

Sequential DoE Framework for Steady State Model Based Calibration

Mohammed Reza Kianifar and Loan Felician Campean
University of Bradford

Dave Richardson
Jaguar Cars Ltd.

ABSTRACT

The complexity of powertrain calibration has increased significantly with the development and introduction of new technologies to improve fuel economy and performance while meeting increasingly stringent emissions legislation with given time and cost constraints. This paper presents research to improve the model based engine calibration optimisation using an integrated sequential Design of Experiments (DoE) strategy for engine mapping experiments. This DoE strategy is based on a coherent framework for a model building - model validation sequence underpinned by Optimal Latin Hypercube (OLH) space filling DoEs. The paper describes the algorithm development and implementation for generating the OLH space filling DoEs based on a Permutation Genetic Algorithm (PermGA), subsequently modified to support optimal infill strategies for the model building - model validation sequence and to deal with constrained non-orthogonal variables space.

The development, implementation and validation of the proposed strategy is discussed in conjunction with a case study of a GDI engine steady state mapping, focused on the development of an optimal calibration for CO₂ and particulate number (Pn) emissions. The proposed DoE framework applied to the GDI engine mapping task combines a screening space filling DoE with a flexible sequence of model building - model validation mapping DoEs, all based on optimal DoE test plan augmentation using space filling criteria. The case study results show that the sequential DoE strategy offers a flexible way of carrying out the engine mapping experiments, maximising the information gained and ensuring that a satisfactory quality model is achieved.

CITATION: Kianifar, M., Campean, L. and Richardson, D., "Sequential DoE Framework for Steady State Model Based Calibration," *SAE Int. J. Engines* 6(2):2013, doi:10.4271/2013-01-0972.

INTRODUCTION

The development of engine technologies to improve the performance, fuel economy and drivability while meeting increasingly stringent emissions legislation has resulted in an increased complexity of powertrain calibration with significant time and cost implications. One of the main challenges in the calibration process is to find the best combination of engine parameters to compromise between the conflicting engine calibration requirements, such as emissions and fuel consumption. To address this calibration challenge with a satisfactory expenditure of costs and time, strategies such as model based calibration have been introduced [1, 2].

This paper focuses on model based steady state calibration which is underpinned by statistical modelling of

the engine responses of interest using test data collected at fixed engine speed / load operating points. Design of Experiments (DoE) methods have been widely utilised to enhance the effectiveness of the steady state testing [1], by delivering engine response models of sufficient fidelity and accuracy to support the development of a steady state calibration through mathematical optimisation.

In simple terms the main purpose of the DoE methods is to collect the maximum possible information with the least measurement effort [2]. DoE methods commonly used for steady state engine mapping experiments include Optimal DoEs (e.g. D-Optimal and V-Optimal) and space filling DoEs [3,4,5,6,7,8]. The space filling DoEs, particularly Optimal Latin Hypercube (OLH) designs, have been increasingly used given that they enable more flexibility to characterise the engine behaviour over a wider design space (e.g. "global"

models, over the whole engine speed-load space), and with no prior knowledge required regarding the type of model that would be adequate to represent the engine response trends [6,7,8,9].

In order to satisfy the required model accuracy, conventional DoE strategies for steady state engine mapping typically use in excess of 100 test points (often as many as 150) at each engine speed/load point selected for mapping, with further 10-20 test points collected for model validation [2, 7]. This represents a significant testing effort, and does not necessarily guarantee that an acceptable model is achieved for each engine response of interest.

Sequential DoE strategies, usually involving a Model Building - Model Validation (MB-MV) sequence underpinned by a metric of DoE optimality, have been researched and developed to minimise the testing effort. Such methods have been successfully introduced in other engineering fields, such as electronics, chemistry and aerodynamics [10,11,12,13,14,15,16,17]. Sequential DoE strategies provide an opportunity to check the model accuracy at each iteration to terminate the data collection process when the target accuracy is achieved.

The research idea behind this paper is that sequential DoE strategies could lead to efficiency gains compared to conventional steady state engine mapping testing (based on a fixed, usually large DoE test plan), thus delivering potential savings from smaller test plans while ensuring that the desired model quality is reached.

