# Crashworthiness Optimization in LS-OPT: Case Studies in Metamodeling and Random Search Techniques

Nielen Stander* Willem Roux* Mathias Giger**
Marcus Redhe** Nely Fedorova† Johan Haarhoff‡

*Livermore Software Technology Corporation, Livermore, CA
**Division of Mechanics, University of Linköping, Linköping, Sweden
†Snezhinsk Institute of Physics and Technology, Snezhinsk, Russia
‡Department of Mechanical Engineering, University of Pretoria, Pretoria, South Africa

Keywords: Metamodeling, Response Surface Methodology, Neural Networks, Kriging, Stochastic Search

***Abstract***

This crashworthiness optimization study compares the use of three metamodeling techniques while using a sequential random search method as a control procedure. The three methods applied are (i) the original Successive Linear Response Surface Method, (ii) the Neural Network method and (iii) the Kriging method. It is shown that, although NN and Kriging seem to require a larger number of initial points, the three metamodeling methods have comparable efficiency. The random search method is surprisingly efficient in some instances, but by nature much less predictable.

## 1. Introduction

Version 2.1 of LS-OPT [1] features a number of new methods applicable to design modeling and optimization. Techniques have been added to introduce more accurate and flexible metamodeling tools as well as the possibility of conducting exploration and optimization using a search method, i.e. without resorting to the construction of response surfaces. Although the addition of new methods is generally beneficial, it may also complicate the user's decision on which tools to use.

Metamodeling techniques are necessary in design approximation when the simulation runs for the physical modeling are extremely expensive. These techniques allow exploratory techniques such as optimization, variable screening, tradeoff studies and reliability and robustness assessment to be conducted using surrogate design information. Several techniques are available in LS-OPT, namely the response surface method (RSM), Artificial Neural Networks (NN) and Kriging. Each has its advantages, pitfalls and idiosyncrasies. The purpose of this study is to provide guidelines for using the various methods. The present study only focuses on the optimization of nonlinear dynamic problems in crashworthiness design using LS-DYNA and compares the different methods for efficiency and accuracy.

As part of the study, a sequential random search method is used as a control procedure so that response surface optimization and search results can be compared. In this evaluation, it should be taken into account that the metamodeling techniques, when sufficiently accurate, have the additional advantage that they can be used as surrogate design models for design exploration such as reliability or tradeoff studies. On the other hand, these techniques are less reliable when extremely noisy designs occur, or if there is a high degree of nonlinearity, especially instability. Such behavior, e.g. the display of major buckling modes, may be the result of poor conceptual design rather than being inherent to crash behavior. There is some evidence of chaotic behavior in the examples presented.

The examples shown encompass crashworthiness of a full vehicle as well as head and knee impact. The study concludes with some remarks and guidelines.

## 2. Metamodeling techniques

Metamodeling techniques allow the construction of surrogate design models for the purpose of design exploration such as variable screening, optimization and reliability. LS-OPT provides the choice of three types of metamodeling techniques, namely response surfaces, Neural Networks (NN's) and Kriging. All these approaches can be useful to provide a predictive capability for optimization or reliability. In addition, linear polynomials, although perhaps less accurate, are highly suitable for variable screening. At the core, these techniques differ in the regression methods that they employ to construct the surrogate models. The polynomial response surface method uses linear regression, while neural networks use nonlinear regression methods requiring optimization. Kriging is a Gaussian Process that uses Bayesian regression, also requiring optimization.

When using polynomials, the user is faced with the choice of deciding which monomial terms to include. The polynomial model then determines the number of simulations required. Because the typical choice is either linear or quadratic, there may be a large difference in the number of simulations required in either case, restricting the flexibility for large $n$. In addition, polynomials, by way of their nature as Taylor series approximations, are not natural for the creation of updateable surfaces. This means that if an existing set of point data is augmented by a number of new points which have been selected in a local subregion (e.g. in the vicinity of a predicted optimum), better information could be gained from a more flexible type of approximation that will keep global validity while allowing global and local refinement. Such an approximation could provide a more natural approach for combining the results of successive iterations. Potential candidates are NN's, Kriging and Space Mapping.

### 2.1 Polynomial response surface methodology (RSM)

This technique is well known [2] and commonplace in today's design environment. The method consists of experimental design and regression using polynomials. LS-OPT features a successive response surface technique (SRSM [3]) that is iterative in nature and constructs a new response surface for each iteration. Because of cost, linear response surfaces are commonly used.
To select appropriate sampling points, the $D$-Optimality criterion was chosen using the default number of $1.5(n+1)+1$ points per experimental design [1].

### 2.2 Kriging

This method is named after D.G. Krige, who applied empirical methods for determining true ore grade distributions from distributions based on sampled ore grades. In recent years, the Kriging method has found wider application as a spatial prediction method in engineering design. A detailed mathematical formulation of Kriging is given by Simpson [4]. The basic postulate of this formulation is:

$$y(\mathbf{x}) = f(\mathbf{x}) + Z(\mathbf{x})$$

where $y$ is the unknown function of interest, $f(\mathbf{x})$ is a known polynomial and $Z(\mathbf{x})$ the stochastic component with mean zero and covariance:

$$Cov[Z(\mathbf{x}^i),Z(\mathbf{x}^j)] = \sigma^2\mathbf{R}([R(\mathbf{x}^i,\mathbf{x}^j)]).$$

