BURGE HUGHES WALSH

Training and Consultancy in Systems Design and Process Improvement

Systems Engineering

ROBUST DESIGN

Stuart Burge
Burge Hughes Walsh Partnership

Contents

Table of Figures

Introduction to Robust Design - What is it and what does it do?

Robust Design is more than a tool; it is complete methodology that can be used in the design of systems (products or processes) to ensure that they perform consistently in the hands of the customer. +-It comprises a process and tool kit that allows the designer to assess the impact of variation that the system is likely to experience in use, and if necessary redesign the system if it is found to be sensitive.

The creator of Robust Design is Dr Genichi Taguchi who in the 1950's worked for the Electrical Communications Laboratory (ECL) of the Nippon Telegraph and Telephone Corporation. During his twelve years there, he developed his ideas and understanding of Robust Design. One critical insight was the recognition that small changes (variation) in a design parameter can significantly affect the performance. Taguchi also realised that the traditional approach to handling this variation through the use tolerances and specification limits can be misleading. Taguchi was able to put his understanding on a sound theoretical basis through the concept of the *loss function*. But his stroke of genius was the understanding that the inherent non-linear nature of most systems can be exploited to find design solutions that are "robust" against the variation they are likely to experience in use. Rather than tightening tolerances when faced with a sensitive design the system should be redesigned to find an alternative solution that was insensitive. He distilled this into an approach he called *Parameter Design*.

Robust Design is now an established methodology for most serious engineering companies and has made a major contribution to product quality. It is one of the reasons your motorcar starts first time every day and your mini-system has such good sound quality. Taguchi was interested in product design but it is clear now that Robust Design is applicable to any type of system, such as process or service intensive systems. The same principle can be applied to the service quality in a hotel, restaurant or bank.

Taguchi's Loss Function

Taguchi, when working at ECL, noticed that apparently minor variations in a system design parameter can have a significant effect on overall system performance. He also recognised that the "traditional" approach to handling such sensitivities by setting tight tolerances is flawed.

To explain Taguchi's philosophy it is worthwhile considering a simple example. Imagine that a lawn mower designer has selected a rotating blade, like that shown in Figure 1, as the mechanism for cutting grass.

Figure 1: A lawn mower blade

One of the responsibilities of the designer is to determine numerical values for the blade design parameters, such as:

- Blade length
- Blade thickness
- Blade width
- Blade shape
- Blade material
- etc

These numerical values are called the "target" or "nominal" values and represent the ideal values that typically the designer has calculated to give the required performance. Most designers, however, know that these target values will not be achieved in practice because of one or more causes of variation that include:

- Environmental; e.g. temperature, pressure, humidity etc
- Manufacturing: no two components can be made the same
- Wear: over time wear causes the values to change

To "cope" with this expected variation designers will set limits (called specification limits or tolerances) as shown in Figure 2.

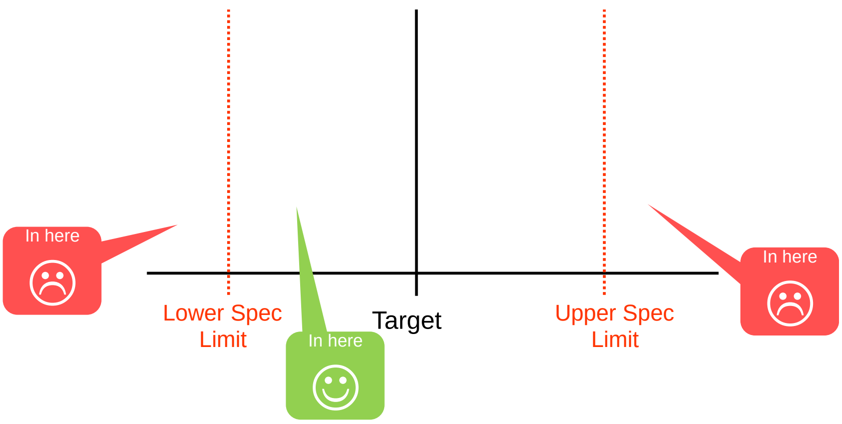

Figure 2: The use of specification limits (tolerances) to manage variation

The purpose of specification limits, or tolerances, is to identify when an actual blade is so far from its intended design target value, that the product performance is unacceptable and potentially leading to customer complaints. Taguchi noted that these specification limits were often chosen from experience and judgment or simply copied from the previous design – there was little science in establishing the "best" limits!

From a quality viewpoint, items outside the specification limits are to be rejected. On identifying a reject, there are three courses of action:

1. Scrap the reject and make a replacement. The money for this replacement will come from the company's profit.
2. Rework the reject to bring it inside the limits. The money for this rework will come from the company's profit.
3. Agree a concession, a relaxation of the specification limits on a case-by-case basis. The money for this procedural work will come from the company's profit.

The important point to note is that if an item is outside the specification limits it is going to cost and the money will come from this year's profit. However, an item inside the limits will not require the expenditure of money.

This cost of variation is called the Quality Loss and based upon the scenario described above results in the economic model shown in Figure 3. In Figure 3, when the variation in the design parameter reaches the specification limit, there is a step change in cost. It may not be equal either side of the target, but inside the limits the loss is zero!

