Engineering Design via Surrogate Modelling

A Practical Guide

Alexander I. J. Forrester, András Sóbester and Andy J. Keane

University of Southampton, UK

WILEY

A John Wiley and Sons, Ltd., Publication

Engineering Design via Surrogate Modelling

Engineering Design via Surrogate Modelling

A Practical Guide

Alexander I. J. Forrester, András Sóbester and Andy J. Keane

University of Southampton, UK

WILEY

A John Wiley and Sons, Ltd., Publication

This edition first published 2008

Registered office
John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex,
PO19 8SQ, United Kingdom

For details of our global editorial offices, for customer services and for information about how to apply for permission to reuse the copyright material in this book please see our website at www.wiley.com.

Library of Congress Cataloging in Publication Data

Forrester, Alexander I. J.
Engineering design via surrogate modelling : a practical guide / Alexander I.J. Forrester, András Sóbester, and Andy J. Keane.
p. cm.
Includes bibliographical references and index.
ISBN 978-0-470-06068-1 (cloth : alk. paper) 1. Engineering design—Mathematical models. 2. Engineering design—Statistical methods. I. Sóbester, András. II. Keane, A. J. III. Title.
TA174.F675 2008
620′.0042015118—dc22
2008017093

A catalogue record for this book is available from the British Library

ISBN 978-0-470-06068-1

Set in 10/12pt Times by Integra Software Services Pvt. Ltd. Pondicherry, India
Printed and bound in Great Britain by Antony Rowe Ltd, Chippenham, Wiltshire

Contents

Preface

Think of a well-known public personality whom you could easily identify from a photograph. Consider now whether you would still recognize them if most of the photograph was obscured, except for the corner of an eye, a small part of their chin and, perhaps, a half of their mouth. This is a game often played on television quiz shows and some contestants (and viewers at home) often display an uncanny ability to come up with the correct name after only a few small sections of the picture are revealed.

This is a demonstration of the brain's astounding ability to fill in blanks by subconsciously constructing a *surrogate model* of the full photograph, based on a few samples of it. The key to such apparently impressive feats is that we actually know a great deal about the obscured parts. We know that the photograph represents a human face, that is the image is likely to be roughly symmetrical, and we know that somewhere in the middle there must be a pattern we usually refer to as a 'nose', etc. Moreover, we know that it is a famous face. The 'search space' thus reduced, the task seems a lot easier.

The surrogate models that form the subject of this book are educated guesses as to what an engineering function might look like, based on a few points in space where we can afford to measure the function values. While these glimpses alone would not tell us much, they become very useful if we build a number of assumptions into the surrogate based on our experience of what such functions tend to look like. For example, they tend to be continuous. We may also assume that their derivatives are continuous too. With such assumptions built into the learner, the surrogate model becomes a very effective low cost replacement of the original function for a wide variety of purposes.

Surrogate modelling has had a great impact on the way the authors think about design and, after many years of combined experience in the subject, it has become a fundamental element of our engineering thought processes. We wrote this book as a means of sharing some of this experience on a practical level. While a lot has been written about the deeper theoretical aspects of surrogate modelling (indeed, references are included throughout this text to the landmarks of this literature that have informed our own thinking), what we strove to offer here is a manual for the practitioner wishing to get started quickly on solving their own engineering problems. Of course, like any sharp tool, surrogate modelling can only be used in a scientifically rigorous way if the user is constantly aware of its dangers, pitfalls, potential false promises and limitations – the present text goes to great lengths to point these out at the appropriate times.

To emphasize the practical dimension of this guide, we accompany it with our own *MATLAB*® implementation of the techniques described therein. Snippets of this code are included in the text wherever we felt that, through the 'maths-like' and compact nature of *MATLAB*, they contribute to the explanations. These, as well as all the rest of the code, can be found on the book website at `www.wiley.com/go/forrester`. Template scripts are also provided, ready for the user to replace our objective function modules with his or her own. It is worth noting here that our own example functions, while mostly representing 'real life' engineering problems, were designed for easy experimentation; that is they take only fractions of a second to run. We expect, however, that most of the applications the codes will be used for 'in anger' will be several orders of magnitude more time-consuming.

This is a self-contained text, though we assumed a basic familiarity with calculus, linear algebra and probability. Additional 'mathematical notes' are included wherever we had to refer to more advanced topics within these subjects. We therefore hope that this book will be useful to graduate students, researchers and professional engineers alike.