With $L$ the number of sampling points, $\mathbf{R}$ is the $L$ x $L$ correlation matrix with $R(\mathbf{x}^i,\mathbf{x}^j)$ the correlation function between data points $\mathbf{x}^i$ and $\mathbf{x}^j$. $\mathbf{R}$ is symmetric positive definite with unit diagonal. The correlation function used in this study is Gaussian:

$$R(\Theta)=\prod_{k=1}^{n} e^{-\Theta_k d_k^2}$$

where $n$ is the number of variables and $d_k = x_k^i - x_k^j$, the distance between the $k^{th}$ components of points $\mathbf{x}^i$ and $\mathbf{x}^j$. There are $n$ unknown $\theta$-values to be determined. Once the correlation function has been selected, the predicted esitimate of the response $y(\mathbf{x})$ is given by:

$$y = \hat{\beta} + \mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1}(\mathbf{y}-\mathbf{f}\hat{\beta})$$

where $\mathbf{r}^T(\mathbf{x})$ is the correlation vector (length $L$) between a prediction point $\mathbf{x}$ and the $L$ sampling points, $\mathbf{y}$ represents the responses at the $L$ points and $\mathbf{f}$ is an $L$-vector of ones (in the case that $f(\mathbf{x})$ is taken as a constant). The vector $\mathbf{r}$ and scalar $\hat{\beta}$ are given by:

$$\mathbf{r}^T(\mathbf{x}) = [R(\mathbf{x},\mathbf{x}^1),R(\mathbf{x},\mathbf{x}^2),\ldots,R(\mathbf{x},\mathbf{x}^L)]^T, \qquad \hat{\beta} = (\mathbf{f}^T\mathbf{R}^{-1}\mathbf{f})^{-1}\mathbf{f}^T\mathbf{R}^{-1}\mathbf{y}.$$

The estimate of variance from the underlying global model is:

$$\hat{\sigma}^2 = \frac{(\mathbf{y}-\mathbf{f}\hat{\beta})^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{f}\hat{\beta})}{L} \quad .$$

The maximum likelihood estimates for $\Theta_k$, $k$ = 1,2,…,$n$ can be found by solving the following constrained maximization problem:

$$\text{Max } \Phi(\mathbf{T}) = \frac{-[L\ln(\hat{\sigma}^2)+\ln|\mathbf{R}|]}{2}, \text{ subject to } \mathbf{T}>0.$$

where both $\hat{\sigma}$ and |**R**| are functions of $\mathbf{T}$. This optimization problem is solved using the LFOPC algorithm [1] of LS-OPT in multi-start fashion. **R** is adaptively regularized because of potential ill-conditioning. The net effect is that the approximating functions no longer interpolate the observed response values exactly, but still closely approximate the observations.

### 2.3 Feedforward Neural Networks

Neural methods are natural extensions and generalizations of regression methods [5]. Like RSM, they model relationships between a set of input variables and an outcome. They can be thought of as computing devices consisting of numerical units (*neurons*), whose inputs and outputs are linked according to specific topologies. A neural model is defined by its free parameters – the inter-neuron connection strengths (*weights*) and biases. These parameters are typically *learned* from the training data using an appropriate optimization algorithm. The training set consists of pairs of input (design) vectors and associated outputs (responses). The training algorithm tries to steer network parameters towards minimizing a distance measure, typically the mean squared error (MSE)

$$MSE=\sum_{i}^{L}(\hat{y}_i-y_i)^2/L$$

of the model computed on the training data. $L$ is the number of data points.

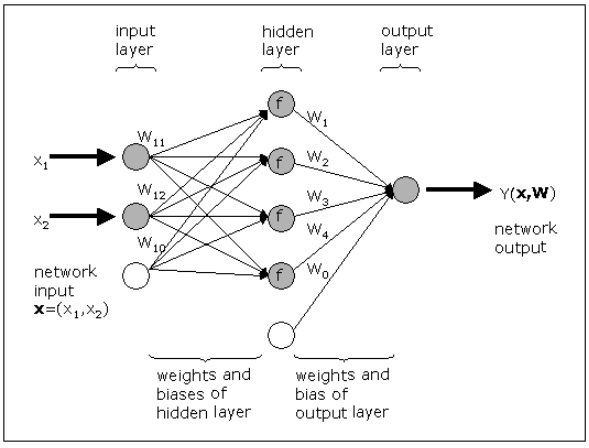



Figure 2-1: Schematic of a neural network with 2 inputs and a hidden layer of 4 neurons with activation function $f$

Feed-forward (FF) neural networks have a distinct layered topology. Each unit performs a biased weighted sum of their inputs and passes this value through a transfer (activation) function to produce the output. The outputs of each layer of neurons are the inputs to the next layer. In a feed-forward network, the activation function of intermediate ('hidden') layers is generally a sigmoid function, network input and output layers being linear. In the case of the examples presented, FF neural networks with this classic topology were used.

Neural networks have been mathematically shown to be universal approximations of continuous functions and their derivatives (on compact sets). In other words, when a network converges towards the underlying function, all the derivatives of the network converge towards the derivatives of this function.

Standard non-linear optimization techniques including a variety of gradient algorithms are applied to adjust the FF network's weights and biases. The second-order *Levenberg-Marquardt* algorithm appears to be the fastest method for training moderate-sized FF neural networks (up to several hundred adjustable weights). However, when training larger networks, the first-order RPROP algorithm becomes preferable for computational reasons.