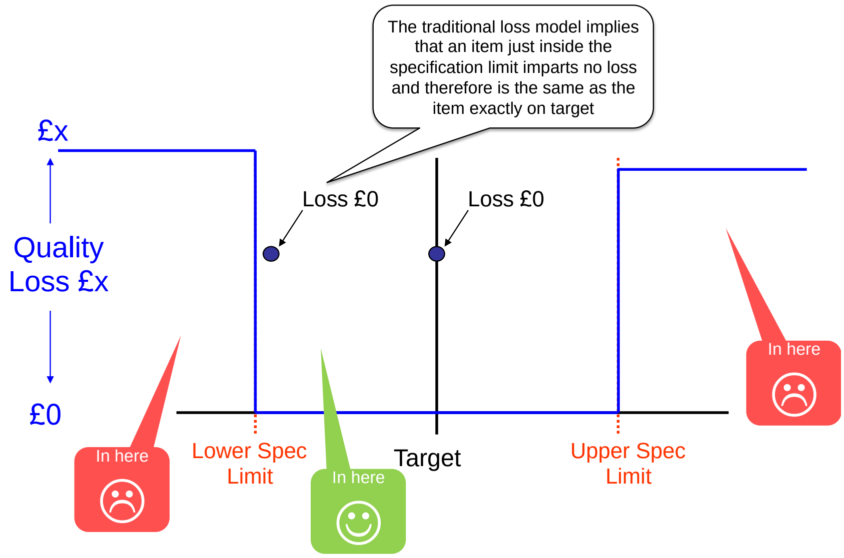

Figure 3: The traditional Quality Loss model

Taguchi was unhappy with this quality loss model, because it inferred that an item exactly on the designer's target would have a zero loss, as would one just inside the specification limits. Remember that these specification limits are often determined by "engineering judgment" or what was on the previous design. Taguchi argued that an item near the specification limit could not be the same as one exactly on target! The cost model was wrong and a better model was needed.

In deriving a new cost model, Taguchi's key insight was that a loss would always be incurred when a design parameter (denoted by *y*) deviates from its target value (denoted by *m*) **regardless** of how small the deviation is. Taguchi argued the quality loss is zero only when $y = m$ and defined a loss function as $L(y)$ where:

$$L(y) = L(m + \{y - m\}).$$

Which could be expanded by Taylor's series:

$$f(x + h) = f(x) + hf'(x) + (h^2/2!)f''(x) + (h^3/3!)f'''(x) + \ldots$$

by making $x = m$ and $h = \{y - m\}$. Thus:

$$L(y) = L(m) + L'(m)\{y - m\} + L''(m)\{y - m\}^2/2! + \ldots.$$

Returning to Taguchi's premise, when $y = m$, the loss is zero – i.e. $L(y) = 0$, and since $\{y-m\}$ will be small, terms with power higher than two can be ignored resulting in:

$$L(y) = L''(m)\{y - m\}^2/2! = k(y - m)^2$$

The term $(y - m)$ represents the deviation from the target value and so the loss *A* due to a deviation $\Delta = (y - m)$ is:

$$A = k\Delta^2$$

In other words, the loss is quadratic as shown in Figure 4.

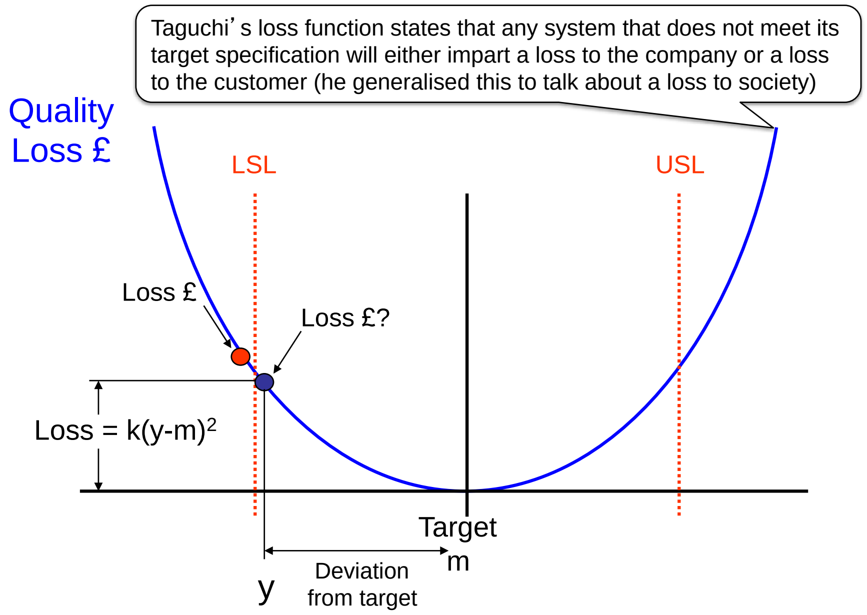

Figure 4: Taguchi's loss function

The profound insight that comes from Figure 4 is that even when we are inside the “specification limits” there will be a quality loss measured in pounds, yen, or dollars. Specification limits have no economic value. Dependent on which side of the target you are, sometimes the company loses, and sometimes the customer will lose. Taguchi summed this up by stating that any product that does not meet its target specification will impart a loss to society. He also said “a company that ships a product with a high quality loss is worse than a thief!”

