Understanding Robust Design Spaces using Using Self Organizing Maps

Thole Sidhant Pravinkumar[1]*, Palaniappan Ramu[2]

[1]*Dept. of Engineering Design,Indian Institute of Technology, 600036, Chennai, India*
** Corresponding author: tholesidhantp@gmail.com*
[2]*Dept. of Engineering Design, Indian Institute of Technology, 600036, Chennai, India*

Abstract
Metamodels are widely used to alleviate the computational expense involved in solving optimization formulations with expensive computer models. Metamodels are built based on the responses evaluated at Design of Experiment (DoE) points. Hence, the DoE and the number of points in the DoE play a vital role in the quality of the metamodel. Understanding function non-linearity is essential to decide the number of points required in the DoE which in turn governs the order of the metamodel that can be built. Self organizing maps (SOM) are predominantly used as visualization tools and for clustering. Here we propose using it for understanding correlation and nonlinearity in the design space, with small samples. The advantage of SOM is that the relationship between the response and any design variable is reduced to a two dimensional plot allowing for easy visualization. This is extremely useful in high dimensional problem where nested axis plots and tile plots are not adequate. Further we propose to use SOM based design space exploration to identify zones of interest for robust design. Robust design refers to small response variation regardless of the intensity of variation in design variables , which often translates to flat or small gradient regions in the response function. Based on limited input samples, we use SOM to explore design space and understand the flat regions.
Keywords : *Robust Design, Design Space Exploration, Self Organizing Maps*

1. Introduction

Design Space Exploration (DSE) is a process in which design alternatives are explored for a variety of purposes such as finding optimal designs, identifying regions of interest, sampling for DoE, dimension reduction etc. DSE is widely used in engineering design tasks to understand complex design spaces or to decompose the design space. DSE might refer to: performing sensitivity studies to understand factor effects or response function non-linearity, creating nested plots to visualise response function variation in design space, conducting factor analysis to estimate factor effects or building surrogates to identify regions of interest in design space. The bottom line is to understand the non-linearity of response function in design space and capture it as accurately as possible. However, all these methods suffer from curse of dimensionality and the number of samples required increases exponentially with the dimensions. Hence, it is desirable to develop approaches that can allow designers to visualise function non-linearity in any dimension. Self Organising Maps (SOM) [1] is an unsupervised neural network, which is generally used for data mapping and currently being successfully used in data mining and visualization. SOM helps in representing multidimensional data onto the 2-dimensional grid, SOM can map non-linear relationship within multi-dimensional data into the simple geometric relationship among their nodes in the 2-dimensional grid. [2] used SOM along with fuzzy clustering to reduce design space into relatively smaller promising regions and performed optimization in the zones of interest. They analyze the relation between design variables and objective function using SOM built on initial samples followed by resampling in reduced design space. [3] used rough set theory and SOM to analyze optima in the design space. [4] explored adaptive sampling methods using SOM to identify set of input variables which would give desired responses. [5] used SOM to visualize pareto data of robust design optimization to visualize spread of the data and analyze tradeoffs between multiple objectives.
Designers usually seek robust designs than optimal designs in the presence of uncertainties. Robust designs guarantee performance in spite of uncertainties.That is, the variability of output response is between some allowable threshold in spite of variabilities in the input space. Robust region is usually characterized by flat regions in design space. Hence, it is desirable to capture flat regions in design space because they are likely to correspond to robust designs. It is to be noted that there may be multiple flat regions in design space. Our interest lies in understanding the function non-linearity in design space with small samples and further identify the region of interest for Robust Design Optimization (RDO) using SOM. The rest of the paper is organised as follows: section 2 discusses the concept and algorithm for SOM, robust design concepts are discussed in Section 3 along with discussion on implementation of SOM for identifying regions corresponding to robust design followed by conclusions in Section 4.

