Robust Design: An introduction to Taguchi Methods

The theoretical foundations of Taguchi Methods were laid out by Genichi Taguchi, a Japanese engineer who began working for the telecommunications company, ECL (Electrical Communications Lab, a part of NT&T), in 1950's. After World War II, his team was involved in the design of telephone system components, and successfully produced designs that performed better than their main rivals, Bell Labs, of the US.

What do we mean by "performed better"? In Taguchi's view, the traditional definitions of quality were inadequate and perhaps vague. He developed his own definitions of these concepts over several years of working with various projects on product and process design.

Robustness: Taguchi's definition of a robust design is: "a product whose performance is *minimally sensitive* to factors causing variability (at the lowest possible cost)".

Taguchi's view was that in traditional systems, robustness (or in general, quality) was measured by some performance criteria, such as:

meeting the specifications
% of products scrapped
Cost of rework
% defective
failure rate

However, these measures of performance are all based on make-and-measure policies. They all come too late in the product development cycle. Robust design is a systematic methodology to design products whose performance is least affected by variations, i.e. noise, in the system (system variations here means variations due to component size variations, different environmental conditions, etc.)

Some statistical tools are necessary to generate robust designs. These are broadly covered in standard courses on Design of experiments. We will study the basic ideas.

The problem with traditional measures of Quality

(1) As mentioned above, methods such as "failure rate", "meeting the specs", "% scrapped" are essentially tools for testing. These methods also use statistics (methods that are called Statistical Quality Control, Control charts, etc.) However, these methods do not give guidance for product design. We get the information that x% of products are failing, and perhaps the primary failure modes (main reasons for failure); or we get data about which spec was out-of-tolerance under testing conditions. By studying this information, the designer then tries to modify the design component or module that appears to be causing the failure. However, after this engineering change, some other performance factor may become more sensitive to noise. This approach of design works as follows: design → test → find problem → solve problem → test → find problem → … until no

problem is discovered. The approach is called “plug-the-leak” or “whack-the-mole” approach. It is time-consuming and costly.

(2) The %-defective fallacy

Another problem with the traditional SQC approaches is called the %-defective fallacy by Taguchi, and is best illustrated via the following example. In the 1980’s Sony was manufacturing television tubes in two plants. Sony-Japan was following Taguchi’s principles, and the main product specification, color density, which was achieved by the plants, followed the bell-shaped curve shown in the figure below. Some tubes were outside the acceptable range (m-5, m+5), and therefore there was some waste. Sony-USA was using a SQC based approach that rejected parts outside the range (m-5, m+5), and the production system was tuned such that there were almost no parts outside the range (i.e. zero %-defective); however, the color-density followed a roughly uniform distribution.

It was discovered that customers perceived Sony-Japan TV sets to be better quality, even though this plant had a higher %-defective rate. This perception can be explained as follows: a larger percentage of the customers buying the Sony-Japan TV sets got products that were very close to the ideal spec (color-density = m), and likewise, very small % of the customers of Sony-Japan ended up with C-grade TV’s. Thus, statistically, the customer feedback was better for the products coming from the Japan factory.