Potentially, every design parameter in a system will have its own characteristic loss function, but the ‘degree of loss’ is dependent upon the constant *k*. Thus, for different *k* values there will be a different “shape” to the loss function (still quadratic) that reflects the sensitivity of the chosen design point. Some will be sensitive some will not – some will be robust to variation as shown in Figure 5.

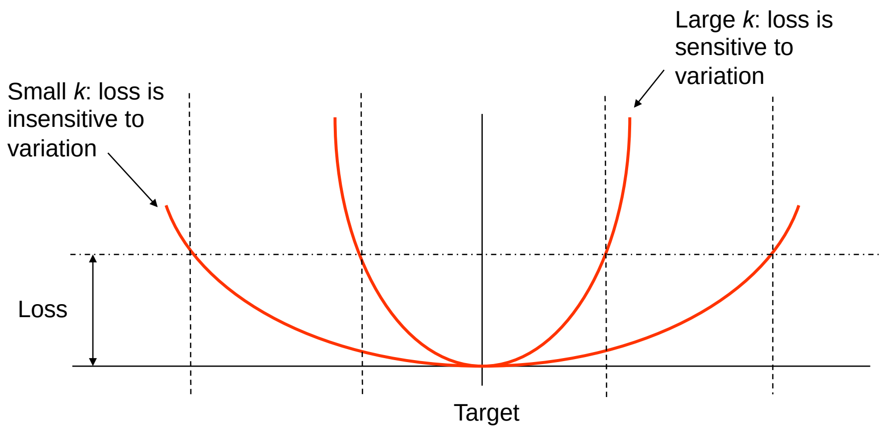

Figure 5: Sensitive and insensitive loss functions

Figure 5 shows two extreme cases. The first is a sensitive, or un-robust design, where variations in the particular design parameter around the target value will cause significant quality losses. Faced with the knowledge of such a situation, the typical reaction of the designer is to tighten the specification limits. There is no doubt this will work, the system will work as expected in the hands of the user, but the tightening of the tolerances adds cost that either must be borne by the user or the producer. Someone is going to lose out.

The second extreme is the flatter loss function. Here the performance will only degrade when the variation is large. If variation occurs over the expected range, the user will experience very little degradation in performance. This is a robust design, where performance is maintained at the desired level over the expected range of operating conditions.

In other words having a loss function that is insensitive to variation is desirable because the end user, the customer, will experience consistent performance. It was this epiphany that led Taguchi to his Robust Design process. Clearly, it is the responsibility of the designer to determine the loss function, and if it is sensitive not to react by setting tighter specification limits but to redesign to a new design point where it is insensitive to variation.

To reinforce this critical message consider the system shown in Figure 6. This particular system has only two design parameters x_1 and x_2. When x_1 and x_2 are changed, they will put the design on another point on the “mountain”. The customer requires a “certain“ performance, which is represented by the plane that cuts through the “mountain”. Anywhere above the “plane” is acceptable, anywhere below is not!

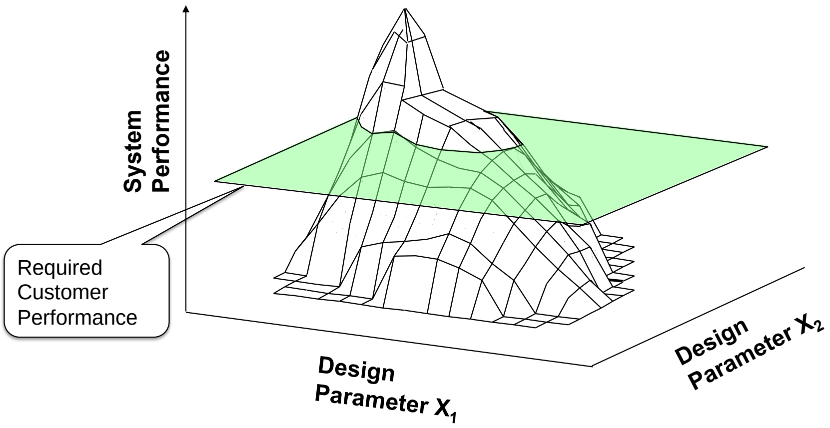

Figure 6: A system design with two design parameters

Faced with such a situation and the pressure to deliver “excellent” performance, many a designer would aim for the peak of the “mountain”. This is simply because it is “world-class” performance, it cannot be bettered! Indeed, having found the peak, the designer could work “backwards” to find appropriate values for x_1 and x_2 as shown in Figure 7.