The aim of this paper is to present the development, implementation and validation of a sequential DoE strategy based on optimal space filling designs (OLH DoEs) for steady state engine testing. The paper starts with a review of sequential DoE strategies available in the literature, and a discussion of the principle of the suggested approach for steady state engine testing based on the optimal sequential augmentation of an initial OLH DoE within the MB-MV iteration. The algorithm development and software implementation of this sequential DoE strategy, based on Permutation GA algorithms, is outlined next. The paper then describes the application of the sequential DoE strategy to a series of mapping experiments for a GDI engine, covering screening and mapping experiments through Model Building - Model Validation (MB-MV) iterations, outlining the specific challenges of planning engine experiments, which prompted further refinements to the framework and algorithms. The paper ends with a comprehensive discussion of the results and directions for further work.

SEQUENTIAL DOE FRAMEWORK

Literature Background

In general, sequential DoEs can be divided into two main categories:

1. Optimal sequential design: In an optimal sequential DoE, the model type and its parameters are known in advance (e.g. polynomials). This allows the algorithms to use the

behaviour of the chosen model type to guide the sampling points into the right direction within design space. For example, the D-optimal designs minimise the covariance of the model parameters estimates [11, 12]. The main issue with these DoEs is that if the assumed model type is not suitable for the response, the DoE plan is not efficient and the enhancement in model accuracy through collecting more data is not guaranteed.

2. Evolutionary sequential design: Generic sequential DoEs make no assumptions about the model type, number of sample points or system behaviour. In evolutionary sequential DoEs the information from the previous iterations is used to decide where to select the next test points [10]. This approach is potentially useful for many practical engineering problems (including engine testing and development) where the type of model is not known in advance and experimentation is expensive.

The evolutionary sequential DoEs can be further classified into:

i. Exploitation-based sequential design [13,14,15]: Exploitation-based DoEs use an error measure from the previous steps to guide the sampling points to the specific regions of design space, e.g. areas with discontinuous system behaviour or areas containing optima. The main problem with exploitation-based DoEs is their tendency to over-focus on specific areas, which could leave regions of the design space undersampled.

ii. Exploration-based sequential design methods [16-17]: Exploration-based sequential DoEs, give equal importance to all regions of design space and aim to fill it up as evenly as possible at each sequence. In this method, the location of the test points from the previous iteration is used as feedback for sampling new test points, ensuring that not too many or too few samples are collected from the same regions of the design space. These DoEs are not specifically linked with any response models and aim to distribute the points evenly through the design space.

Considering the fact that the engine calibration is a complex engineering problem with many uncertainties, an optimal sequential DoE will not always provide a useful option, given that knowledge or specification of the model type is not known in advance. This is commonly the case when new calibration variables are introduced (e.g. following introduction of new technology or ECU strategies, such as multiple injections for a GDI engine) or new engine responses are required to be modelled (e.g. particulate numbers, Pn, to deal with new emissions legislation). Also, an exploitation based sequential DoE may result in early dismissal of potential calibration solutions given that some parts of the high dimensional design space could be left unexplored.

The only attempt to apply sequential DoEs for steady state engine mapping and calibration, known to the authors, is based on the Sobol Sequence method [18, 19]. The problem

with using the Sobol Sequence method in an iterative process is that it is not adaptive with respect to ‘learning’ from previous stages.

In this research we have adopted an exploration based sequential design strategy, based on an optimal DoE augmentation using space filling criteria in conjunction with OLH DoEs. The principle of the approach and its implementation will be described in the following sections.

OLH Based MB-MV DoE Sequence for Engine Mapping Experiments

The approach adopted in this research is to use OLH space filling DoEs as the basis for successive iterations of Model Building (MB) - Model Validation (MV) test plans. The OLH principle is to generate a DoE based on the Latin squares principle that minimises a chosen metric for space fillingness (such as the maximum or minimum distance between test points). In order to generate an optimal sequence of DoEs for successive MB-MV sequences, it is proposed that the space filling criteria is maintained throughout the MB-MV iteration. What this means is that given an MB generated as an OLH DoE, an MV DoE is generated as an OLH for which the optimality criterion is the space filling metric for the overall DoE sequence, i.e. including all MB and MV DoE test points.

