NISS

Circuit Optimization Via Sequential Computer Experiments: A Case Study

Robert Aslett, Robert J. Buck, Steven G. Duvall, Jerome Sacks, and William J. Welch

Circuit Optimization Via Sequential Computer Experiments: Design of an Output Buffer

Robert Aslett
Intel Corporation, JFT-102
2111 N.E. 25th Avenue
Hillsboro, OR 97124-5961, U.S.A.

Robert J. Buck
Department of Mathematics and Statistics
Western Michigan University
Kalamazoo, MI 49007, U.S.A.

Steven G. Duvall
Intel Corporation, RN2-40
P.O. Box 58119
2200 Mission College Boulevard
Santa Clara, CA 95052-8119, U.S.A.

Jerome Sacks
National Institute of Statistical Sciences
P.O. Box 14162
Research Triangle Park, NC 27709-4162, USA

William J. Welch
Department of Statistics and Actuarial Science
University of Waterloo
Waterloo, Ontario N2L 3G1, Canada

SUMMARY

In electrical engineering, circuit designs are now often optimized via circuit simulation computer models. Typically, there are many response variables characterizing the circuit's performance. Each response is a function of many input variables, including factors that can be set in the engineering design and noise factors representing manufacturing conditions. We describe a modelling approach appropriate for the simulator's deterministic input-output relationships. Nonlinearities and interactions are identified without explicit assumptions about the functional form. These models lead to predictors to guide the reduction of the ranges of the designable factors in a sequence of experiments. Ultimately, the predictors are used to optimize the engineering design. We also show how visualization of the fitted relationships facilitates understanding of the engineering trade-offs between responses. The example used to demonstrate these methods, the design of a buffer circuit, has multiple targets for the responses, representing different trade-offs between the key performance measures.

Keywords: Circuit simulator; Computer code; Computer model; Engineering design; Parameter design; Stochastic process; Visualization.

1 Introduction

Design of electronic circuits using computer models has been widespread since the 1970's. The inputs to these models include values of adjustable engineering variables (or parameters) for the sizes of the electrically active devices (typically transistors) and values of factors representing variability due to manufacturing noise. The outputs characterize the circuit performance, e.g., time delays for propagation of signals through the circuit.

The engineering problem is to choose values of the sizes of the designable devices such that the circuit performs consistently well in the presence of variability from manufacturing noise. This is known as robust engineering design (Taguchi, 1986). Taking account of manufacturing variability is known as "statistical" design in electrical engineering.

The output from a circuit simulator is deterministic. Given the same input values, replicate runs will produce the same output values. One specified manufacturing condition is represented by one set of values for the noise factors. The noise factors would have to be varied over several runs to generate variability in the outputs.

Circuit design has typically involved manual optimization, where the engineer iterates three steps: speculation about how the parameters of the circuit might be changed in order to improve performance, implementing the changes, and verifying the impact of the changes through circuit simulation. Ad hoc exploration of design options can consume a significant portion of the available engineering resources in a design project.

Linking simulators to optimization algorithms has obvious appeal. Development time and engineering effort can be reduced, and it is easier to handle a large number of designable engineering parameters. Furthermore, systematic exploration of design options is more likely to deal adequately with trade-offs among circuit performances and variation from noise factors.

Early methods taking account of manufacturing-noise factors (see Brayton et al., 1981 for a survey) operated directly on circuit simulators. These methods were gener-

ally too inefficient for routine solution of real, large-scale circuit optimization problems, typically requiring large numbers of simulations, for example to obtain Monte Carlo estimates of yield. More recent efforts have focused on indirect methods of optimization, with optimizers operating on empirical models fitted to outputs from circuit simulation (e.g., Alvarez et al., 1988; Welch et al., 1990; Yu et al., 1991). Originally, these methods relied upon classical methods for the design and analysis of experiments. Polynomial regression models may give poor approximations to the highly nonlinear input-output relationships that are often present, however.

Deterministic simulators are also widespread in other engineering disciplines. For instance, Su et al. (1996) used generalized linear regression models to design a lamp filament via a deterministic finite-element computer code. With a three-level experimental design and dummy variables they were able to deal with moderate nonlinearities and interactions. They also noted that different assumed error distributions produced little change in goodness of fit with data from a deterministic simulator.

The lack of random error makes experiments involving deterministic computer codes quite distinct from physical experimentation. Methods for the design and analysis of computer experiments were described by Currin et al. (1991), Sacks, Schiller, and Welch (1989), Sacks, Welch, Mitchell, and Wynn (1989), and Welch et al. (1992). Bernardo et al. (1992) applied these methods to the design of a voltage-shifter circuit. Their problem involved a total of 14 input variables and four output variables.

The example presented here, the design of output buffer circuits, is a larger problem. It has 36 inputs, consisting of 20 designable device sizes and 16 noise factors. The eight performance outputs of major interest are various time delays and voltage "spikes," outputs whose values tend to trade-off against each other. A major complication of the example is that a generic buffer can be used in many different applications by adjusting device sizes. Subject to a maximum delay defined by the specific application, we want to minimize the voltage spikes. Thus, the aim is to find several good subregions in the 20-dimensional space of device sizes, each covering a range of maximum delays and

2

with its own set of approximations to the simulator's input-output relationships. For a specific application and hence a particular constraint on the delay, an engineer would take the appropriate subregion and its approximating functions and quickly optimize the circuit. This takes considerably less computer time than optimizing via the circuit-simulation code itself. Earlier attempts at this problem using factorial experiments and polynomial models had failed to locate good engineering designs.

In Section 2 we describe the output-buffer engineering requirements, including the criteria for optimization. Section 3 outlines the overall sequential strategy. In Section 4 we describe the experimental designs used at each stage. Development of accurate predictors for the input-output relationships is crucial to guide the sequence of experiments. Thus, Section 5 is concerned with modelling, prediction, assessing prediction accuracy and visualization of the predictors. In Section 6 we apply these methods to the buffer circuit and describe the six stages of the experiment. Some confirmation results show that the final approximating models are sufficiently accurate to produce circuits that meet or exceed the performance of buffers produced by experienced designers using manual methods. We conclude in Section 7 with some discussion, including further methods under investigation.

2   The Output-Buffer Problem

2.1   *Output buffers*

An output buffer forms the interface between an integrated circuit chip and the environment in which the chip operates. Buffer circuits similar to that in the example are used in several generations of microprocessors as well as numerous other chips at Intel.

The time between the low to high transition of the input and the high to low transition of the output is called the "high-low" delay of the buffer. There is a similarly defined "low-high" delay. These delays are key measures of buffer performance.

Due to changing currents in the package wires, voltage spikes will be induced on two power supply lines. For each supply line, there are high-low and low-high voltage

Variable	Min	Max
T1, T2	100	2400
P1	2	200
N1, P2	2	400
N2	2	200
P3, N3, P4, N4	2	100
P5, N5, P6, N6, TP1, TN1	2	50
ENP1, ENN1	2	20
DP1, DN1	2	40

Table 1: Designable variables and their initial ranges (microns).

spikes. If large enough, they could be interpreted as changes in logic state by the external environment, and so must be controlled. Thus, the maximum voltage spike is also an important measure of buffer performance. (A voltage spike is often called "output noise", but we avoid this term here to avoid confusion with manufacturing noise.)

2.2 *Inputs*

A total of 36 input variables were considered: 20 controllable device sizes and 16 uncontrollable factors to represent processing conditions. The device variables and their initial ranges are given in Table 1. The names refer to the roles of the designable transistor gates in the circuit. Very wide initial ranges were chosen in order to cover a substantial range of maximum delays.

The 16 uncontrollable noise factors, $U_1, \ldots, U_{16}$, have an approximately Gaussian joint distribution, rescaled here to have standard normal margins. The first eight of these factors are correlated; the other eight vary independently. Estimated moments of the joint distribution, including the correlations, were available from data gathered from the manufacturing process. The noise factors are independent of the designable parameters, thus the same estimated moments apply to all regions of the design space.

2.3 Outputs and Engineering Objectives

The output performances of major interest and the initial engineering objectives were as follows.

- Primary time delays, *TL* and *TH*, measured in nanoseconds (ns). These time delays, for the high-low and low-high transitions respectively, should be no more than a given application-specific maximum, t_{max}, i.e.,

$$TL \leq t_{max} \quad \text{and} \quad TH \leq t_{max}.$$

In the applications anticipated, t_{max} would range from about 5 to 12 ns.

