and transistors, and the x_v would be the variabilities around the nominal values of resistors and transistors. The external variables could be operating temperature and power-supply voltage.

Given that, we can rewrite equation (1) as:

$$Y = H(x_c, x_n) \quad (3)$$

where x_c represents the nominal value of design variables (control factors), and x_n the union of all the other variables (noise factors). Under equation (3), we have built in the standard control- and noise-factor notation.

Let us assume that we measure product performance as Y's discrepancy with respect to a target value Y_0. Then, the robust-design problem can be posed as:

$$\min_{x_c} E_{x_n}(Y - Y_0)^2 \quad (4)$$

where the function E_{x_n} is the statistical expectation over the distribution of x_n.

In other words, the objective is to find those values for the control factors that minimize the expected deviation of Y around its target value Y_0 in the presence of noise factors.

To solve equation (4) directly, one must choose a mathematical optimization algorithm[5] that, starting from an initial choice of x_c, sequentially picks new points along carefully chosen directions, until it cannot further reduce the above expected squared error.

However, the proposed solution hides a problem: The expectation in equation (4) needs to be calculated for every new point x_c to be tried, which can be computationally expensive. If we can use the CAD system to obtain values of Y for any choice of x_c and x_n, then we can approximate the expectation using a Monte Carlo average. A closed-form expression for the expectation in terms of the x_c is seldom known, because the function H in equation (1) is not known explicitly. Instead, the function H is simulated within the CAD system as a numerical solution to a set of differential equations that govern the product's physical behavior. Therefore, a Monte Carlo approximation to the expected value would involve potentially hundreds of calls to the CAD system, each expensive in itself.

In spite of this, we would still like to solve the robust-design problem in terms of equation (4), because

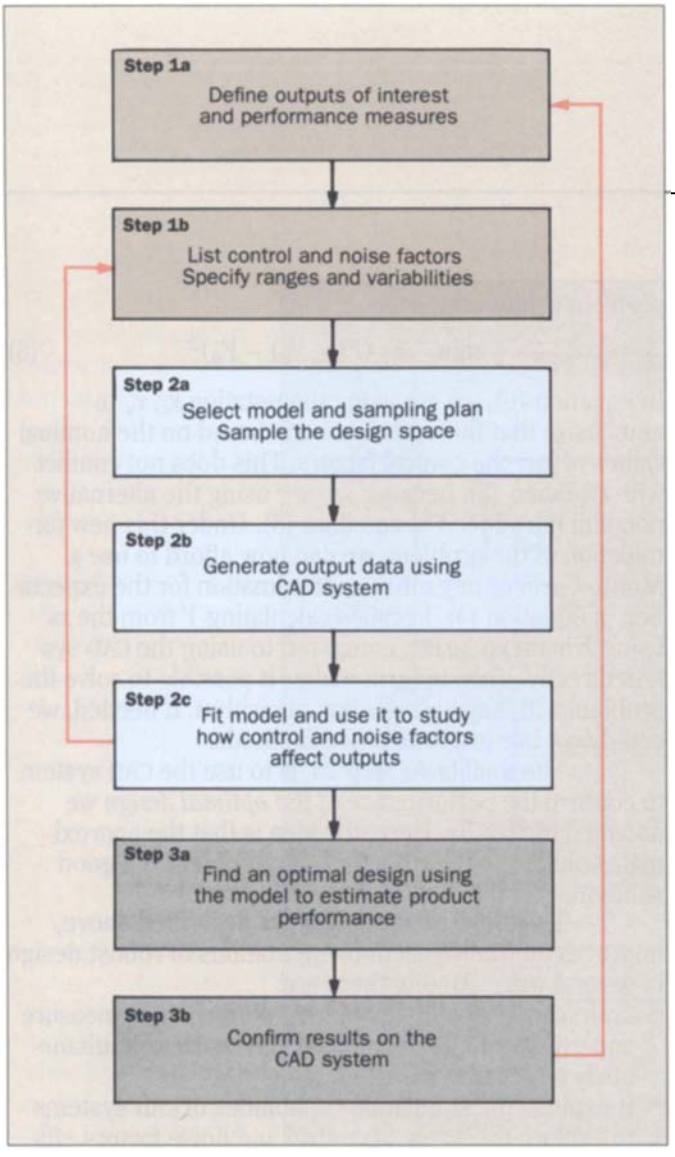

Figure 5. The CADQ flow diagram expands the steps in Figure 3. The strategy here is to find *Y* (the value of the product's responses or performance characteristics) for a selected number of designs and then use this information to interpolate *Y* for other designs. Information obtained on each iteration is used to refine the ranges and variabilities, until a solution is found.

