INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITE DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

EXPLORING DESIGN SPACE USING MULTI-INSTANCE MODELLING

B. Singh[1], J. Matthews[1], G. Mullineux[1] and A. J. Medland[1]
[1]University of Bath, UK

ABSTRACT
The ability to visualise and explore the space of feasible design solutions is important in all design tasks. It is often complicated by the large number of design parameters involved and the fact that their interaction is poorly understood. In many cases there are examples of design instances which are known to perform successfully. Other instances can be explored using suitable parametric models. An approach is proposed which can help to build up some appreciation of the design space. This is to start with the known successful instances and interpolate (or "morph") between these and determine the resultant performance which can be visualised as a surface. This is demonstrated by application to selection of mechanisms with a prescribed output, starting with examples chosen from a catalogue.

Keywords: Design space, machine design, constraints, multi-instance modelling, optimisation, rationalisation, visualisation

1 INTRODUCTION
The typical design process usually involves the manipulation of a large number of design parameters which constitute larger degrees of freedom. For example, in the mechanical area of mechanism design, the simplest motion generator is a four bar chain and this has nine degrees of freedom in total. When subsystems interact, the complexity further increases with the compounding of the degrees of freedom and the introduction of constraints imposing relations between them.
Small and medium sized companies which design and manufacture machines often do not have the resources to fully analyse their designs [1] and as a consequence do not fully understand the ranges of product that their machines can handle successfully. The result is that if a customer requires a machine to perform some form of variant task, the response is to design and produce a variant machine [2] rather than investigate whether an existing design can cope. This leads to a collection of one-off designs which then have to be supported during their operational lives.
The difficulty in analysing the capabilities of such ranges of designs is that the number of degrees of freedom is large and so sophisticated computer-based procedures are required [3]. These can operate as a "black box" into which a model of the design is entered and from which some form of optimal design is created. The designer however receives no feedback as to how sensitive the design is to small variations and what variant designs might be possible.
A number of researchers have investigated means for providing the designer with information about the design space which is being used and about the progress of optimisation schemes and the sensitivity of the results. Some of these approaches are reviewed in section 2. One particular area in which visualisation is important is in "computational steering" [4] where a designer interacts and controls an optimisation process such as those occurring in the resolution of design constraints [5,6,7].
While the optimisation of a complete design may be complex, the existence of one-off examples of workable designs, here called "instances", can be used to perform a simplified investigation. This paper investigates two related areas. The first is the use of these "instances" of allowable designs to investigate the full design space. The second is the use of the approach for identifying possible families of related designs and the implications of these for rationalising product ranges.
The basic approach is discussed in section 3. It is to model the known instances of a successful design. As suggested before, a large number of design parameters are likely to be present. However the fact that the instances are good designs suggests that some of the required relations between the parameters are already satisfied (albeit on a heuristic basis). Given these instances, interpolation or "morphing"

between them can be used to generate other instances whose performance can be evaluated. This information can be provided to the designer as surface plots. Some examples are given in section 4 based on the selection from a catalogue of mechanisms to provide a specified output motion. Finally some conclusions are drawn.

2 DESIGN SPACE VISUALISATION TECHNIQUES

This section discusses some of the various visualisation techniques that have been adopted for design space exploration. Straightforward methods include the use of bar charts, pie charts, scatter data plots and surface plots. These have the advantage of simplicity. However they are only helpful when number of the variables to be plotted is small. One way to add extra dimension to the conventional methods used is to incorporate visual aids like colour, shape and relative size. However there is still need for multi-dimensional data representation that incorporates heuristics and design knowledge [8]. Some of the techniques developed for exploring high dimensional problem spaces in the engineering design are now presented.

The first is nested performance charts [9]. In standard performance charts, performance is plotted against the design variables. This allows understanding of the effect of a certain design variable on the overall performance of the system. The drawback of this technique is that it is limited to one or two design variables. A nested design chart is essentially an array of performance charts. Each involves (say) two parameters and the performance value. A third design variable is allowed to change (over discrete values) as one move across the array, and fourth as one move up and down. In this way the effects on the individual charts as the third and fourth variables change is made clear. A limitation associated with this method is that at least two design variables must be discretised.

One way to incorporate heuristics and design knowledge in multi-dimensional data representation is with the use of “computational steering” [4,10,11]. Traditional optimisers operate as “black box” solvers which present the designer with end results from a prescribed input. However the designer gets no information about the alternative solutions and trend followed by the optimisation process. Computational steering aims to provide the ability to visualise how solution procedure is progressing. This lets the designer alter the parameters during the analysis and so influence the optimisation process. However the main drawback is the need for very high computational power for its calculations and data transfer.

A modified computational steering paradigm is “visual design steering” [12]. The approach is divided into two stages. Firstly the problem is reduced by eliminating (temporarily) those design variables and constraints that do not have a large impact. Secondly this reduced problem is visualised and a technique of “graph morphing” is used which enables useful insight to be gained which allows the selection of a start point for subsequent searching which is closer to the global optimum. It is claimed that there is often a 50% reduction in the number of function evaluations required.

A “cloud visualisation technique” has also been proposed [8] that is also based on computational steering. Here the designer can view and interact with all previously generated information. It provides the ability to navigate, interact with, and customise the data being presented and hence steer the optimisation process by providing new points to investigate.

For some particular design tasks it can be possible to “normalise” the problem so that some of the design parameters are removed and hence the remaining ones are easier to visualise. For example, in the kinematic design of parallel mechanisms, normalisation can be applied to reduce the number of parameters by one [13]. The range of allowable values for the others is clearly determined which helps clarify visual feedback.