While numerous colleagues have assisted us in the writing of this volume, a few names stand out in particular. We would like to thank Prasanth Nair and David Toal of the University of Southampton, Max Morris of Iowa State University, Donald Jones of the General Motors Co., Natalia Alexandrov of NASA, Tom Etheridge of Astrium, Lucian Tudose of the Technical University of Cluj Napoca, Danie G. Krige and Stephen J. Leary of BAE Systems for their suggestions and for reading various versions of this manuscript.

Finally, a disclaimer. Surrogate modelling is a vast subject and this text does not claim nor, indeed, can hope to cover it all. The selection of techniques we have chosen to include reflect, to some extent, our personal biases. In other words, this is the combination of tools that works for us and we earnestly hope that it will for the reader too.

Alexander Forrester, András Sóbester and Andy Keane
Southampton, UK

Disclaimer

About the Authors

Dr Alexander I. J. Forrester is Lecturer in Engineering Design at the University of Southampton. His main area of research focuses on improving the efficiency with which expensive analysis (particularly computational fluid dynamics) is used in design. His techniques have been applied to wing aerodynamics, satellite structures, sports equipment design and Formula One.

Dr András Sóbester is a Lecturer and EPSRC/Royal Academy of Engineering Research Fellow in the School of Engineering Sciences at the University of Southampton. His research interests include aircraft design, aerodynamic shape parameterization and optimization, as well as engineering design technology in general.

Professor Andy J. Keane currently holds the Chair of Computational Engineering at the University of Southampton. He leads the University's Computational Engineering and Design Research Group and directs the Rolls-Royce University Technology Centre for Computational Engineering. His interests lie primarily in the aerospace sciences, with a focus on the design of aerospace systems using computational methods. He has published over two hundred papers and three books in this area, many of which deal with surrogate modelling concepts.

Foreword

Over the last two decades, there has been an explosion in the ability of engineers to build finite-element models to simulate how a complex product will perform. In the automotive industry, for example, we can now simulate the injury level of passengers in a crash, the vibration and noise experienced when driving on different road surfaces, and the vehicle's life when subjected to repeated stressful conditions such as pot holes. Moreover, our ability to quickly modify these simulation models to reflect design changes has greatly increased. The net result is that the potential for using optimization to improve an engineering design is now higher than ever before.

One of the major obstacles to the use of optimization, however, is the long running time of the simulations (often overnight) and the lack of gradient information in some of the most complicated simulations (especially crashworthiness). Due to the long running times and the lack of analytic gradients, almost any optimization algorithm applied directly to the simulation will be slow.

Despite this slowness, one could still bite the bullet and invest one's computational budget in applying an optimization algorithm directly to the simulations. But this is unlikely to be satisfying, because rarely does a single optimization result settle any design issue. For example, if the result is *not* satisfactory, one may want to gain insight into what is going on by performing parameter sweeps and plotting input-output relationships. Or one might want to repeat the optimization with a modified formulation (different starting point, different constraints). All this, of course, requires doing more simulations. On the other hand, if the result *is* satisfactory, one still want might to do further investigations to see if a better tradeoff can be struck between competing objectives. Again, this requires more simulations. Clearly, if one uses up all the available resources solving the first optimization problem, all these follow-up studies would not be possible, or at least lead to missed deadlines.

The basic idea in the 'surrogate model' approach is to avoid the temptation to invest one's computational budget in answering the question at hand and, instead, invest in developing fast mathematical approximations to the long running computer codes. Given these approximations, many questions can be posed and answered, many graphs can be made, many tradeoffs explored, and many insights gained. One can then return to the long running computer code to test the ideas so generated and, if necessary, update the approximations and iterate.

While the basic idea of the surrogate model approach sounds simple, the devil is in the details. What points do you sample to use in building the approximation? What approximation method do you employ? How do you use the approximation to suggest new, improved designs? How do you use the approximations to explore tradeoffs between objectives? What do you do if your simulation has numerical noise in it? And, equally important: Where do I get the computer code to do all these things?

In *Engineering Design via Surrogate Modelling: A Practical Guide*, the authors answer all of these questions. They are like cooks giving you a recipe for an entire meal: appetizer, salad, entrée, wine, dessert, and coffee. It is not an isolated recipe for bread rolls such as you might find in the cooking section of the Sunday paper. The authors start at the very beginning, with variable screening to determine which variables to include in the study. One then learns how to develop a sampling plan for developing the initial approximations. Several approximation methods are then discussed, but the authors' preference for Kriging is clear. They then show how to use Kriging approximations to do unconstrained optimization, constrained optimization, and tradeoff studies. At each step, sample code is provided in *MATLAB*, which is also available in electronic form on an associated website.