2. Self Organizing Maps

SOM is an unsupervised learning algorithm that provides visualization of high dimensional data by reducing it to the two-dimensional map.SOM is easy to understand,it can be used for clustering,prediction,data representation. The main advantage of SOM is that it preserves the topology of high dimensional data. SOM is similar to nonlinear Principal Component Analysis (PCA) but with the advantage of visualization. SOM differs from ANN on that fact that it does not have error-based learning. SOM works on competitive learning methodology. Once input data is presented to SOM, only one node which is closest to the input point will win. For every input data point, the nodes will compete with each other and the SOM grid gets trained. During the training, the values for the input variables are gradually adjusted to preserve neighborhood relationships (thereby capturing the topology) that exist within the input space. SOM has nodes at output layer arranged in rectangular or hexagonal lattice. SOM output has multiple components such as U Matrix, component planes, hit rate. U (Unified Distance) Matrix provides information about the distances between samples in design space. Component plane plots allows to understand the variation in each input variable and the output response. The algorithm for generating SOM is discussed next.

One starts with a collection of combination of input variables and the respective responses that form the dataset. Each row in a dataset corresponds to a design point and the respective response. The idea of SOM is to use a two dimensional grid of nodes to map each row of the dataset based on a distance criteria. SOM is a two dimensional array of nodes (neurons) where each node is characterised by a vector of weights. Weights here refers to the coordinate of the output node. The number of weights each output node will carry corresponds to the size of the dataset. The next step is, for each row of the dataset, find the node that is closest to it by computing the Euclidean distance between the dataset and the output nodes as presented in Eq.1.

$$D = \sqrt{\left(v_i - w_i\right)^2} \tag{1}$$

where D is the euclidean distance, v_i is a row of the dataset, w_i are weight vector of the output node. The output node that has the least D is considered the Best Matching Unit (BMU) for the dataset row under consideration.

$$v_i = [x_i, y], i \in [1,2,3,...d]$$

$$w_i = [m_{i1}, m_{i2}, m_{i3}, ...m_{i(d+1)}]$$

where d is number of input variables in the problem and y is the output variable (which can also be multiple) and m_i are the weights , m_{d+1} is weight corresponding to the output variable. The radius of the neighborhood from BMU is to be found at every iteration and all the nodes in that radius will have their weights updated using the updation rule according to the equations 2,3,4.

$$w_i(t+1) = w_i(t) + \varphi(t)L(t)\left(v_i(t) - w_i(t)\right) \tag{2}$$

$$L(t) = L_0 \times e^{\left(\frac{-t}{\lambda}\right)} \tag{3}$$

$$\varphi(t) = e^{-\left(\frac{D^2}{R^2}\right)} \tag{4}$$

$\varphi(t)$ is to ensure the weight change is more when the distance of updating node is less from BMU. The radius decays with iterations as in Eq. 5

$$R(t) = R_0(t) \times e^{\left(\frac{-t}{\lambda}\right)} \tag{5}$$

Where R_0 is maximum distance in Lattice and λ is the ratio of the total number of iterations to the R_0 .This process is repeated for all input vectors. SOM is demonstrated on two examples in the following. The focus here is to use SOM output to understand the non-linearity of the response function in the design space. Among other things, this information is useful in deciding the number of points required in a DoE which in turn affects the metamodel that can be built, and its accuracy. We use the SOM toolbox for the implementation.

2.1 Understanding the non-linearity of the response function in design space

Consider the function in Eq. 6. Here, we pretend that we have no information about the function. A DoE with 20 LHS points and respective response values are considered. SOM is trained over the dataset.

$$z = x^2 + y^2 \ , x \in [-5,5] \tag{6}$$

400 output nodes are considered. Fig.1 (a) shows the trained SOM over the original surface given in Eq. (6). Fig.1 (b) shows the U matrix and component plane plots. It can be observed from z component plane plot that the biquadratic nature of the function is captured. It can be seen that as x increases from top to bottom in the component plane z first decreases and then increases, which is similar to the original function. From the y component plane, it can be observed that as y is increasing from left to right, the z first decreases and then increases. In addition, the z component plane shows increasing trend as we go in radially outward direction. This exercise demonstrates that SOM can be used to understand the non-linearity of the response function in design space with respect to the design variables.