- Secondary-path time delays, *TLO* and *THO* (ns). There are two "output enable" time delays, again for the high-low and low-high transitions respectively. Each should be no more than 10% of the corresponding primary delay, i.e.,

$$TLO \leq 0.1\,TL \quad \text{and} \quad THO \leq 0.1\,TH.$$

To deal with these constraints, we introduced two constraint slacks which have to be nonnegative,

$$TLC = 0.1\,TL - TLO \geq 0 \quad \text{and} \quad THC = 0.1\,TH - THO \geq 0. \tag{1}$$

- Voltage spikes, *VSL*, *VCL*, *VSH*, and *VCH*, measured in volts (V). These correspond to the two supply lines (called VSSP and VCCP) and the two transitions. Subject to the above constraints, we want to minimize over the device sizes the maximum of the four voltage spikes.

The above names for the output variables are shortened from the names more typically used by electrical engineers; e.g., *TL*, *TLO*, and *VSL* would often be called *TDLEN*, *TDLOENN*, and *VSSPTDL*, respectively.

All of the above output variables depend on the designable inputs, *T1*, ..., *DN1* in Table 1, and the 16 noise factors, $U_1, \ldots, U_{16}$. We needed to take account of the noise

factors in defining the engineering objectives. For given values of the device sizes, i.e., for a specific circuit design, the joint distribution of $U_1, \ldots, U_{16}$ induces a distribution of, say, TL values. To penalize variability from the noise factors, define TL^+ to be the mean plus three standard deviations of the TL distribution, with similarly defined terms for the other outputs. These quantitites depend only on the device sizes, $T1, \ldots, DN1$. The above objectives and constraints then become: minimize over the device sizes

$$\max(VSL^+, VCL^+, VSH^+, VCH^+) \tag{2}$$

subject to

$$TL^+ \leq t_{\max}, \quad TH^+ \leq t_{\max}, \quad TLC^+ \geq 0, \quad \text{and} \quad THC^+ \geq 0, \tag{3}$$

for a given $t_{\max}$. Because of the trade-off between the primary time delays and the voltage spikes, a larger value for $t_{\max}$ in (3) allows a better minimum to be found for (2).

There were several other outputs of interest, which added further constraints to the problem. For example, impedances ZUP and $ZDOWN$ had to be each less than 50Ω. It turned out these further outputs are simple functions of inputs $T1$ or $T2$ and hence merely revised the $T1$ and $T2$ ranges (see Section 6.1). Although these responses raised no interesting modelling issues, their impact on the $T1$ and $T2$ ranges turned out to be important, as described in Section 6.3.

3 The Sequential Strategy

The ultimate aim was to build approximations for the responses as functions of all simulator inputs, allowing an engineer to optimize quickly the objective function (2) subject to the constraints (3) for a specific primary-delay goal. It was recognized early in the project that accurate predictions are practically impossible with high-dimensional input unless the region of interest is reduced to a manageable size. Thus, in order to construct approximations that are accurate for a range of delay goals,

6

multiple models were constructed. Each provided accurate predictions for a subrange of delays and was fitted from simulator runs in a specific subregion of device sizes.

The sequential optimization methodology described in Bernardo et al. (1992) was extended to handle the range of performance goals. The investigation proceeded in stages, with each stage consisting of the following steps:

1. Experimental design. Choose combinations of values for the inputs (device sizes and noise factors) in the current input region and run the simulator.

2. Predictor construction. Fit a model and obtain a predictor for each simulated response.

3. Evaluation. Assess the accuracy of prediction.

4. Visualization. Use graphical methods to visualize the effects of individual input variables on the predictors. Where estimated interactions are judged to be important, also consider joint effects of pairs of variables, etc.

5. Optimization. If there is sufficient predictor accuracy, perform tentative optimizations.

6. From the visualization and, possibly, from the optimizations, identify promising new subregions in the device-size space. Each subregion will be appropriate for a limited range of primary delay goals.

The subregion(s) identified at Step 6 define the experimental region(s) for Step 1 of the next stage. The joint normal distribution for the noise factors, $U_1, \ldots, U_{16}$ cannot be reduced in spread. The device-size variables, with very wide initial ranges, dominate the input-output relationships at the early stages. Thus, accuracy of prediction will tend to improve as the device-size space is reduced.

4 Experimental design

At each stage, Latin hypercube experimental designs (McKay et al., 1979) were generated. These designs were proposed specifically for analysing the output from deterministic computer simulation codes.

For the initial, Stage 1 experiment, for example, we constructed a Latin hypercube with 120 simulator runs. The range for each device-size variable was represented by the midpoints of 120 equal intervals covering the range. Each of the 120 values occurs once in the experiment. For each noise factor, the values in the design were given by $\Phi^{-1}(\frac{i-0.5}{120})$ for $i = 1, \ldots, 120$, where Φ is the standard normal cumulative density function. Thus the range for each input was fully explored, with uniform interest across the ranges of the device sizes, and marginal normal distributions reflecting operating conditions for the noise factors.

For a completely random Latin hypercube, the 120 values for each variable would have been in random order when combining the factors in the experimental design. This would have produced correlations randomly varying about zero between pairs of variables. For the noise factors, however, we would like to reflect the correlation structure in manufacturing. The remaining correlations, between two designable variables or between a designable variable and a noise factor, are ideally zero to avoid partial confounding. Iman and Conover (1982) described how to transform a starting, random Latin hypercube into one with a desired correlation structure using the Cholesky decomposition of the correlation matrix. By iterating their procedure, correlations very close to the target correlation structure were achieved.

Figure 1 shows two-dimensional projections of the Stage 1 design for the first two device sizes, $T1$ and $T2$, and for the first two noise factors, U_1 and U_2. The $T1$-$T2$ space is fairly uniformly covered, and the correlation between U_1 and U_2 is apparent. There are no replications, which would be uninformative in the absence of random error, in these low-dimensional projections of a Latin-hypercube design. Thus, if only a few input factors dominate an input-output relationship, all runs will still be useful.

Such a "space-filling" design is well suited to the model and predictor in Section 5. The predictor gives larger weight to design runs close to the point where we want to predict. Even in the context of polynomial regression models fitted by least squares, work going back to Box and Draper (1959) has established the desirability of spreading runs throughout the experimental region for prediction when variability from random error is unimportant.

5 Modelling

5.1 *Gaussian Stochastic Process Models*

At each stage, approximating functions were constructed for each output as a function of all 36 inputs. We applied the methodology described by Sacks, Welch, Mitchell, and Wynn (1989). We now outline the main ideas.

Let $\mathbf{x} = x_1, \ldots, x_{36}$ denote the vector of device-size and noise inputs, and let y denote one of the outputs (e.g., *TL*). We treat $y(\mathbf{x})$ as a realization of a stochastic process,

$$Y(\mathbf{x}) = \beta_0 + Z(\mathbf{x}), \tag{4}$$

where β_0 is an unknown constant, and $Z(\mathbf{x})$ is a random function with mean zero, variance σ_Z^2, and correlation $R(\mathbf{x}, \mathbf{x}')$ between the two Z values at input vectors $\mathbf{x}$ and $\mathbf{x}'$. If the simulator input-output function acts like the realization of a stochastic process, then this mathematical artifice may provide very good approximations in practice. Moreover, the stochastic model gives a basis for estimating uncertainty of prediction. An alternative Bayesian interpretation, described by Currin et al. (1991), is that properties of the correlation function can be chosen to represent prior knowledge about the behaviour of $y(\mathbf{x})$.

Central to this model is the correlation function, $R(\mathbf{x}, \mathbf{x}')$. We take

$$R(\mathbf{x}, \mathbf{x}') = \Pi_{l=1}^{36} \exp(-\theta_l |x_l - x_l'|^{2-\alpha_l}), \tag{5}$$

where $\theta_l \geq 0$ and $0 \leq \alpha_l < 2$ are parameters to be estimated. (These ranges give

positive definite correlation matrices.) Qualitatively, this correlation structure implies that two vectors $\mathbf{x}$ and $\mathbf{x}'$ close together in the input space give rise to two values of the output function that are highly correlated (i.e., similar), as would be expected if the function is smooth. Conversely, $\mathbf{x}$ and $\mathbf{x}'$ vectors remote from each other lead to two output values with near-zero correlation (i.e., they are unrelated).

If we assume further that the stochastic process in (4) is Gaussian, the estimation of the model parameters, β_0 and σ_Z^2 in (4) and $\theta_1, \ldots, \theta_{36}$ and $\alpha_1, \ldots, \alpha_{36}$ in (5), is straightforward using maximum likelihood.