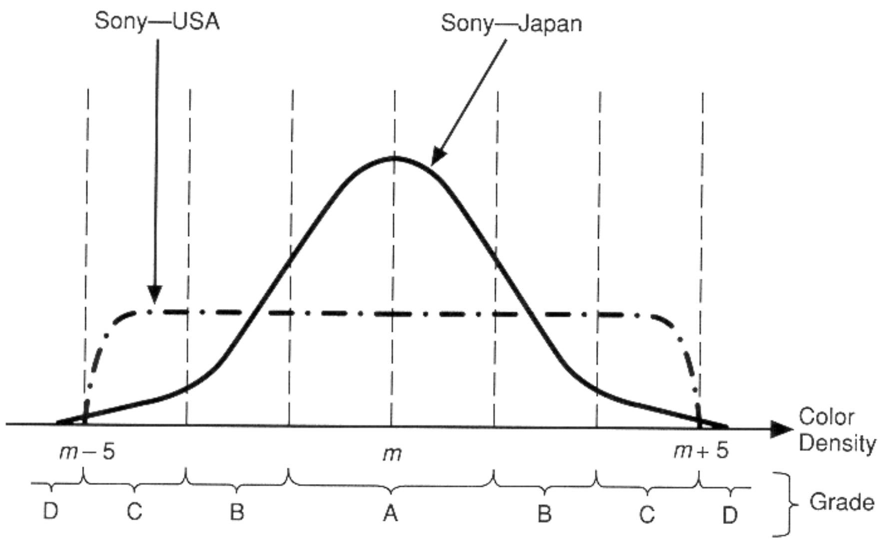

Figure 1. The %-defective fallacy

Two examples of Robust designs

Example 1. Caramel candy

A company producing caramel candy discovered that the “chewability” of the candy, which should be just right for the candy to be enjoyable, varied too much with the external temperature (see figure 2). The goal was to reduce the temperature dependence.

Over ten ingredients were mixed to produce caramel. Different mixing ratios would yield different properties of the resulting product. After much experimentation (a ~25 year old Taguchi was consulting the company in late 1940's). After experimentation, the plasticity of the new candy was much more stable relative to ambient temperature changes, as seen in figure 2.

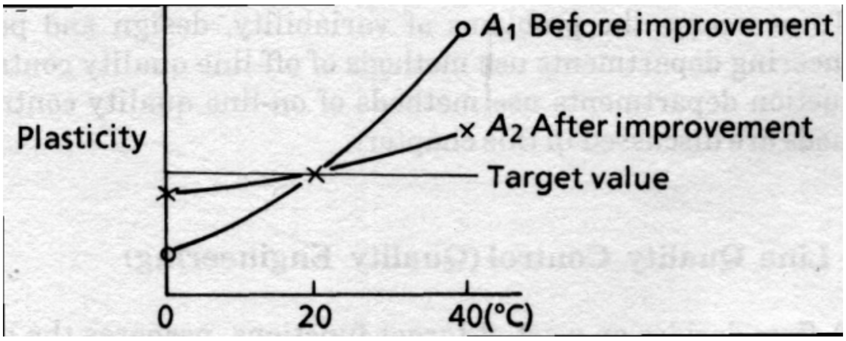

Figure 2. Plasticity of candy versus ambient temperature variations (noise)

Example 2. Amplifier design

An electronics engineer is designing a power amplifier, which is required to output 115V DC output for a nominal input voltage of 100V AC. The two main components are a transistor (which is characterized by its forward current gain, h_{FE}), and a resistor (characterized by its resistance, R). Different combinations of values for h_{FE} (design parameter A) and R (design parameter B) can provide this.