3 MULTI-INSTANCE MODELLING

It is rare that any design activity starts entirely from first principles. Most design work is in some way a modification of an existing design concept. Usually that concept has resulted in some form of successful product, and possibly variations of that product. So there is some pre-history which has resulted in a number of “instances” of a successful design.

One particular example of this is found with companies (especially small and medium sized companies) which design and manufacture machines for specific tasks. When a customer requires a new machine, the first step is to see whether an existing design meets the new specification. If none is available then a new variant design is created and hence a new instance of a machine is produced.

Indeed other instances may have been generated as part of the design process and then rejected because performance was not what was required.
The idea which underlies the approach described in this paper is that designers often have information about known (successful) designs. So there it may be beneficial to use such information as a guide to what the full design space looks like. That space usually has a large number of dimensions, corresponding to the number of design parameters, which may easily be unwieldy. Very many combinations of the design parameters lead to infeasible designs. Basing design exploration upon successful instances can mean that relationships implicitly required between the parameters are imposed without the need for a formal investigation of what these may be. In this way, although a large number of design parameters are likely to be present, the fact that the instances are good designs suggests that some of the required relations between the parameters are already satisfied (albeit on a heuristic basis). The assumption is made that the instances share essentially the same "topology" although the "geometry" of each is different.
It is also necessary to have one or more performance metrics by which any design instance can be evaluated. Such metric might include: cost, weight, ability to reach target values. A suitable computer-based model of the design is required which is parametric. This allows different instances to be investigated and the performance metric(s) evaluated.
Given a collection of "base" instances, the next stage is to investigate the effect of interpolating or "morphing" between them. If p is a particular parameter of the design and there are n instances, then there n values of the parameter among the base instances. Denote these by $p_1, p_2, ..., p_n$. To find a new morphed instance, a set of weighting values $\alpha_1, \alpha_2, ..., \alpha_n$ are taken whose sum is unity, the same set for all the parameters. The new value of parameter p is then taken as

$$\alpha_1 p_1 + \alpha_2 p_2 + ... + \alpha_n p_n \qquad (1)$$

and the similar combination for all the other design parameters. These new values can be used within the parametric model and the morphed instance created. It may not work successfully. However if it does, its performance metric(s) can be determined.
This can be done in discrete steps, for different choices of the weights α_i, and the performance evaluated for each new instance. The simplest case is morphing between two base instances, which produces a curve of performance measures. This can be extended to morphing between three base instances which creates a surface of performance measures. In the examples in the next section, the surface is plotted above an equilateral triangle whose vertices represent the base instances. The height of the surface gives the value of the metric. Any morphed instance corresponds to a point within the triangle and the weights can be regarded as the barycentric coordinates of this with the respect to the triangle.
If the weights are all non-negative, then the corners of the surface patch represent the base instances which are used. The weights can of course be taken as negative, although, in this case, one is extrapolating away from the base instances rather than interpolating between them.
Consider the case of interpolation. The surface (or curve) may be continuous and reasonably "flat". This suggests that the base instances used all lie within a single "family" of possible designs. The optimal member of this family can be found as that new interpolant which provides the best performance value(s). It also suggests that this optimal design is relatively insensitive to variations in its parameters. If the surface rises or falls steeply away from some of the vertices, this suggests that the original base instances lie in different families and there is some fundamental difference between these designs. The sensitivity of the base instances is likely to be high. An extreme case is when the surface is discontinuous or has holes in it. This suggests that there are situations where the design has failed and the base instances are definitely distinct in some way. There is the danger that variation in parameters may lead to a design which fails.
This sort of analysis can have implications in the area of product rationalisation. If a company produces a range of variants of a particular design, it may be interested in knowing whether that range can be reduced. If several instances create a roughly flat surface, then, as suggested above, there is commonality between them. The best morphed design based upon these may be capable of fulfilling the tasks of the original ones and hence these can be replaced by a single design.
It is also possible to morph between four or more base instances. This can naturally be carried out in a high dimensional space for example a tetrahedron can be used to represent four instances. However for ease of visualisation a surface in three dimensions is used. Here the surface is again plotted over a

(regular) polygon whose vertices represent the base instances. It needs to be noted that with more base instances, there is more than one choice of weights that can specify any given point within the polygon. (One way to make a definite choice is to use one of the extensions of the idea of barycentric coordinates used to define surfaces patches with arbitrary numbers of sides [14].) Conversely, a single surface cannot capture all the possible combinations of the base instances. In particular ordering of the instances will change the plot.

An alternative to representing the base instances by the vertices of a regular polygon is to place them in the plane according to some form of natural ordering. The examples in the next section, deal with the selection of a mechanism to generate a specified output curve. In this case, the relevant base instances could be represented by a vertex whose position corresponds to that of the particular mechanism. In the case when there are many design objectives, one can be used as the performance metric and one or two of the others used to determine the position of the vertices for the base instances.

4 CASE STUDY: CATALOGUE SELECTION

This section discusses a particular case study example. This is based around the use of a catalogue of mechanisms. A common design task is the selection of a mechanism to achieve a prescribed motion. Before the advent of computer aids, one starting point for the design of such mechanisms was an atlas of standard mechanisms and the output curves that they generate (e.g. [15]). Today such paper-based catalogues can be set up electronically. One way to do this is to create a parametric model of one or more standard mechanism types. Each is then run with a range of choices of the parameters. If the parameter choice is inappropriate, then the mechanism does not cycle correctly. When proper operation occurs, then the mechanism is stored in a disc file in terms of its type and parameters, and with it is stored the output path.