5.2 Prediction

From the fitted model, a best linear unbiased predictor, $\hat{Y}(\mathbf{x})$, can be constructed. It interpolates the responses from the n runs in the experiment, as it should for a deterministic relationship. Let $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}$ be the n points in the experimental design, and let $\mathbf{y} = (y_1, \ldots, y_n)^T$ denote the corresponding simulated output values. The predicted response at an untried $\mathbf{x}$ is computed from

$$\hat{Y}(\mathbf{x}) = \hat{\beta}_0 + \mathbf{r}(\mathbf{x})^T \mathbf{R}_{\mathcal{D}}^{-1} (\mathbf{y} - \hat{\beta}_0 \mathbf{1}), \tag{6}$$

where $\mathbf{r}(\mathbf{x})$ is an $n \times 1$ vector of correlations with element i given by $R(\mathbf{x}, \mathbf{x}^{(i)})$ in (5), $\mathbf{R}_{\mathcal{D}}$ is an $n \times n$ matrix of correlations for the responses at the design points with element i, j given by $R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ in (5), $\mathbf{1}$ denotes an $n \times 1$ vector of 1's, and $\hat{\beta}_0 = \mathbf{1}^T \mathbf{R}_{\mathcal{D}}^{-1} \mathbf{y} / \mathbf{1}^T \mathbf{R}_{\mathcal{D}}^{-1} \mathbf{1}$ is the generalized least squares estimator of β_0.

In numerous applications, including the circuit-simulator example of Bernardo et al. (1992), accurate prediction has followed from this modelling strategy without the need to make assumptions about the forms of nonlinearities and interactions.

5.3 Assessing prediction accuracy

A mean squared error of prediction,

$$s^2(\mathbf{x}) = \sigma_Z^2 \left[1 - \begin{pmatrix} 1 & \mathbf{r}(\mathbf{x})^T \end{pmatrix} \begin{pmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & \mathbf{R}_{\mathcal{D}} \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ \mathbf{r}(\mathbf{x}) \end{pmatrix} \right], \tag{7}$$

also follows from the model (4).

We use mainly cross validation, however, to assess prediction accuracy. Let $\hat{Y}_{-i}(\mathbf{x}^{(i)})$ denote the leave-one-out cross-validation prediction of y_i using the other $n-1$ cases. It can be computed from $\hat{Y}(\mathbf{x}^{(i)})$ in (6) when element i is removed from the vectors $\mathbf{y}$, $\mathbf{r}(\mathbf{x})$, and $\mathbf{1}$, and row i and column j are removed from $\mathbf{R}_{\mathcal{D}}$. Comparison of $\hat{Y}_{-i}(\mathbf{x}^{(i)})$ with y_i for $i = 1, \ldots, n$ gives a visual indication of prediction accuracy. As a summary measure, we can also compute the cross-validation root mean squared error of prediction,

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}[y_i - \hat{Y}_{-i}(\mathbf{x}^{(i)})]^2}. \tag{8}$$

Similarly, let $s_{-i}(\mathbf{x}^{(i)})$ denote the root mean squared error when case i is removed in (7). We can compute standardized cross-validation residuals,

$$e_i = \frac{y_i - \hat{Y}_{-i}(\mathbf{x}^{(i)})}{s_{-i}(\mathbf{x}^{(i)})},$$

for $i = 1, \ldots, n$. If the model (4) holds, then $e_1, \ldots, e_n$ should behave approximately like a sample from the standard normal distribution. In particular, they should lie within about $[-3, 3]$. This provides a model diagnostic.

5.4 *Visualization*

Visualization of the predictor plays an important role in revising the experimental ranges from one stage to the next, especially in early stages of the sequential strategy.

The estimated main effect of an input is defined by averaging or integrating out all the other inputs from $\hat{Y}(\mathbf{x})$. It is convenient here to write the vector of all 36 inputs, $\mathbf{x}$, as $(\mathbf{c}, \mathbf{u})$, where $\mathbf{c}$ denotes the 20 controllable device sizes and $\mathbf{u}$ is the vector of 16 uncontrollable (in actual manufacturing) noise factors. Thus, we write the predictor $\hat{Y}(\mathbf{x})$ as $\hat{Y}(\mathbf{c}, \mathbf{u})$. This division of the inputs is useful as $\mathbf{u}$ has a joint Gaussian distribution in manufacturing and hence in the integration. The estimated main effect of the jth device size, a function of its value c_j, is computed from

$$\hat{\mu}_j(c_j) = \int \hat{Y}(\mathbf{c}, \mathbf{u}) \Pi_{k \neq j} \frac{dc_k}{b_k - a_k} f(\mathbf{u}) \, d\mathbf{u} \tag{9}$$

11

where a_k and b_k are the respective lower and upper limits of c_k, and $f(\mathbf{u})$ is the noise distribution. Thus, we integrate out all noise factors with respect to their joint distribution and all device sizes except the jth with respect to uniform distributions. The estimated main effect is computed for a set of c_j values covering the range and is plotted against c_j.

The correlation function (5) and hence $\mathbf{r}(\mathbf{x}) \equiv \mathbf{r}(\mathbf{c}, \mathbf{u})$ in the predictor (6) are products of terms involving a single input variable. Thus, in (9), the integration with respect to the rectangular space of $\mathbf{c}$ becomes a product of one-dimensional integrals and is numerically straightforward. Integration with respect to the Gaussian distribution, $f(\mathbf{u})$, in (9) is approximated numerically by a Monte Carlo average. The $\mathbf{u}$ combinations in the Latin hypercube experimental design are convenient for this purpose as they have approximately correct first and second moments.

A numerical summary of the importance of the estimated main effect for device size j is obtained by comparing the variability of $\hat{\mu}_j(c_j)$ over the set of c_j values with the total variability of $\hat{Y}(\mathbf{c}, \mathbf{u})$ over the entire input space. The calculation of the 36-dimensional integral for the total variability is again facilitated by the product correlation function.

To compute a similar main effect for a noise factor u_j that is correlated with the other noise factors would require integration with respect to an appropriate conditional distribution. This is not undertaken, however, as we are interested in visualization for the purpose of reducing the ranges of the device sizes; the noise factor ranges cannot be changed from stage to stage.

The estimated joint effect, $\hat{\mu}_{jj'}(c_j, c_{j'})$, for device sizes j and j', as functions of their values c_j and $c_{j'}$, can be similarly computed. We integrate out all inputs except these two from $\hat{Y}(\mathbf{c}, \mathbf{u})$. The estimated interaction,

$$\hat{\mu}_{jj'}(c_j, c_{j'}) - \hat{\mu}_j(c_j) - \hat{\mu}_{j'}(c_{j'}) + \hat{\mu}_0, \tag{10}$$

where $\hat{\mu}_0$ is the integral of $\hat{Y}(\mathbf{c}, \mathbf{u})$ over all inputs, measures the nonadditivity of device sizes j and j' in the estimated joint effect. If the variability in the interaction over the

c_j and $c_{j'}$ ranges is a trivial proportion of the total variability in the predictor, it is sufficient to consider the corresponding estimated main effects. Otherwise, we need to make a contour plot of the estimated joint effect, $\hat{\mu}(c_j, c_{j'})$, as a function of both c_j and $c_{j'}$.

6   The Output-Buffer Experiment

The experiment consisted of six stages. The first two stages narrowed the space of device sizes down to a manageable size. At later stages, greater accuracy of prediction was possible, allowing optimization of the engineering objectives via the fitted predictors.

We describe Stage 1 in some depth to illustrate the modelling and visualization methodology. Stage 2 proceeded similarly, and we give a briefer account. At Stage 3, visualization of the engineering trade-offs led to relaxing some constraints. Stage 4 discovered a subregion of the device-size space appropriate for small time delays. Prediction accuracy became satisfactory for all responses, enabling optimization of the voltage spikes subject to small upper bounds on the primary delays. Stages 5 and 6 identified further subregions for higher primary delays.

6.1   *Stage 1*

An initial Latin hypercube of 120 runs was generated in the 36-dimensional input space. The choice of 120 runs was somewhat arbitrary. Earlier experience with the stochastic-process model in various applications had indicated that 10 times the anticipated number of active variables is often adequate even in the presence of nonlinearities and interaction. Thus 120 runs would be enough for up to 12 active variables per response.

The simulator failed to give output for some responses at two (extreme) settings. In addition, we discarded six runs with at least one primary delay greater than 20ns. This cut-off was chosen to remove outlying observations of little engineering interest

13

when we impose much smaller values for the primary-delay constraint. They would have degraded the accuracy of fitted models in regions where the response is lower and hence of concern. In our experience, such data editing decisions often have to be made at early stages of experimentation when input ranges are very wide. Thus, 112 runs were left for analysis.