this formulation allows us to deal with many different design problems in a unified way. For instance, we could use performance measures other than square deviation and could consider constraints on the x_c without invalidating equation (4). We want to avoid problem-dependent formulations because they compromise our ability to implement algorithms and software for general use.

Approximation Principle. Clearly, we need to find a way to solve equation (4) that avoids the computational expense of using the CAD system too often. Our solution is based on a simple principle that we call the *approximation principle*: Use the CAD system only to find the value of Y for a selected number of designs, and use this information to interpolate the value of Y for other designs. We now describe this strategy in more detail.

Figure 5 contains a flow diagram of the CADQ process. Steps 1a and 1b here are a finer breakdown of Step 1 in Figure 3. In Step 1b, *ranges* refer to the set of possible values for each component in x_c, and *variabilities* refer to the interval on which the x_n are allowed to vary. Steps 2a, 2b, and 2c correspond to subactivities in Step 2 of Figure 3. We implement the steps in Figure 5 based on the approximation principle stated in the previous paragraph. That is, we will find a closed-form function F, such that:

$$Y \approx F(x_d, x_e) \qquad (5)$$

within the range specified in Step 1 (Figure 3) to a satisfactory degree of approximation of H in equation (1). Here, as in equation (1), x_d and x_e refer to design and external variables, respectively.

In other words, we are trying to model the physical system in its original form—i.e., product outputs as a function of all the inputs. If we find such an F, we then can proceed with the rest of the robust-design process. But by using F as a surrogate for H in equation (1), we avoid the expense of having to use the CAD system in the optimization step.

Here, we also see one main difference between the CADQ and Taguchi approaches. CADQ models the outputs as a function of all the inputs (design and external

variables) and then uses the approximation function for performance calculations and optimization. Taguchi, instead, calculates performances directly from the raw data and then models the performances (S/N ratios) as a function of the control factors. CADQ's approach offers more simplicity and flexibility. It usually is simpler to model the outputs as a function of the inputs than to model a nonlinear function of the outputs, such as an S/N ratio. Also, once the designer finds a good approximation, he or she can use it for calculating a variety of different performance measures, if necessary. This is more flexible than having to refit the model every time a new performance measure is considered.

We start the process of building F (Step 2a) by selecting a parametric family of functions (polynomials, splines, etc.) from which we will choose a best fitting F. Then, we choose a sample of points in the (x_d, x_e) space. The use of an inexpensive sampling plan with good statistical properties is crucial to the success of the overall strategy. The sampled points and their associated values of Y are to be regarded as data, and the parametric family of functions is to be regarded as a model. Any correspondence with statistical-model-fitting terminology is *not* at all coincidental.

In Step 2b, we use the CAD system to generate values of the outputs Y for each of the sample points selected in Step 2a.

In Step 2c, we choose a best fitting F based on the data collected. We then use this F as a surrogate for the CAD system to study the effect of both control and noise factors on the output response Y. Here is where the computational savings start to show.

As Figure 5 suggests, we might need to go back to Step 1 if we cannot find a satisfactory fit. In particular, this occurs when the original design ranges are too large. However, as we will discuss shortly, the information collected in Step 2c can be used to resize the design ranges appropriately.

Once we have found a good approximation F, we can proceed to solve equation (4). In Step 3a, the problem is now stated as:

$$\min_{x_c} E_{x_n}(F(x_c, x_n) - Y_0)^2 \qquad (6)$$

In equation (6), we are using the notation x_c, x_n to emphasize that the optimization is based on the nominal values of just the control factors. This does not conflict with equation (5), because we are using the alternative notation introduced in equation (3). Under this new formulation of the problem, we can now afford to use a Monte Carlo or any other approximation for the expectation in equation (4), because calculating Y from the xs using F is inexpensive, compared to using the CAD system directly. This, in turn, makes it possible to solve the problem with any optimization algorithm. If needed, we could now use gradient-based methods.

