

## USING SELF ORGANIZING MAPS AS A DESIGN EXPLORATION TOOL

Peter Matthews and Ken Wallace

### Abstract

A significant amount of design work is based on modifying current products to meet new specifications. The problem with such work is answering the question: ‘how do the new design specifications constrain the design space?’. This paper proposes a tool that (1) provides a means for visualizing the design domain, and (2) provides an interactive means for constraining the design space and visualizing the results. Visualization is achieved by using a Self Organizing Map. An interactive user interface for editing the design constraints as part of the exploration process has been developed. The method is illustrated using two case studies: the first is a hand-crafted flat screen display design domain and the second is the design of a gas turbine combustor. Preliminary studies indicate encouraging results, and also provide direction for future development of this tool.

*Keywords: Parametric Modelling; Information Analysis; Machine Learning; Product Families; Virtual Design.*

### 1 Introduction

A large portion of design work can be classified as *redesign*. This is where the overall function and concept of the product has been previously determined and developed, but a variant on this design is required. This variant will, at a certain abstraction level, appear to be identical to all other products conforming to this product family. Below this abstraction level, a description can be made that describes the overall differences between the individual members of this family. It is at this level that the design exploration tool is targeted. At this point the design has been structurally well defined, but the lacks the detail required to finalize the product.

A common means for exploring the design space when developing a new product is to take a closely matching previous design and then modify it to meet the new criteria. Expert designers achieve this modification using a tacit model of the design space, this model being a result of several years experience in the domain. This model typically is in the form of coarse trade-offs and a general understanding for the domain. There are two problems for a novice designer here: firstly, the challenge in rapidly identifying a suitable previous design, and secondly, the ability to know how best to modify it to meet the new requirements. This paper describes a design exploration tool that requires little prior expertise of the design domain. By using a database of prior design examples, a model of the design space is constructed using a Self Organizing Map (SOM). By describing the design criteria as a set of constraints, the tool identifies a region of the design space where it is highly likely to find suitable solutions. These constraints are added interactively, and can be modified on the fly. This allows a designer to rapidly explore design

regions by appropriately constraining the design space.

The feasible design space is visualized using the SOM toolkit developed by Helsinki University of Technology [1]. This provides a means for visualizing a high dimensional space using a two-dimensional medium, namely a computer VDU. The remainder of this paper describes the background work, arguing for the selection of the SOM for visualization purposes. An overview of the SOM is then provided, followed by how this was implemented as an interactive design navigation tool. Two case studies are then presented to illustrate various aspects of the navigation tool. Finally, the paper is concluded by reviewing the tool.

## 2 Background

This research aims to provide a means for exploring a design space, without requiring great expertise within the particular design domain. For a computational tool to be of assistance, the design domain must first be provided with a representation that can be analysed computationally. The selection of this representation must be compatible both with the design process and the analysis technique. This section will discuss both these issues.

The ability to parametrically describe a family of designs is a key element to this work. Without this, it would not be possible to numerically analyse a collection of previous designs. In addition, the design parameterisation must be compatible with the design process stage at which this exploration is to take place. Fortunately, design parameterisation is a mature subject, and has been discussed in previous ICED conferences, e.g. [2, 3]. These references were focused at capturing the design form at the conceptual and embodiment stages. It is important for this work is that the key design parameters and criteria have been identified. It is these variables that define the design family, and they are then expressed as a real valued vector.

Once the design space has been expressed as a vector space, a natural question to pose is how to visualize a sample of previous designs. Humans are very able at interpreting visual information, and therefore the aim is to take advantage of this ability to navigate through the design domain. However, the key challenge lies with the high dimensional nature of a typical design space. The problem is how to display a high dimensional space using a two-dimensional medium. It is worth noting that it is important to restrict the display medium to two dimensions for two reasons: the first is that this is the most accessible format. The most basic method for displaying data with a two dimensional medium is by using a collection scatter diagrams, where all pairs of data components are plotted against each other. The second reason for restricting the display to two dimensions is a numerical argument: as more dimensions are used for displaying the original data, the more combinations there are. Specifically, consider an  $N$ -dimensional design space and assume that it is possible to display  $k$  dimensions at once. The total number of combinations to display is given by:

$$\binom{N}{k} = O(N^k) \quad \text{while } k < \frac{N}{2} \quad (1)$$

The above expression takes its maximal value when  $k = \frac{N}{2}$ , and returns to 1 when  $k = N$ . Therefore, it is clear that a visualization approach should attempt to display *all* dimensions at once. Three such methods are Chernoff plots, parallel axes, and Self Organizing Maps. Chernoff plots (or faces) display each single data point using a cartoon human face [4]. The values of each component of the data point are rendered as a separate feature of the face, for

example, the first component is represented by the width of the mouth, the second is given by the size of the left eye, the third is the size of the right eye, etc. While this provides a compact representation of a high dimensional data point, there are serious problems for applying this to design data. The main problem is that it is impossible to determine any relationships between design variables, due to the difficulty of comparing a potentially very large number of faces (there is one face for each design in the sample set). Parallel axes plot the value of each component on a separate axis and a line is drawn through each of these points to link the single data point up [5]. Multiple designs can then be drawn on these axes, resulting in a large number of link-lines. It is now clearer to visualize some relationships, however, this will be dependent on the order the parallel axes are drawn. Self Organizing Maps (SOM) plot a projection of the data onto a series of two-dimensional lattices, one for each design component [6]. These lattice plots are commonly referred to as ‘component maps’. Relationships can be readily identified where sets of similarly looking component maps are found. Such pairs of component maps are analogous to the case where a scatter plot’s points lie principally in a line. The disadvantage of the SOM is that it is an approximate display rather than an exact plot as is the case for the first two. However, this is the least limiting method and is therefore has been adopted for this work.

### 3 Self Organizing Maps

The SOM provides a means for re-representing a given high-dimensional dataset in a topographic manner using a regular lattice of new data points. The purpose of doing this is to simplify the original dataset and obtain a concept of neighboring points by virtue of the connectedness of the lattice. The lattice is chosen such that it is of a lower dimension than the space containing the data points. Typically, the lattice is set to two dimensions, allowing it to be easily displayed. The dimensionality of the lattice largely determines how nodes are connected to each other. One property of the lattice is that there can be no ‘knotting’ of the lattice. This is important as then it is possible to display the lattice with uniform spacing between the lattice points and this will be isomorphic with the lattice as it exists in the higher dimensional data space. It this property that is relied on when displaying the two-dimensional lattice to visualize the design space.

The lattice is positioned in the design space using a training sequence. An initial lattice is generated to span the extent of the dataset. This is then fined tuned using the individual data points. As each data point is ‘presented’ to the lattice, the lattice is shifted to better represent that point. More formally, the procedure is as follows: consider a datapoint  $\mathbf{x} \in \mathbf{X}$  taken from the dataset. Denote the lattice points  $\mathbf{m} \in \mathbf{M}$ . The nearest lattice point to the data point is given by:

$$\mathbf{m}_c = \underset{\mathbf{m} \in \mathbf{M}}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{m}\| \quad (2)$$

where  $\operatorname{argmin}$  returns the lattice point  $\mathbf{m} \in \mathbf{M}$  that is closest to the data point. Next, this lattice point is moved closer to the data point as follows:

$$\mathbf{m}'_c = \mathbf{m}_c + \alpha(\mathbf{x} - \mathbf{m}_c) \quad (3)$$

and  $\mathbf{m}'_c$  then replaces  $\mathbf{m}_c$  in the lattice set  $\mathbf{M}$ . The coefficient  $\alpha$  represents the ‘plasticity’ of the lattice and is initially set at slightly less than 1 and is reduced to 0 as the training process

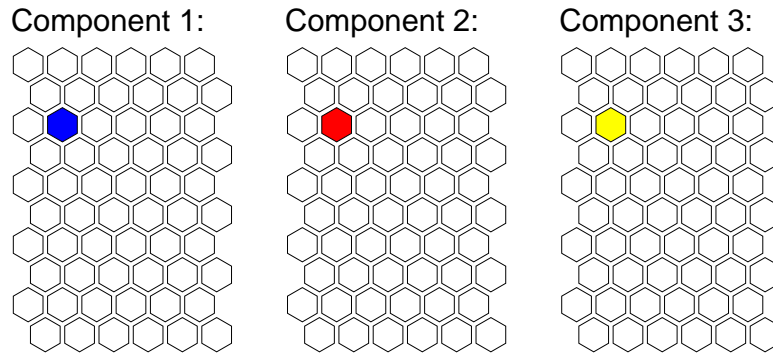


Figure 1. Extracting a design prototype from the SOM component maps: the highlighted lattice point in each of the component maps above contains the value taken on by the component, and all three together form the design 'description'.

continues. In addition, the neighbors of the winning node are also moved towards the datapoint  $x$ , but to a lesser extent. This is done using the above equation, but with a smaller value for  $\alpha$  so that the neighbors are moved by a corresponding lesser extent. The process iterates over all data points several times until  $\alpha$  reaches zero. This typically will occur after iterating over the full dataset about 50 times.