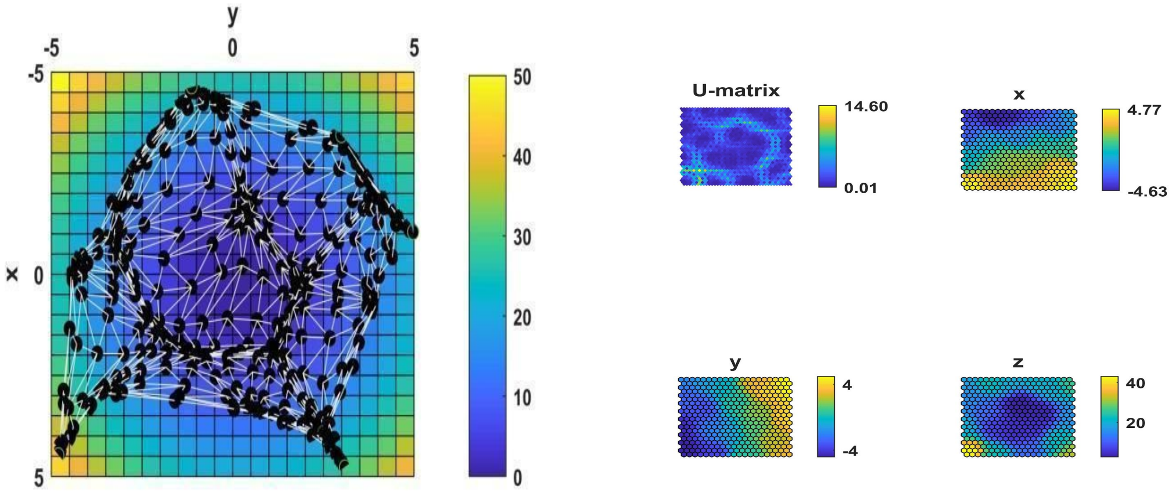

Figure 1. (a) SOM net over surface (b) U Matrix and component plane plots

Modified SOM algorithm

Another instance of 20 LHS samples is considered**.** Fig. 2 shows the trained SOM. It can be observed that the net is self intersecting. This is referred to as aberration [6] and leads to errors in predictions or understanding the function. With less than 2 dimension one can plot the SOM net to check for self intersection. For higher dimensions the aberration is evident from the component plane. That is, the component plane of the input dimensions are not ordered if there is aberration. If there is uniform increase or decrease in all input variable component plane in any one particular direction we can say that net is not self intersecting. In the event of self intersection, there should be abrupt change in the color of component plane it can be concluded that SOM net is self intersecting.For example, in Fig.2b, x component plane has abrupt change in trend*.* x should increase from left to right but it is first increasing then decreasing. Hence, one can suspect self intersection of net. Smoother maps tend to convey more information from the same data. Since, SOM is usually used to extract information on correlation or factor analysis, self intersection of the net was immaterial though it affects the topology preservation quality of self organizing map. However, in this work we have a different objective on understanding the relationship between the input variables and response, flatness etc and we seek to avoid self intersection of SOMs. We use an algorithm proposed in [7] ,where the BMU is found with help of input variables only. Response vector or output is used for updating only.This ensures that there will no intersection and folding in map.The results from the modified algorithm is discussed below:

Ackley function with 5 input variables and one output as shown in Eq. 7 is considered to demonstrate the modified SOM algorithm. 20 LHS data point are considered and SOM net is trained as per the modified algorithm.Fig.3 shows the U-matrix and component plane after training. It can be seen from the z component plane that there is valley around the centre of the z component plane and the value increase as we travel radially outwards from the valley. We can get the region of interest by values of weights associated with corresponding areas of interest. Examining the component planes, one can conclude how the response function varies with respect to the input variables providing a sense of sensitivity information.