Regularization may be done by controlling the number of network weights ('model selection'), by imposing penalties on the weights ('ridge regression'), or by various combinations of these strategies. Model selection requires choosing the number of hidden units and, sometimes, the number of network hidden layers. Most straightforward is to search for an 'optimal' network architecture that minimizes the cross-validation norms, e.g. using generalized cross validation (GCV). Often, it is feasible to loop over 1,2, ... hidden units and finally select the network with the smallest GCV error. In any event, in order for the GCV measure to be applicable, the number of training points $L$ should not be too small compared to the number of adjustable network parameters $M$.

To prevent over-fitting, it is always desirable to find neural solutions with the smallest number of parameters. In practice, however, networks with a very parsimonious number of weights are often hard to train. The addition of extra parameters (i.e. degrees of freedom) can aid convergence and decrease the chance of becoming stuck in local minima or on plateaus. Weight decay regularization involves modifying the performance function *F*, which is normally chosen to be the mean sum of squares of the network errors on the training set. When minimizing the mean squared error (MSE) the weight estimates tend to be exaggerated. A penalty for this tendency can be imposed by adding a term that consists of the sum of squares of the network weights.

$$F = \beta E_D + \alpha E_W ; \qquad E_D = \frac{\sum_{i=1}^{L} (\hat{y}_i - y_i)^2}{2}, \quad E_W = \frac{\sum_{m=1}^{M} W_m^2}{2},$$

where *M* is the number of weights in the neural network model.

### 2.4 Metamodel updating and experimental design

In the examples that follow, the Neural Nets and Kriging surfaces are updated (the default option in LS-OPT). That means that, in each iteration, the data points used to construct the metamodel are accumulated. For polynomials, a new experimental design and surface is constructed in each iteration thereby omitting data points from previous iterations. So far, more sophisticated updating techniques such as Space Mapping [6], a methodology based on Broyden updates of an existing surface, have not been used.

Although NN's and Kriging are similar in purpose, they differ in many respects. Kriging surfaces interpolate exactly at the data points, while NN's do not due to the use of cross validation methods for establishing the best predictive capability. Furthermore, Kriging surfaces can be more expensive to fit, depending on the number of points and the configuration of the points. Ill-conditioning is a problem in the reduction of the correlation matrix, especially when datapoints are near coincident. In LS-OPT the ill-conditioning is addressed by regularization and by using a Space Filling point selection scheme which maximizes the minimum distance between points using Simulated Annealing. The same Space Filling method is used for NN's.

## 3. Sequential Random Search using Latin Hypercube Sampling

The Sequential Random Search (SRS) method in LS-OPT relies only on Latin Hypercube Sampling (LHS) and simple heuristics to narrow the search [7]. In this paper, the performance of the continuous methods is measured against the SRS performance.

The technique uses the standard design formulation:

$$\text{Min. } f(x), \text{ subject to } g_j(x) < 0,\ j=1,\ldots,m$$

A sorting procedure is used to select the design with the lowest (for minimization) or highest (for maximization) objective from all the feasible designs. If no feasible design exists, the least infeasible design is chosen. An experimental design such as Latin Hypercube Sampling (LHS) allows a sequential random search procedure. LS-OPT automatically moves the region of interest by centering it on the most recent best design. The scheme also involves automatic subdomain reduction in which the subdomain is reduced by the zoom parameter $\eta$ (see LS-OPT User's Manual [1]) if the best design is the same as the baseline design [7]. Otherwise $\gamma_{\text{pan}}$ (typically = 1) is used as the panning range. All the variable ranges are reduced by the same amount.

The following example illustrates, what is perhaps typical, convergence performance of the methodology as a function of the number of variables. The example is an unconstrained minimization problem with starting point [1,1,1,…,1], solution [0,0,0,…,0] and an initial range of $[0.5;1.5]^n$ and the objective to minimize:

$$1/n \sum_{i=1}^{n} x_i^2$$

for $n$ = 20, 50 and 100. In Figure 3-1 SRSM is compared with the random search method for 20, 50 and 100 variable optimization problems. In this example SRSM uses the default number of simulations per iteration, namely 32, 77 and 152 respectively. *D*-optimal point selection is used. The random search uses 20 LHS simulations per iteration. As expected, the cost increases with $n$ for both SRSM and SRS. Note the logarithmic trends of the convergence for both methods. Each interval on the vertical axis represents an order or magnitude in accuracy. No attempt was made to optimize the number of simulations per iteration.