The nature of the training algorithm does require that the data is properly conditioned. This primarily consists of normalizing the dataset so that all components have the same standard deviation. In addition, it is possible to omit the occasional observation in the dataset. This is important, as most industrial datasets are incomplete, missing measurements due to faulty measuring equipment or other such reasons.

There are a number of parameters that define various aspects of the SOM learning algorithm. These are determined largely by the amount of data available and how the resulting network is to be used. As the network is most typically going to be used to display the design space using a two-dimensional medium, this restricts the lattice to two dimensions. The next key parameter to determine is the lattice geometry. Rectangular lattices are most frequently used, as these are the simplest to implement. The total size of the lattice is roughly proportional to the square root of the number of data points provided. The ratio of the length of the two sides of the rectangle are in proportion to the size of the first two principal components of the (normalized) training data.

Once the SOM has been trained, the next step is to display the lattices. The shape of the regular lattice is used as a template for the display purposes. Recall that each lattice point is a vector in the original design space. Hence, each lattice point can be considered as a fully described prototypical design. For each design variable, the value that design variable takes is plotted on the lattice by using a colour scale (see Figure 1). The final result is a two-dimensional representation of the design space. By reading these plots, it is possible to navigate the design space. Specifically, target design regions can be identified by finding suitable regions in a subset of component maps, and then mapping this region onto all component maps (see Figure 2). The values taken on by the design variables in each of these maps represent the target design region.

The above approach has been implemented by allowing a designer to mask out areas of the design space that are undesirable. The mask is created from a designer's specified set of constraints. This mask is then applied to all component maps, and then displaying the remaining target region. The implementation details are described in the next section.

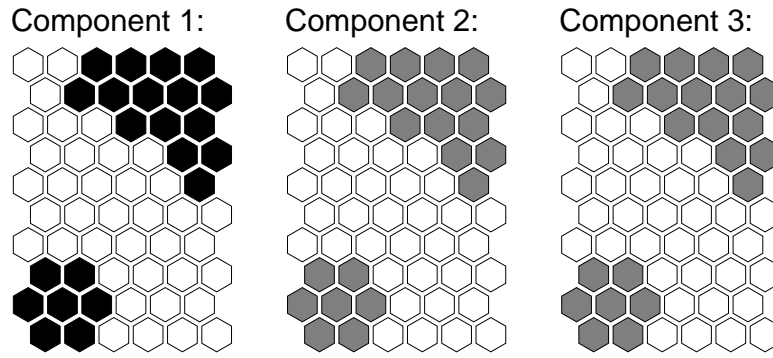


Figure 2. A constraint has been placed onto Component 1 (black lattice nodes), which in turn has been mapped onto the remaining component maps (grey lattice nodes). The design space has been constrained to these filled nodes.

## 4 Implementation

The implementation of the SOM is achieved using the source code supplied by the Helsinki University of Technology [1]. This package provides all the core functions required to create and display a SOM. In addition, the display function has the ability to size the individual lattice nodes when displayed. This was exploited by displaying the constrained nodes smaller than the unconstrained nodes. A benefit of this was that the constrained nodes were still visible, and it was clear what values had been constrained out, making it possible to identify where it would be beneficial to slacken a constraint.

In addition to the code provided by the HUT, a constraint editor was needed. A very simple constraint editor was devised to provide this functionality. This editor provided a list of all the design variables (see Figure 6). The user selects a design variable from the list, and the right-hand half of the window displays the current constraint setting for that variable. Initially, this is set to allow the full range of values, i.e. is unconstrained. On the right-hand side of the window, two sliders permit the designer to set the minimum and maximum permitted values for that design variable. By pressing the ‘Filter’ button, these are then passed to the display function. The display function refreshes the SOM component maps on screen, masking out the design regions where the constraints would be violated. In addition, there are also buttons that generate a text report of the feasible design space and reset all the constraints.