The path can be stored in terms of a number of points around it. An alternative is to treat the path as a closed planar curve and to form its (complex) Fourier coefficients [16]. These coefficients can then be stored instead of points on the path. However, the storage can be refined. The first few Fourier coefficients have a geometric interpretation. For example, those of the fundamental frequency are the coordinates of the centroid of the path curve. Those of the next harmonic represent the size of the path and its rotational orientation. It is possible [16] to "normalize" the path coefficients so as to move its centroid to the origin, rotate it to a standard orientation, and to scale it to a standard size. These operations are equivalent to rigid body transforms (isometries) and a global scaling applied to the mechanism itself. If just the normalised mechanism and path coefficients are stored, this removes the need to hold "duplicates" which are the same apart from translation, rotation or scaling. This clearly reduces the size of the catalogue and makes retrieval more straightforward.

When a new path is given and a mechanism sought to create it, the first stage is to find its Fourier coefficients and normalise these. These are then compared with those stored in the catalogue. Comparison is by taking the sum of the squares of differences of corresponding values (Euclidean distance). Mechanisms with low sum values are good. Such mechanisms are then "unnormalised" and these provide candidate mechanism which can achieve the path required.

In some cases, the "best" mechanism found in this way is good enough to be carried forward into the next stages of the design process. In other cases, there may be other limitations on what can be done, and the so a list of possible candidates needs to be inspected. If none is found to be exactly suitable, then one strategy is to try adjusting the parameters of one such candidate mechanism in order to try to improve the selection. Such adjustment can be made manually (assuming a suitable parametric modeller is available). Another approach is to use some form of automatic optimisation scheme. If it is just a question of path matching, then the objective function for the optimisation is the comparison value between the Fourier coefficients and the variables are the parameters describing the mechanism.

One drawback with automatic optimisation is that there may be a large number of degrees of freedom: for instance a four bar linkage has nine independent parameters. Another is that if the search is automatic, the designer has little information about how well the search has performed and the sensitivity of the result to small changes in the mechanism parameters or in the path specification. Such feedback can be provided to the designer by giving some means to visualise the design space.

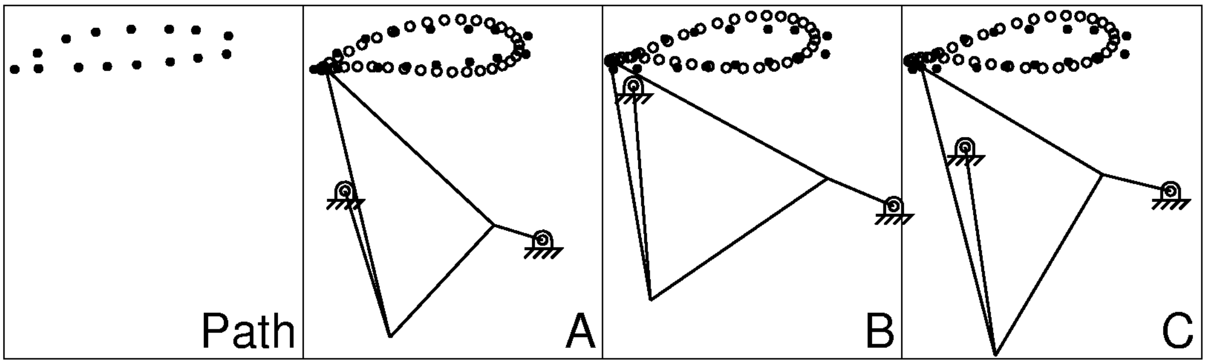

Figure 1. Given path and three base mechanisms

As an illustration, consider the three mechanisms shown in Figure 1. These are all obtained from a catalogue as providing paths which match closely the prescribed path also shown in the figure. It is clear that these mechanisms are similar to each other. These three are taken as successful design instances and morphing between them is undertaken as previously described. Each newly created instance is cycled and its output compared with the given path. The comparison value for each is held and used to plot the surface shown in Figure 2.