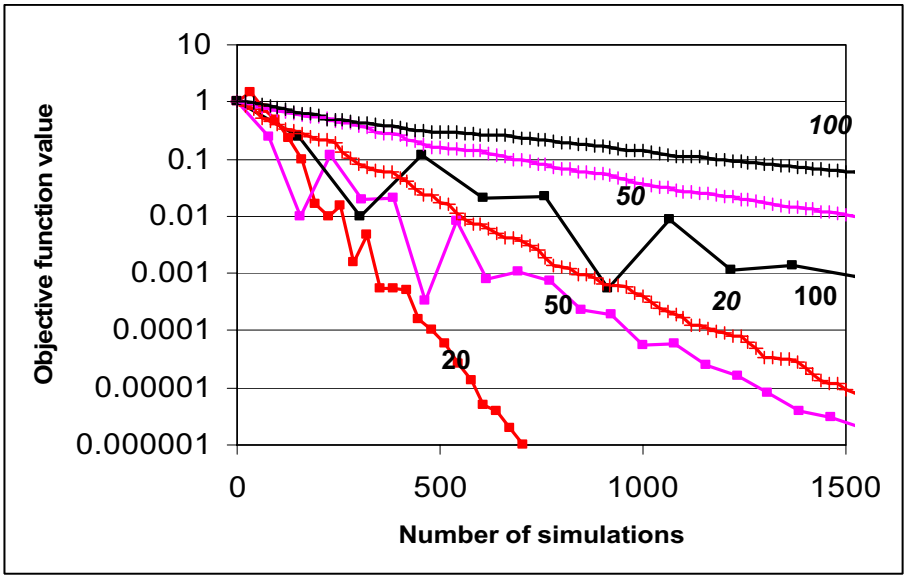



Figure 3-1: Minimization of a quadratic polynomial. Efficiency comparison of linear response surface (¦ ) and random search (+) methods for 20, 50 and 100 variables. Each point is an iteration.

## 4. Examples

As far as was possible, the examples were analyzed using what was thought to be the most efficient approach. In all the examples, the less important variables were first screened from the superset. This was done over one or two iterations. Further iterations were then conducted with the reduced set. Because a variable screening approach was not available for the search method, the search was started with the best optimum resulting from the variable screening runs. The number of simulations per iteration was chosen according to the recommended defaults for the surface and 19 runs per iteration for the search method. The actual number used can be observed on the plots.

### 4.1 Full vehicle MDO using Crashworthiness and Vibration Criteria

The crashworthiness simulation considers a model containing approximately 30,000 elements of a National Highway Transportation and Safety Association (NHTSA) vehicle undergoing a full frontal impact [8]. A modal analysis is performed on a so-called 'body-in-white' model containing approximately 18 000 elements. The crash model for the full vehicle is shown in Figure 4-1 for the deformed ($t$ = 78ms) states, and with only the structural components affected by the design variables, both in the undeformed and deformed (time = 72ms) states, in Figure 4-2. In the NVH (Noise, Vibration and Harshness – although only vibration is used) model, only body parts that are crucial to the vibration mode shapes are retained. The design variables are all thicknesses or gauges of structural components in the engine compartment of the vehicle parameterized directly in the LS-DYNA input file. LS-DYNA 970 is used for both the crash and NVH simulations, in explicit and implicit modes respectively.

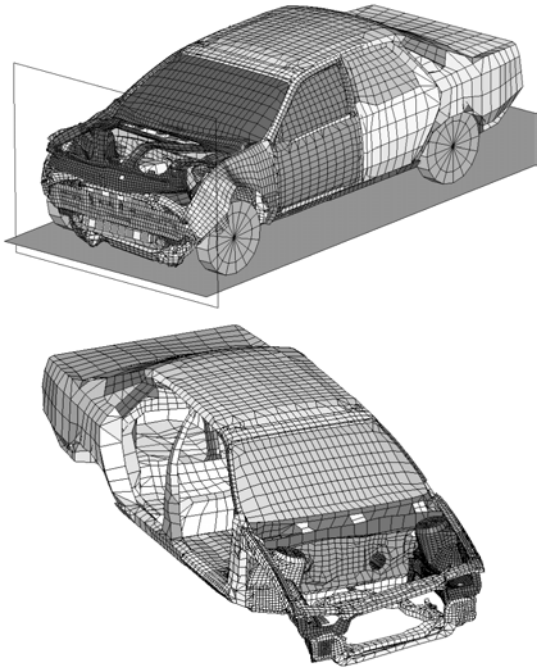


Figure 4-1: Crash (a) and Modal Analysis showing 1st torsional mode(b)

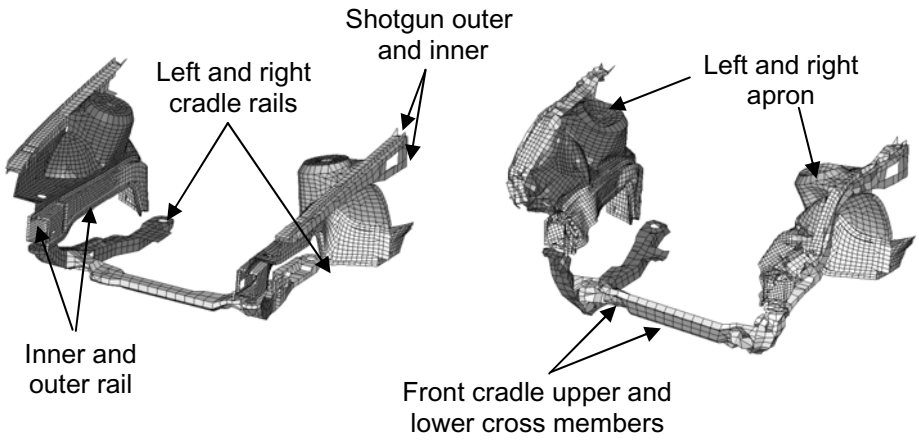



Figure 4-2: Structural components affected by variables –
Undeformed and (b) deformed (time = 72ms)

