Advances in Engineering Software 000 (2017) 1–12

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Research paper

Computing permissible design spaces under consideration of functional responses

W. Graf*, M. Götz, M. Kaliske

Institute for Structural Analysis, Technische Universität Dresden, Germany

ARTICLE INFO

Article history:
Received 16 February 2017
Revised 19 May 2017
Accepted 29 May 2017
Available online xxx

Keywords:
Numerical design
Inverse problem
Cluster analysis
Functional responses
Permissible design spaces

ABSTRACT

The design of structures is one of the major tasks for engineers. The objectives of numerical design – computing robust and reliable structures – can be realized by means of analyzing different variants, application of optimization tasks, or solving inverse problems. The method of choice depends on the objective of design, the problem and available data/resources.

This contribution introduces an application oriented procedure for the solution of an inverse problem, on the basis of existing data. This data consists of design vectors and related responses. This contribution is focused on functional responses. With this procedure independent areas of permissible designs can identified. These areas are approximatively described by hyperrectangles, which are multi-dimensional boxes. The procedure uses various algorithms within the context of data mining. The relevant methods will be discussed shortly.

The proposed field of application is the detection permissible design spaces in early design stages. Therefore the applicability of the presented procedure is shown by practical numerical design studies.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction

Optimization concepts have in common, that a parametrization of the design problem (input parameters) and an identification of significant result variables (output parameters) are necessary. In this context the terms variable, component and parameter are used as synonyms. The solution of an inverse problem is the computation of specific input parameters yielding a target design. This contribution shows an application oriented algorithm for an approximate solution of the inverse problem. The procedure is based on a set of pre-evaluated designs, out of a Design of Experiments (DoE). The designs can be classified into permissible and non-permissible designs. The two steps (generation of point set and decision) are shown in the two leftmost boxes in Fig. 1.

A response is an output of an engineering model, while parametrized response variables are called performance variables. Two types of results are existing. On the one hand a discrete result, meaning, that for each design a specific amount of result variables is given. An example for this type of result variables is the mass of a system, which is in general not temporal changing. On the other hand, the result could have continuous properties, e.g. the result could be time-dependent. This type of variables is in common FE post-processing programs often called "history variable". This contribution formulates different approaches for handling these functional responses. A definition of permissibility should be made, in order to obtain the results. This means, that the point set is divided into permissible and non-permissible results and inversely related designs. Due to the non-linearity of the objective functions, the permissible areas can be disconnected, see Fig. 1.

To identify these independent areas of permissible designs, cluster technologies are applied to partitioning the permissible points. An inverse problem can be solved, by identification of permissible regions within the design space. These regions can be described approximatively by hyperrectangles, see [1–3]. Due to the axis-parallelism of hyperrectangle, the permissible regions can be described without interaction. A permissible hyperrectangle is computed for each cluster. The set of permissible hyperrectangles is called a permissible design space, in which designs that yield permissible responses can be found. Synonyms for permissible design spaces are: solution space [4], feasible space, feasible region, feasible design areas [5] or permissible domains.

The available approaches to compute permissible design spaces are grouped into analytical and data mining based. The first group, is connected with interval arithmetics [6]. An approach for robust designs in linear programming can be found in [7]. The second group applies and combines various data mining technologies. Therefore methods of Computational Intelligence, as partitioning

* Corresponding author.
E-mail address: wolfgang.graf@tu-dresden.de (W. Graf).

http://dx.doi.org/10.1016/j.advengsoft.2017.05.015
0965-9978/© 2017 Elsevier Ltd. All rights reserved.

Nomenclature

ξ	mapping operator, objective function
$\mathfrak{D}(p,q)$	distance of discrete or functional responses p and q
$\#\{\mathcal{A}\}$	operator for the number of elements in the set $\mathcal{A}$
$\mathbf{D}$	design space
$\mathcal{D}$	set of design vectors in the design space
d	design vector
$\mathbf{K}$	continuous space
$\mathcal{K}$	set of support points in the continuous space
k	support point
$\mathbf{Z}$	result space
$\mathcal{Z}$	set of result vectors in result space
z	result vector
$\mathcal{C}$	cluster configuration (set of point sets = set of clusters)
$\mathcal{H}$	set of permissible hyperrectangles
n_d	number of design dimensions
n_z	number of result dimensions
n_k	number of continuous dimensions
n_s	number of samples
n_t	number of support points
n_c	number of clusters

and pattern analysis, are often used. The presented procedure is related to the second category. The aim of the procedure is an unsupervised identification of large areas of permissible designs, without a priori knowledge about the number of independent regions. These two steps (identification of partitions and computing permissible design spaces) are shown in the two rightmost boxes in Fig. 1.

The procedure is based on a general four step algorithm and can be found in Section 2.