Simple plotting revealed that the impedances *ZUP* and *ZDOWN* mentioned in Section 2.3 are trivially related to device size *T1* or to *T2*. The requirement that both *ZUP* and *ZDOWN* be less than 50 Ω led to a lower bound of 330 microns for *T1* and a lower bound of 560 microns for *T2*.

The primary delays, *TL* and *TH*, the secondary-delay constraint slacks, *TLC* and *THC*, and the voltage spikes, *VSL*, *VCL*, *VSH*, and *VCH*, were modelled as in Section 5.

Examining the cross-validation predictions described in Section 5.3 indicated that prediction accuracy was fairly good for some responses, but less so for others. Figure 2(a), for example, shows the actual *VSL* values versus their cross-validation predictions. This voltage spike and *VCH* tend to be larger than the others and hence the most important in the objective function (2). The figure shows fairly good prediction for this response. Moreover, as shown in Figure 2(b), the standardized cross-validation residuals lie within $[-3, 3]$, giving some credibility to the model (4). Figure 3 shows analogous plots for *TH*, an output with poorer prediction accuracy, though the magnitude of the prediction error is again well modelled.

For comparison, second-order regression models were also fitted by least squares. Because there are only 112 data points and potentially 703 terms in the model, a first-order regression in the 36 input variables was initially fitted. After deleting terms insignificant at the 0.05 level, a second-order model was fitted with the remaining input variables. Finally, insignificant terms from this model were deleted. Table 2 includes the cross-validation root mean squared error given in (8) for the Stage 1 stochastic-process models and for the regression models. The regression models are clearly much less accurate.

| | Stage | | |
Response	1	2	4
VSL (V)	0.13 (0.21)	0.07 (0.12)	0.002 (0.005)
VCL (V)	0.18 (0.24)	0.13 (0.25)	0.011 (0.020)
VSH (V)	0.20 (0.39)	0.11 (0.22)	0.033 (0.058)
VCH (V)	0.08 (0.12)	0.02 (0.05)	0.007 (0.009)
TL (ns)	0.93 (1.74)	0.14 (0.41)	0.032 (0.090)
TH (ns)	1.05 (1.80)	0.22 (0.51)	0.030 (0.112)
TLC (ns)	0.12 (0.25)	0.03 (0.07)	0.009 (0.019)
THC (ns)	0.21 (0.56)	0.05 (–)	0.011 (–)

Table 2: Cross-validation root mean squared error for the stochastic-process predictor at various stages. Figures in parentheses are for a second-order, least-squares regression model (there were insufficient runs to fit the regression models for *THC* at Stages 2 and 4). *TLC* and *THC* were redefined after Stage 1.

Prediction accuracy, even from the stochastic-process model, was judged to be insufficient for formal optimization. This is not surprising given the very wide initial ranges for the designable device sizes. Thus we relied on visualization of the predictor to guide the choice of narrower ranges for the next stage.

For instance, Figure 4 shows the estimated main effect, computed from (9), of input *P1* on the voltage spike *VSL*. The estimated main effect of *P1* accounts for 53.0% of the variability in the *VSL* predictor across the 36-variable input space and is therefore judged to be the dominant factor for *VSL* at this stage. The plot indicates that *P1* should take a low value to minimize *VSL*. The other responses also had to be considered, however, in choosing a new range for *P1*, as summarized in Figure 5. Recall that small voltage spikes, small primary time delays, and positive values of the secondary time-delay constraint slacks in (1) are desirable. The estimated effects for *VSL* and for the secondary time-delay slacks suggest small values of *P1* (Figures 5(a) and 5(c)). At very small *P1* values, *TL* rises sharply (Figure 5(b)). As a compromise between these somewhat conflicting considerations, the lower 10% and the upper 40% of the original *P1* range was removed for Stage 2. Analogous plots were produced and

examined for all 20 designable inputs.

Some important interactions were also identified. The interaction, computed from (10), between $P3$ and $N3$ accounts for 16.2% of the variability in the VSH predictor, for example. Figure 6 shows the estimated joint effect, i.e., the predictor with all other factors integrated out, of these two factors on VSH. Their estimated main effects account for 10.7% and 6.5% of the VSH predictor variability; together with the interaction effect, they account for a total of 33.5%. The plot suggests that small values of $P3$ in combination with large values of $N3$ are undesirable. The only other estimated interactions accounting for at least 5% of predictor variability are two more relating to VSH and one relating to VCL.

The noise factors, $U_1, \ldots, U_{16}$, are largely ignored in these plots of estimated effects, as they are simply being averaged out. At Stage 1, however, this was not a major concern, because we found small estimated effects relative to those for the designable inputs. The 16 noise factors together account for modest percentages of predictor variability: the largest percentage is 14% for VCL. Similarly, the estimated interactions between the designable inputs and the noise factors were small. As the ranges of the designable inputs were narrowed in subsequent stages, the noise factors became relatively more important.

A new range was chosen for each designable input using main-effect plots like Figure 5 and checking the decisions against joint-effect plots like Figure 6 where necessary. The new ranges for Stage 2 (and subsequent stages), relative to those for Stage 1, are summarized in Figure 7.

6.2 *Stage 2*

A 90-point Latin hypercube design was generated for the Stage 2 region. Model fitting proceeded as for Stage 1.

Making positive the secondary time-delay constraint slacks in (1) proved to be very difficult, with the danger that these secondary considerations would dominate

the problem. An engineering decision was made to relax the constraints so that each secondary time delay is no more than 20% (rather than 10%) of the corresponding primary delay. Thus, we redefined the constraint slacks to be

$$TLC = 0.2\,TL - TLO \geq 0 \quad \text{and} \quad THC = 0.2\,TH - THO \geq 0.$$

Table 2 shows the cross-validation root mean squared errors of prediction at Stage 2. The stochastic-process predictors again perform well relative to second-order regression models. There was much improvement in accuracy compared with Stage 1, because we were predicting over a smaller region. Further improvement was still required, however. For practical purposes, it was felt that a root mean square error of 0.05 ns for the primary delays and 0.02 V for the voltage spikes would be adequate.

Inspection of estimated main-effect and, where necessary, joint-effect plots indicated that some variables apparently important at Stage 1 were less important at Stage 2, while others emerged as relevant for one response or another. The main patterns carried over, however, from Stage 1. The noise factors and their interactions with the device sizes still appeared to be relatively unimportant.

In addition to the visualization of estimated effects, some tentative optimizations were performed. For given maximum primary delay time, $t_{\max}$, the minimization of the voltage spikes in the objective function (2) subject to the primary and secondary delay constraints in (3) was carried out using the fitted predictors. For given values of the device sizes, $T1, \ldots, DN1$, we predicted TL, for example, using the values of $U_1, \ldots, U_{16}$ in the experimental design as a Monte Carlo sample. Taking the sample mean plus three times the sample standard deviation of TL over these 90 noise combinations gave an estimate of TL^+. Optimization was carried out using the NPSOL algorithm (Gill et al., 1986). Trial and error with various values of $t_{\max}$ gave two points in the $T1, \ldots, DN1$ space for $t_{\max}$ as low as 5.4 ns and another point for lower values of $t_{\max}$. Regions around these points were constructed.

6.3 Stage 3

Ninety runs were made for each of the three regions labelled Stages 3.1, 3.2, and 3.3 in Figure 7. The results were clearly disappointing in terms of the engineering criteria. For example, the 90 runs in the first experimental design gave *VCH* values all above 0.30 V with *TL* and *TH* primary delays mainly in the range 5.0–7.0 ns. Far smaller voltage spikes were expected and later found. In Section 6.5 we report confirmation runs where the primary time delays are in similar ranges but the *worst-case* voltage spike (mean plus three standard deviations) is *less* than 0.31 V for all four voltage spikes.

In a review of the project to date, two problems were uncovered. First, re-examination of the earlier estimated-effect plots showed some inappropriate ranges for Stage 3. Consider, for example, *N1*. Figure 8 shows the estimated main effects for Stage 2. We see that small values of *N1* reduce *TL* substantially, with little effect on *TH*. The voltage spike effects appear to be small. The indication is that the *N1* range should have been lower for Stage 3 than at Stage 2. Similar comments apply to the *P2* range.

