



Your Short Cut to Knowledge

The following is an excerpt from a Short Cut published by one of the Pearson Education imprints.

Short Cuts are short, concise, PDF documents designed specifically for busy technical professionals like you.

We've provided this excerpt to help you review the product before you purchase. Please note, the hyperlinks contained within this excerpt have been deactivated.

Tap into learning—NOW!

Visit www.informit.com/shortcuts for a complete list of Short Cuts.



SAMS

Cisco Press

**IBM
Press™**

que®

Taguchi Methods for Robust Software Design

parameters simultaneously. His use of orthogonal matrices permits a multifactorial analysis that is far more efficient than a conventional “bottleneck” analysis. It also allows the study of factor interactions.

Taguchi Methods for Robust Software Design

We will begin with a brief review of Taguchi Methods as they were developed over the past 50 years, originally for robust hardware design. Statistical quality control first began in manufacturing in the early 1900s by testing the output of production lines in an effort to reduce process and product variability. Deming and others soon concluded that although downstream testing might reduce the number of defective products shipped, it could never achieve any intrinsic process improvement or product quality. However, many years passed before it was accepted that upstream preproduction experiments done during design could contribute to the optimization of industrial processes, quality improvement of products, or reduction of cost and waste. As you will see, this is critical to the design of robust, trustworthy software. Software development really has no manufacturing component; it is the direct result of design and continual

redesign and upgrading. Conventional wisdom dating from Aristotle says that in designing an experiment, you should change only one parameter at a time. This is the essence of the Western tradition of logical analysis. But this analytic approach can’t discover parameter interactions because if a system is more than the sum of its parts, analysis will always fail to discover its essence. Taguchi Methods let you change many factors at a time, albeit in a systematic way, to discover both that factor’s major effects and its interaction or systemic effects. As soon as a factor (parameter) and its effects are well understood, the designer can take appropriate steps through parameter design to eliminate potential defects in the resulting product. This technology was developed by Professor Genichi Taguchi, director of the Japanese Academy of Quality and four-time recipient of the Deming Prize. He devised a quality improvement technique that uses statistical experimental design for the effective characterization of a process or product at design time, together with a statistical analysis of variability. This allows quality to be designed as far upstream as possible, at the initial design and prototyping stages of product development.¹

Taguchi defines *quality* in a negative manner as the loss imparted to society from the time a product is shipped.

SECTION 3

Taguchi Methods for Robust Software Design

This includes the cost of customer dissatisfaction, including loss of reputation, and goodwill. Apart from direct loss due to warranty and service costs, an indirect loss occurs due to market-share loss and any costs needed to overcome consequent loss of competitive advantage. Taguchi's loss function establishes a value basis for the development of quality products. A product does not fail to exhibit quality only when it is outside specification, but also when it deviates from its *target value*. Thus, quality improvement now means minimizing the variation of product performance about its target value. Figure 1 depicts Taguchi's famous quadratic quality loss function, in which loss is proportional to the square of the deviation from the target value:

$$L(Y) = (M/D^2) (Y - t)^2$$

In the formula for this curve, $L(Y)$ is the loss to society when a product's performance (Y) deviates from its target (t). M is the producer's monetary loss when the customer's tolerance (D) is exceeded. The objective of Taguchi Methods for product design improvement is to identify *controllable* (or design) factors and their values or parameters. By optimizing these parameters, any variation in product performance can be minimized and the product made robust with respect to changes in operating

and environmental conditions. Also, the Taguchi approach deals with *uncontrollable* (or noise) factors in the product's operating environment that affect product performance. This too enhances product robustness.