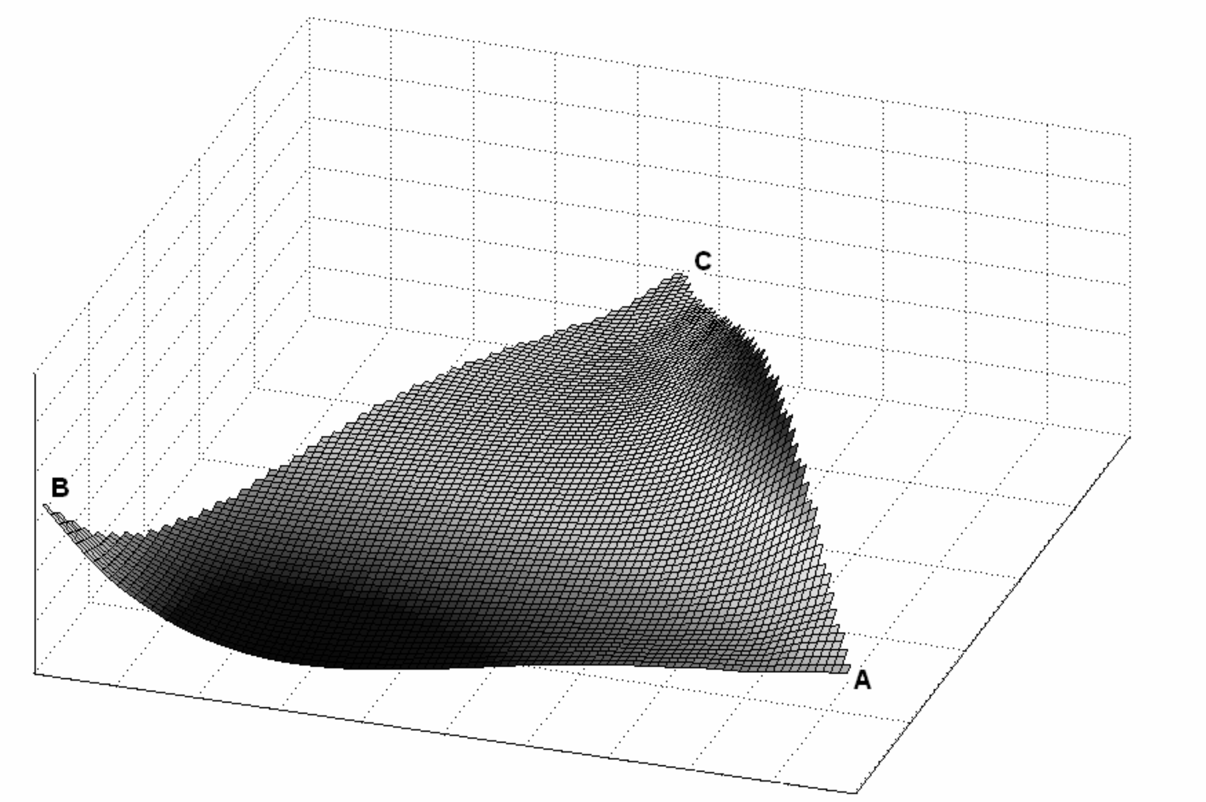

Figure 2. Result of morphing between the base mechanisms

It is clear that the surface is roughly flat. However there is a minimum value of the comparison value within the triangle and this represents a better mechanism providing a better path match than the original three. The flatness of the surface suggests that the three candidate mechanisms are similar and that any solution in triangular region is likely to be insensitive to small changes.

Three other mechanisms producing good matches to a given path are shown in Figure 3. These again come directly from a search of the catalogue. When these are morphed and the resultant mechanisms tested, the surface obtained is that shown in Figure 3.