**Formulation**

Minimize Mass,

subject to

Maximum intrusion($\boldsymbol{x}_{crash}$) > 551.8mm
Stage 1 pulse($\boldsymbol{x}_{crash}$) > 14.34g
Stage 2 pulse($\boldsymbol{x}_{crash}$) > 17.57g
Stage 3 pulse($\boldsymbol{x}_{crash}$) > 20.76g
37.77Hz < Torsional mode frequency($\boldsymbol{x}_{NVH}$) < 39.77Hz *(Fully-shared variables)*
38.27Hz < Torsional mode frequency($\boldsymbol{x}_{NVH}$) < 39.27Hz *(Partially-shared variables)*

The starting parameters for optimization are shown in Table 4-1. Variable screening charts were used to select the variables. The full design space is used as starting range for all the variables and all the methods.

| Variable | Lower Bound | Baseline | Upper Bound | Crash | NVH |
|---|---|---|---|---|---|
| Cradle Rails | 1 | 3 | 3 | ? | ? |
| Cradle Crossmember | 1 | 3 | 3 | ? | |
| Shotgun (inner) | 1 | 2.5 | 2.5 | | ? |
| Shotgun (outer) | 1 | 2.5 | 2.5 | | ? |
| Rail (inner) | 1 | 3 | 3 | ? | |
| Rail (outer) | 1 | 3 | 3 | ? | |
| Aprons | 1 | 2.5 | 2.5 | ? | |

Table 4-1: Design parameters and variable participation for full vehicle

**Point selection.** The NN's and Kriging use 30 points per iteration, a number decided on by using the number of points required for a full quadratic approximation as a guideline. For RSM, 13 points were used for the first two iterations (variable screening) and 10/7 (Crash/NVH) per iteration beyond. 19 points were used for SRS.

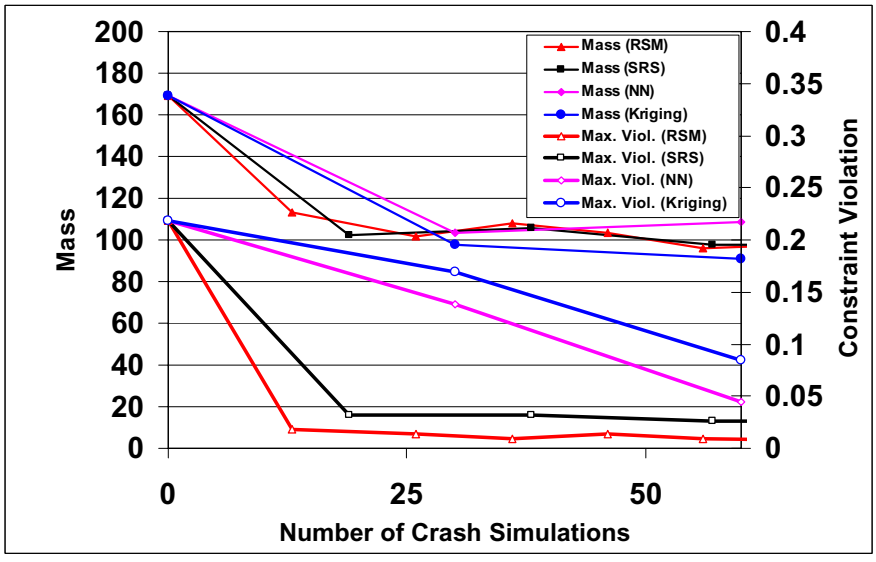
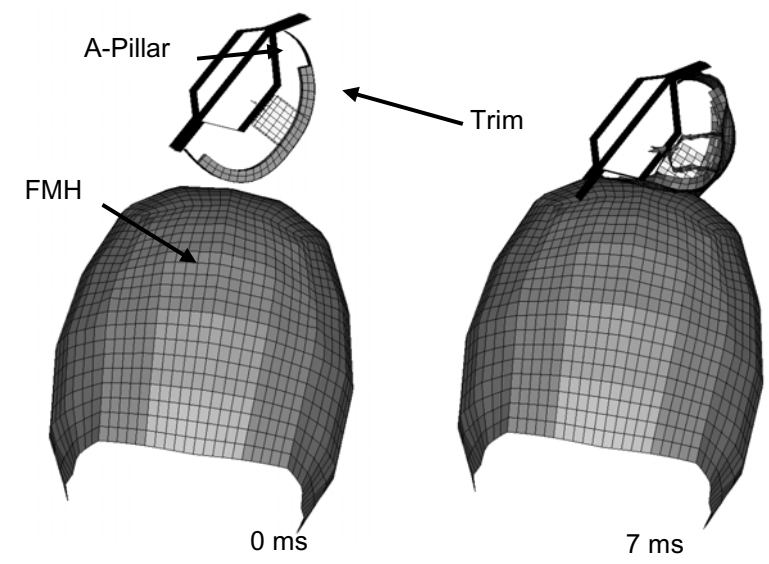


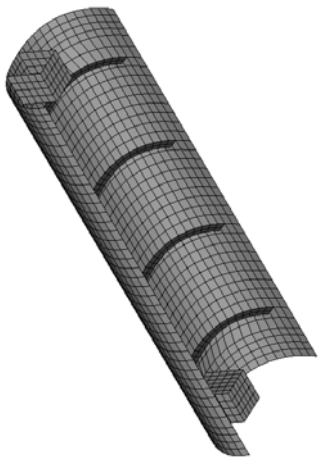

Figure 4-3: Comparison of metamodeling techniques and random search for full vehicle optimization

**Results.** Figure 4-3 shows the comparison of the methods for the Mass and maximum constraint violation. The optimum design does not become feasible for any of the methods, but the maximum violation is reduced considerably from 23% to 2%. The optimum design variables differ significantly for the different methods (not shown), indicating the presence of local minima. RSM performs the best, while the NN and Kriging methods have significantly slower convergence.