Secondly, the project review indicated a need to reconsider again the engineering trade-offs. Figure 9 shows the estimated main effects of *T1* at Stage 2. Small values of *T1* reduce *VSL* and *TH* but increase *TL*. Analogous plots of the estimated *T2* effects show a similiar pattern: small values of *T2* reduce *VCH* and *TL* but increase *TH*. To obtain smaller values of *VSL* and *VCH*, which tend to be the largest voltage spikes, *T1* and *T2* should be reduced. The Stage 2 lower limits for *T1* and *T2* were set to accommodate the *ZUP* and *ZDOWN* constraints (see Section 2.3). These constraints were relaxed, allowing smaller values for *T1* and *T2* for subsequent stages.

6.4 Stages 4, 5, and 6

A 90-point Latin hypercube design was used to generate the Stage 4 experiment. As shown in Table 2, we found considerable improvement in accuracy relative to earlier

Maximum delay (ns)		VSL (V)	VCL (V)	VSH (V)	VCH (V)	TL (ns)	TH (ns)
5.9	Predicted	0.356	0.090	0.187	0.261	4.854	4.893
	Actual	0.353	0.096	0.171	0.254	4.932	4.861
6.4	Predicted	0.292	0.063	0.088	0.234	5.287	5.174
	Actual	0.295	0.072	0.057	0.245	5.353	5.211

Table 3: Predicted and actual outputs at two sets of device sizes found by optimization subject to different maximum delays, $t_{\max}$.

stages. The much-reduced ranges for the device-size variables limited their effects, making modelling easier. With smaller effects from the device-size variables, the noise factors and their interactions with the device sizes were estimated to be relatively much more important than at Stages 1 or 2.

Encouraged by these results, we optimized the objective (2) subject to the constraints (3) via the predictors for primary-delay constraints of 5.9 and 6.4 ns. These optimizations considered variation in the noise factors via Monte Carlo sampling, as described in Section 6.2. Table 3 compares the predicted outputs with the actual values from confirmation runs at the two sets of device sizes identified; there is fairly good agreement. Here, the predicted values were means over the Monte Carlo sampling of the noise factors, $U_1, \ldots, U_{16}$, while the confirmation runs had $U_1, \ldots, U_{16}$ set to zero, i.e., nominal conditions. More extensive confirmations are reported in Section 6.5.

We also tried optimizing for a primary-delay constraint of 6.9 ns. The solution had *P1*, *N2*, and *P4* constrained by their Stage 4 lower bounds. Thus, we collected new data for Stage 5, with lower ranges for these device sizes; the other ranges remained unchanged (see Figure 7). Using predictors fitted from another 90-point Latin hypercube sample, we repeated the optimization with primary-delay constraints of 6.4, 6.9, 7.4, 7.9, 8.4, and 8.9 ns, giving six corresponding combinations of the device-size variables. Six confirmation runs showed disappointing accuracy for *TH* and for *VSH*. We added them to the Stage 5 Latin hypercube to improve prediction accuracy near

the six apparent optima. (The correlation parameters in (5) were not refitted by maximum likelihood). New confirmation runs for the reoptimized device-size configurations showed improved accuracy and were deemed satisfactory for a primary-delay constraint of up to 8.4 ns.

To find good solutions for higher values of the time-delay constraint we repeated the steps in Stage 5. A Stage 6 experimental design of 90 points, with revised ranges for *T1*, *T2*, *P1*, *N2*, and *P4* (see Figure 7), was constructed. After optimizing for various values of the constraint, the confirmation runs were added to the design. Further confirmation runs for the re-optimized device sizes indicated that accuracy was satisfactory for delay constraints up to 13.4 ns.

6.5 *Confirmation*

The confirmation runs carried out at Stages 4, 5, and 6 were at nominal conditions for the noise factors. Before use in a specific application, more careful confirmations were necessary.

For example, one application had a time-delay constraint of 7.0 ns. The appropriate fitted predictors, those from Stage 5, were used to optimize the device sizes. The circuit simulator was run with the suggested device sizes and Monte-Carlo sampling of the noise factors. The means and standard deviations for the various outputs agreed well enough for practical purposes with the predicted means and standard deviations shown in Table 4.

For at least one application, a hand-optimized circuit produced by an experienced engineer was available. Comparison showed that the method proposed here produced means and standard deviations for the time delays and for the voltage spikes that were 5–10% smaller.

20

Ouptut	Mean		Standard deviation	
	Simulator	Predicted	Simulator	Predicted
VSL (V)	0.229	0.233	0.0286	0.0246
VCL (V)	0.0506	0.0537	0.00618	0.0088
VSH (V)	0.152	0.122	0.0357	0.0054
VCH (V)	0.236	0.230	0.0244	0.0218
TL (ns)	5.84	5.72	0.431	0.414
TH (ns)	5.42	5.35	0.417	0.523

Table 4: Means and standard deviations from simulator runs compared with predictions from the fitted models.

7 Discussion

This example demonstrates the usefulness of using statistical techniques in conjuntion with engineering simulation models. Indeed, attempting to optimize many performance characteristics over a very high dimensional space, while minimizing variability from processing noise, makes a systematic approach essential. We believe the stochastic-process model is well suited to large-scale problems with many inputs and outputs. The data-adaptive nature of the model obviates the need to specify nonlinearities and interactions, a daunting task when repeatedly modelling eight responses as functions of 36 explanatory variables.

There are, of course, other data-adaptive methods, e.g., generalized additive models (Hastie and Tibshirani 1990). The stochastic-process model was motivated specifically to deal with deterministic input-output relationships, however, and has proved in many applications to give relatively good accuracy when there are many inputs and few runs.

Some lessons were learnt during this example about the sequential strategy. Stage 3, with hindsight, could have been better planned. The Stage 3 regions were identified by optimizations using the Stage 2 predictors, but at Stage 2 accuracy was known to be unsatisfactory. An optimizer will tend to favour predictions that are in error by being too optimistic, i.e., searching introduces bias. This is more of a problem when errors are large. Another problem is that optimization locates a *point*. If accuracy is

insufficient, we want reduced *ranges* for the next stage.

When accuracy is inadequate, visualization of the predictors appears to be more useful for choosing ranges for the next stage. Another advantage of visualization is that the engineering trade-offs between conflicting performance measures can be seen. The engineering criteria had to be changed several times during the output-buffer project; this is not unusual, we suspect.

The overall strategy was sequential in stages. Within Stages 5 and 6 we also optimized sequentially. Confirmation runs at points predicted to be optimal were incorporated into the predictor, and the constrained optimizations were re-run. This raises the possibility of a fully sequential strategy, after an initial experimental design, which would be more automated. Such methods have worked well in much simpler contexts (Schonlau, Welch, and Jones, 1997). Extensions to deal with noise factors and multiple engineering targets are required to deal with systems like the output-buffer problem presented here. To deal with more device sizes, Bates et al. (1996) described a method for decomposing circuits prior to statistical modelling.

Acknowledgements

References

Alvarez, A.R., Behrooz, L.A., Young, D.L., Weed, H.D., Teplik, J., and Herald, E.R. (1988) Application of statistical design and response surface methods to computer-aided VLSI device design. *IEEE Transactions on Computer-Aided Design*, **7**, 272–288.

Bates, R.A., Buck, R.J., Riccomagno, E., and Wynn, H.P. (1996) Experimental design and observation for large systems. *J. R. Statist. Soc.* B, **58**, 77–94.

Bernardo, M.C., Buck, R., Liu, L., Nazaret, W.A., Sacks, J., and Welch, W.J. (1992) Integrated circuit design optimization using a sequential strategy. *IEEE Transactions on Computer-Aided Design*, **11**, 361–372.

Box, G.E.P. and Draper, N.R. (1959) A basis for the selection of a response surface design. *J. Amer. Statist. Ass.*, **54**, 622–654.

Brayton, R.K., Hachtel, G.D., and Sangiovanni-Vincentelli, A.L. (1981) A survey of optimization techniques for integrated-circuit design. *Proceedings of the IEEE*, **69**, 1334–1363.

Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991) Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *J. Amer. Statist. Ass.*, **86**, 953–963.

Gill, P.E., Murray, W., Saunders, M.A., and Wright, M.H. (1986) User's guide for NPSOL, Version 4.0. Report SOL 86-2, Department of Operations Research, Stanford University, Stanford.

Hastie, T.J. and Tibshirani, R.J. (1990) *Generalized Additive Models*. London: Chapman and Hall.

Iman, R.L. and Conover, W.J. (1982) A distribution-free approach to inducing rank correlation among input variables. *Comm. Statist. Simul. Comp.*, **11**, 311-334.