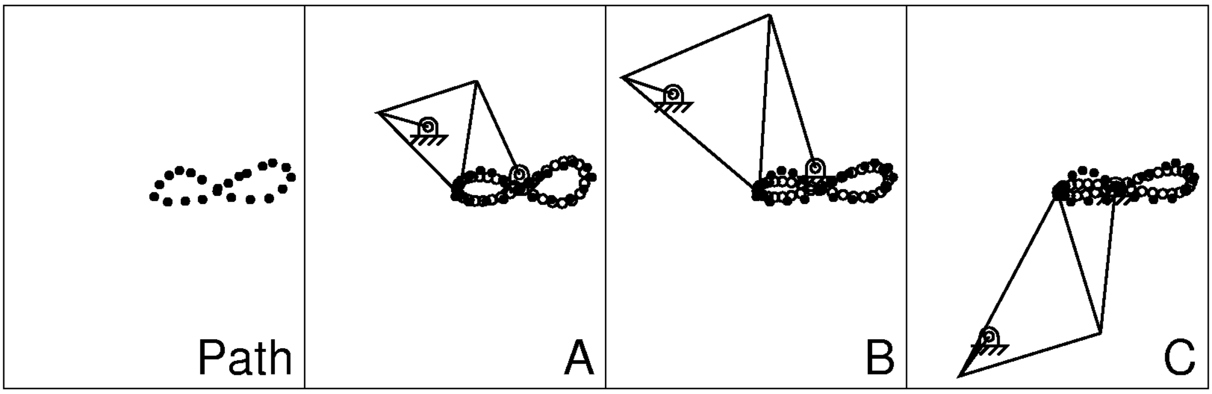

Figure 3. Given path and a set of three base mechanisms

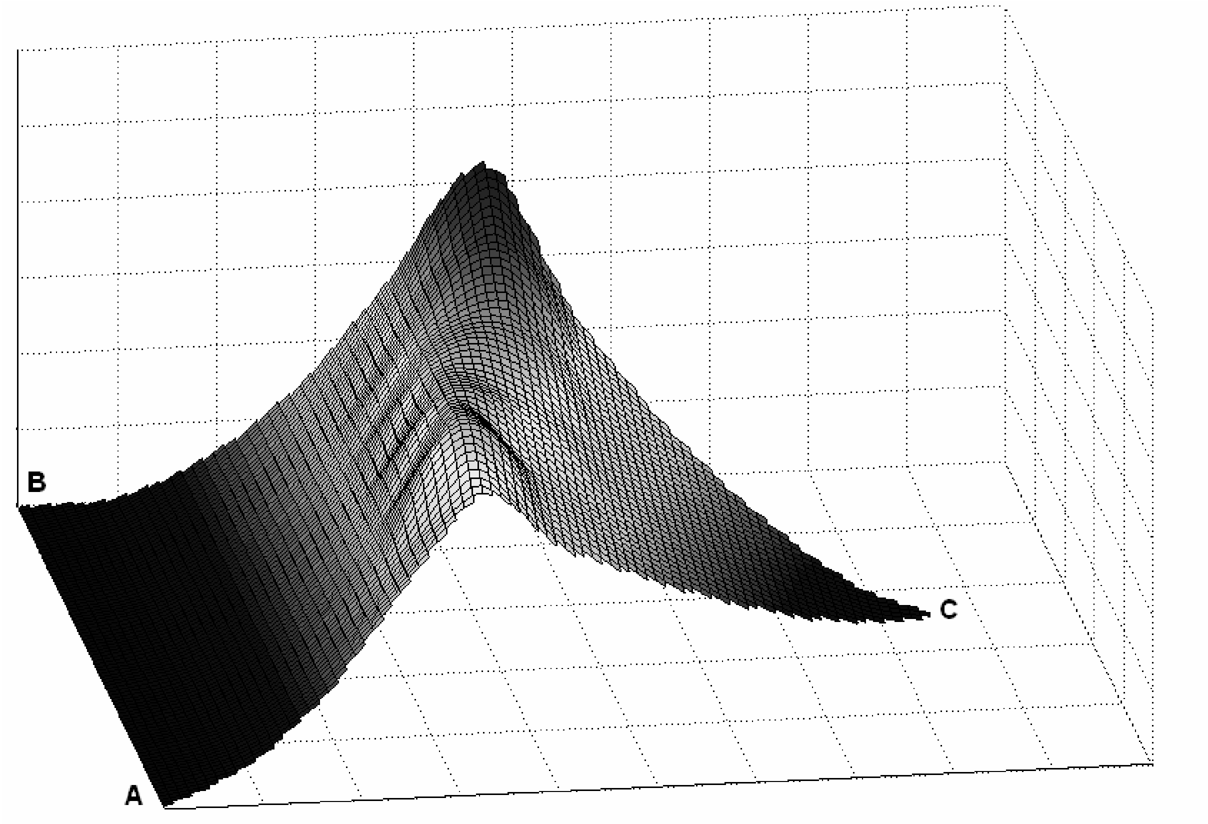

Figure 4. Surface resulting from morphing the different set

What is now seen is that the surface is considerably less flat. There is a ridge separating one vertex from the other two. This suggests that the isolated vertex represent a mechanism which, in some sense, belongs to a different class. The other two can be thought of variations of each other. A lowest point in the surface is still available and represents a better choice than any of the initial three. The separation shown by the surface is perhaps not surprising given that mechanism C is clearly “lower” than A and B in Figure 3. Another three mechanisms and their surface are shown in Figures 5 and 6. Here the mechanisms are apparently similar but still a separating ridge appears in the surface.