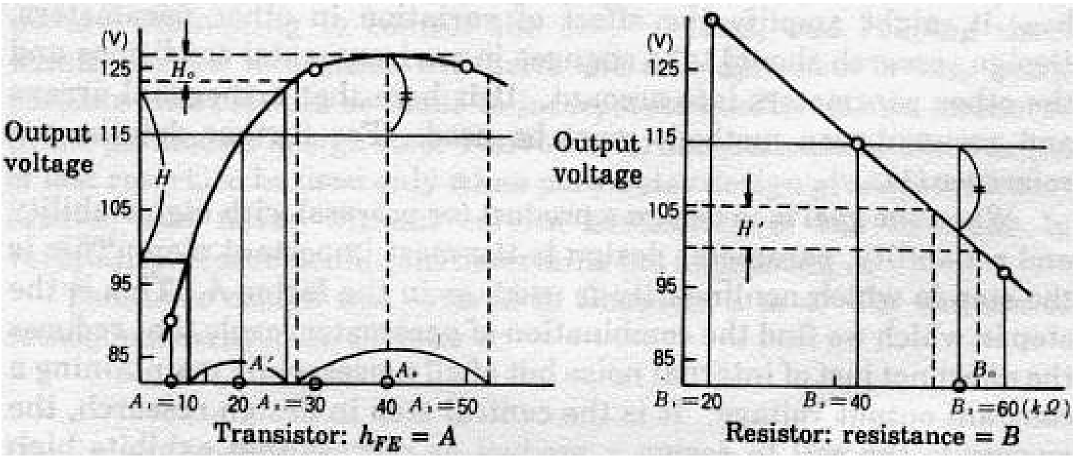

Figure 3. Parameter selection for amplifier design

Assume that the engineer builds a prototype circuit with a transistor that has $h_{FE} = 10$, but only gets an output voltage of 90V (Figure 3a). The designer can now select a smaller resistance, or different transistor to obtain the desired response. Since the output is more sensitive to the h_{FE}, the designer may try different transistors, and eventually find that at $h_{FE} = 20$, he obtains the correct response, 115V DC.

Question: why does the designer first try to iterate with the parameter to which the output is more sensitive?

Question: Is the new design a good design?

Answer: No, according to Taguchi. The designer's goal should be to reduce output variability *in the presence of noise*. Typically, for a cheap transistor, over its designed life of 10 years, the h_{FE} will vary up to ±30%. At A= 20, this corresponds to output variation between (98, 121)V over the life of the amplifier.

Instead, if the designer used a transistor with h_{FE} = 40, then, even with a ±30% variation in h_{FE} between (28, 52), the output voltage of the amp will range between (122, 127). Of course, this is nominally higher than the required output, but the nominal output voltage can be adjusted down by selecting a larger resistance, i.e. by varying parameter B.

Note that since the resistor has a linear response, the system sensitivity does not change with a change in the nominal value of parameter B.

Design using Orthogonal arrays

In the amplifier example above, we were able to select the nominal parameter values where the design is least sensitive to noise. However, consider the case where the system response to both, A and B is non-linear. Then, selecting parameter A at the least sensitive point may force us to fix parameter B at a more sensitive region, and vice versa. In such cases, how do we adjust the two parameters to get the right design? In general, what if there are multiple parameters that are correlated (think of the caramel candy example with 10 ingredients)?

Robust design goal: to come up with a combination of design parameter values such that (a) the functional objectives are met, and (b) the response of the design is 'least sensitive' to any possible combination of noise factors.

The brute force method:

1. For each possible combination of design parameters (one combination ⇔ one design)
   try all possible levels of noise
   record the maximum deviation from ideal output response
2. Select the combination that has the least deviation

There are two issues here: firstly, there may be more than one output responses (i.e. the design may need to satisfy multiple functional requirements). In this case, how do we define "least deviation" in step 2?

One method could be to select the worst-performance output response. [Why?]

The bigger issue is that it may be too expensive (in cost and time) to perform all possible experiments. Consider a design with m modules, and each module can be designed in one of p ways. The total design possibilities = p^m. Further, how many levels of noise must be tested? Let us assume that the noise response is monotonic, and therefore we only need to test for extreme variations in the noise levels, e.g. nominal $\pm 3\sigma$. If there are *n* noise variables, then again we need to perform 2^n experiments, *for each design*!

How to reduce the number of experiments?

Taguchi introduced a systematic methodology to do so. The key idea is the use of orthogonal matrices. Before introducing the idea, let's look at an example.

Example

Suppose that we want to design a web-page, and are considering a good combination of background- and text-colors. We know that some combinations are inherently 'better' in terms of the readability* of the page. There are approximately 32 different colors we can select from. To test each combination will require 32*32 = 1024 experiments, each involving perhaps several subjects.

Possible approaches:

(a) First go through all the background colors and select one, then try all possible text-colors and see which combination for that choice of background is the best.

(b) Select the text color first, and then try several background colors.

(c) Set up a criterion for 'readability'; select a few of your favorite colors (e.g. pink and yellow) for background. Try a few text-colors against, say, pink-background; the first text-color choice that meets the requirement is the selected combination.

(d) …

* Readability: We may set up a quantitative test for the readability, by, e.g. smallest distinguishable font-size for a given reader from a given distance, or farthest distance from which a given font can be read without errors.

What are the problems with the above approach(es)?