McKay, M.D., Conover, W.J., and Beckman, R.J. (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, **21**, 239–245.

Sacks, J., Schiller, S.B., and Welch, W.J. (1989) Designs for computer experiments. *Technometrics*, **31**, 41–47.

Sacks, J., Welch, W.J., Mitchell, T.J., and Wynn, H.P. (1989) Design and analysis of computer experiments (with discussion). *Statist. Sci.*, **4**, 409–435.

Schonlau, M., Welch, W.J., and Jones, D.R. (1997) Global versus local search in constrained optimization of computer models. Research Report RR-97-09, Institute for Improvement in Quality and Productivity, University of Waterloo, Waterloo.

Su, H., Nelder, J.A., Wolbert, P., and Spence, R. (1996) Application of generalized linear models to the design improvement of an engineering artefact. *Quality and Reliability Engineering International*, **12**, 101-112.

Taguchi, G. (1986) *Introduction to Quality Engineering*. Tokyo: Asian Productivity Organization.

Welch, W.J., Buck, R.J., Sacks, J., Wynn, H.P., Mitchell, T.J., and Morris, M.D. (1992) Screening, predicting, and computer experiments. *Technometrics*, **34**, 15–25.

Welch, W.J., Yu, T.K., Kang, S.M., and Sacks, J. (1990) Computer experiments for quality control by parameter design. *J. Qual. Technol.*, **22**, 15–22.

Yu, T.K., Kang, S.M., Sacks, J., and Welch, W.J. (1991) Parametric yield optimization of CMOS analogue circuits by quadratic statistical circuit performance models. *International Journal of Circuit Theory and Applications*, **19**, 579–592.
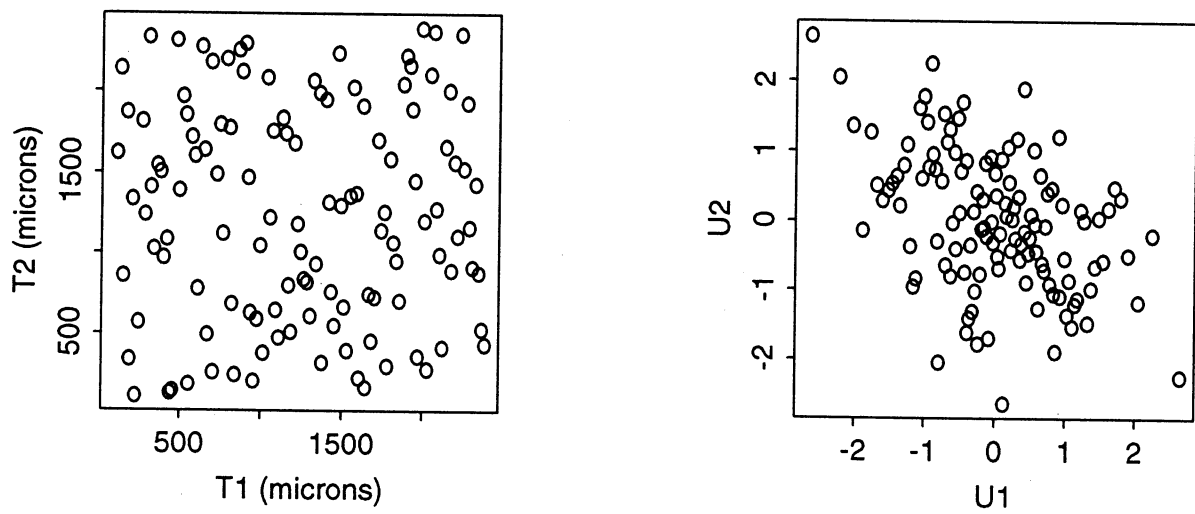
Figure 1: Two-dimensional projections of the Stage 1 experimental design for the first two device sizes, *T1* and *T2*, and for the first two noise factors, U_1 and U_2.
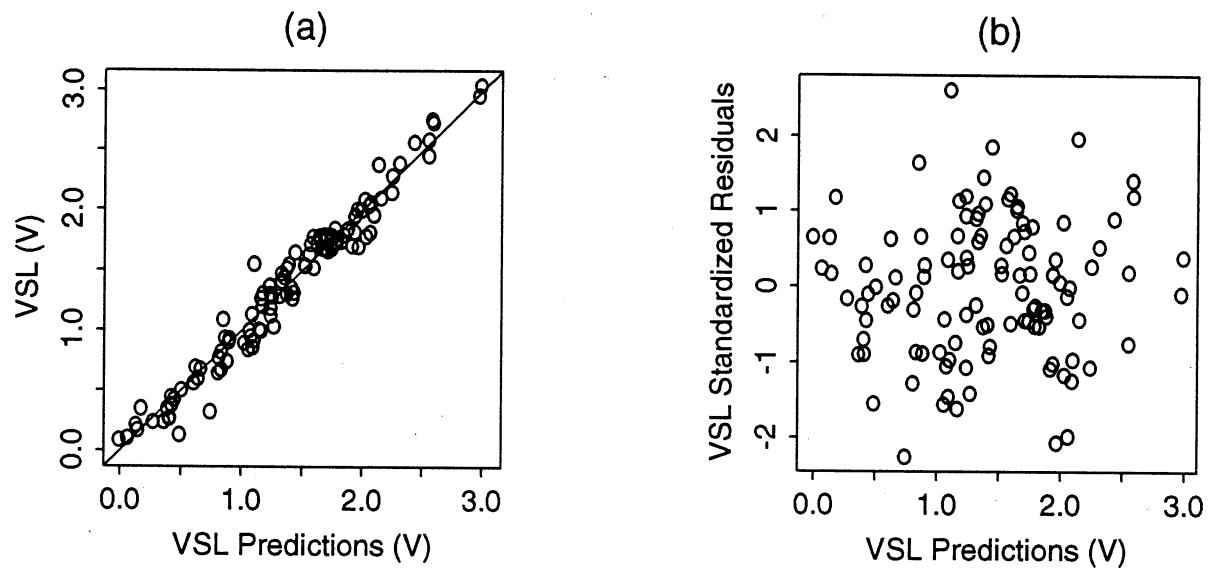
Figure 2: Modelling of *VSL* at Stage 1: (a) Actual values versus their cross-validation predictions and (b) Standardized cross-validation residuals versus cross-validation predictions.
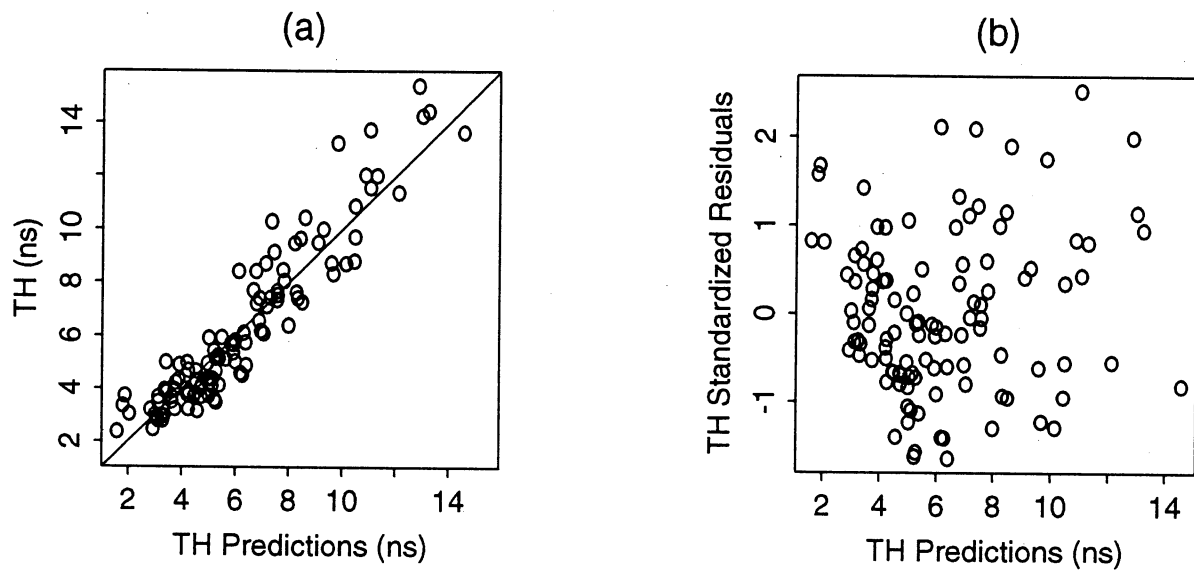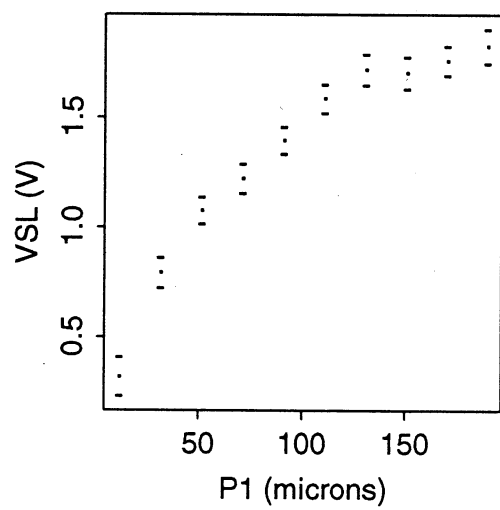
Figure 3: Modelling of *TH* at Stage 1: (a) Actual values versus their cross-validation predictions and (b) Standardized cross-validation residuals versus cross-validation predictions.