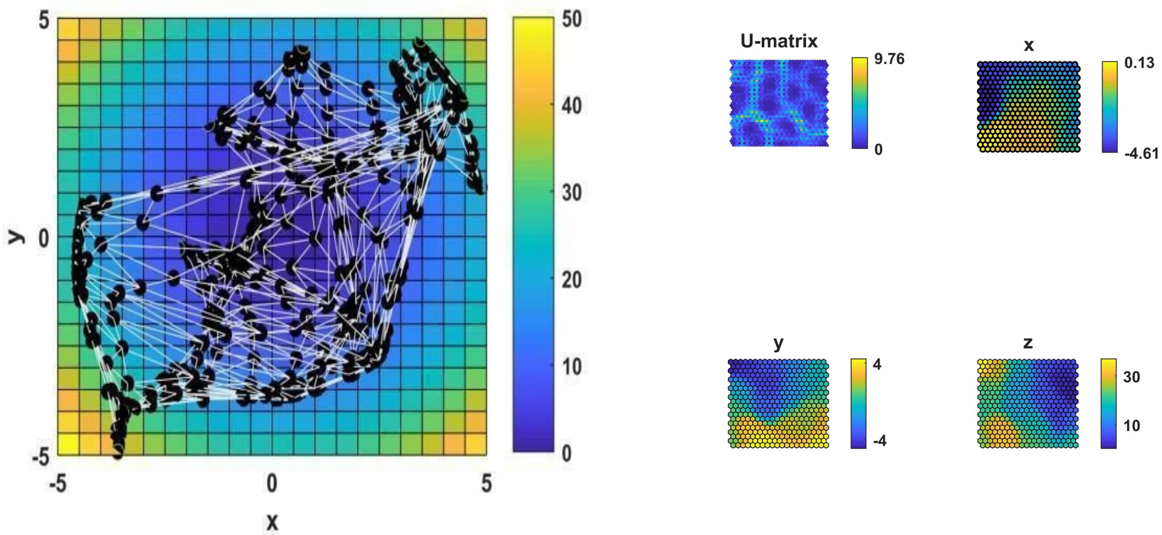

Figure 2. (a) Folded SOM net over surface (b) U Matrix and Component Planes

$$z = -20\exp\left(-0.2\sqrt{\left(\frac{1}{5}\sum_{i=1}^{5} x_i^2\right)} - \exp\left(\frac{1}{5}\sum_{i=1}^{5}\cos\left(2\pi x_i\right) + 20 + \exp(1)\right)\right), x_i \in [-1,1] \tag{7}$$

The point to be noted in the modified algorithm is that for a fixed DoE, the trend of component plane plots are preserved. This is not guaranteed in the conventional algorithm.
Fig.4 shows the Umatrix and component plane obtained by the conventional SOM algorithm. It can be clearly seen that there is no uniform trend in the component plane hinting the self intersection and folding in the trained SOM net.

3. Robust Design

Uncertainties are inevitable in design and it causes poor performance of product.These uncertainty can be caused by variation in material properties,load conditions etc.This uncertainty then gets propagated to output. Traditional design approach takes factor of safety into account and seeks a conservative design.To handle these uncertainty in better way one can use probabilistic techniques.Robust Design Optimization (RDO) is one of the way to deal with this without removing source of uncertainties. As shown in Fig.5, design optimization will choose A as solution but even slight variation in the input x causes a large variation in output. Whereas RDO will choose B as output where if there is variation in input x will cause lesser variation in output as compared to the A. The characteristic of the region corresponding to robust design is flat or near flat with less gradient. Hence, in RDO, it is advantageous to know flat regions apriori so that more samples can be used in the flat region thereby increasing the chances of identifying a robust optima. We propose using modified SOM to visualize the design space and identify flat regions. As discussed above, SOM is obtained based on the DoE points. Once the weights are obtained, K Means algorithm is used for clustering. Number of clusters is decided based on Davies-Boulding index.For each cluster the standard deviation of the weights of the response is computed. When the standard deviation is less, it is likely to point to flat regions or near flat regions leading to robust regions in design space.
Results are validated by comparing trends in standard deviations of responses calculated using original function on large