The final task, Step 3b, is to use the CAD system to confirm the performance of the *optimal design* we obtained in Step 3a. Here, the idea is that the approximate solution to the robust-design problem *is* a good solution.

The CADQ methodology, as described above, improves on traditional implementations of robust design in several ways. Among them are:

- CADQ can deal with any kind of performance measure and even multiple performance measures simultaneously or separately.
- It exploits the simulation capabilities of CAD systems to explore the space of control and noise factors efficiently, without resorting to the expensive product-array technique.
- The optimal design can be anywhere in the design range. The search is not constrained to an arbitrary, discrete set of designs.

To summarize, we want to emphasize that CADQ is heavily dependent on the availability of a good CAD system to simulate the characteristics of the product. Without such a system, it would be difficult or even impossible to pursue the approximation approach on which CADQ is based. If product simulation is not possible, then physical prototypes must be constructed to study the

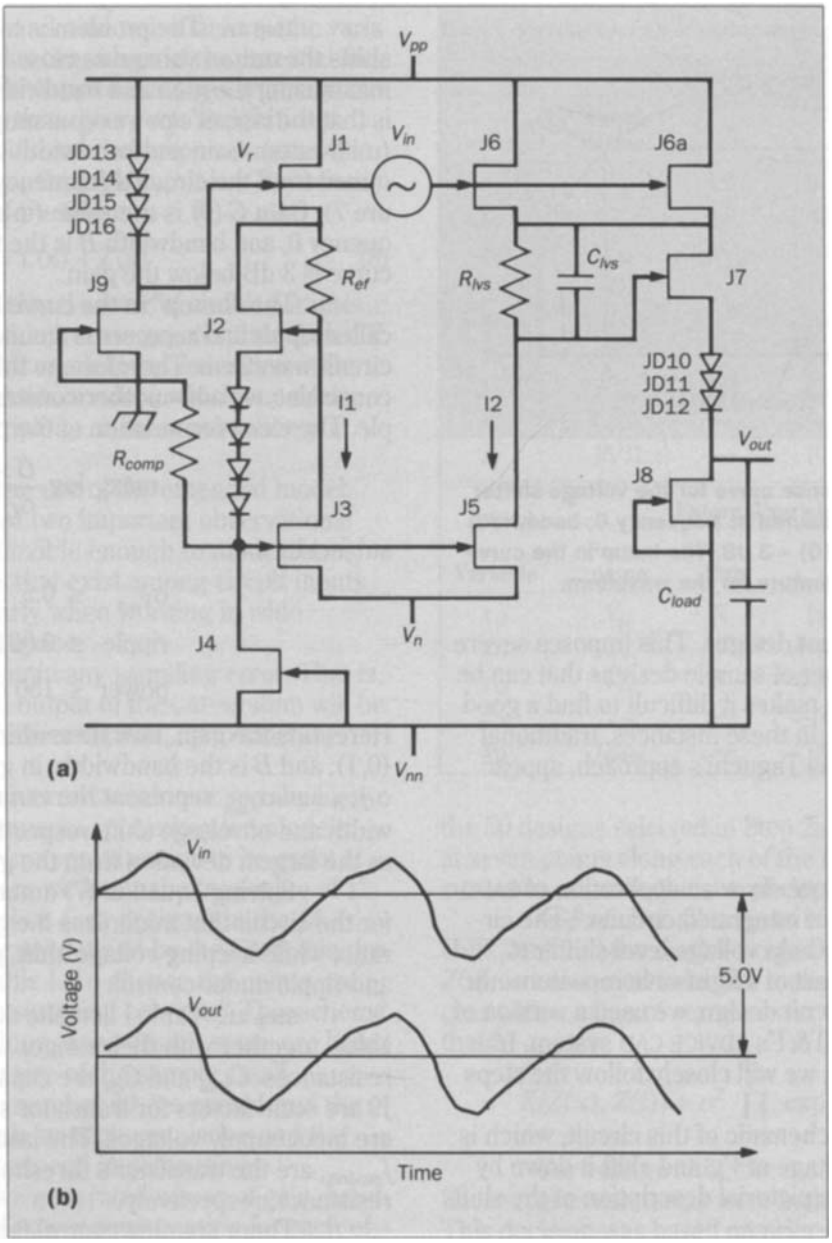

Figure 6. CADQ implementation for a GaAs voltage-level-shifter circuit. This circuit shifts the voltage at V_{in} down by 5V to V_{out}. (a) Schematic of the circuit; (b) pictorial description of the shifting mechanism.