Figure 4: Estimated main effect of *P1* on *VSL* at Stage 1. The bounds are pointwise approximate 95% confidence intervals.
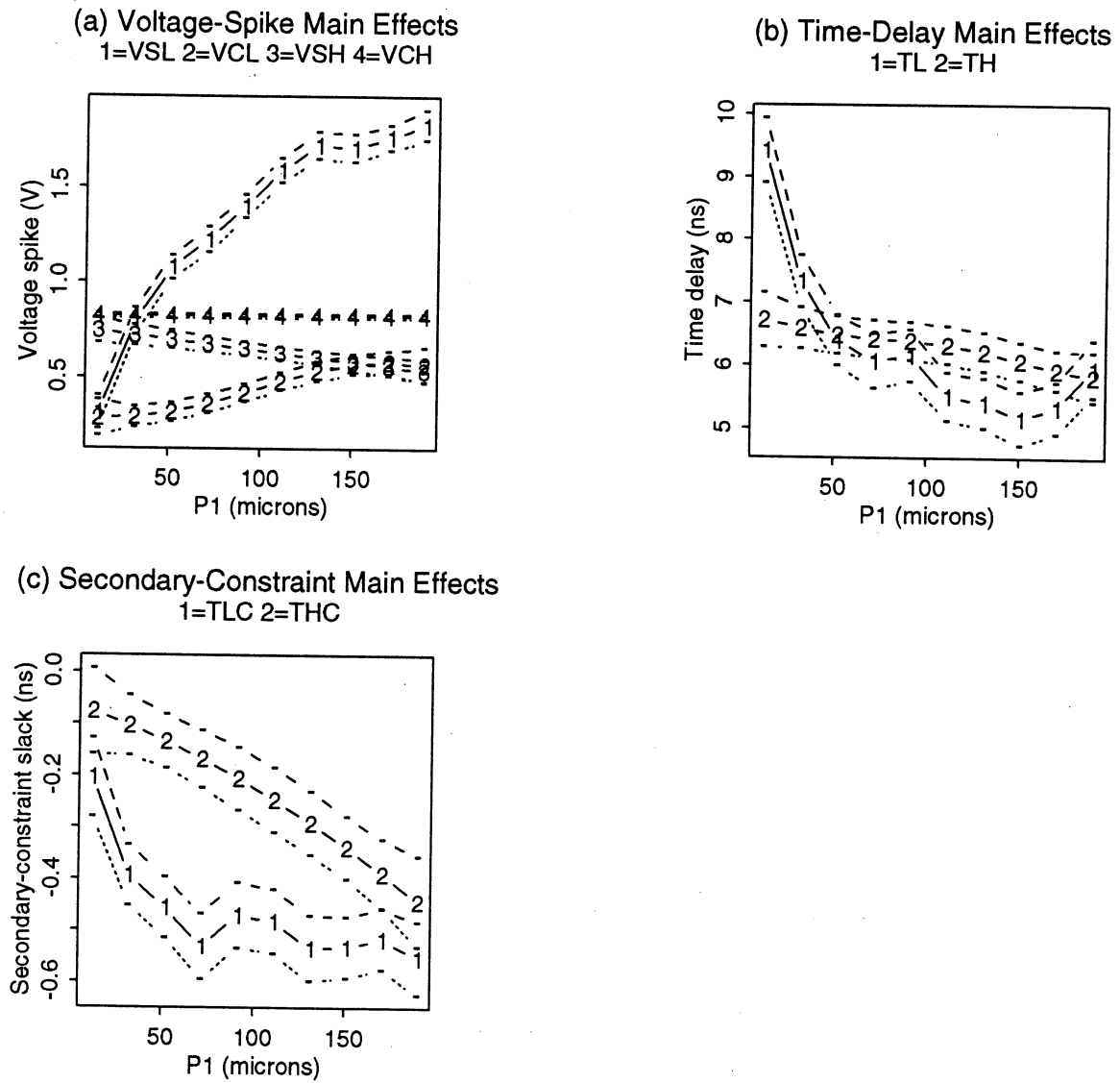
(a) Voltage-Spike Main Effects
1=VSL 2=VCL 3=VSH 4=VCH

(b) Time-Delay Main Effects
1=TL 2=TH

(c) Secondary-Constraint Main Effects
1=TLC 2=THC

Figure 5: Estimated main effects of *P1* at Stage 1 on (a) the voltage spikes, (b) the primary time delays, and (c) the secondary time-delay constraint slacks. The bounds are pointwise approximate 95% confidence intervals.
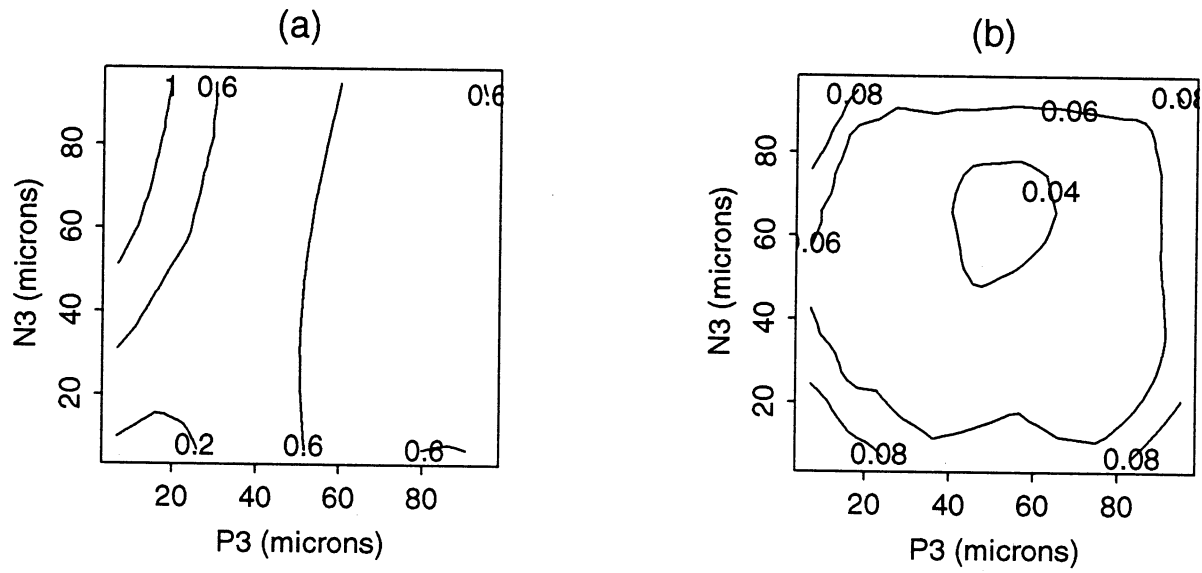
(a)

(b)

