Solving optimisation problems in metal forming using Finite Element simulation and metamodelling techniques

M.H.A. Bonte, A.H. van den Boogaard, J. Huétink*

Abstract

During the last decades, Finite Element (FEM) simulations of metal forming processes have become important tools for designing feasible production processes. In more recent years, several authors recognised the potential of coupling FEM simulations to mathematical optimisation algorithms to design optimal metal forming processes instead of only feasible ones.

Within the current project, an optimisation strategy is being developed, which is capable of optimising metal forming processes in general using time consuming nonlinear FEM simulations. The expression "optimisation *strategy*" is used to emphasise that the focus is not solely on solving optimisation problems by an optimisation algorithm, but the way these optimisation problems in metal forming are modelled is also investigated. This modelling comprises the quantification of objective functions and constraints and the selection of design variables.

This paper, however, is concerned with the choice for and the implementation of an optimisation algorithm for solving optimisation problems in metal forming. Several groups of optimisation algorithms can be encountered in metal forming literature: classical iterative, genetic and approximate optimisation algorithms are already applied in the field. We propose a metamodel based optimisation algorithm belonging to the latter group, since approximate algorithms are relatively efficient in case of time consuming function evaluations such as the nonlinear FEM calculations we are considering. Additionally, approximate optimisation algorithms strive for a global optimum and do not need sensitivities, which are quite difficult to obtain for FEM simulations. A final advantage of approximate optimisation algorithms is the process knowledge, which can be gained by visualising metamodels.

In this paper, we propose a sequential approximate optimisation algorithm, which incorporates both Response Surface Methodology (RSM) and Design and Analysis of Computer Experiments (DACE) metamodelling techniques. RSM is based on fitting lower order polynomials by least squares regression, whereas DACE uses Kriging interpolation functions as metamodels. Most authors in the field of metal forming use RSM, although this metamodelling technique was originally developed for physical experiments that are known to have a stochastic nature due to measurement noise present. This measurement noise is absent in case of deterministic computer experiments such as FEM simulations. Hence, an interpolation model fitted by DACE is thought to be more applicable in combination with metal forming simulations. Nevertheless, the proposed algorithm utilises both RSM and DACE metamodelling techniques.

As a Design Of Experiments (DOE) strategy, a combination of a maximin spacefilling Latin Hypercubes Design and a full factorial design was implemented, which takes into account explicit constraints. Additionally, the algorithm incorporates cross validation as a metamodel validation technique and uses a Sequential Quadratic Programming algorithm for metamodel optimisation. To overcome the problem of ending up in a local optimum, the SQP algorithm is initialised from every DOE point, which is very time efficient since evaluating the metamodels can be done within a fraction of a second. The proposed algorithm allows for sequential improvement of the metamodels to obtain a more accurate optimum.

As an example case, the optimisation algorithm was applied to obtain the optimised internal pressure and axial feeding load paths to minimise wall thickness variations in a simple hydroformed product. The results are satisfactory, which shows the good applicability of metamodelling techniques to optimise metal forming processes using time consuming FEM simulations.

1 Introduction

During the last decades, Finite Element (FEM) simulations of metal forming processes have become important tools for designing feasible production processes. In more recent years, several authors recognised the potential of coupling FEM simulations to mathematical optimisation algorithms to design *optimal* metal forming processes instead of only *feasible* ones.

The basic concept of mathematical optimisation is presented in Figure 1. Basically, it consists of two major phases: the *modelling* and the *solving* of the optimisation problem. The modelling phase exists of:

1. Selecting a number of design variables the user is allowed to adapt
2. Choosing an objective function, i.e. the optimisation aim

*Faculty of Engineering Technology (Applied Mechanics group), University of Twente, P.O. Box 217, 7500 AE, Enschede, The Netherlands, email: m.h.a.bonte@utwente.nl