(1) Unexamined choices: there are several choices of background/text colors that may not get a reasonable number of trials, but could have yielded a good design. Although we cannot examine all combinations, it may not be reasonable to totally ignore some background color choices, while trying too many combinations with others.

(2) Noise: Almost all terminals are set up at different levels of contrast, brightness, etc. by their users. Often, a design that is 'good' on one terminal may not be as good on a different one. Such variability in the displays is termed by Taguchi as environmental noise.

Later we shall see how to test for effects of noise on the design parameter choices, but for now, let us concentrate only on the first issue.

What combinations of Design Parameters to Test: The Inner Array

For those of you who will take classes on Design of experiments, you will learn other systematic methods of generating the "statistically best" subset of experiments in order to arrive at near-optimal designs. Technically, Taguchi's method is somewhat different than these techniques, but (i) several underlying principles remain the same, and (ii) for most practical purposes, it works remarkably well – there are very large number of practical, industrial examples of successful designs generated using Taguchi's method.

The principles in selecting the proper subsets are based on two simple ideas: *balance* and *orthogonality*.

Balance:
Assume that a variable (i.e. a design parameter under investigations) can take n different values, $v_1 \ldots v_n$. Assume that a total of m experiments are conducted. Then a set of experiments is *balanced with respect to the variable* if: (i) $m = kn$, for some integer k; (ii) each of the values, v_i, is tested in exactly k experiments.
An experiment is *balanced* if it is balanced with respect to each variable under investigation.

Orthogonality:
The idea of balance ensures giving equal chance to each level of each variable. Similarly, we want to give equal attention to *combinations of two variables*. Assume that we have two variables, A (values: $a_1, \ldots, a_n$) and B (values $b_1, \ldots, b_m$). Then the set of experiments is orthogonal if each pair-wise combination of values, (a_i, b_j) occurs in the same number of trials.

Let us get some intuition about these ideas using some examples. Consider a design with three variables, each of which can be set at two different values. For convenience, we denote these values as levels, 1 and 2. A complete investigation requires 23 = 8 experiments, as shown in Table 1a. On the other hand, if each experiment is expensive, we can get much useful data using four experiments as indicated in Table 1b.

Table 1a.			
Run	**A**	**B**	**C**
1	1	1	1
2	1	1	2
3	1	2	1
4	1	2	2
5	2	1	1
6	2	1	2
7	2	2	1
8	2	2	2

Table 1b.			
Run	**A**	**B**	**C**
1	1	1	1
2	1	2	2
3	2	1	2
4	2	2	1

Exercise: Show that the array in table 1b is orthogonal.

Exercise: Comment qualitatively on what type of effects cannot be studied in the second case.
[Hint: see if the value of the third variable changes when the other two are kept constant.]

It turns out that in many design applications, the function of the product depends upon several different design parameters, and for each parameter, we can usually focus on two or three different "good" values, or levels. Thus we are mostly dealing with situations where we have n-variables, and need to examine 2, or 3 levels for each variable. This can lead to a large number of combinations, so we try to experiment with some of the combinations, not all (technically, we conduct partial factorial experiments instead of a full-factorial experiment).

To do this in a systematic method, we shall need to determine the following:
(a) How many experimental design configurations must be tested?
(b) What is the sequence in which we test the different design configurations?

If we are not careful in setting up these trial runs, then we may end up with a poor design. The study of the subject of Design of experiments focuses on the best techniques to conduct such experiments. The goal is to get the maximum information out of a limited number of experiments through the use of statistical analysis. While we do not have the time to go into details of this topic in this course, let us look at some examples to understand the main issues that are involved.

Example:
A landscape designer is studying the growth rate of a type of plant. Two types of seeds are available, which exhibit different growth rate. An important factor is the amount of water provided each day. Suppose the experiment is conducted on four different land areas to eliminate the effect of soil differences. Which set of experiments is better ?