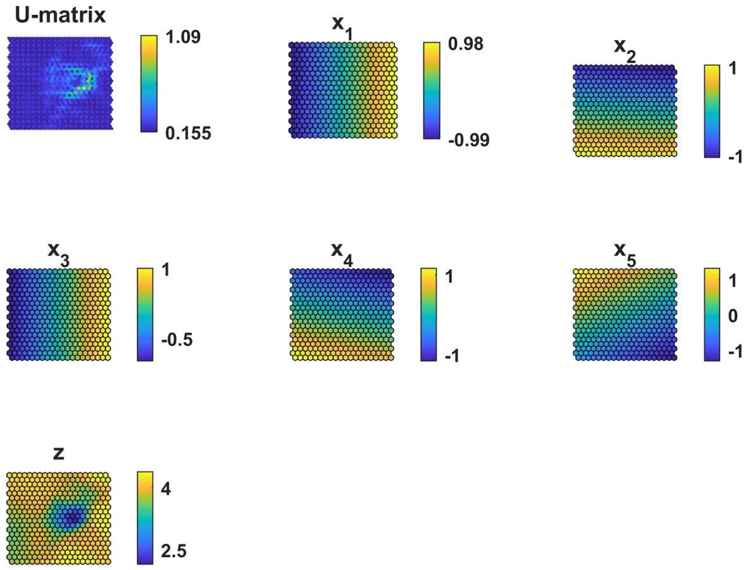

Figure 3.U Matrix and component plane plots using modified SOM algorithm

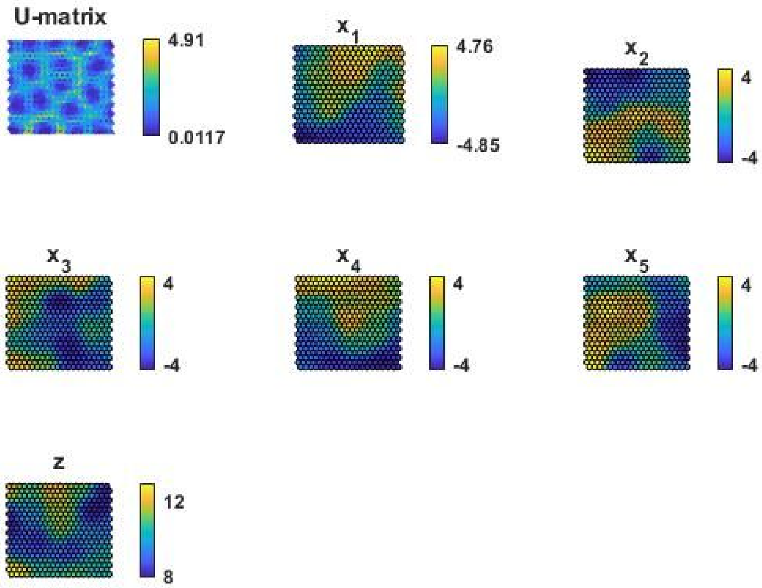

Figure 4. U Matrix and component plane using conventional algorithm

number of samples sampled in each of the clusters. Standard deviation from original function evaluations is denoted by σ_2 and σ_1 denote standard deviations from the response coordinate of weight vectors belong to respective clusters.

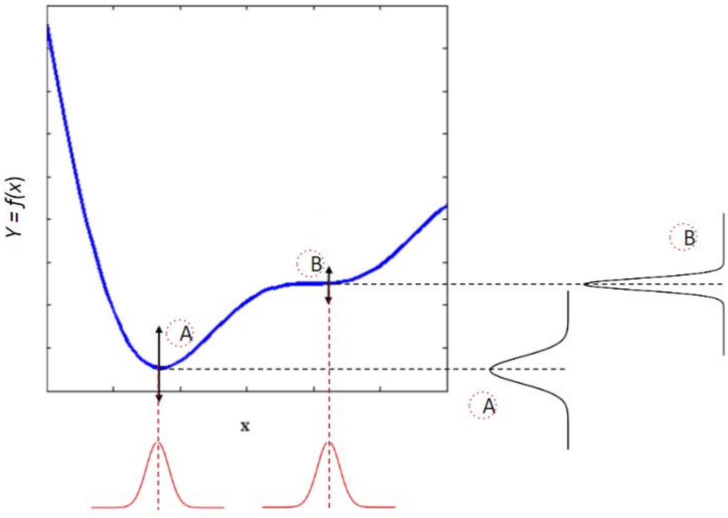

Figure 5. Concept of Robust Design

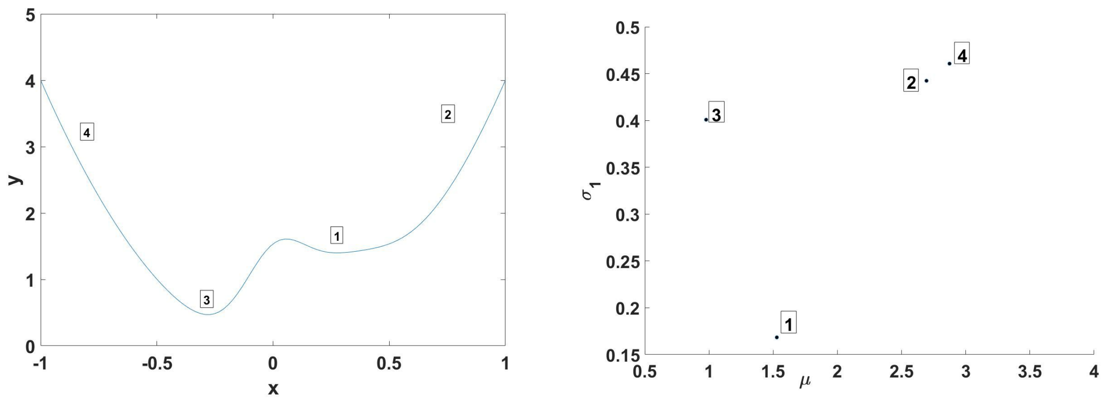

Figure 6. (a)Aspenberg Function (b) μ vs σ_1

Consider the Aspenberg function [7] in Fig. 6 and given by Eq.8.