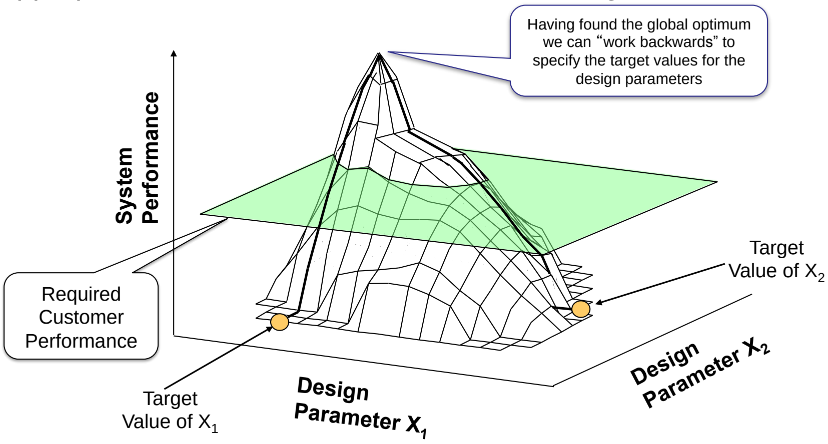

Figure 7: Designing the 'best' possible system

Theoretically, the situation shown in Figure 7 is the best possible design since it gives the best possible performance. BUT it is entirely reliant on being able to:

- achieve the target values for x_1 and x_2
- maintain the target values for x_1 and x_2

Neither of these is possible in the real world because of variation caused by environmental factors, manufacturing variation and wear. In other words the target values will vary considerably and we “will fall off the mountain peak” as shown in Figure 8. Taguchi called this “real world” variation NOISE.

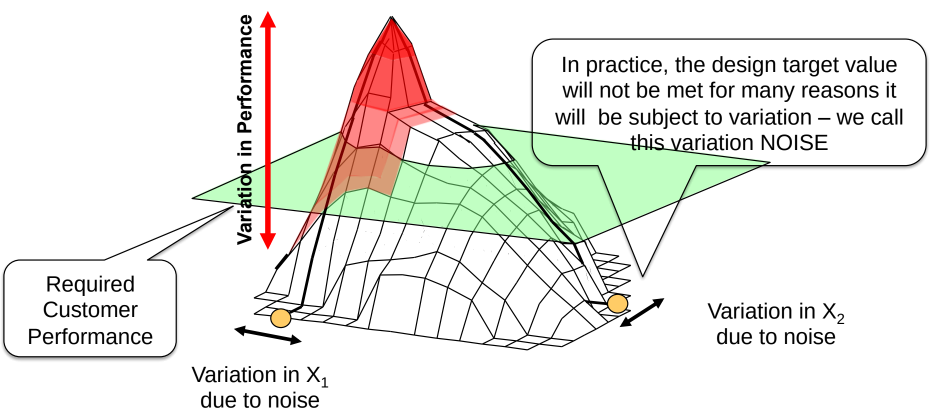

Figure 8: The impact of noise on system performance

Because of the noise, the variation in system performance experienced by the customer is huge. One day the performance will be world class, the next awful. Customers hate inconsistent performance – they like consistent performance. This is a sensitive situation and the loss function at this peak is the steep one in Figure 5.

Returning to the earlier argument around Figure 5, having found yourself at the peak and discovering the extreme sensitivity to noise one solution is to set "tighter" tolerances around the design parameters. There is no doubt this will work as shown in Figure 9, but at a greater cost!

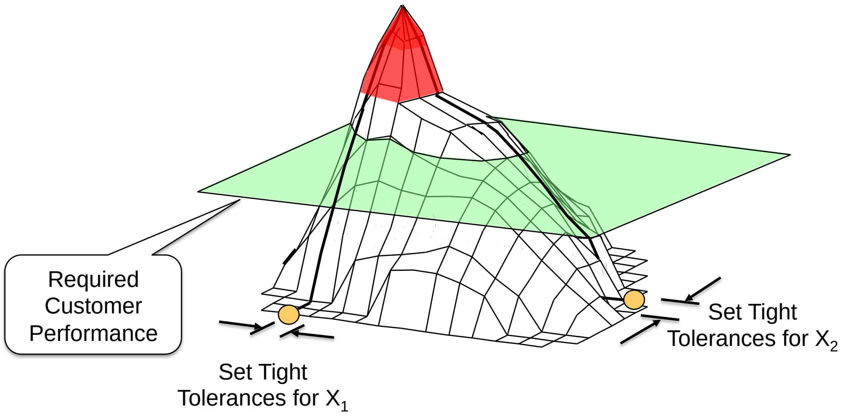

Figure 9: Tightening tolerances - an expensive solution

By now, however, you will have realised there is another solution which is to move the whole design to new target values that put it on the plateau. By placing the design on the plateau we find:

- We achieve the customers performance requirements
- We are insensitive to the likely variation (the noise)

We will of course not achieve the "best" possible performance, but it does exceed their requirements consistently. This plateau, shown in Figure 10, is a "robust optimum".