Experiment 1		
Plot	**Seed Type**	**Water**
1	A	2
2	A	2
3	B	1
4	B	1

Experiment 2		
Plot	**Seed Type**	**Water**
1	A	2
2	A	1
3	B	2
4	B	1

After getting the data on growth rates from the different plots, in the case of Experiment 1, we shall be unable to distinguish the differences in growth rate due to seed type and amount of watering (e.g. perhaps growth rate for type B would be higher if it got more water). We say that in this case, the factor "Seed Type" is ***confounded*** with the factor "Water".

Example continued:

Assume that we still conduct four runs, but also want to study the effect of amount of fertilizer used, as shown in the following table:

Experiment 3			
Plot	**Seed Type**	**Fertilizer**	**Water**
1	B	2	1
2	A	1	1
3	B	1	2
4	A	2	2

Let us study which factors get confounded in this case. Clearly this experiment was designed with some care – if we look any pair of factors, there appears to be no confounding. However:
(Seed X Fertilizer) is confounded with Water,
(Seed X Water) is confounded with Fertilizer, and
(Water X Fertilizer) is confounded with Seed.

Obviously, with only four experiments, it is not easy to eliminate all such interaction effects. However, one must carefully study the problem before settling on the experiments. It is possible that some factors are already known to be totally independent. In this case, we really will not care if these factors are confounded in our experimental runs. This idea forms an important aspect of Taguchi methods.

The main idea: if we have identified some factors that affect the performance of the final product in combination we need to run extra experiments where the different combinations of their values are constant, while other factors are changing.
For example, if we add two new runs on experiment 3 (rows 5 and 6 below), then we can get more information about the growth rate of different seeds when both fertilizer and water levels are high, or low.

Experiment 3			
Plot	**Seed Type**	**Fertilizer**	**Water**
1	B	2	1
2	A	1	1
3	B	1	2
4	A	2	2
5	B	1	1
6	B	2	2

The addition of new experiments is designed to add more runs with combinations of some pair of factors being held at a constant level while other factor(s) change.

Instead of designing the experiments from the basics, we shall make use of some standard orthogonal arrays. Several examples of such arrays are presented in the Appendix B of Otto and Wood's textbook.

The figure below shows an $L_8(2^7)$ array. We shall use the notation $L_a(x^b y^c \ldots)$ to denote a series of *a* experimental runs, studying *b* factors that each have *x* levels, *c* factors that each have *y* levels, etc. Thus our array below can be used to study up to seven factors, each at two settings, by conducting a set of eight experiments.

	factor							
		A	**B**	**C**	**D**	**E**	**F**	**G**
Experiment No	**1**	1	1	1	1	1	1	1
	2	1	1	1	2	2	2	2
	3	1	2	2	1	1	2	2
	4	1	2	2	2	2	1	1
	5	2	1	2	1	2	1	2
	6	2	1	2	2	1	2	1
	7	2	2	1	1	2	2	1
	8	2	2	1	2	1	1	2

Consider a design where we have seven independent factors. We can then study the effect of varying each factor by looking at the experimental results corresponding to the correct columns. e.g. The mean effect of changing the factor D from setting 1 to setting 2 is identified by the difference in the mean from experiments (1, 3, 5, 7) and the mean from experiments (2, 4, 6, 8). However, no interaction effects can be identified in this case.

What if we have only 5 factors? We will still use this array, and just ignore the settings of the last two columns, i.e. you can think of F and G as dummy factors whose two states are invariant.

Assume, however, that we know that factors A and B have some interaction, i.e. different combinations of levels of A and B cause the product to behave differently. In this case, we cannot directly estimate the differences in, say, C, by looking at combinations of (A,B) values – e.g. when (A, B) == (1, 1), then the level of C is also not varying. To estimate the interaction effect, we will assign one column for this interaction (say column C). Thus, we can study a design with less number of factors, but can get some more information about the nature of interaction between A and B. Likewise, if we know that each pair of factors has some interaction, then we will sacrifice one column for each, meaning we can study a maximum of four factors using this array, assigned as follows:
A, B, AXB, C, AxC, BxC, D).