No different than any cook, the authors have their biases: they like particular ways of sampling, particular ways to use Kriging for optimization, etc. To their credit, however, in several sections the authors go out of their way to mention other possible approaches and to provide references for you to follow up if you are interested.

In my view, the book can appeal to two audiences. For those experienced in the field of surrogate models, the book provides a glimpse at what the authors, as experienced practitioners, consider to be the state of the art. For those just beginning, the book provides a self-contained introduction to the field.

Like any cookbook, the book is a place to start, not to finish. I suspect that people reading this book will take some recipes as they are, will modify others to suit their taste, and will ignore still others in favor of their own recipes. But I am convinced that even the most experienced persons in the field will find new things that pique their interest (this was certainly true for myself). So, to all those beginning this book, may I say, *Bon Appetit*!

Donald R. Jones
General Motors Co.

Prologue

Engineering design is concerned with the making of decisions based on analysis, which directly impact the product or service being designed. To accomplish this, engineers typically engage in a great deal of analysis to understand the background to their decisions. It is often necessary for months of analysis by dedicated teams to be undertaken to inform key product decisions. It is against this backdrop that the current book has been written. For example, in modern aerospace design offices the computational power needed to support advanced decision making can be prodigious and, even with the latest and most powerful computers, designers still wish for greater understanding than can be gained by straightforward use of the familiar analysis tools, such as those coming from the fields of computational fluid dynamics or computational structural mechanics.

One way of gaining this desirable increased insight into the problems being studied is via the use of surrogate (or meta) models. Such models seek to provide answers in the gaps between the necessarily limited analysis runs that can be afforded with the available computing power. They can also be used to bridge between various levels of sophistication afforded by varying fidelity physics based simulation codes, or between predictions and experiments. Their role is to aid understanding and decision taking by wringing every last drop of information from the analysis and data sources available to the design team and making it available in a useful and powerful way. This book aims to discuss the application of such surrogate models using some of the most recent results stemming from the academic and industrial research communities. To place these ideas in context we begin with a (far from exhaustive) summary of where surrogate models typically find use in engineering design.

The simplest, and currently most common, use of surrogate models is to augment the results coming from a single, expensive simulation code that needs to be run for a range of possible inputs dictated by some design strategy (perhaps a planned series of runs or those suggested by some search process). The basic idea is for the surrogate to act as a ‘curve fit’ to the available data so that results may be predicted without recourse to use of the primary source (the expensive simulation code). The approach is based on the assumption that, once built, the surrogate will be many orders of magnitude faster than the primary source while still being usefully accurate when predicting away from known data points. Note that there are two key requirements here: (1) a significant speed increase in use and (2) useful accuracy. Clearly, these factors are often in tension with each other and the user will often have to balance these competing needs carefully.

Another increasingly common use for surrogates is to act as calibration mechanisms for predictive codes of limited accuracy. It is quite common when producing a software model of some physical process to have to simplify the approach taken so as to gain acceptable run times. For example, in computational fluid dynamics there are a whole raft of different simulation approaches that run from simple but very rapid potential flow solvers, through Euler codes to Reynolds averaged Navier–Stokes methods to large eddy simulations and on to direct numerical simulation of the full equations. A surrogate may well be trained to bridge between two such codes by being set up to represent the differences between a simple but somewhat inaccurate code and a more accurate but slower approach, the idea being to gain the accuracy of the expensive code without the full expense. Such 'multi-fidelity' or 'multi-level' approaches can be extended to dealing with data coming from physical experiments and their correlation with computational predictions – indeed, much early work in this field stems from long term agricultural experiments where data coming from crop trials had to be interpreted.

A third use of surrogate models is to deal with noisy or missing data. It is a commonplace experience that results coming from physical experiments are subject to small random errors. These need to be dealt with when the data are used, often by some process of averaging. It will also often occur in physical experimentation that some experiments fail to yield usable results at all. It is less well known that the results of computational codes also suffer from such problems, though in this case any noise is generally not random. Computational 'noise' stems from the schemes used to set up computational models, notably discretized and iterative approaches where solutions are not fully independent of the discretization or the number of iterations used. Similarly, most numerical schemes are rarely completely foolproof and will sometimes fail in unexpected ways. In these circumstances surrogate models can be used as filters and fillers to smooth data, revealing overall trends free of extraneous fine detail and spanning any gaps.