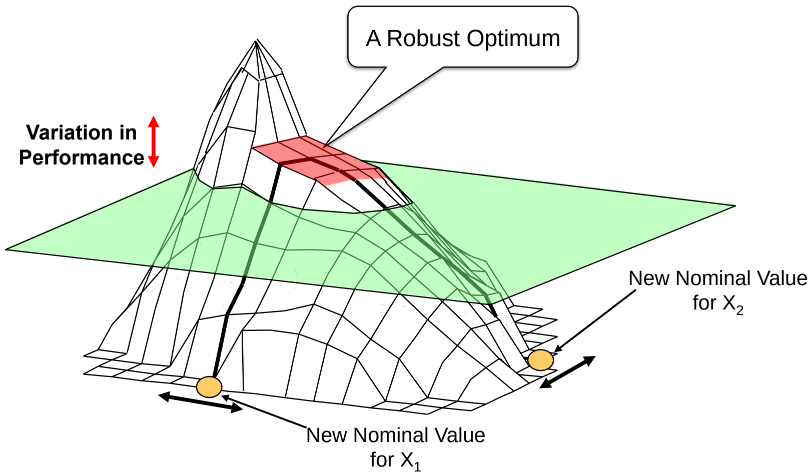

Figure 10: The place to be, the Robust Optimum

The beauty of a robust optimum is that, although we do not achieve the best possible performance, we do consistently exceed our customer's expectations consistently. Customers like consistent performance. When on the plateau, the loss function is the flatter one in Figure 5.

Taguchi's Robust Design Approach

Taguchi codified the ideas expressed above into the three-stage process:

1. System design
2. Parameter design
3. Tolerance design

While this process is broadly correct because it is at such a high level, it is possible to add more detail as to how to perform Robust Design. This more detailed process is shown in Figure 11.

Engineer System is about creating a system design concept that satisfies the customer's requirements. Taguchi talked little about how to do this, but it is critical step. Optimising a "bad" system concept design is nugatory work. To ensure the best possible solution Systems Engineering [1] must be used, as this will demand the full understanding of the problem defined by the customer's requirements, the exploration of candidate solutions ultimately leading to the selection of the "best" system concept solution.