Further applied and discussed methods within this contribution are:

- **Design of Experiments (DoE)** is the starting point of the procedure. By a DoE, input parameter will be defined in an appropriate way and the related responses (discrete or functional) will be computed.
- **Cluster technologies** are widely used in the context of data mining and can be applied to various kinds of data. The only condition is, that the differences of two designs can be quantified by a real number. The different cluster algorithms will not be discussed in this paper, only the relevant similarity measures for discrete and functional responses, which can be found in Section 3.
- **Interval approach** is introduced in this contribution to compute independent design areas. The computation of this permissible hyperrectangle is formulated in Section 4.
- **Visualization of high dimensional data** is necessary to validate the results and to have understandable results. An effective method is the Parallel Coordinate Plot [9], that creates vectors of high-dimensional spaces (as design vectors) on parallel coordinates. Another useful visualization technology is the Self-Organizing Map (SOM) [10]. Theses maps can be used to visualize high-dimensional data in 2-dimensional figures, by highlighting of regions with high point density. This information can be used to validate the results of the clustering methods. A discussion about different visualization technologies will not be in the focus of this contribution.

The examples in Section 5 show the applicability of cluster technologies for functional responses and the detection of permissible hyperrectangles by the proposed algorithm. In Section 6 the capabilities of the procedure are discussed. This paper is based upon [11], but the current paper focuses on the solution of inverse problems.

2. Procedure for detecting permissible design spaces with functional responses

2.1. Definitions

In order to consider discrete, as well as functional responses, it is necessary to clarify the different spaces, sets and vectors. To preserve the overview about the symbols, a nomenclature can be found at the beginning of the paper. The **design space** is defined as

$$\mathbf{D} = I_1 \times I_2 \times \ \ldots \times I_{n_d} \subset \mathbb{R}^{n_d},\ \text{where}$$
$$I_i = [a_i, b_i] \wedge a_i, b_i \in \mathbb{R} \wedge a_i < b_i \,\forall\, i \in \{1, \ldots, n_d\}, \tag{1}$$

with $n_d \in \mathbb{N}$ design dimensions. The variables a_i, b_i indicate the boundaries of each design dimension i. The design space is defined without any interaction between the design parameters, in order to describe it as independent intervals, called bounding box. The design space contains a design vector $d = [d_1\ d_2\ \ldots\ d_i\ \ldots\ d_{n_d}]^{\mathrm{T}}$, $d \in \mathbf{D}$.

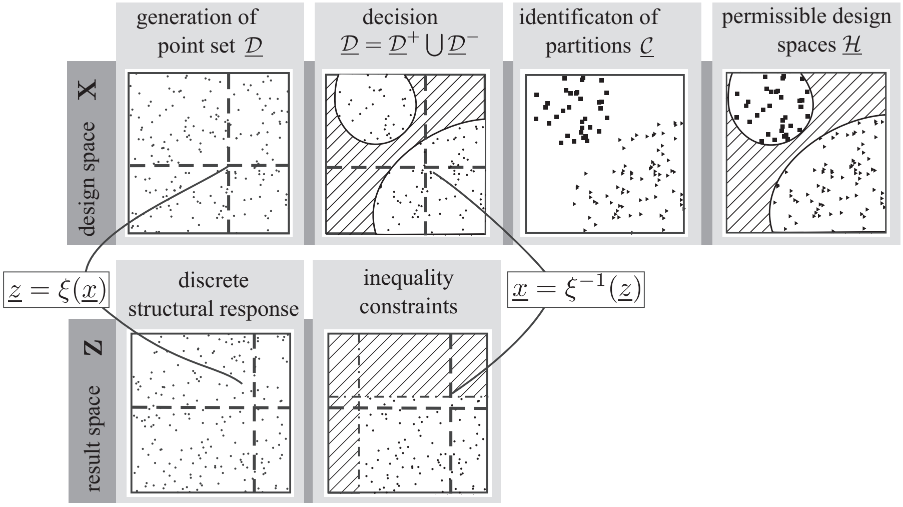
generation of point set $\underline{\mathcal{D}}$
decision $\underline{\mathcal{D}} = \underline{\mathcal{D}}^+ \bigcup \underline{\mathcal{D}}^-$
identificaton of partitions $\underline{\mathcal{C}}$
permissible design spaces $\underline{\mathcal{H}}$
design space $\mathbf{X}$
$\underline{z} = \xi(\underline{x})$
discrete structural response
inequality constraints
$\underline{x} = \xi^{-1}(\underline{z})$
result space $\mathbf{Z}$

Fig. 1. Algorithm for detecting permissible design spaces for discrete responses.

Please cite this article as: W. Graf et al., Computing permissible design spaces under consideration of functional responses, Advances in Engineering Software (2017), http://dx.doi.org/10.1016/j.advengsoft.2017.05.015

Download English Version:

https://daneshyari.com/en/article/6961379

Download Persian Version:

https://daneshyari.com/article/6961379

Daneshyari.com