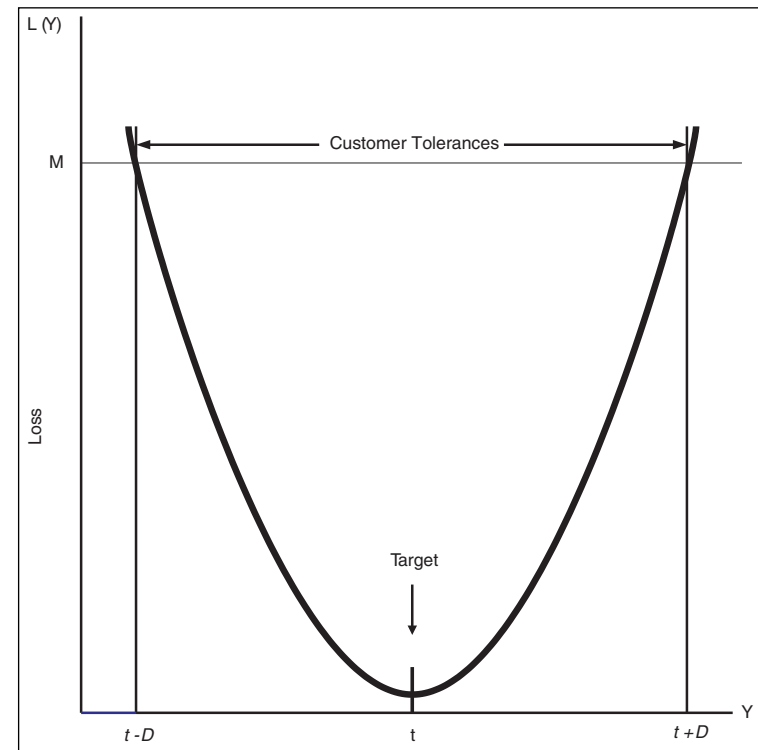


FIGURE 1 Taguchi's Loss Function $L(Y)$

Taguchi Methods for Robust Software Design

Key to the design of statistical experiments in Taguchi Methods is the use of orthogonal matrices, which are described later. This technology was developed by Plackett and Burman in 1946 for multifactorial experiments and later was adopted by Taguchi.² Controllable factors that reduce performance due to noise factors but keep it on target are identified. The effect of the noise factors is reduced or eliminated rather than the noise factors themselves because they are defined as uncontrollable. Performance variation is simulated in these statistical experiments by systematically varying noise factors for each of the settings of the controllable factors in the study. This is called a reduced multifactorial study in Taguchi Methods because not all the factors of a complex product or system can be studied at once. Figure 2 is a simplified paradigm of the Methods that shows the parts and the process. Further technical details appear later in this article. In Figure 2, the rows of the *inner* (fractional orthogonal) *array* are controllable factor-level settings in which every level setting of each factor chosen for the statistical experiment with every other level of all the others chosen factors the same number of times. At every level combination of the controllable factors, observations are obtained while changing the settings of the noise

factors. For the purposes of the experiment, we can change the uncontrollable factors to discover design tolerances, just as the electrical engineer changes margins on a circuit design to discover failure modes and improve robustness. The *outer arrays* shown in the figure are used to determine the level combinations of the noise factors. This allows the design engineer or her statistician to simulate the effect of variability and optimize the settings to minimize performance variability from the target value and thus make the design more robust. For each of the m rows in the inner array, the n rows of the outer array provide n observations on the response being investigated, or nm values for the entire experiment.

The outer array is not a random subset of values from the noise space but rather test levels of the noise factors chosen to cover or span the noise space. If the distribution of a noise factor is known with a mean M and standard deviation S , and the factor is linear, Taguchi suggests it be tested at two levels: $(M - S)$ and $(M + S)$. However, if its effect is not linear, it should be tested at three levels: $M - S\sqrt{(3/2)}$, M , and $M + S\sqrt{(3/2)}$. If a prototype hardware or software product is available, the parameter design experiment can be conducted by actual trials. If the product is unavailable, the experiment must be

SECTION 3

Taguchi Methods for Robust Software Design

conducted on the design itself using a response model.³ Such a model simply relates product performance to both signal (controllable) and noise (uncontrollable) factors. The results of the statistical experiment are the performance measures shown on the right side of the figure. The noise performance measures (NPM) give the variation in response at each setting to determine the controllable factors that can minimize the effects of noise on performance. The target performance measures (TPM) identify the controllable factors that have the largest effect on mean performance response and thus can be modified to bring mean response to the design target. Before Taguchi, such statistical experiments focused on the TPM only. Taguchi's major contribution was to include the NPM and a suitable measure for it—the *signal-to-noise ratio* (SNR), which estimates the inverse of the coefficient of variation, or ratio M/S —the mean divided by the standard deviation. Taguchi's formula for SNR is

$$\text{SNR} = \log_{10} (M^2/S^2)$$

in which M and S are the mean and the standard deviation of the Y_{ij} , or the data column in Figure 2.