### 4.2 Head Impact

This unconstrained problem [1] is outlined in Figure 4-4. Shown is a Free Motion Headform (FMH) impacting the A-pillar of a vehicle covered on the interior with plastic trim. The aim of the optimization is to reduce the Head Injury Criterion, HIC-d = 166.4 + 0.75466*HIC15 as measured at the FMH's center of gravity) by modifying the trim design. The five design variables used are the trim thickness, rib height and thickness, number of ribs and rib span (distance between the first and last rib). The inclusion of the number of ribs as a design variable makes this an integer-based optimization problem addressed by choosing the nearest integer before running the preprocessor. Adaptive meshing is incorporated in the parameterization of the mesh through the TrueGrid [9] preprocessor to ensure good mesh quality for all possible designs.





**Variables**
- Trim thickness
- Rib height,
- Rib thickness,
- Number of ribs,
- Spacing between ribs

Figure 4-4 : Head impact: Design variables and trim deformation due to impact of FMH

| Variable | Lower Bound | Baseline | Upper Bound | Initial Range |
|---|---|---|---|---|
| Trim thickness | 2 | 2 | 3.5 | 1.5 |
| Rib thickness | 0.8 | 1 | 2 | 1 |
| Rib height | 5 | 6 | 20 | 10 |
| Number of ribs | 3 | 10 | 15 | 6 |
| Span | 130 | 180 | 180 | 40 |

Table 4-2: Starting design and design space parameters for head impact example

**Point selection.** The NN's and Kriging use 20 Space Filling points per iteration. This choice is frugal when compared to the vehicle example, simply made to see how this choice would affect the number of simulations. RSM and SRS use 10 *D*-Optimal and 19 LHS points respectively. The design space parameters are shown in Table 4-2.The full design space is used as the initial range for NN and Kriging.

**Results.** Figure 4-5 depicts the efficiency comparison, showing that all the methods perform similarly.

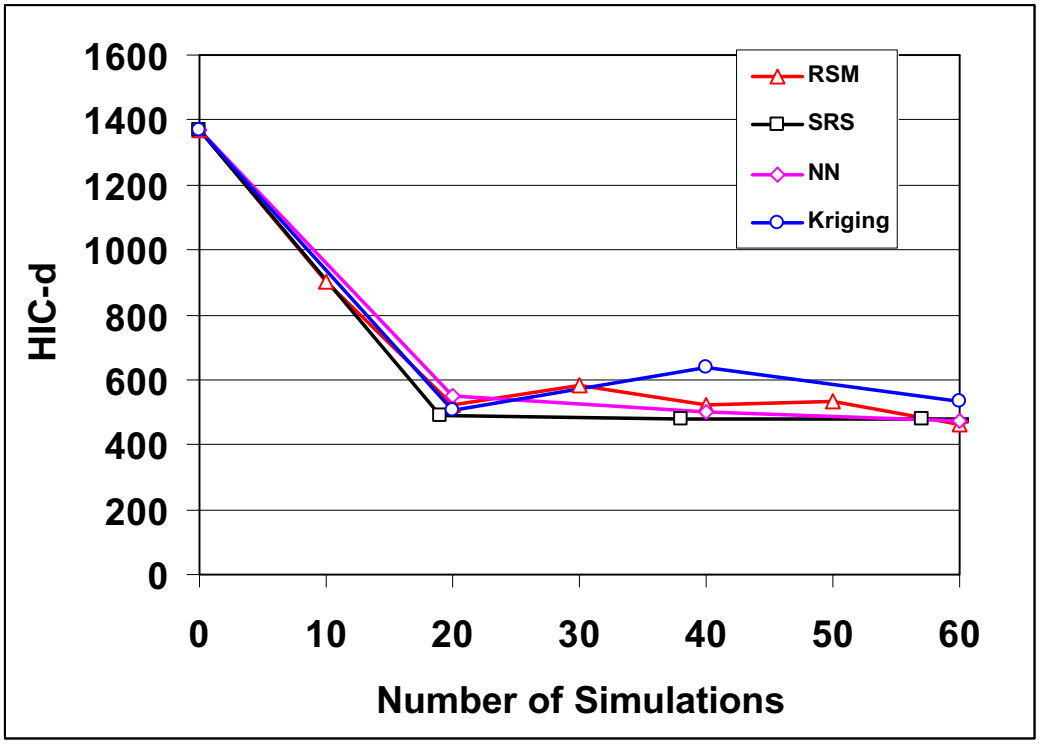



Figure 4-5: Comparison of metamodeling techniques and random search for head impact optimization

### 4.3 Knee Impact Problem

Figure 4-6 shows the finite element model of a typical automotive instrument panel (IP) [10]. For model simplification and reduced per-iteration computational times, only the driver's side of the IP is used in the analysis, and consists of around 25 000 shell elements. Symmetry boundary conditions are assumed at the centerline, and to simulate a bench component "Bendix" test, body attachments are assumed fixed in all 6 directions. Also shown in Figure 4-7 are simplified knee forms that move in a direction as determined from prior physical tests. As shown in the figure, this system is composed of a steel knee bolster that also serves as a steering column cover with a styled surface, and two steel energy absorption (EA) brackets attached to the cross vehicle IP structure. The brackets absorb a significant portion of the lower torso energy of the occupant by deforming appropriately. A steering column isolator (also known as a yoke) is used as part of the knee bolster system to delay the wrap-around of the knees around the steering column. The last three components are non-visible and hence their shape can be optimized. The 11 design variables are shown in Figure 4-7.