## 5 Case Studies

Two case studies are presented to illustrate this design exploration tool. The first is the design of a flat screen display panel. This is a hand-crafted design problem, which is fully understood and is used for testing and verifying the approach. The second is the design of a gas turbine combustor. This is taken from industry, and is less well understood, but provides an opportunity for industrial designers to trial the tool.

### 5.1 Flat screen display

The flat screen design domain is described using eight design variables. Four of these are design parameters (i.e. parameters that can be directly changed by the designer, see Figure 3) and

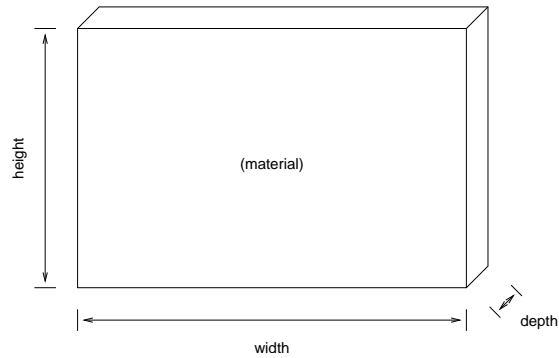


Figure 3. Design parameters of the flat screen display. The design evaluation criteria are: weight, cost, expected life, and expected sales volume.

the other four are design evaluation characteristics. The design parameters are: width, height, depth, and material. The evaluation characteristics are: weight, cost, expected life, and expected sales volume. The relationships defining weight and expected life were defined strictly in terms of the design variables. Expected life and expected sales volumes were partly defined by ‘external influences’. These two were related to each other and expected life. This was aimed to provide an open aspect to the design system, representing potential design parameters that were not, or could not, be explicitly expressed using only the given design parameters.

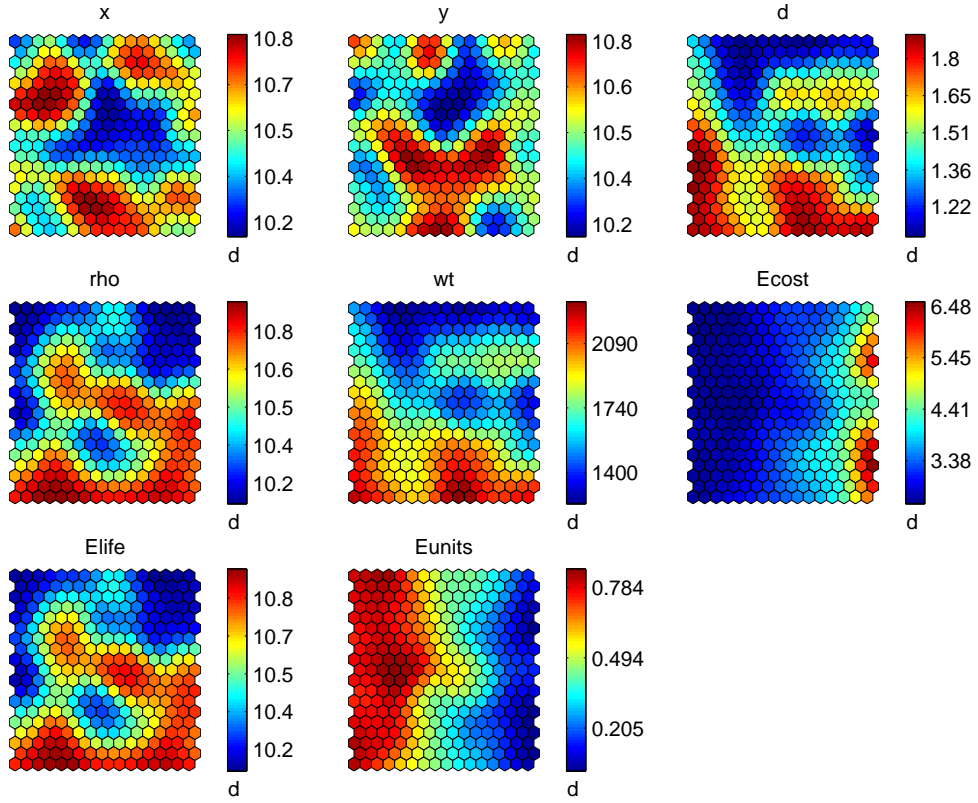
The aim of this case study was to investigate how well the design space could be navigated. Using the display design model, a database of 2000 designs was created. This database represented previous design experience, and used to train the SOM with. A number of design specifications were devised to test navigating with constraints from a number of alternative starting points.