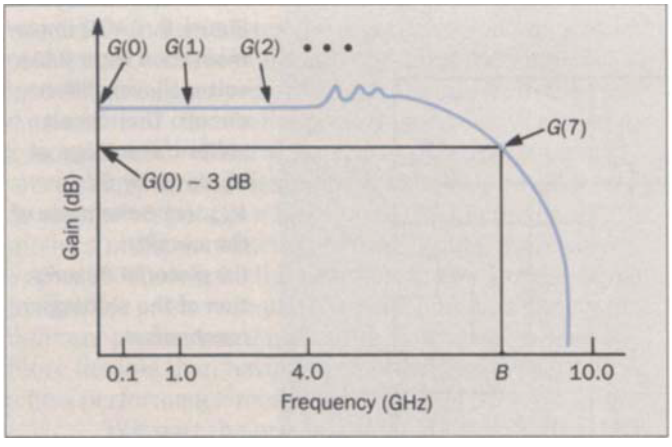

Figure 7. Frequency-response curve for the voltage-shifter circuit. Gain, *G*(0), is measured at frequency 0; bandwidth (*B*) is the frequency at *G*(0) – 3 dB. The hump in the curve is ripple, an undesirable feature for the waveform.

characteristics of different designs. This imposes severe limitations on the number of sample designs that can be studied in Step 2, which makes it difficult to find a good approximation function. In these instances, traditional implementations, such as Taguchi's approach, appear more appropriate.

An Application of CADQ

In this section, we show an application of CADQ to the statistical design of integrated circuits.[6] The circuit in our example is a GaAs voltage-level-shifter IC, which is to be used as part of a lightwave repeater unit. To obtain an optimal circuit design, we used a version of CADQ that works with AT&T's ADVICE CAD system. In describing the example, we will closely follow the steps described in Figure 5.

Figure 6a is a schematic of this circuit, which is designed to take the voltage at V_{in} and shift it down by 5V to V_{out}. Figure 6b is a pictorial description of the shifting mechanism.

Step 1a. The problem is to design a circuit that shifts the output voltage as close to 5V as possible, while maximizing the gain and bandwidth. A design constraint is that the circuit's power consumption be below 150 mW (milliwatts). Gain and bandwidth values are readily determined from the circuit's frequency-response curve (Figure 7). Gain G (0) is the value (in dB) of the curve at frequency 0, and bandwidth B is the frequency at which the curve is 3 dB below the gain.

The "hump" in the curve before it starts falling is called *ripple* and represents an undesired feature for the circuit waveform. Therefore, to the power-consumption constraint, we add another constraint on the size of ripple. The exact formulation of the problem is:

$$\max_{x_c} \log \frac{G \times B}{\sigma_{G \times B}} \qquad (7)$$

subject to:

$$\sigma_{V_{shift}} \leq 0.5$$

$$\text{ripple} \leq 0.001 \text{ dB}$$

$$\text{power} \leq 150 \text{ mW}$$

Here, G is the gain, now measured in the linear scale (0,1); and B is the bandwidth, in gigahertz (GHz). The $\sigma_{G \times B}$ and $\sigma_{V_{shift}}$ represent the variances of gain by bandwidth and of voltage shift, respectively. Ripple is defined as the largest deviation from the gain along the curve.

Solving equation (7) amounts to finding a design for the circuit that minimizes the gain-by-bandwidth S/N ratio, while keeping voltage shift, power consumption, and ripple under control.

Step 1b. Table I lists the design and external variables, together with their ranges. R_{ef}, R_{comp}, and R_{lvs} are resistances. C_{load} and C_{lvs} are capacitances. J1, J6, J7, and J9 are scale factors for transistor size. V_{pp}, V_n, and V_{nn} are input supply voltages. The last two factors, J_{modvto} and J_{modres}, are the transistor's threshold voltage and ohmic resistance, respectively.