$$y = 4x^2 + e^{-30x^2} + e^{-10(x-0.25)^2} \tag{8}$$

Fig.6 (a) shows there exist deterministic optima in region where $x \in [-0.4, 0]$ and robust region lies in region $x \in [0.25, 0.5]$. SOM net is obtained based on 15 LHS samples.

Table 1. Summary of Cluster Statistics

Cluster	σ_1	σ_2	x_{ranges}
Cluster 1	0.1684	0.1064	[-0.0596,0.6358]
Cluster 2	0.4424	0.5797	[0.6767,1.0039]
Cluster 3	0.4008	0.3763	[-0.6732,-0.1005]
Cluster 4	0.4607	0.6022	[-1.0004,-0.7141]

Proposed approach gives 4 cluster as output. Standard deviations of all clusters are given in Tab.1. As it can be clearly seen, cluster 1 has least standard deviation and the ranges of x corresponding to this cluster corresponds to the region of robust design.Fig.6 (b) shows μ vs σ_1 plot (μ is the mean of weight corresponding to response). One can refer to this plot and choose the cluster depending on one's requirement. That is, depending on one's interest to reduce mean or standard deviation, one can choose the regions. For instance, one would choose cluster 1 if looking for least variation in response and would choose cluster 3 in case of least response value but with an acceptable variation of 0.4.

Next, we consider 20 LHS samples of Aspenberg function with 2D input. Equation of function is given in Eq.9. Fig.8 (a) shows the top view of the Aspenberg function. It can be observed that it has deterministic optima in region where $x \in [-0.5, 0], y \in [-0.5, 0]$ .There exist two robust regions at $x \in [0, 0.5], y \in [-0.5, 0]$ and . $x \in [-0.5, 0], y \in [0, 0.5]$.