To illustrate, consider the designing a display that *must* be thin. In addition, it would be desirable to sell as many units as possible and for these units to have a long expected life. This represents the order in which the constraints will be placed onto the design space. The aim is to identify the best possible region of the design space that should be explored in greater detail.

The tool is initialized, using the above database of 2000 elements. The user is presented with the constraint editor window and the unconstrained SOM (see Figure 4). The first constraint, requiring the design to be thin, is entered into the constraint editor (see Figure 6). This is then used to mask out parts of the SOM, as can be seen in Figure 5. This procedure is then repeated, first to constrain the expected number of units to be high and then constraining the expected life to be long. These resulted the masked SOM component maps in Figure 7.

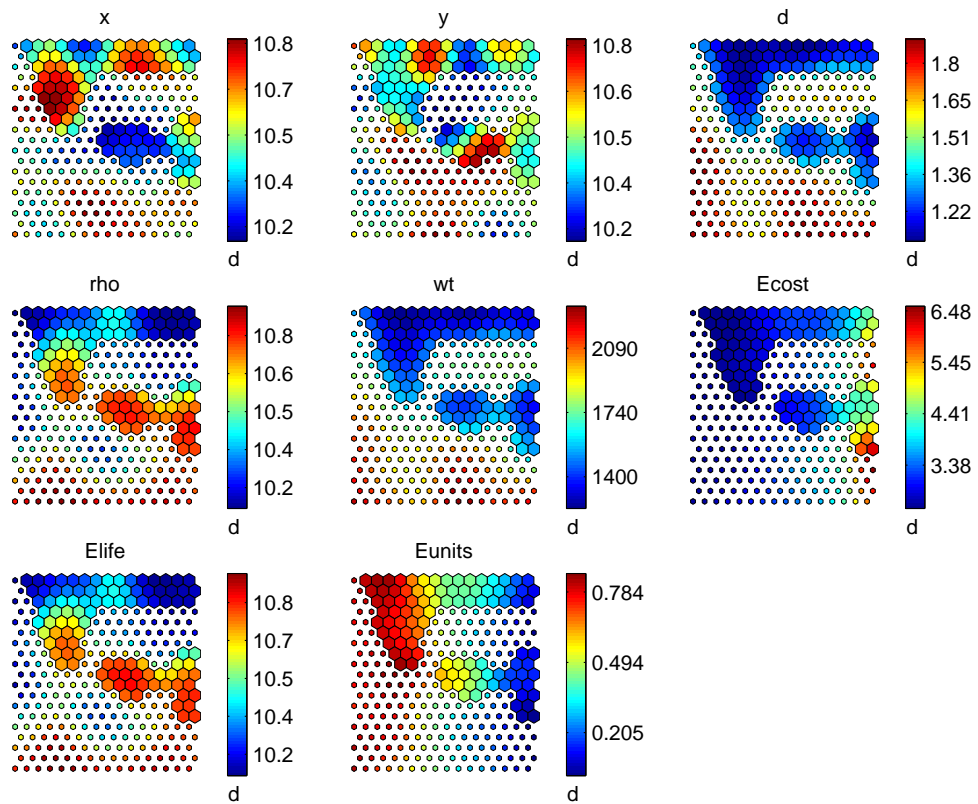
At each iteration, the remaining design region is reduced, thereby focusing the designer into a smaller region of interest. It is interesting to note that at the final stage, there are two disconnected regions of interest (see Figure 7). This mainly separates an area of design with large  $x$  values and one with small  $x$  values. By investigating the corresponding regions on the other component maps, it can be seen that the large  $x$  valued designs can be designed with a ‘lesser’ material (as determined by  $\rho$ ), cost less, last longer, and sell more units than the small valued  $x$  designs. By presenting this result to a domain expert, it is possible to gain further insight to this phenomena.

As can be seen from this process, after each constraint has been added the masked area grows. It is possible to overconstrain the problem, resulting in a completely masked out SOM. In this event it is necessary to slacken constraints to regain a feasible design region.