There are nine control factors (the nominal

values of x_1 through x_9) and nine noise factors (the variability in x_1 through x_4 plus the values of the external factors x_{10} through x_{14}). We decided that the variability in the nominal values of x_5 through x_9 was insignificant.

Step 2a. The model we chose approximates the outputs as a function of x_1 through x_{14}. It is a Bayesian interpolation model[7,8] of the form:

$$Y = L(\mathbf{x}) + Z(\mathbf{x}) \qquad (8)$$

where $\mathbf{x}$ is the 14-dimensional vector of circuit parameters, L is a polynomial of degree no greater than 2, and Z is a random function. This model is an extension of the popular quadratic-regression models, which consider only the L and add a sampling- or measurement-error component.

We can justify the use of the extended model, equation (8), with at least two important observations:

- Polynomials are not flexible enough to model the kinds of relationships that exist among circuit inputs and outputs, particularly when working in wide regions of the design space.
- Our data does not contain any sampling error. That is, given a value of $\mathbf{x}$, the output of the CAD system will be the same for several different trials, up to machine accuracy.

In the absence of random error, we do not see any justification for using least-squares-approximation models. Indeed, we would like our approximation to be exact at the sampled points, where we know the true value of Y.

We chose 50 points (i.e., designs) in the 14-dimensional space of inputs defined by the Cartesian product of the ranges in Table I. To choose the points, we used a Latin-Hypercube sampling scheme.[9] This scheme guarantees that all portions of the design space are being observed, yet keeps the size of the sample small.

To arrive at the number 50, we considered the time the CAD system took for each simulation and the total time we wanted to expend in computing.

Step 2b. Here, we generated values of gain, bandwidth, voltage shift, and power consumption for each of the 50 designs selected in Step 2a. We also took readings at seven points along each of the frequency-response curves to estimate the size of ripple.

Table I. Variables for GaAs Voltage-Shifter Circuit

Design variables				
Variable	Factor name	Unit	Design range	Initial design
x_1	R_{ef}	kΩ	[0.8000, 6.4000]	2.400
x_2	C_{load}	pF	[0.0300, 0.3000]	0.100
x_3	C_{lvs}	pF	[0.0050, 2.0000]	0.100
x_4	J1	—	[0.0202, 0.0495]	0.030
x_5	J7	—	[0.0150, 0.0450]	0.030
x_6	J9	—	[0.0075, 0.0225]	0.015
x_7	R_{comp}	kΩ	[1.3500, 5.4000]	2.700
x_8	R_{lvs}/R_{ef}	—	[0.2500, 1.0000]	0.300
x_9	J6/J1	—	[0.5000, 2.0000]	1.500
External variables				
Variable	Factor name	Unit	Variability range	Center value
x_{10}	V_{pp}	V	[4.5000, 5.5000]	5.000
x_{11}	V_n	V	[–3.5000, –2.7000]	–3.000
x_{12}	V_{nn}	V	[–5.7200, –4.6800]	–5.200
x_{13}	J_{modto}	—	[–0.9375, –0.5625]	–0.750
x_{14}	J_{modres}	—	[1.0500, 2.4500]	1.750

Step 2c. For all outputs of interest except voltage shift, we fitted the model—equation (8)—with $L(\mathbf{x}) = c$. $Z(\mathbf{x})$ was assumed to be the realization of a Gaussian process with a mean of zero and a covariance kernel K, such that:[8]

$$K(Z(u), Z(t)) = \sigma^2 \prod_{i=1}^{14} \exp\left(-\theta_i \mid u_i - t_i \mid^2\right) \qquad (9)$$

For voltage shift, the model was different in that linear-regression terms were included for some inputs. This decision was based on prior information about the