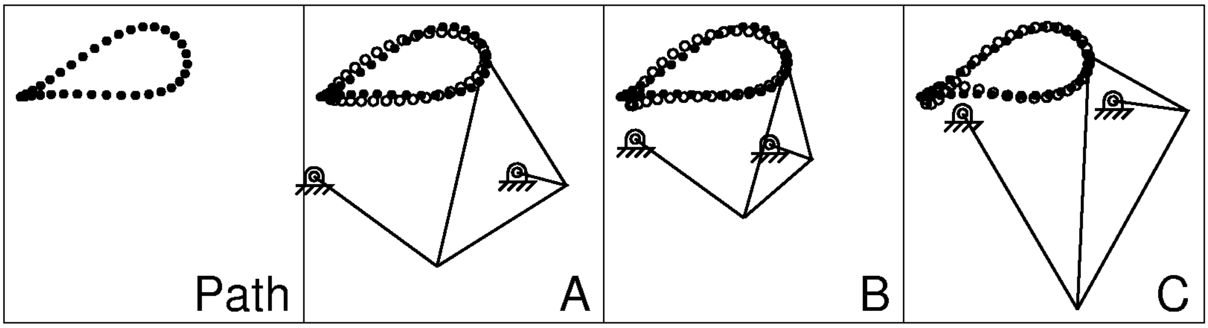

Figure 5. Another path and three base mechanisms

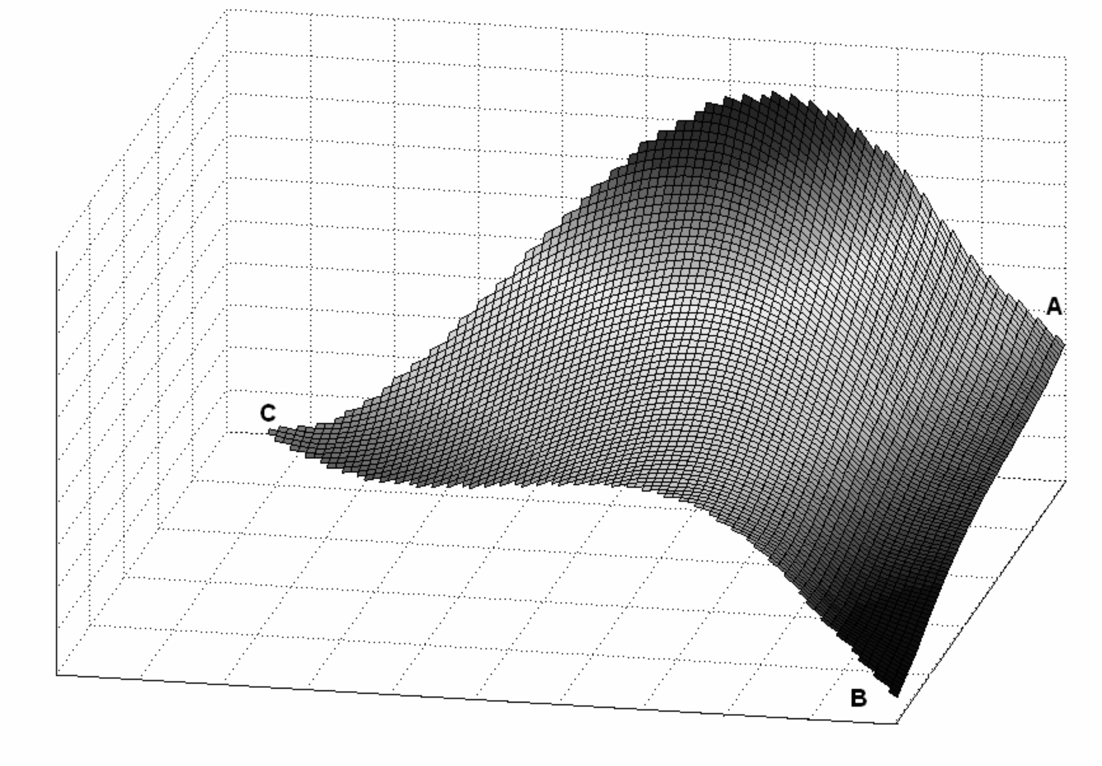

Figure 6. Surface representation of morphing between the base mechanisms

Figure 7 shows four candidate mechanisms selected from the catalogue. These can also be morphed and each new instance evaluated against the prescribed path. As noted before the order in which the original four are taken does now affect the resultant surface. One such is shown in Figure 8. It is seen that the surface is roughly flat with a minimum, corresponding to a better choice, lying on the interpolation between mechanisms A and B.

Figure 9 shows the same example with the exception of mechanism D. Mechanism D has been intentionally replaced by one whose path is close to the given figure of eight but which is roughly elliptical. The resulting surface is shown in Figure 10. It is clear by comparing Figures 8 and 10 that the relationship between mechanism A, B and C holds the same as of the earlier case. A better choice is still lying on the interpolation between mechanisms A and B. However performance of the mechanism D is changed significantly; Point D on the surface is raised by a factor of 10. There is a ridge separating mechanism D from others which indicates that this belongs to a different class. It also validates the objective function used which is based upon the comparison value between the Fourier coefficients of the prescribed path and the generated path.