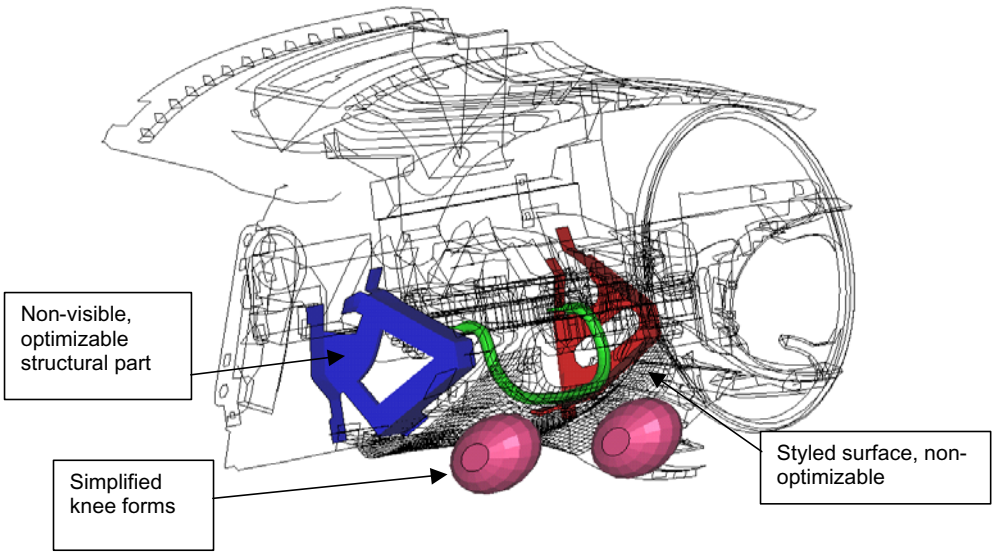



Figure 4-6: Typical instrument panel prepared for a "Bendix" component test

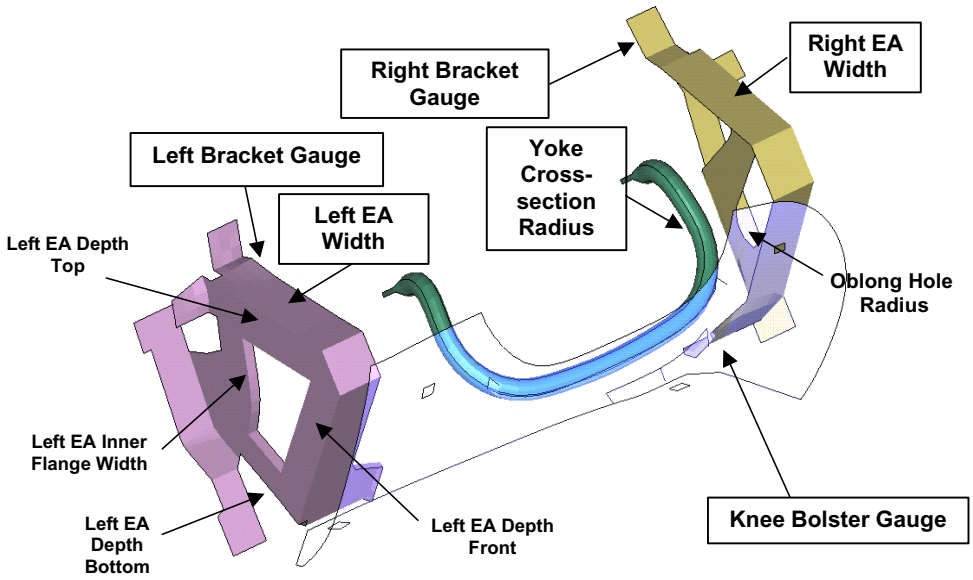



Figure 4-7: Typical major components of a knee bolster system and definition of design variables. The variables shown in bold were selected from an ANOVA screening process.

The simulation is carried out for a 40 ms duration by which time the knees have been brought to rest. The Bendix component test is used mainly for knee bolster system development; for certification purposes, a different physical test representative of the full vehicle is performed. Since the simulation used herein is at a subsystem level, the results reported here are used mainly for illustration purposes.

**Definition of optimization problem.** The optimization problem is formulated as follows:

Minimize Max. (Knee_$F_L$, Knee_$F_R$)
Subject to

| | | |
|---|---|---|
| Left Knee intrusion | < | 115mm |
| Right Knee intrusion | < | 115mm |
| Yoke displacement | < | 85mm |

The knee forces were normalized to 6500N. Minimization over both knee forces is achieved by constraining them to impossibly low values (in this case 0.5*6500N). The optimization algorithm will therefore always minimize the largest of the two knee forces, since it first attempts to minimize the constraint violation. The knee forces have been filtered SAE 60 Hz, to improve the approximation accuracy. The starting design and design space data are given in Table 4-3. The initial range was only applied to the RSM and SRS methods; otherwise the full design space was used.

| Variable | Lower Bound | Baseline | Upper Bound | Initial Range |
|---|---|---|---|---|
| Left Bracket Gauge | 0.7 | 1.1 | 3 | 2 |
| Left Flange Width | 20 | 32 | 50 | 10 |
| Right Bracket Gauge | 0.7 | 1.1 | 3 | 2 |
| Right Flange Width | 20 | 32 | 50 | 10 |
| Bolster Gauge | 1 | 3.5 | 6 | 3 |
| Yoke Radius | 2 | 4 | 8 | 2 |

Table 4-3: Starting design and design space parameters for Knee impact problem

**Point selection.** The NN's and Kriging use 30 Space Filling points per iteration. RSM and SRS use 11 *D*-Optimal and 19 LHS points respectively. Two separate trials were conducted using the SRS method.