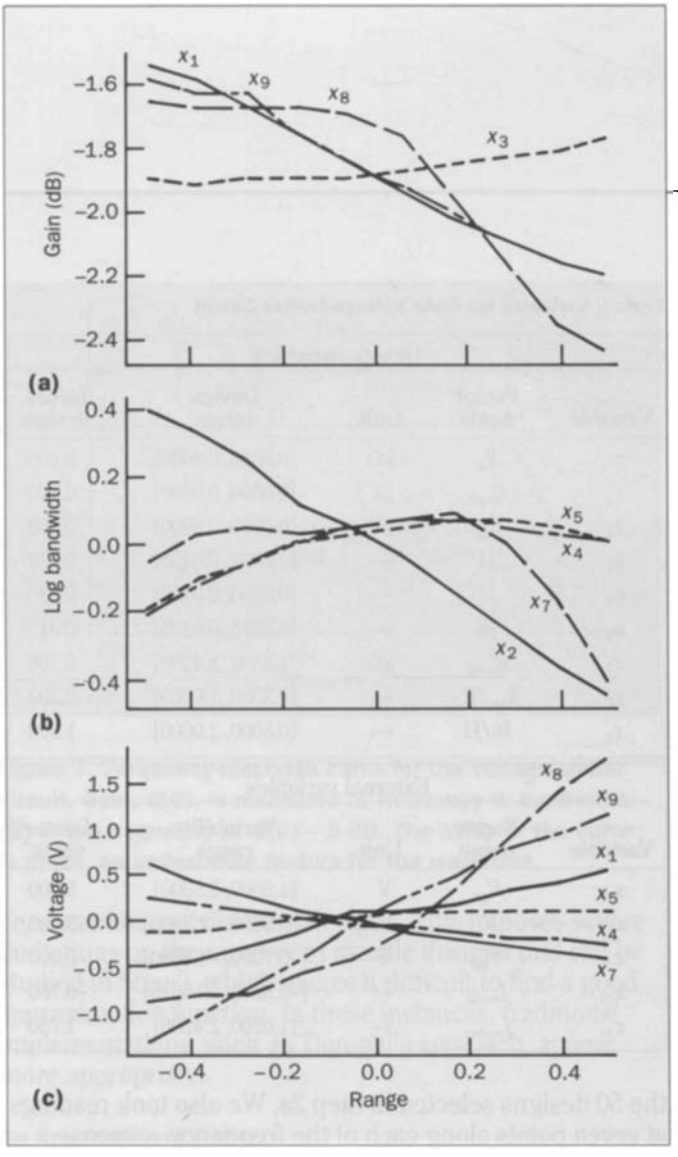

Figure 8. Main-effects plots for the voltage shifter's significant factors: (a) gain, (b) logarithm of bandwidth, and (c) voltage. Ranges for all factors are standardized to the interval [–0.5, +0.5]. The x_n labels on curves refer to the design and external variables in Table I. These curves show the relative importance of each input on the outputs. To reduce the design problem's dimensionality, we could screen out the less active inputs (i.e., curves with minimal slope).

relationship between this output and the inputs.

The proposed models led to an interpolation function for each of the outputs that exactly fits the sampled data and is smooth everywhere. The choice of covariance kernel in equation (9) rules out any jagged approximation curves, which is consistent with our knowledge about the smoothness of the outputs. (Further details on the fitting procedure[10] are outside the scope of this paper.)

We carried out four iterations from this step back to Step 2a before we obtained a satisfactory fit for each of the outputs. In each iteration, we used the approximations to produce plots like those in Figure 8, which indicate the relative importance of each input on the outputs. These plots are used for two main purposes:

- Identify and eliminate less active inputs (i.e., those with minimal slope). This helps us reduce the dimensionality of the optimization problem.
- Eliminate portions of the design region where the outputs are unacceptable. This helps us reduce the size of the design region, increasing the goodness of fit.

Based on the plots in Figure 8, we could decide to retain control factors x_1, x_2, x_7, x_8, and x_9 for modeling $G \times B$ and could eliminate the rest. Also, we could restrict the range on control factor x_2, C_{load}, to values below its current center, which is where we see larger values of bandwidth.

In general, with information obtained by the end of each iteration, we built a new design region and refitted the models.

Step 3a. We solved the optimization in equation (7) using a random-search algorithm.[10] To calculate $\sigma_{G \times B}$ and $\sigma_{V_{shift}}$, we took a Monte Carlo sample of 100 points around the current optimal point at each iteration in the search. Table II shows the optimal circuit design.