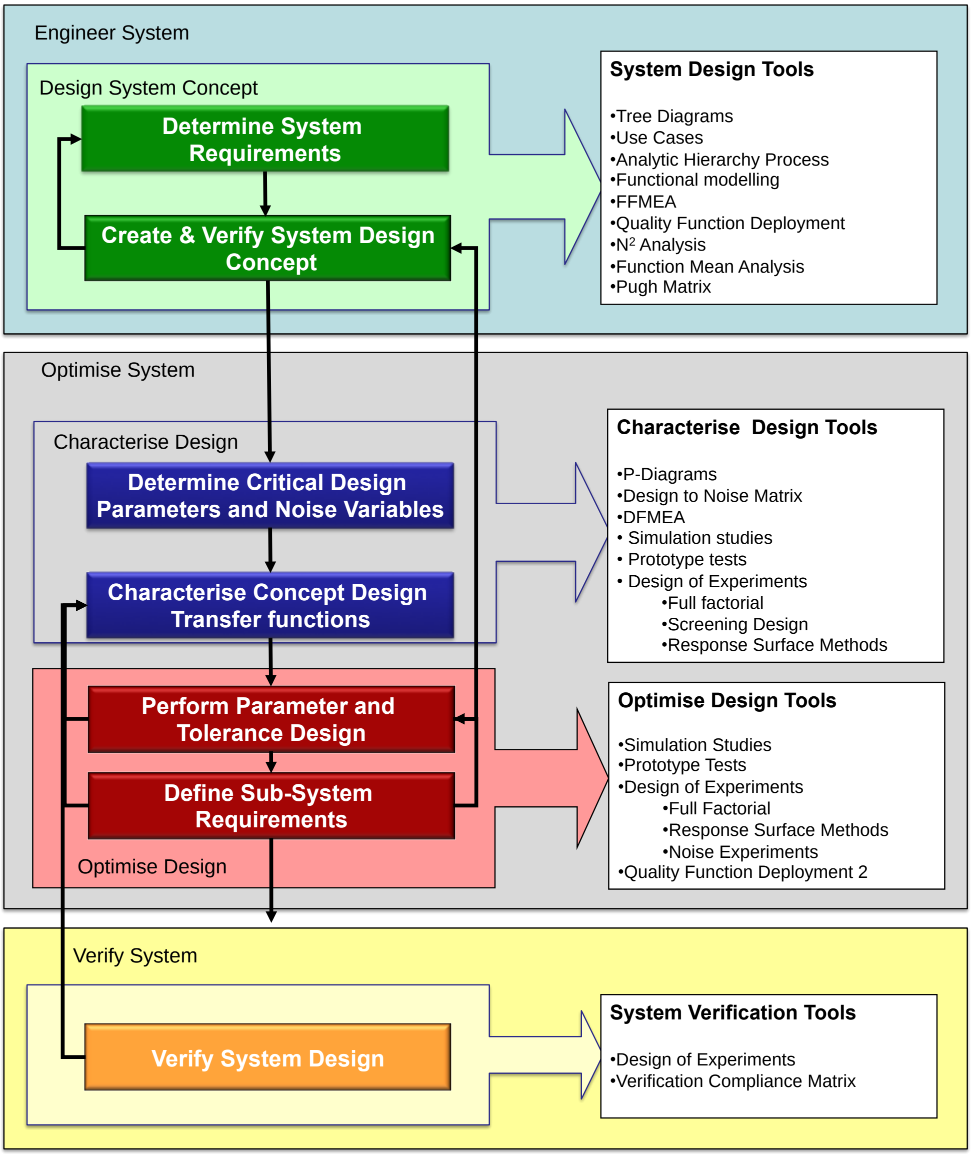

Figure 11: The Whole Robust Design Process

Optimise System comprises two activities. The first is to *Characterise Design*. This is, in simple terms, finding out what the mountain (range) looks like. I want to know, either explicitly or implicitly, the mathematical relationship between my design parameters, noise factors, user commands and the critical system performance measures that are important to the customer. Taguchi reduced any Robust Design problem down to the diagram shown in Figure 12.

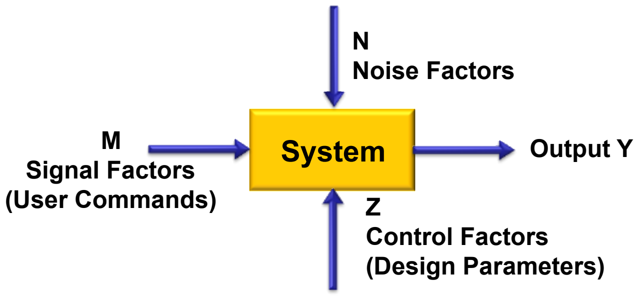

Figure 12: The universal Robust Design problem

Taguchi argued that there are three types of factors which determine the output of a system:

Signal Factors M (or user commands) are parameters set by the user to set or command the intended value for the output of the system. So for example, on a lawn mower the user may have a throttle lever to change the speed of rotation of the cutter blade.

Control Factors Z (or design parameters) are parameters that can be freely specified by the designer. In fact, it is the designer's responsibility to determine the best values for these parameters. For example, the lawn mower design may decide the blade length is 520mm.

Noise Factors N are factors or parameters that cannot be, or chosen not to be, controlled by the designer. They cause the Output Y to deviate from the target specified by the Signal Factor M.

Taguchi said that the job a designer is to determine the target values for the design parameters (control factors) such that the system output achieves the user commanded level irrespective of the noise factors. Pictorially, as shown in Figure 13, we want the design on a plateau and not on a peak. It is therefore necessary to search the mountain range to find the plateaus and avoid the peaks. This searching of the design space – the mountain range – is what Taguchi called Parameter Design.

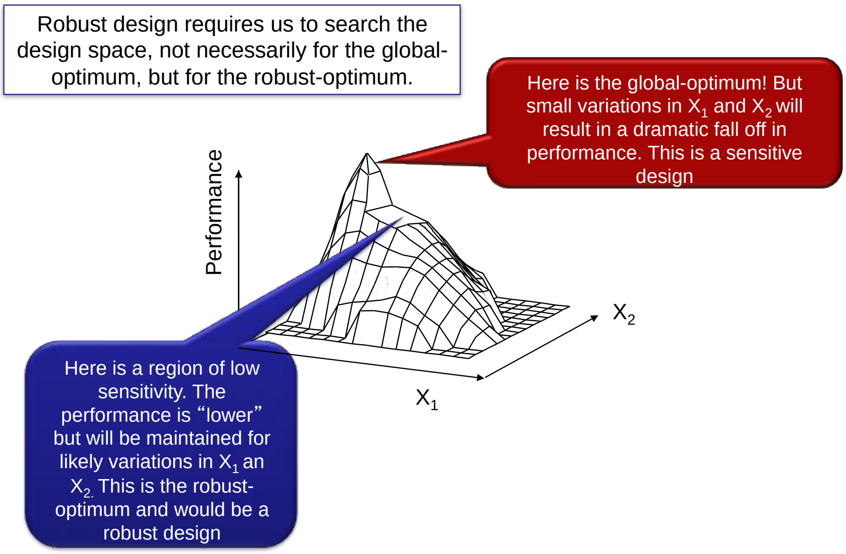

Figure 13: The Robust Design task

Figure 13 shows a very simple system with only two design parameters where it is possible to visualise the relationship between the design parameters and the system output. Real systems have much more than two design parameters, they cannot be visualised in the way Figure 13 shows. Effectively the robust design task is akin to a blind person searching a mountain range to find the plateaus and avoid the peaks.

Given a concept design from the Systems Engineering, the first step in Robust Design is to determine all the design parameters, noise factors and signal factors that relate a particular system output that in turn relates to customer satisfaction. Here, the tool of choice is a form of Figure 12 called a *Parameter Diagram* or *P-Diagram.* Figure 14 shows an example P-Diagram for the grass cutting system of a Lawn Mower.