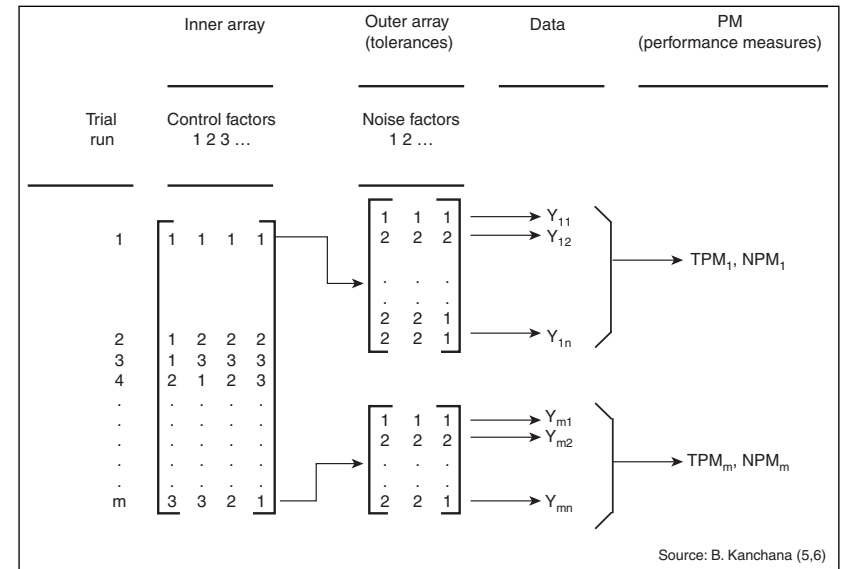


FIGURE 2 Experimental Paradigm for Taguchi Optimization

Table 1 lists the steps of Taguchi Methods for either hardware or software design. This article gives examples of both to illustrate the use of the Methods. The principal novel idea of Taguchi Methods is that the statistical testing of a new or improved product should be carried out at the design stage to make the product robust to variations in both its manufacturing and usage environments. Because software is not manufactured in the ordinary sense of the term, the focus here is on design

SECTION 4

An Example from Engineering Design

for trustworthiness in use. Hence, we will not repeat or even summarize the vast literature on the successful application of Taguchi Methods in manufacturing, but rather focus on their application to the design of robust software.

TABLE 1 Steps in Taguchi Methods

Step Number	Step	Description
1	Define the problem.	Clearly state the problem to be solved.
2	Determine the objective.	Identify performance responses to be optimized.
3	Brainstorm.	Identify both signal and noise factors.
4	Design the experiment.	Choose factors and build orthogonal arrays.
5	Conduct the experiment.	Perform trials and collect data.
6	Analyze the data.	Evaluate TMP and NPM for each trial run.
7	Interpret the results.	Identify variability and target control factors and select their optima.
8	Confirm the experiment.	Prove that new parameter settings enhance robustness.

An Example from Engineering Design

Although our interest here is software design, a simpler yet realistic example from electrical circuit design may be very helpful in giving you insight into Taguchi Methods. This application shows the trade-off potential between the three stages of Taguchi design and also gives you insight into the increasing replacement of electronic circuits by software algorithms. Analog filters for radio and other signal processing have long since been replaced by digital (software-programmed) filters. Modern communication receivers are simply computer programs that tune to and receive signals on a chip-based computer. Similarly, the most advanced radio transmitters are computers that, rather than computing a number, prepare and transmit a signal at a chosen or preprogrammed frequency. As an example, we take a case study from the literature by Kanemoto.⁴

Much of electrical circuit design involves adjusting part values and their tolerances to achieve a target performance over a certain range with the minimum number of parts and the least constraint on part tolerances. Normally, electronic components have tolerance values of