This strategy is significantly different from other OLH based MB-MV strategies, such as that described in [20], in that the joint MB-MV DoE is no longer an OLH. While each of the individual DoEs (i.e. the initial MB DoE and each of the subsequent MV DoEs in the sequence) satisfy the Latin hypercube requirements when taken in isolation (e.g. based on n levels for the MB DoE, and m levels for any MV DoE), the joint DoE created by the merger of the MB and MV test points does not strictly fulfil the Latin hypercube principle because it will be based on $n+m$ interlaced and in general unevenly distributed levels. The optimality criterion for the MV DoE is to minimise the space filling metric defined for the joint MB-MV sequence. An important advantage of the strategy adopted in this research is that it provides flexibility with respect to the number of iterations required for the sequence, which can be arbitrarily terminated when the response models are deemed satisfactory by the analyst.

This proposed strategy provides a good fit with the practical requirements of the steady state engine testing problem. Thus, according to the proposed framework, a smaller model building (MB) OLH DoE experiment can be planned (e.g. using < 50 test points) and executed, followed by a validation (MV) DoE experiment, e.g. of 15 test points. The MV is also an OLH (defined over 15 levels), but the optimality criterion is to minimise the space fillingness metric across the merged MB + MV set ($50 + 15 = 65$ test points). Engine response models are fitted for the engine test data collected from the MB experiments, typically using non-parametric or semi-parametric models (such as Kriging or Radial Basis Function (RBF) [21]), and the quality of the

models is evaluated via the prediction error for the validation set of test data (e.g. using the Root Mean Squared Error for the prediction residuals, RMSE) [20]. If the model accuracy requirements are not met for any of the engine responses of interest, a further validation DoE (MV2) is planned using the same principle of an OLH MV DoE in which the space fillingness metric is minimised across the MB + MV + MV2 set. A new model is fitted using the initial MB + MV set, and validated against the new MV set (MV2), based on validation RMSE for prediction residuals. This process is repeated iteratively until the model accuracy requirement is met. The principle of the approach is illustrated in the flowchart presented in Figure 1.

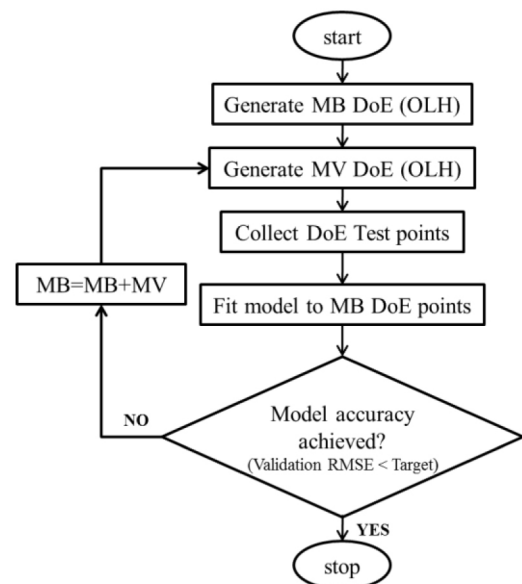


Figure 1. MB-MV Strategy Flowchart

IMPLEMENTATION

The proposed sequential DoE strategy was implemented in two stages: firstly, an algorithm was developed to generate the initial MB OLH DoE using a Permutation Genetic Algorithm (PermGA); secondly, an algorithm to generate the subsequent MV OLH DoEs was developed by modifying the MB OLH PermGA algorithm. Both of these algorithms will be described in the following sections.

PermGA Algorithm to Generate OLH DoEs

The generation of an OLH DoE with an optimal distribution of a given number of test points in the design space can be regarded as a discrete optimisation problem [22]. The main challenges with such optimisation problems are: (i) formulation of the optimisation objective function; and (ii) development of an effective evolutionary algorithm for the discrete optimization problem.

Optimisation objective function

As discussed, both the model building and model validation DoEs are individual OLH designs, i.e. satisfying the condition of one point per level [8] and generated based on optimizing a space filling or uniformity metric. Several uniformity metrics have been described in literature [22-23]; the metric chosen for this implementation was based on the Audze-Eglaiss (AELH) potential energy concept, which was shown to have good ability to distribute the points uniformly within the design space regardless of the sample size [22, 24]. The idea of the AELH function is based on the magnitude of the repulsive forces which is inversely proportional to the square of the distance between the mass units in the system, which can be assumed to be the points in the DoE [20]. Thus, the AELH objective function can be written as follows [23,24,25]:

$$U = \sum_{p=1}^n \sum_{q=p+1}^n \frac{1}{L_{pq}^2} \quad (1)$$

where U is the potential energy, L_{pq} is the Euclidian distance between any 2 points p and q , $p \neq q$, and n being the number of levels (points) in the OLH DoE. Minimizing the potential energy ensures uniform distribution of the sample points within the design space.