Figure 6: Joint effect at Stage 1 of *P3* and *N3* on *VSH*: (a) Estimated effect and (b) standard error of the estimated effect.
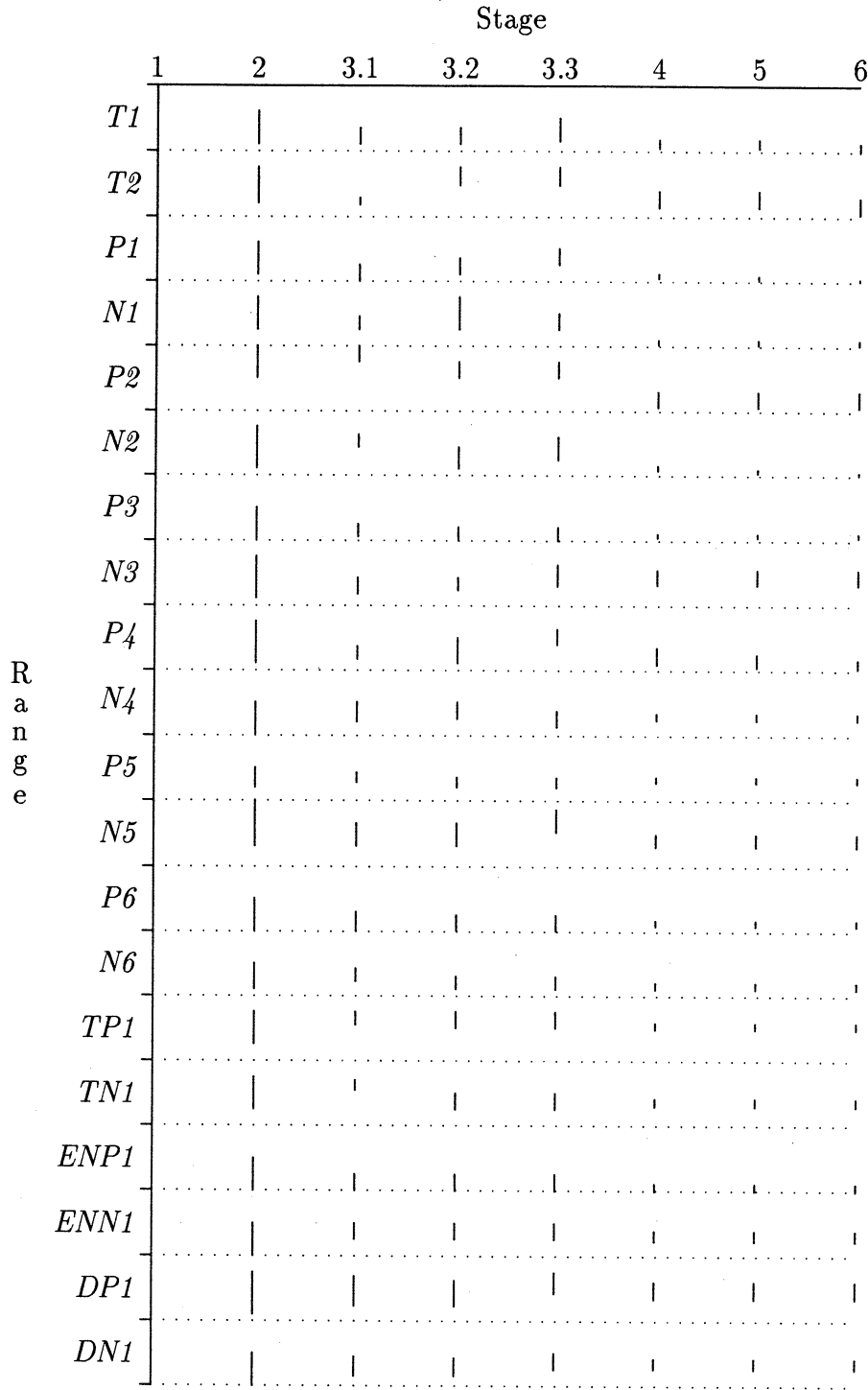
Figure 7: Ranges for the designable device sizes at each stage of the experiment. The Stage 1 ranges in Table 1 are normalized to have the same length; the bars for subsequent stages are relative to Stage 1.
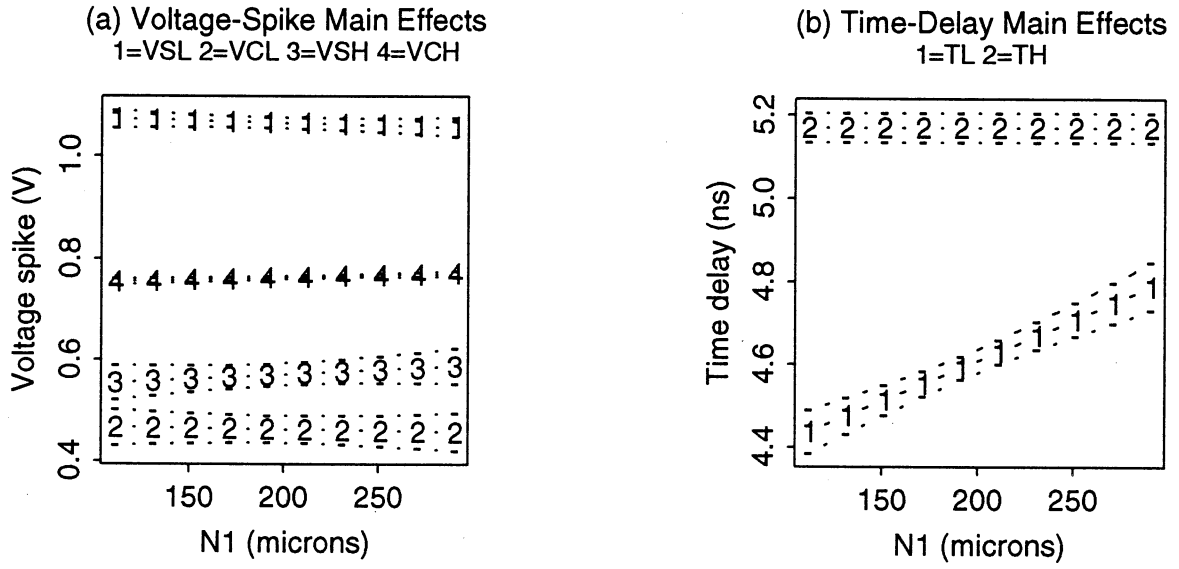
(a) Voltage-Spike Main Effects
1=VSL 2=VCL 3=VSH 4=VCH

(b) Time-Delay Main Effects
1=TL 2=TH

Figure 8: Estimated main effects of *N1* at Stage 2 on (a) the voltage spikes and (b) the primary time delays. The bounds are pointwise approximate 95% confidence intervals.

(a) Voltage-Spike Main Effects
1=VSL 2=VCL 3=VSH 4=VCH
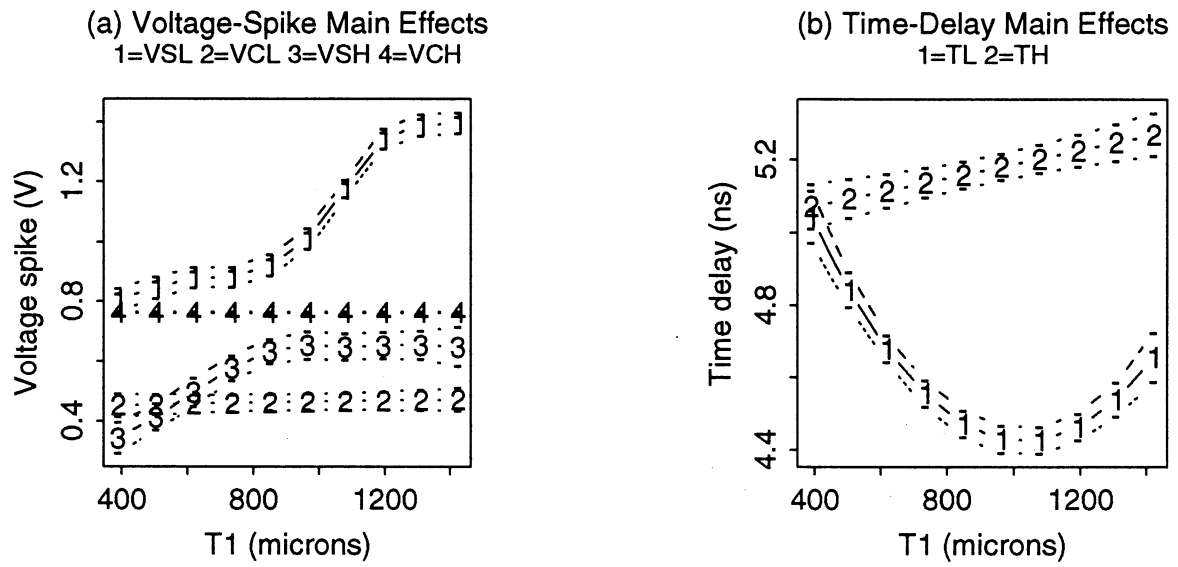
(b) Time-Delay Main Effects
1=TL 2=TH

Figure 9: Estimated main effects of *T1* at Stage 2 on (a) the voltage spikes and (b) the primary time delays. The bounds are pointwise approximate 95% confidence intervals.