Finally, surrogate models may be used in a form of data mining where the aim is to gain insight into the functional relationships between variables open to the design team and results of interest. If appropriate methods are selected and applied to sets of data, surrogates can be used to demonstrate which variables have most impact and what the forms of such effects appear to be. This can allow engineers to focus on those quantities that have most importance and also to understand such quantities with greater clarity. Sometimes such understanding comes directly from the equations resulting from surrogate construction; alternatively surrogates may be used in visualization schemes to map and graph different projections of the data more rapidly than would be possible by repeated runs of the available analysis codes.

In all the above cases the basic steps of the surrogate modelling process remain essentially the same, and are illustrated in the flowchart in Figure P.1, where each stage is related to the chapter that describes it.

Firstly, some form of data set relating a series of inputs and outputs is obtained, typically by *sampling* the design decision space, making use of the available, and often expensive, analysis codes. In other words, a number of possible candidate designs are generated and analysed, using whatever computational or experimental means are at hand.

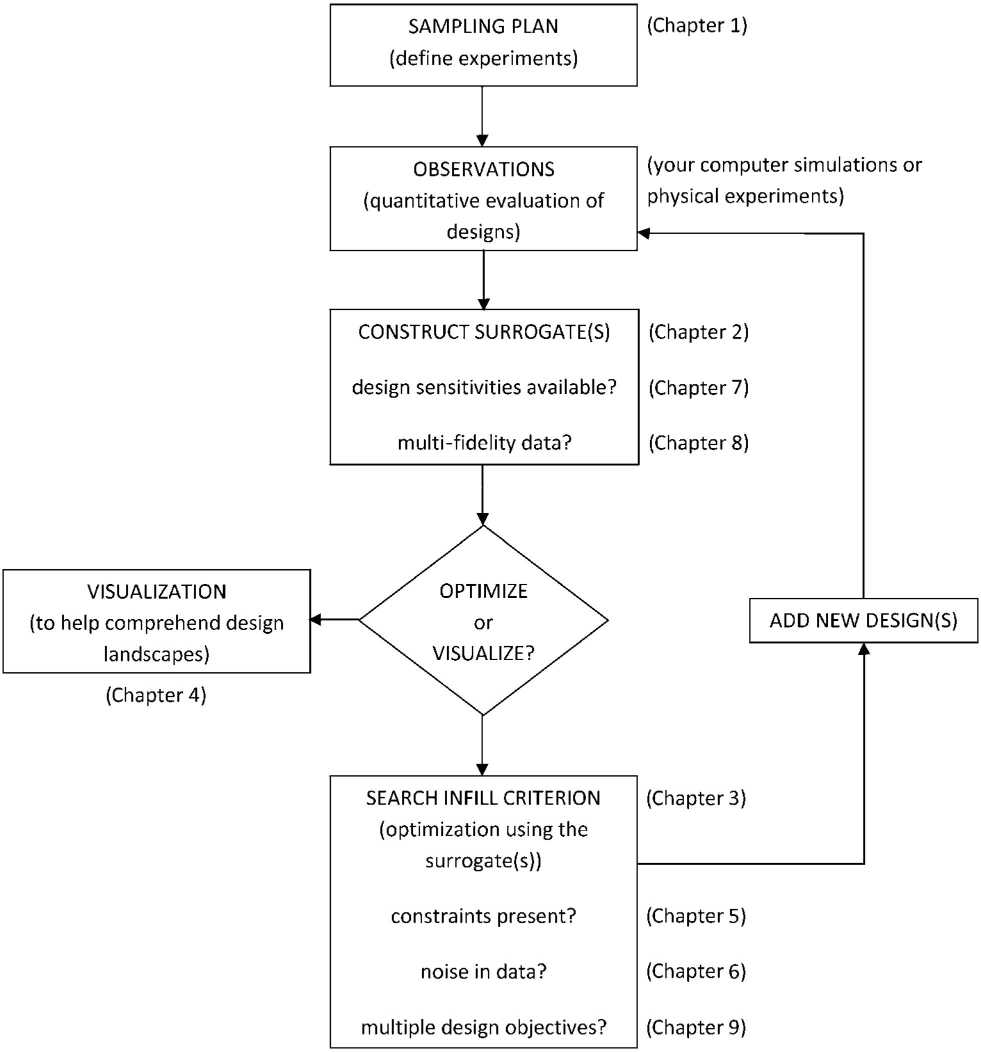

Figure P.1. The surrogate modelling process

Following this, a suitable surrogate model form must be selected and fitted to the available data – this process lies at the heart of this book. Its parameters must be estimated, it must be assessed for accuracy and a number of schemes can be used to do this. Note a key limitation of the surrogate approach at this point: if the problem being dealt with has many dimensions the number of points needed to give reasonably uniform coverage rises exponentially – the so-called *curse of dimensionality*. Currently the only way around this problem is either to limit the ranges of the variables so that the shape being modelled by the surrogate is sufficiently simple to be approximated from very sparse data or, alternatively, to freeze many of the design values at hopefully sensible values and work with just a few at a time, iterating around those being made active as the design process progresses (for example, in aircraft design, dealing with aerodynamic quantities at one stage with structural variables fixed and then swapping these around).

Since the initial design selections made to produce the first set of data will almost inevitably miss certain features of the landscape, the construction of a useful surrogate often requires further, judiciously selected calls to the analysis codes. These additional calls are termed *infill points* and the process of applying them is known as *updating*. The selection of new points is usually made either in areas where the surrogate is thought to be inaccurate or, alternatively, where the surrogate model suggests that particularly interesting combinations of design variables lie. The selection of such points is often made using an optimization-based search over the surrogate. The updating of the surrogate with infill points may be carried out a number of times until the surrogate is fit for purpose (or perhaps the available budget of computing effort has been exhausted).