We say that a factor that is constant has 0 degrees of freedom (**dof**); if it can take two different values, then it has 1 dof, and if it can take *n*-different values, it has (*n-1*) dof. Let's say we want to isolate some information about two interacting (i.e. not independent) factors, A and B, each with three levels. In this case, the number of degrees of freedom of A, B = 2 each; the number of degrees of freedom of the interaction is 2x2 = 4 (why?). Each column of an $L_a(3^b)$ array gives us 2 dof's. Thus we need one column for A, one column for B, and two more columns for AxB. In other words, if these are the only two factors, we can use an $L_9(3^4)$ array, with columns [A, B, AxB, AxB], as shown below:

	A	B	AxB	AxB
x1	-1	-1	-1	-1
x2	-1	0	0	0
x3	-1	1	1	1
x4	0	-1	0	1
x5	0	0	1	-1
x6	0	1	-1	0
x7	1	-1	1	0
x8	1	0	-1	1
x9	1	1	0	-1

Reducing the Sensitive to Noise: the Outer Array

The above discussions mostly focused on how to generate different design alternatives, and the sequence in which to test them. We now concentrate on the word “test”. We may be looking for how close the actual behavior of the design is to the nominal value. However, the key idea of Taguchi methods is to select designs that are insensitive to noise. In other words, for each design, we must now perform tests at different noise levels.

What is noise? We shall define noise as any environmental/temporal factor (e.g. temperature change, humidity change, aging etc) that can affect the performance of the product. These are factors that are not in the control of the designer.

Once we have determined the different types of noise factors, we must decide how many levels of noise we must test the product at. If the response varies linearly (or even monotonically) with the noise, it may be sufficient to test at some extreme values of the noise levels (e.g. $\mu \pm 3\sigma$). However, when the response is not so well behaved, we may need to test at more levels of noise.

Consider now that there are *n* types of noise, and each must be tested at *k* levels. Then we must perform k^n tests for each design alternative. This may be impractical, and therefore we again will perform partial factorial experiments. The design of the noise array is performed using the same ideas as the design of the design parameter array. The array depicting the combination of noise levels to be tested for each design is often referred to as the outer array. A very common example of an outer array is the $L_4(2^3)$ array, which indicates that we perform 4 sets of experiments for each design configuration, and collect information for combinations of three types of noise, each of which has two levels.

It is customary to write the inner and outer arrays in one chart-like figure. This chart allows us to record all experimental results in a tabular format. To do so, the outer array is written in a transposed form. The Figure below shows an L8(27) inner array coupled with a L4(23) outer array. The design parameters are A,…G, while the three noise factors are P, Q and R.

								Outer Array				
								R	1	2	2	1
Inner Array								**Q**	1	2	1	2
Design parameters (Control factors)								**P**	1	1	2	2
	A	**B**	**C**	**D**	**E**	**F**	**G**					
1	1	1	1	1	1	1	1					
2	1	1	1	2	2	2	2					
3	1	2	2	1	1	2	2					
4	1	2	2	2	2	1	1					
5	2	1	2	1	2	1	2					
6	2	1	2	2	1	2	1					
7	2	2	1	1	2	2	1					
8	2	2	1	2	1	1	2					

Discussion

There are several important issues about Taguchi methods that I have not discussed in these notes, but which are essential in completing this study. It is difficult to go into these topics without doing a full course on the design of experiments. For the interested students, I will mention a few items here.
In settling which design parameter (factor) to assign to which column in the orthogonal array, one needs to take some amount of care. The underlying issues here are related to statistical concepts of blocking and randomizing. There are some procedures that can assist in making such assignments, for instance the method of linear graphs. Another related issue is that of resolution.

In the next set of notes, we shall look at some of the very basic statistical background required to interpret the results of a series of experiments conducted using Taguchi's method.

Reference: The primary sources of these notes were
Introduction to Quality Engineering: Designing Quality into Products and Processes, Genichi Taguchi, 1986, Asian Productivity Organization
Product Design, Kevin Otto and Kristin Wood, Prentice Hall, New Jersey, 2001
Robust Engineering, Genichi Taguchi, Subir Chowdhury, and Shin Taguchi, 2000, McGraw Hill, New York
Taguchi Methods, Glen Stuart Peace, Addison Wesley, 1993