Optimisation algorithm

Several global optimisation algorithms have been utilized for generating OLH designs [22,23,24,25,26]; however, previous research has shown that a permutation Genetic Algorithm (PermGA) is more efficient for high dimensional OLH DoE problems due to a better convergence rate corresponding to the varying number of variables and levels [22, 24]. In other words, PermGA can generate better distributed points for high dimensional DoEs while reducing the computational costs. Given that engine calibration problems have an increasing dimensionality, a PermGA was developed and implemented for the generation of the OLHs. Figure 2 shows a flow diagram of the PermGA algorithm implemented in the MATLAB environment.

The main differences between the standard GA and a PermGA consist of (i) the way the initial population is generated, and (ii) the way the genetic operators - crossover and mutation, are formulated [22, 26]. While the generation of the initial population can simply consist of random Latin hypercubes (RLH), the PermGA operators such as crossover and mutation, are used in the OLH DoE algorithm to promote the overall uniformity and stratified space-filling properties. Three different crossover functions can be applied for PermGA: simple crossover, cycle crossover and inversion. Bates [22] has discussed that employing a combination of crossover functions, i.e. cycle crossover followed by inversion, performs considerably better. This strategy was adopted for the Matlab PermGA algorithm developed. An elitist strategy was also implemented, i.e. "elite" individuals with the highest fitness in a population are identified during

each generation and "preserved". This strategy ensures that the best individuals are not lost during subsequent generations and the maximum value of fitness will not decrease during iterations.

PermGA for OLH Design Augmentation

The strategy for the optimal augmentation of the MB OLH DoE, generating optimal "infill" points for the MV set, will use the same uniformity metric, i.e. the AELH function, and the same PermGA algorithm. The main challenge is to formulate the objective function to take into account the position of the MB points already fixed in the design space.

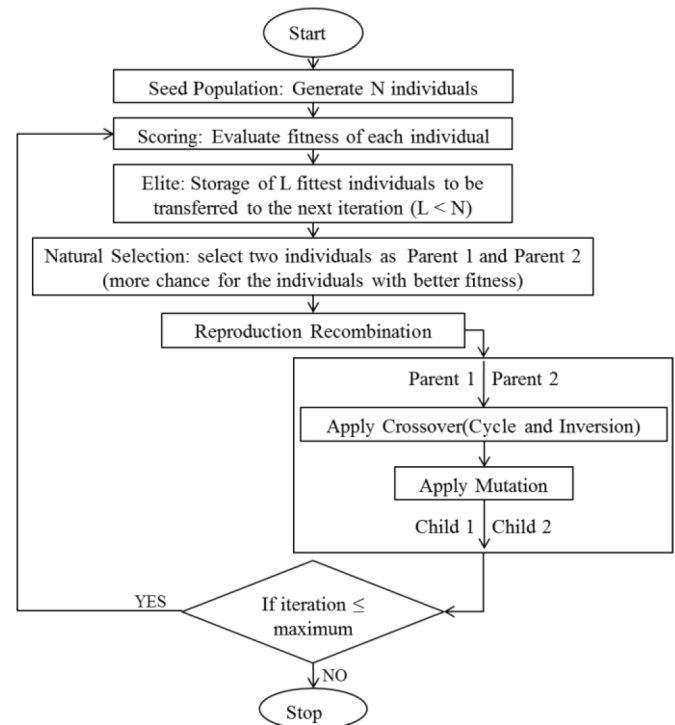


Figure 2. PermGA Flowchart

This means that the fitness function should include both the m new test points for the MV DoE, and the existing n test points from the MB OLH DoE. A modified fitness function was developed accordingly, as shown in equation 2.

$$U = \left(\sum_{p=1}^m \sum_{q=p+1}^m \frac{1}{L_{pq}^2} + \sum_{p=1}^m \sum_{n=1}^n \frac{1}{L_{pn}^2} \right) \quad (2)$$

where U is the overall potential energy, L_{pq} is the distance between any two new points p and q where $p \neq q$ in the MV DoE set, and L_{pn} is the distance between each new point p and the existing n points in the MB DoE set.

The outcome of a MB-MV DoE iteration is an OLH DoE with an optimal uniform distribution of points across the design space, i.e. with the new MV points optimally filling the under-sampled areas in the original design. Figure 3 illustrates a convergence plot for the PermGA algorithm for