Having constructed (and hopefully tested) a suitably accurate model, it is then finally exploited or explored in some fashion, perhaps being embedded in a modified solver or as a subject for use along with optimization or visualization tools. The processes of exploration, exploitation and updating may well be closely interlinked so that the surrogate remains usefully accurate as the design process evolves. Moreover, data coming from previous design processes may well also be melded into the system if appropriate.

It turns out that it is rare for a completely fixed approach to be appropriate in all cases of interest, since the data itself may well influence the directions taken. This will call for knowledge, care and experience from those constructing and using the surrogates – hopefully the following sections and chapters will help support this process. A good understanding of the capabilities and limitations of the various techniques presented will be the hallmark of the knowledgeable designer.

Part I

Fundamentals

1

Sampling Plans

Engineering design problems requiring the construction of a cheap-to-evaluate 'surrogate' model $\widehat{f}$ that emulates the expensive response of some black box f come in a variety of forms, but they can generally be distilled down to the following template.

Here $f(\mathbf{x})$ is some continuous quality, cost or performance metric of a product or process defined by a k-vector of design variables $\mathbf{x} \in D \subset \mathbb{R}^k$. In what follows we shall refer to D as the *design space* or *design domain*. Beyond the assumption of continuity, the only insight we can gain into f is through discrete *observations* or *samples* $\left\{\mathbf{x}^{(i)} \rightarrow y^{(i)} = f(\mathbf{x}^{(i)}) | i = 1, \ldots, n\right\}$. These are expensive to obtain and therefore must be used sparingly. The task is to use this sparse set of samples to construct an approximation $\widehat{f}$, which can then be used to make a cheap performance prediction for any design $\mathbf{x} \in D$.

Much of this book is made up of recipes for constructing $\widehat{f}$, given a set of samples. Excepting a few pathological cases, the mathematical formulations of these modelling approaches are well-posed, regardless of how the *sampling plan* $\mathbf{X} = \left\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\right\}$ determines the spatial arrangement of the observations we have built them upon. Some models do require a minimum number n of data points but, once we have passed this threshold, we can use them to build an unequivocally defined surrogate.

However, a well-posed model does not necessarily *generalize* well, that is it may still be poor at predicting unseen data, and this feature *does* depend on the sampling plan $\mathbf{X}$. For example, measuring the performance of a design at the extreme values of its parameters may leave a great deal of interesting behaviour undiscovered, say, in the centre of the design space. Equally, spraying points liberally in certain parts of the inside of the domain, forcing the surrogate model to make far-reaching extrapolations elsewhere, may lead us to (false) global conclusions based on patchy, local knowledge of the objective landscape.

Of course, we do not always have a choice in the matter. We may be using data obtained by someone else for some other purpose or the available observations may come from a variety of external sources and we may not be able to add to them. The latter situation often occurs in conceptual design, where we wish to fit a model to performance data relating to existing, similar products. If the reader is only ever concerned with this type of modelling problem, he or she may skip the remainder of this chapter. However, if you have the possibility of

Engineering Design via Surrogate Modelling: A Practical Guide A. I. J. Forrester, A. Sóbester and A. J. Keane