Map: SOM 12-Feb-2003, Data: Flat Display, Size: 19 16

Figure 4. Unconstrained SOM from the display design case study.



Map: SOM 12-Feb-2003, Data: Flat Display, Size: 19 16

Figure 5. Result of constraining the  $d$  design variable to be small. The large nodes represent the feasible design space, while the small nodes represent the space that fails the constraints.

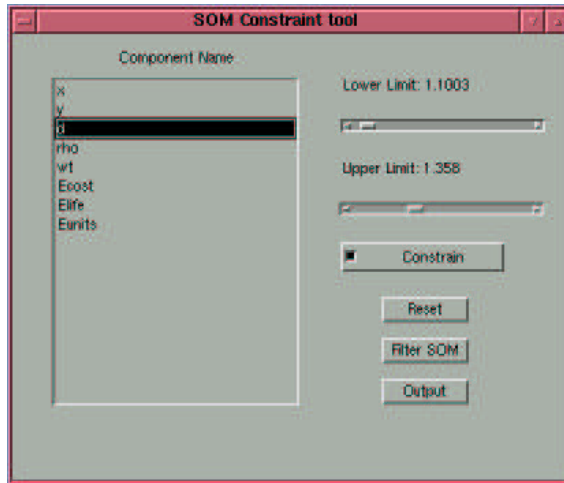


Figure 6. Constraining the  $d$  variable using the editor to:  $1.1003 < d < 1.358$ .

## 5.2 Gas turbine combustor

The combustor of a gas turbine aero engine is a well defined module of the overall engine. The key function the combustor performs is the heating of the air flow through the engine by burning fuel in this air. The main design challenge is providing a good air-fuel mix throughout the combustor. The air flow into the combustor performs two tasks: first is to allow the combustion process to take place and the second is to provide cooling films for the metal surfaces of the combustor which would otherwise melt (see Figure 8).

The aim of this case study was to investigate the graphical user interface in an industrial setting. The system was provided with database of previously developed combustor mixing elements (e.g. holes, ducts, etc. that are placed on the combustor wall). The designers were given a previous combustor specification to meet as a test case. The aim was for the designers to use the tool provide guidance for determining the geometry of the various holes placed in the combustor walls.

The designers were mainly interested in investigating the nature of the tool, and spent a large amount of time testing single constraints. These were used to mask areas of the SOM compo-

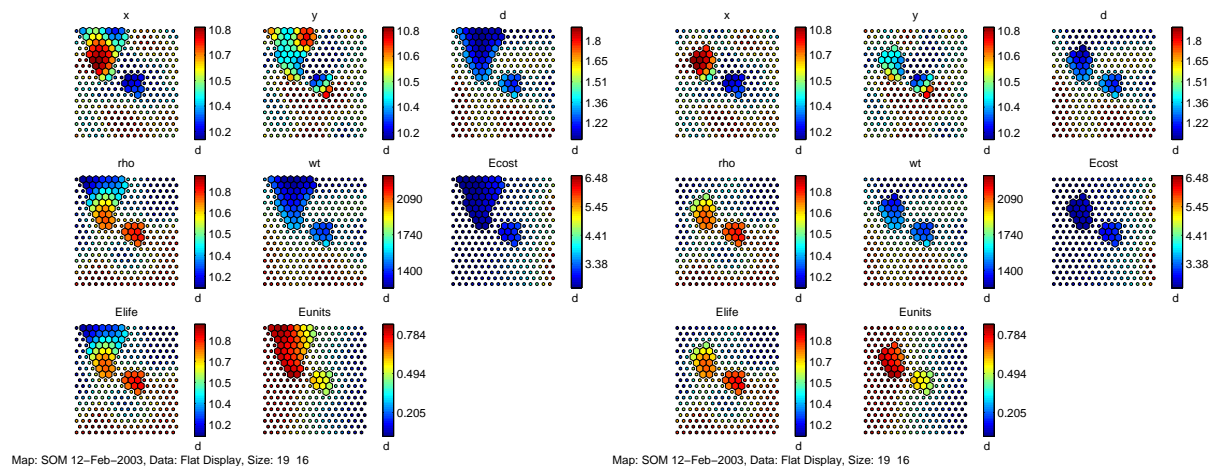


Figure 7. Result of constraining expected number of units to be high (left), followed by constraining expected life to be long (right).