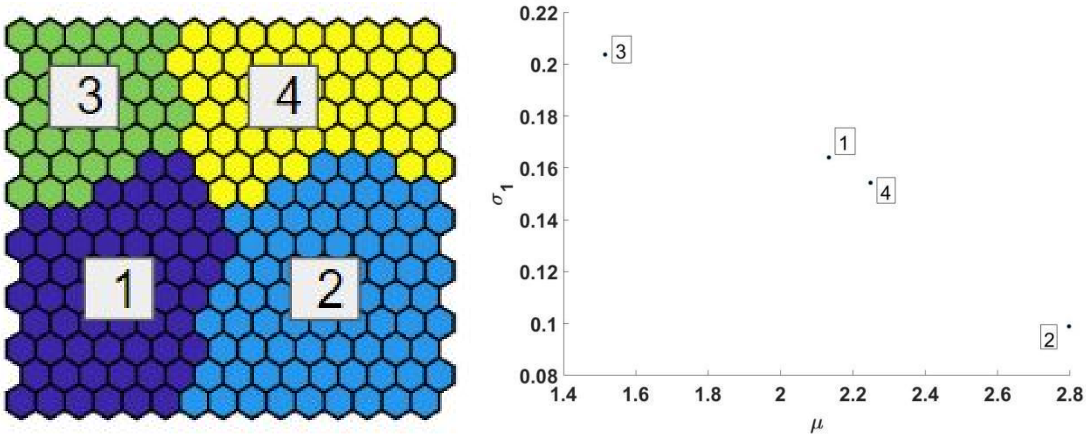

Figure 7. (a) Clusters numbers and (b) μ vs σ_1

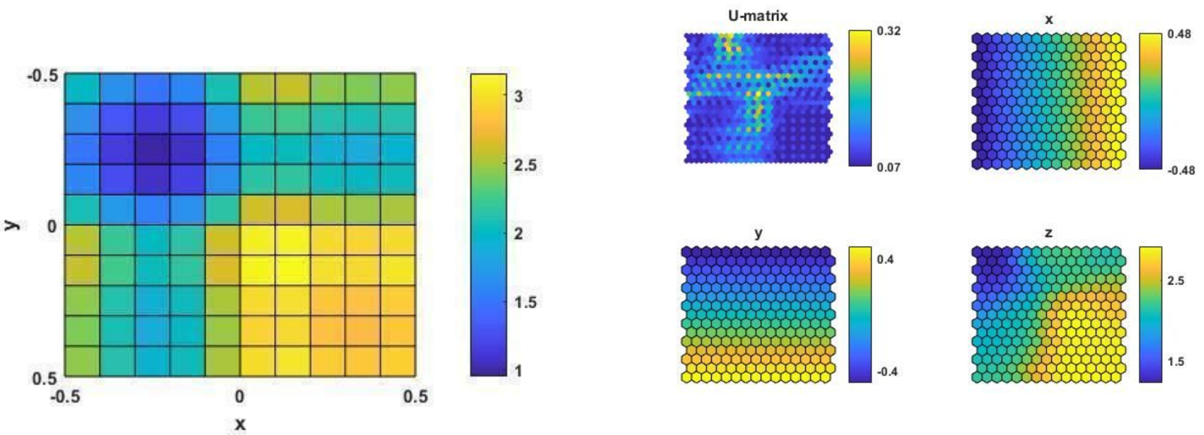

Figure 8. (a) Top View with colormap of Aspenberg Function (b) U Matrix and component planes

$$z = 4\left(x^2 + y^2\right) + e^{-30x^2} + e^{-30y^2} + e^{-10(x-0.25)^2} + e^{-10(y-0.25)^2} \tag{9}$$

After training SOM over samples and performing clustering which is shown in Fig 7 (a) ,the computed standard deviations are compared in Tab.2 It can be observed that cluster 3 shows the largest deviation and other clusters show lesser deviation.

It can be inferred that clusters 1,2 and 4 are comparatively robust than cluster 1.This can also be validated from σ_2 values and top view of function Fig. 8 (a). The above two exercise shows that the modified SOM algorithm along with clustering technique can be used to identify the robust regions in design space.Fig.7 (b) shows μ vs σ_1 plot.

Next, we consider 50 LHS samples of Aspenberg function with with 4D input given by Eq. 10. Proposed approach gives 5 clusters as output. Standard deviations of all clusters are given in Tab.3. Fig. 9 shows the U matrix and component planes. Fig .10 (a) shows cluster numbers.It can be inferred from Tab.3 that Cluster 4 is the having least standard deviation.Cluster 3 can be ranked as second robust region.Fig .10 (b) shows the μ vs σ_1 plot.It can be seen that cluster 4 has higher mean compared to other clusters but least standard deviation.Fig. 9 shows the U matrix and component planes

Table 2. Summary of Cluster Statistics

	σ_1	σ_2	x_{ranges}, y_{ranges}
Cluster 1	0.1640	0.1744	[-0.4758,0.0029] , [-0.1365,0.4880]
Cluster 2	0.0989	0.1208	[-0.0655,0.4816] , [-0.1365,0.4880]
Cluster 3	0.2037	0.2517	[-0.4758,-0.1339] , [-0.4835,-0.0671]
Cluster 4	0.1542	0.1840	[-0.1339,0.4816] , [-0.4835,-0.0671]