Step 3b. The optimal design's performance, as predicted by the approximation function, agreed well with the performance simulated for that design by the ADVICE system. We show both results in Table III.

Figure 9 compares the performance of the initial and optimized circuit designs. These box plots are a graphical representation of the distributions of voltage

Table II. Optimal Circuit Design Obtained by CADQ

Parameter	Unit	Value
Resistance, R_{ef}	kΩ	1.8280
Load capacitance, C_{load}	pF	0.0485
Capacitance, C_{lvs}	pF	0.0775
J1, transistor-size scale factor	—	0.0232
J7, transistor-size scale factor	—	0.0375
J9, transistor-size scale factor	—	0.0087
Resistance, R_{comp}	kΩ	1.4015
R_{lvs}/R_{ef}	—	0.03065
J6/J1	—	1.2268

Table III. Performance of the Optimized Design

Parameter	Unit	Predicted	Simulated
Bandwidth, B	GHz	6.4170	6.413
Voltage shift, V_{shift}	V	4.9760	4.9690
Gain, G	dB	−1.9700	−1.9700
Power	mW	120	120
Ripple	dB	0.0007	0.0000

NOTE: Performance is *predicted* by the approximation function, or *simulated* by the ADVICE system.

shift, gain, and bandwidth for both designs. They show the new design's superiority in terms of nominal performance and robustness.

We also tried to optimize the circuit using Taguchi's approach but could not get a satisfactory solution. That is, we were unable to define a single S/N ratio that accounts for the multiple performance objectives (gain, bandwidth, voltage) and constraints (power dissipation and ripple). Performing a separate analysis for each of the outputs led to radically different designs that could not be reconciled.

These problems are not peculiar to our example but relate to the inability of Taguchi's approach to deal systematically with multiple performances and constraints. In any case, Taguchi's experiment required 729 simulations (27 rows of the control array by 27 rows of the noise array). This is more than three times the number of simulations that CADQ required.

Conclusions

In this paper, we described CADQ, a cost-effective implementation of robust design for CAD environments. CADQ extends the capabilities of CAD simulators and allows designers to consider manufacturing and operating environment variations while designing a product. In this sense, CADQ is geared toward design for quality and design for manufacturability.

Through a real case study, we showed how CADQ can be used in computer-aided circuit design to produce an optimal statistical design for an IC. However, CADQ is not constrained to the design of integrated circuits. It can also be used in areas like circuit-board design and mechanical design, where CAD simulators are also available. Because of its efficiency, CADQ can be used on large design-optimization problems, where more traditional methods are prohibitively expensive.

Finally, although CADQ uses powerful and flexible statistical methods for sampling and model fitting, it also provides graphic aids (such as the main-effects plots in Figure 8) to simplify the analysis of how different factors affect the outputs. If one uses reasonable sampling approximation methods (such as those presented in the preceding section), the software can automate most of the work. Thus, even designers without extensive statistical knowledge can use CADQ.

Acknowledgments

The case study we presented was part of a joint research project with Professor Jerry Sacks, Chona Bernardo, and Bob Buck of the Statistics Department at the University of Illinois, Champaign-Urbana, and William Welch of the Statistics Department at the University of Waterloo, Ontario, Canada.