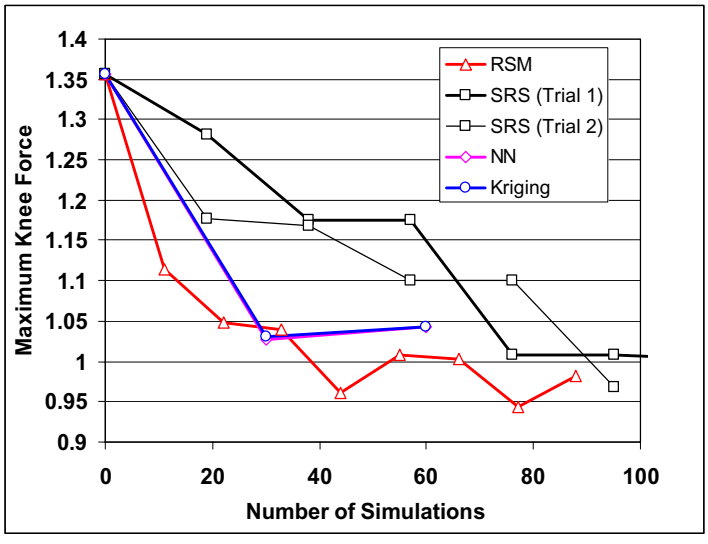



Figure 4-8: Optimization histories for knee impact example.

**Results.** In this case, the metamodels did significantly better than both trials of the SRS method which required about twice as many simulations for the same accuracy. The NN and Kriging methods give almost the same results.

## 5. Conclusions

1. Linear SRSM appears to still be the most reliable choice. However it has the deficiency of not providing global approximations after the analysis.
2. The SRS method performs surprisingly well with the relatively few design variables, but as shown in the knee impact example, can also give unexpected poor results. In this case, a second try confirmed that, although the design is always improved, the results could be somewhat erratic. This unpredictability is largely due to the relatively low sampling rate (~20) used. As shown, the total number of samples is also dependent on the number of design variables.
3. It is recommended that when using NN's or Kriging that the full design space be used as the initial range. The purpose of global approximations is to develop surrogate functions valid throughout the design space. If the starting range is smaller than the design space, patches of the design space may remain unsampled, resulting in extrapolation errors. The space filling point selection method ensures that, no matter where subsequent regions of interest may occur, new points will be placed in sparse areas. RSM should be applied in a small subregion, as always.

## 6. Remarks on ongoing testing

1. As applied in this study, NN's and Kriging are known to perform better with a larger number of points (chosen here roughly as the number of points that would be necessary for a quadratic polynomial fit, i.e. order $n^2$ points). However, it may be possible to start with order $n$ points across the design space, and to update the point selection from that basis using the standard heuristics.
2. For completeness, the quadratic polynomial RSM should be investigated.
3. It may be possible to refine the updating procedure by using Space Mapping [6].

## 7. Reference list


[1] Stander, N., Eggleston, T.A., Craig, K. and Roux, W.J. LS-OPT Version 2 Users Manual, Livermore Software Technology Corporation, Livermore CA, April 2003.

[2] Myers, R.H., Montgomery, D.C. Response Surface Methodology. Process and Product Optimization using Designed Experiments. Wiley, 1995.

[3] Stander, N., Craig, K.J. On the robustness of a simple domain reduction scheme for simulation-based optimization, *Engineering Computations*, 19(4), pp. 431-450, 2002.

[4] Simpson, T.W. A concept exploration method for product family design. Ph.D. Thesis, Georgia Institute of Technology, 1998.

[5] Bishop, C.M. Neural Networks for Pattern Recognition. Oxford University Press, 1995.

[6] Redhe, M. and Nilsson, L. Using Space Mapping and Surrogate Models to optimize vehicle crashworthiness design. Paper 2002-5607, 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, September 4-6, 2002.

[7] Giger, M. An Investigation of Structural Optimization in Crashworthiness Design Using a Stochastic Approach – A Comparison of Stochastic Optimization and Response Surface Methodology, Thesis, Division of Mechanics, Department of Mechanical Engineering, Linköping University, Sweden, 2003.

[8] Craig, K.J. Stander, N., Dooge, D. and Varadappa, S. MDO of automotive vehicle for crashworthiness and NVH using response surface methods, Paper 2002-5607, 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, September 4-6, 2002.

[9] TrueGrid Manual, XYZ Scientific Applications, Inc. Livermore, CA, 2000.

[10] Akkerman, A., Thyagarajan, R., Stander, N., Burger, M., Kuhn, R., Rajic, H. Shape optimization for crashworthiness design using response surfaces. Proceedings of the 1st International Workshop on Multidisciplinary Design Optimization, Pretoria, South Africa, 8-10 August 2000, pp. 270-279.