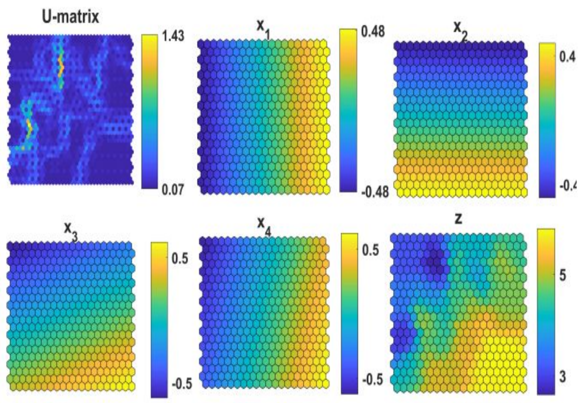

Figure 9. U Matrix and component planes

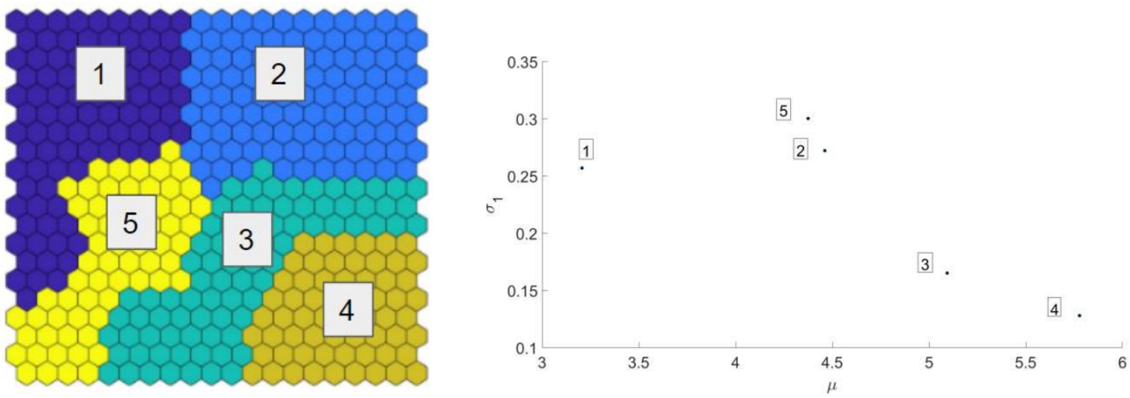

Figure 10. (a) Cluster numbers and (b) μ vs σ_1

$$z = 4\sum_{i=1}^{4} x_i^2 + \sum_{i=1}^{4} e^{-30x_i^2} + \sum_{i=1}^{4} e^{-10(x_i-0.25)^2} \quad (10)$$

Table 3. Summary of Cluster Statistics

	σ_1	σ_2
Cluster 1	0.2567	0.4555
Cluster 2	0.2722	0.3688
Cluster 3	0.1651	0.2268
Cluster 4	0.1280	0.1503
Cluster 5	0.3003	0.3217

4. Conclusions:

This work discusses using SOM for response function visualisation in design space. The underlying idea is to identify function characteristics such as peaks, valleys, non-linearity of the design space and correlation between the input variables**.** A modified SOM algorithm is used to address the issue of aberration in classical SOM. This SOM is further clustered based on the weights aiding in identifying flat regions in design space that correspond to robust designs. Further standard deviations of weight corresponding to response is calculated and then compared with standard deviations of response values obtained from large number of function evaluation in all clusters as validation. Further μ - σ_1 plots are provided to understand function variability in design space. This allows the designer to make intelligent design decision such as if one wants minimum value of response or minimum variation in response.

References:

[1] T. Kohonen, "Self-Organizing Maps," *Springer Ser. Inf. Sci.*, vol. 30, p. 501, 2001.

[2] H. Qiu, Y. Xu, L. Gao, X. Li, and L. Chi, "Multi-stage design space reduction and metamodeling optimization method based on self-organizing maps and fuzzy clustering," *Expert Syst. Appl.*, vol. 46, pp. 180–195, 2016.

[3] X. Z. Chu, L. Gao, H. B. Qiu, W. D. Li, and X. Y. Shao, "An expert system using rough sets theory and self-organizing maps to design space exploration of complex products," *Expert Syst. Appl.*, vol. 37, no. 11, pp. 7364–7372, 2010.

[4] K. Ito, I. Couckuyt, and T. Dhaene, "Design space exploration using Self-Organizing Map based adaptive sampling," *Appl. Soft Comput. J.*, vol. 43, pp. 337–346, 2016.

[5] S. Parashar, V. Pediroda, and C. Poloni, *Self Organizing Maps (SOM) for Design Selection in Robust Multi-Objective Design of Aerofoil*. 2008.

[6] E. Lopez-Rubio, "Improving the quality of self-organizing maps by self-intersection avoidance," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 24, no. 8, pp. 1253–1265, 2013.

[7] Thole S, Ramu P (2019) Design Space Exploration using Self Organizing Map, National Conference on Multidisciplinary Design Analysis and Optimization, March 22-23, 2019, Banglore, India

[8] D. Aspenberg, J. Jergeus, and L. Nilsson, "Robust optimization of front members in a full frontal car impact," vol. 0273, 2013.