We want to thank many of our colleagues at

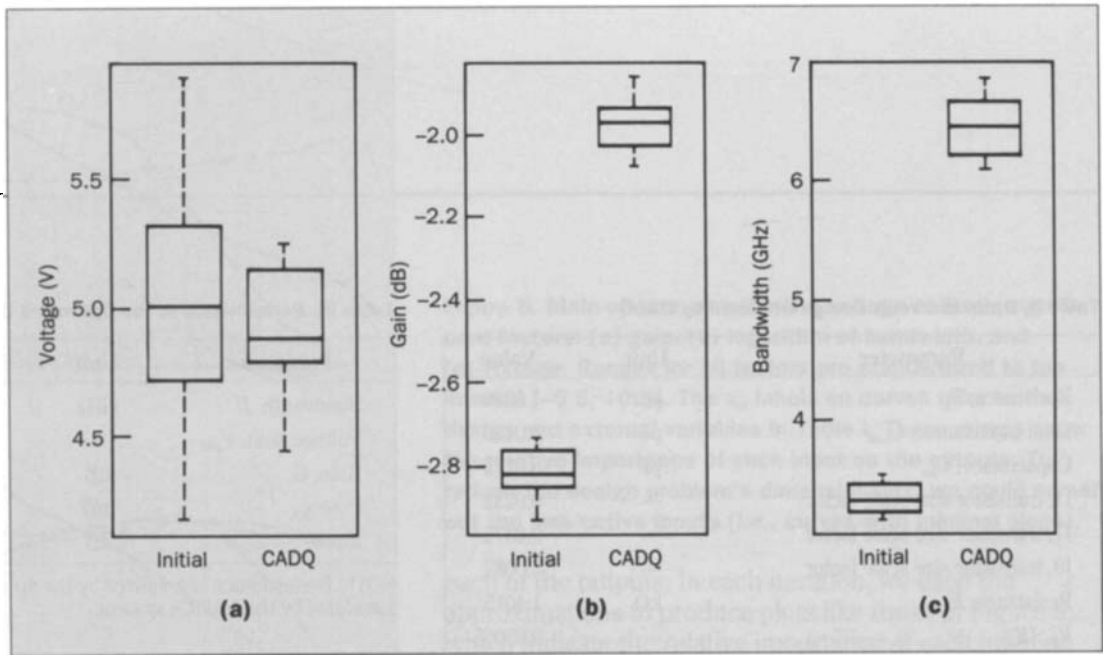

Figure 9. Comparative performance of the CADQ and initial designs. The distributions in these box plots show that the optimized (CADQ) design has better nominal performance and is more robust.

AT&T Bell Laboratories for their contributions to this work; in particular, Jeff Hooper, Rajiv Keny, Dave Martinich, Anne Shoemaker, and Kwok Tsui.

References

1. K. Dehnad, "Quality Control, Robust Design and the Taguchi Method," Wadsworth Statistics and Probability Series, Wadsworth and Brooks/Cole, Pacific Grove, California, 1989.
2. M. Phadke, "Quality Engineering using Robust Design," Prentice Hall, Englewood Cliffs, New Jersey, 1989.
3. R. Brayton, S. Director, and G. Hachtel, "Yield Maximization and Worst-Case Design with Arbitrary Statistical Distributions," *IEEE Transactions on Circuit Systems*, Vol. CAS-27, No. 9, 1980, pp. 756-764.
4. G. Taguchi and Y. Wu, *Introduction to Off-line Quality Control*, Central Japan Quality Control Association, Meieki Nakamura-Ku Magaya, Japan, 1979.
5. W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, England, 1988.
6. R. K. Brayton, G. D. Hachtel, and A. Sangiovanni-Vicentelli, "A Survey of Optimization Techniques for Integrated Circuit Design," *Proceedings of the IEEE*, Vol. 69, No. 10, October 1981, pp. 1334-1362.
7. D. M. Steinberg, "Bayesian Models for response surfaces, I," MRC Technical Report No. 2682, University of Wisconsin, Madison, Wisconsin, 1984.
8. J. Sacks, W. Welch, T. Mitchell, and H. Wynn, "Design and Analysis of computer experiments," *Statistical Science*, Vol. 4, No. 4, November 1989, pp. 409-423.
9. M. D. McKay, W. Conover, and R. Beckman, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, Vol. 21, May 1979, pp. 239-245.
10. L. Pronzato, E. Walter, A. Venot, and J. Lebruchec, "A general-purpose global optimizer: Implementation and applications," *Mathematics and Computers in Simulation*, Vol. 26, 1984, pp. 412-422.

Biographies (continued)

China), an M.S. from the University of Texas at El Paso, and a Ph.D. from the University of Arizona, all in electrical engineering. While with AT&T, Mr. Nazaret worked on the application of mathematical, statistical, and optimization techniques to quality improvement, and managed the CADQ project. Now a vice president of East Coast Computer Systems, he is responsible for research and development of storage and data networking subsystems for the UNIX® system market. He has a B.Sc. in mathematics from Universidad Simon Bolivar, Venezuela, and an M.A. and Ph.D. in statistics from the University of California at Berkeley.

(Manuscript received January 12, 1990)