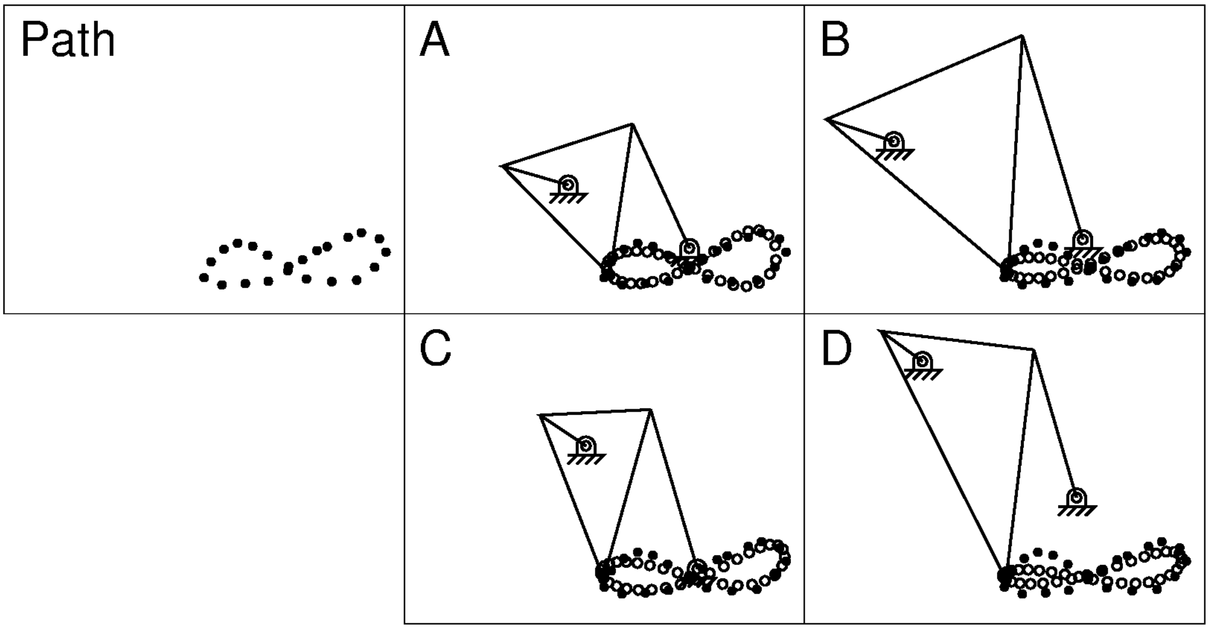

Figure 7. Four base mechanisms

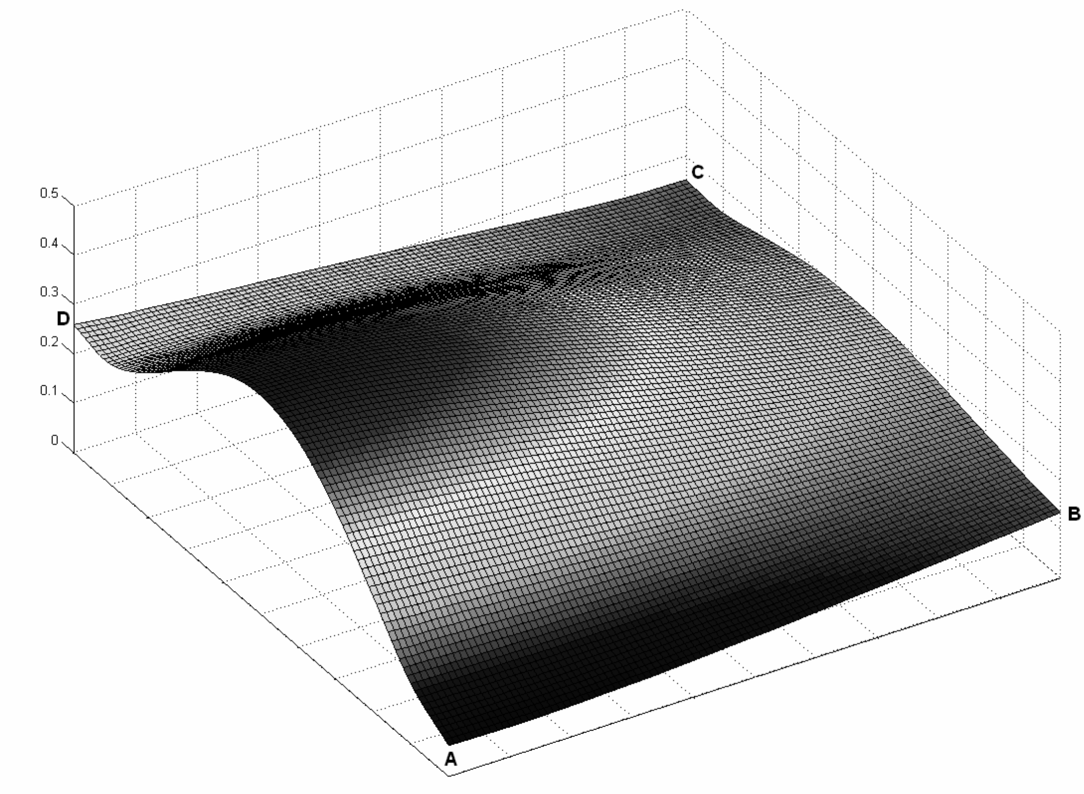

Figure 8. Four-Sided surface

5 CONCLUSIONS

Usually a design task is never entirely new. There are cases of related designs that are known to perform successfully. These represent feasible design instances within the full space of design solutions. The designer needs to understand something of that full space, but full understanding may not be possible due to the large number of design parameters that are involved.

The approach adopted here has been to use the known instances as the basis for an approximate representation of the design space. Interpolation or "morphing" between instances allows a surface to be created which represents a metric of performance. While this is certainly not a complete view of the space, it is suggested that the necessary heuristic constraints which need to be satisfied between the design parameters are satisfied because these are inherited from the base instances used. The approach enables instances which lie in similar or different families to be identified.

The application to the selection of mechanisms to generate a given output motion has been discussed. Here design instances are taken from a catalogue of mechanisms. The morphing procedure allows mechanisms which more closely generate the required motion to be identified.