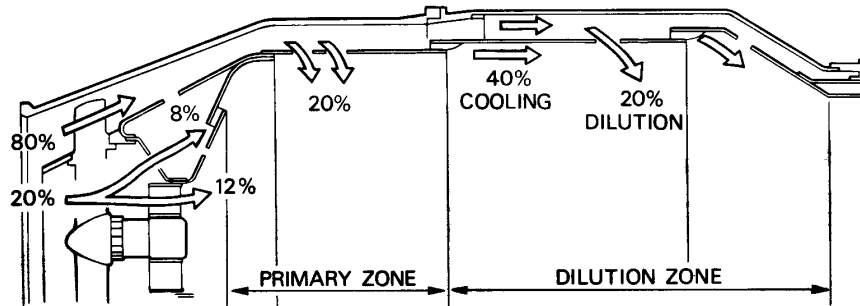


Figure 8. Design schema of a gas turbine combustor (courtesy Rolls-Royce, plc).

nent maps out, and the results were examined closely. Although several of these were initially thought to be providing incorrect guidance, closer investigation in each case revealed a physical reason, proving the correctness of the provided guidance. This in itself is interesting, as a number of ‘surprises’ were uncovered by the designers. This highlights the potential usefulness in using this approach to document the design space, by carefully crafting a series of constraints and describing what effect these have on the design space.

## 6 Conclusions

The initial reviews from this design exploration tool are promising. The use of the SOM provides a very quick means to receiving feedback on suggested constraints. In addition, the use of the tool within an industrial setting provided some encouraging surprises in the form of unexpected phenomena being observed. Upon deeper reflection, these could be explained, demonstrating the potential of using this approach to prompt domain experts into using deeper reasoning when providing documentation.

However, the tool does suffer in the event of an overconstrained design space. Currently, no advice is offered on how best to slacken the constraints. An interactive approach for achieving this needs to be developed. This should be able to merge as seamlessly as possible with the design process, so that results can be rapidly and intuitively obtained.

In addition, comments made by industrial designers trialling the method was of a strong desire to be able to identify changes that have minimal impact on the remainder of the design. This typically arises where a previous design meets all but one criteria. The aim is to modify this design in a manner such that the new design meets the new criteria subject to the minimum number of changes to other aspects of the design. The method described in this paper could be adapted to do so, given a constraint slackening suggestion tool, as described above. The constraints would be set initially to tightly constrain the original design including the new requirement. The constraint slackening tool would then be used in conjunction with the exploration tool described in this paper to guide the designer to the modified design. Work on this approach is currently underway.

## Acknowledgements

This work is funded by the *University Technology Partnership for Design*, a collaborative research project between the universities of Cambridge, Sheffield and Southampton; and with

industrial partners BAE SYSTEMS and Rolls-Royce, plc.

## References

- [1] E Alhoniemi, J Himberg, J Parhankangas, and J Vesanto. SOMToolbox for Matlab. <http://www.cis.hut.fi/projects/somtoolbox/>, December 1997. v1.0beta.
- [2] T N S Murdoch, N R Ball, and P C Matthews. Constraint based templates for design re-use. In A Riitahuhta, editor, *Proceedings of the 11th International Conference on Engineering Design*, volume 3, pages 267–270. Tampere University of Technology, 1997.
- [3] P C Matthews, L T M Blessing, and K M Wallace. Conceptual evaluation using neural networks. In U Lindemann, H Birkhofer, H Meerkamm, and S Vanja, editors, *Proceedings of the 12th International Conference on Engineering Design*, volume 3, pages 1777–1780, August 1999.
- [4] H Chernoff. The use of faces to represent points in  $k$ -dimensional space graphically. *Journal of the American Statistical Association*, 68(342):361–368, 1973.
- [5] A Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1:69–91, 1985.
- [6] T Kohonen. *Self-Organizing Maps*. Number 30 in Springer Series in Information Sciences. Springer-Verlag, Berlin, second edition, 1997.

Dr Peter C Matthews, MIEE

Engineering Design Centre, Cambridge University, Trumpington Street, Cambridge CB2 1PZ, England

Phone: +44 (01223) 332709, Fax: +44 (0870) 133 6961

<http://www-edc.eng.cam.ac.uk/> E-mail: [pm131@eng.cam.ac.uk](mailto:pm131@eng.cam.ac.uk)