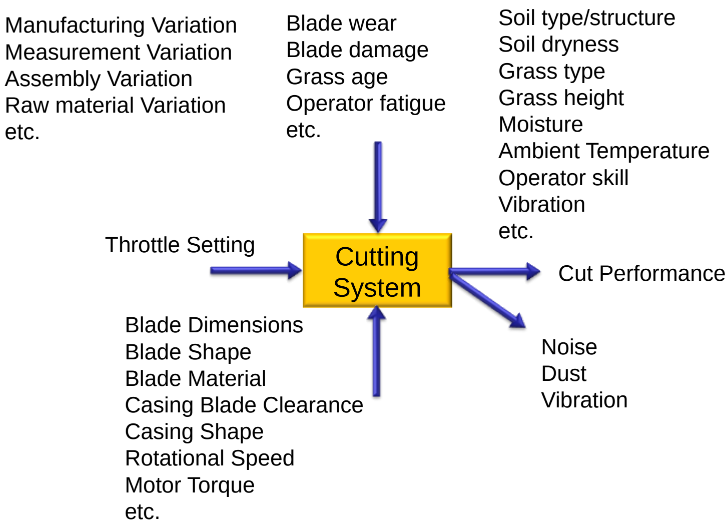

Figure 14: A Partial P-Diagram for the grass cutting system of a lawn mower

A P-Diagram can help a designer (or design team) identify all the possible factors that could have a role to play in delivering good cut performance. Figure 14 is partially complete but shows the overall nature of the “design problem”. It is not uncommon to have 50 – 100 Noise Factors and 20 – 50 Design Parameters. To be able to construct the “mountain range” means that we need determine the *transfer function* between the input factors and the system output. In other words, we need to know mathematically the relationship:

$Y = f(Z, N, M)$

We do not have to know this mathematical expression explicitly (as an actual formula), we can undertake Robust Design if we know the relationships implicitly usually through a simulation model of the system. For engineered systems we build Finite Element Models to determine stress level to ensure the design is strong enough. For process-based system we can build Process Simulation Models to investigate bottlenecks and stock levels.

It is clear, however, that creating a transfer function with several hundred terms is not easy, and in fact we do not do this. Having identified all possible factors, we now determine which of these are critical. A useful tool here is the Design to Noise Matrix, which is a simple tool to allow a designer or design team to assess which factors (design and noise) are most likely to influence the robustness of a nominal design. It is used to provide a quantitative assessment of the influence that both design parameters and noise factors have on the outputs of a system to determine which are critical. Typically, we aim to reduce the number of design and noise factors to between 10 and 20. Any more and the problem becomes intractable.

Even with a reduced number of design and noise factors, to fully map out the mountain range (while not impossible) may take too long. What Taguchi wanted is an efficient way of searching the mountain range – the design space. Fortunately, while he worked at Electrical Communications Laboratory he was a visiting professor at the Indian Statistical Institute, where he worked with Ronald Fisher and C. R. Rao. Fisher and Rao were responsible for the development of the Design of Experiments and in particular the orthogonal array.

Design of Experiments (DoE) is the name given to a collection of experimental approaches that explore the effect changing multiple system parameters have on the output of a system. It power stems from its ability to quantify not only the effects of the individual parameters (the main effects) but also the interactions between parameters. Its origin can be traced back to the pioneering work of Sir Ronald Fisher (1890 – 1962) when working at the Royal Agricultural College Rohamsted. Very much the preserve of statisticians for several decades, it was Taguchi who turned a relatively complex methodology into a practical everyday tool. The advent of statistical software packages like Minitab [2] have now made its use even more widespread.

DoE is one of the most powerful of experimental approaches. It is, however, counterintuitive to the uninitiated. In simple terms, a DoE provides a method for designing and analyzing results from an experiment to understand and quantify the effects that multiple factors have on the outputs of a system. Usually when faced with the scenario where the output of a system could be affected by numerous factors, most engineers would resort to changing just one factor at a time. Changing two or more would be seen as foolhardy because "it would not be possible to separate out the contribution of each factor". A DoE, however, is a system of experimental trials where several factors are changed for each trial in a way that will allow for the separation of the various contributions from each factor. It provides a powerful way of searching a design space. What Taguchi did was to take the work of Fisher and Rao to develop a number of design arrays like that shown in Table 1. Table 1 shows an L6 design array where it is possible to explore six control factors (design parameters) in eight trials at two experimental levels (represented by the 1 and 2). In the early days of using Taguchi's approach, organizations would build physical prototypes where the design parameters considered being important could be adjusted according to the design array. If non-linear behaviour were suspected then three or four experimental levels would be used. The experimental trials would be run and the result analysed to find out which design parameters are important.

Trial	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6
1	1	1	1	2	1	1
2	1	1	2	1	2	2
3	1	2	1	2	2	2
4	1	2	2	1	1	1
5	2	1	1	2	1	2
6	2	1	2	1	2	1
7	2	2	1	2	2	1
8	2	2	2	1	1	2

Table 1: A L6 Design Array

The analysis of the experimental results can be plotted in various ways, one of which is a cube plot. Figure 15 shows a cube plot for the results from a L6 design. The length, width and depth for each cube relates to three of the design parameters. Right and left cubes relate to the fourth, front and back to the fifth, and up and down to the sixth. Note that the eight experimental results are recorded on the cube of cubes and provide insight to the design space.