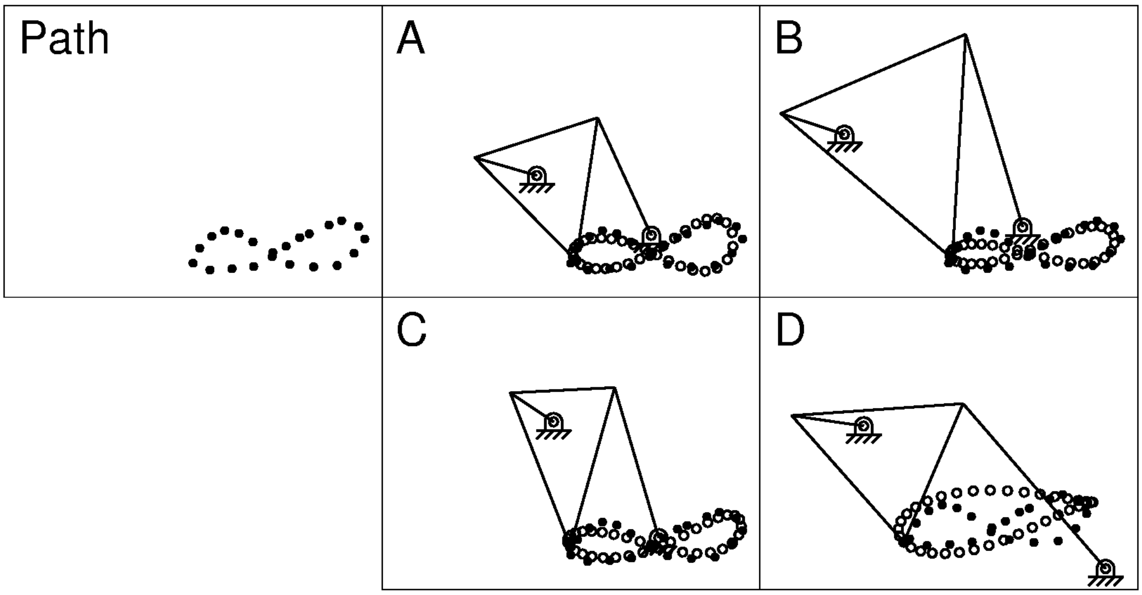

Figure 9. Four mechanisms with one following different path

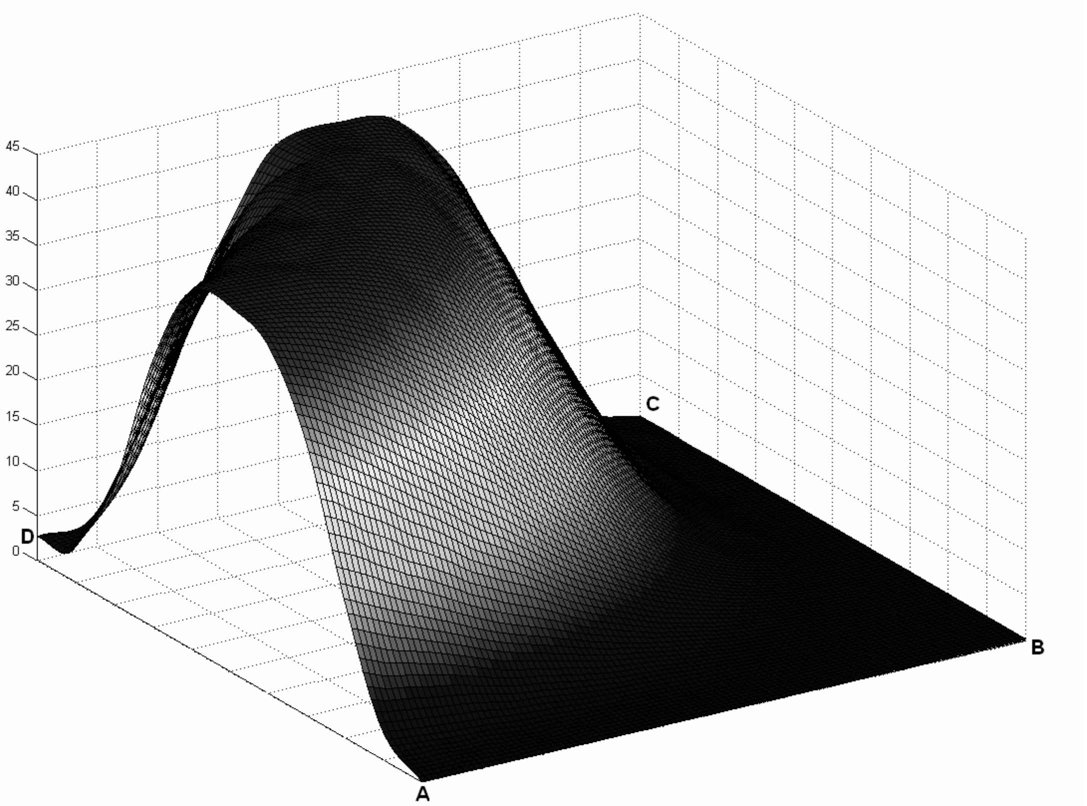

Figure 10. Resulting four-sided surface

ACKNOWLEDGEMENT

The current research work is carried out with the support of a DEFRA research grant given as part of the Food Processing Faraday-KTN. This support is gratefully acknowledged together with that of other collaborators

REFERENCES

[1] Vergeest, J. S. M., Akar, E. and Opiyo, E. Z., Computer-supported shape conceptualization - Interfacing engineers with academic research, *ICED '01*, Glasgow, Aug. 2001.

[2] Cederfeldt, M. and Elegh, F., Design Automation in SMEs - Current state, potential, need and requirements", *ICED '05*, Melbourne, Aug. 2005.

[3] Dalton, J. A., Norman, P. A., Sen, P. and Whittle, S., Extending the design opportunities afforded by modelling frameworks", *ICED '05*, Melbourne, Aug. 2005.

[4] Parker, S. G., Weinstein, D. M. and Johnson, C. R., The SCIRun computational steering software system, *Modern Software Tools for Scientific Computing*, 1997, pp5-44.

[5] Jian-Xin, Ge., Shang-Ching, C. and Xiao-Shan, G., Geometric constraint satisfaction using optimization methods, *Computer-Aided Design*, 1999, 31, pp867-879.

[6] O'Sullivan, B. A., *Constraint-Aided Conceptual Design*, Professional Engineering Publishing Limited, London, 2002.

[7] Hicks, B. J., Medland, A. J. and Mullineux, G., The representation and handling of constraints for the design, analysis and optimization of high speed machinery, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM)*, 2006, 20, pp313-328.

[8] Eddy, J. and Lewis, K., Multidimensional design visualization in multi-objective optimization, *9th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, Georgia. 2002.

[9] Burgess, S., Pasini, D. and Alemzadeh, K., Improved visualisation of the design space using nested performance charts, *Design Studies*, 2004, 25, pp51-62.

[10] Geist, G. A., Kohl, J. A. and Papadopoulos, P. M., CUMULVS: providing fault tolerance, visualization, and steering of parallel applications, *International Journal of Supercomputer Applications and High Performance Computing*, 1997, 11, pp224-235.

[11] van Liere, R., Mulder, J. D. and van Wijk, J. J., Computational steering, *Future Generation Computer Systems*, 1997,12, pp441-450.

[12] Winer, E. H. and Bloebaum, C. L., Visual design steering for optimization solution improvement, *Structural and Multidisciplinary Optimization*, 2001, 22, pp219-229.

[13] Liu, X. J. and Wang, J., A new methodology for optimal kinematic design of parallel mechanisms, *Mechanism and Machine Theory*, in press, corrected proof, 2006.

[14] Gregory, J. A. and Hahn, J., A C2 polygonal surface patch, *Computer Aided Geometric Design*, 1989, 6, pp69-75.

[15] ESDU., *Atlas of Four Bar Linkages*, ESDU International, London, 1989.

[16] McGarva , J. and Mullineux, G., Harmonic representation of closed curves, *Applied Mathematical Modelling*, 1993,17, pp213-218.

Contact: B. Singh
University of Bath
Department of Mechanical Engineering
Bath, BA2 7AY
United Kingdom
Tel: (00) 44 1225 385937
Fax: (00) 44 1225 386928
b.singh@bath.ac.uk