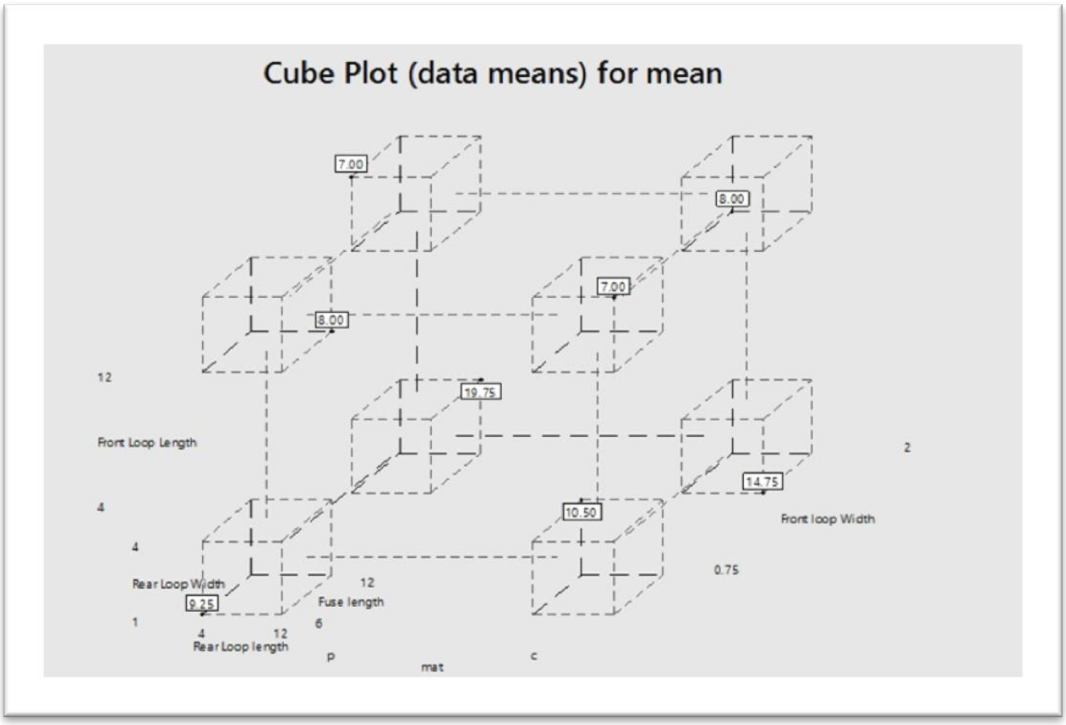

Figure 15: A Cube Plot of results from an L6 Design of Experiments

If we look at the top four cubes in Figure 15, the results are 7.00, 8.00, 7.00 and 8.00. This could be a plateau as the results are quite stable. Whereas the lower level of cubes show considerable variation and present a maximum value of 19.75 on the rear left cube. This could reflect a peak in the design space.

These days it may not be necessary to conduct actual physical experiments. Instead computer based models that simulate the behaviour of a system can be used. Such models combined with Monte Carlo[1] approaches can identify the plateaus associated with a robust design. Here, the Design of Experiments can be used to decide on the simulation runs necessary to efficiently explore the design space.

[1] Monte Carlo is a mathematical approach to the modelling the effects of variation in a system. In particular it looks at the effect of input variation and how this gets transmitted into output variation through the system's transfer function by randomly generating values for the system inputs and calculating their associated output values. This exercise is literally repeated hundreds if not thousands of times to "build up a picture" of the input to output variation. While its origin was in the Manhattan project and performed by hand calculation, modern computers and software make it an invaluable tool for the Robust Designer.

Summary

Robust Design is a complete methodology for designing systems that are insensitive to the variation they are likely to experience in use. The consequence is that such systems will perform consistently in the hands of the user. Robust Design was developed by Dr Genichi Taguchi who adopted a systems view of the problem that led to profound understanding of the impact of variation through Taguchi's loss function. Moreover, Taguchi realised that he could exploit the non-linear behaviour of most systems to find regions of the design space that are insensitive. To find these regions Taguchi developed the use of Design of Experiments as an efficient way of exploring the design space.

References

[1] Blanchard B.S and Fabrycky W.F "Systems Engineering and Analysis" (Prentice Hall International Series in Industrial & Systems Engineering), 2010, ISBN-13: 978-0132217354
[2] Minitab Statistical Software, www.minitab.com, 2014.

The Author

Dr Stuart Burge is one of the founding partners of the consultancy and training company Burge Hughes Walsh (BHW). He is widely known in industry and academia for his expertise in Systems Engineering and particularly system design. Stuart is recognised for his pragmatic approach to Systems Engineering and his ability to explain how to actually do practical Systems Engineering.
Since the formation of BHW in 2000, Stuart has worked with a large number of clients helping them improve their Systems Engineering. This has been through the design and delivery of training courses, coaching and facilitating individuals and project teams and undertaking research into Systems Engineering and System Design.

Stuart is also a Six Sigma Master Black Belt with particular expertise in Design for Six Sigma and Robust Design.