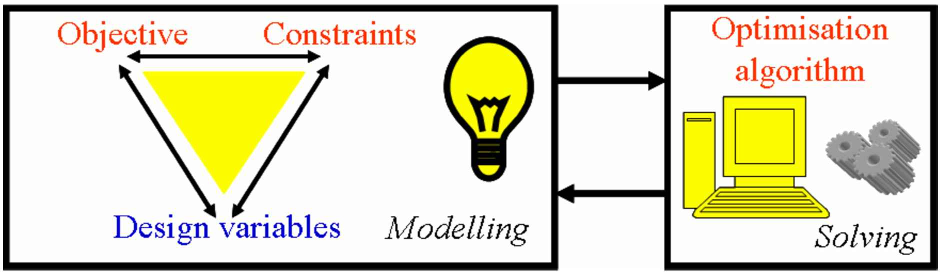

Figure 1: The basic concept of mathematical optimisation: modelling and solving

3. Taking into account possible constraints

These three items are closely related to each other. Both the objective function and the constraints should be quantified by the design variables, which explains the arrows between both the objective function and constraints on the one hand, and the design variables on the other hand in Figure 1. The objective function and constraints are also related to each other in such a way that they are often exchangeable. Let us explain this statement by a metal forming example. Suppose we would like to make a metal formed product and two relevant properties are the product quality and the costs. Then two approaches can be followed: either the quality is maximised while putting a certain limit on the allowed production costs, or the costs could be minimised while ensuring a certain minimum level of the product quality. In the former case, the quality is clearly the optimisation objective and the costs are constraints, whereas it is just the other way around in the latter case.

Next to the modelling phase, mathematical optimisation's second phase is the solving of the optimisation problem. This comprises applying an optimisation algorithm to the modelled optimisation problem, which generally means letting a computer do the hard work. The arrows between the modelling and the solving parts in Figure 1 denote that both phases cannot be seen separately from each other. One should select the right optimisation algorithm for a certain optimisation problem, which is modelled, and one should model the optimisation problem cleverly to adjust it to the optimisation algorithm one is planning to apply. If the optimisation model does not match the algorithm, it is likely that the optimisation problem is not solved efficiently or even at all [17].

This paper focuses on the solving part of optimisation problems in metal forming using time consuming nonlinear FEM simulations. One simulation can easily take hours or even days to execute. It is important to keep this fact in mind when selecting a suitable optimisation algorithm for metal forming processes.

A way of optimising metal forming processes is using classical iterative optimisation algorithms (Conjugate gradient, BFGS, etc.), where each function evaluation means running a FEM calculation, see e.g. [7, 9, 15]. As mentioned above, in case of metal forming these FEM calculations can be extremely time consuming and need to be evaluated sequentially. Furthermore, many classical algorithms require sensitivities, of which the efficient calculation is not straightforward for FEM simulations. A third difficulty concerning iterative algorithms is the risk to be trapped in local optima.

Alternatively, several authors have tried to overcome these disadvantages by applying genetic or evolutionary optimisation algorithms, see e.g. [1, 2, 22]. Genetic and evolutionary algorithms look promising because of their tendency to find the global optimum and the possibility for parallel computing. However, the rather large number of function evaluations that is expected to be necessary using genetic algorithms is regarded as a serious disadvantage [10].

Yet another way of optimisation in combination with expensive function evaluations is using approximate optimisation algorithms, of which Response Surface Methodology (RSM) is a well-known example. RSM is based on fitting a lower order polynomial metamodel through response points, which are obtained by running FEM calculations for carefully chosen design variable settings and finally optimising this metamodel [13]. Metamodels are sometimes also referred to as Response Surface models or surrogate models. Allowing for parallel computing and lacking the necessity for sensitivities, RSM is appealing to many authors in the field of metal forming, see e.g. [3, 4, 14].

Although the practical effectiveness of RSM has been frequently demonstrated, statisticians claim that RSM, being developed for stochastic physical experiments, is theoretically not applicable to deterministic computer experiments such as FEM: running a simulation twice with exactly the same input will generally result in exactly the same answer. They propose the field of "Design and Analysis of Computer Experiments" or DACE instead [19, 20, 21]. DACE is similar to RSM, but interpolates a metamodel through the response points based on Kriging. Allowing for no error, interpolation better suits the deterministic nature of computer experiments. However, DACE is rarely used in the metal forming community, probably due to its complex statistical nature and the lack of readily available software [21].

In this paper an optimisation algorithm incorporating both RSM and DACE metamodelling techniques is proposed for metal forming. Section 2 introduces the basic concept of metamodelling and provides a more detailed description of RSM and DACE. The proposed optimisation algorithm is presented in Section 3 and the applicability to metal forming is demonstrated in Section 4 when it is applied to the optimisation of a hydroforming process. Conclusions are drawn in Section 5.

2 Metamodelling

The principle of metamodelling is presented in Figure 2 [5]. The basic idea is to evaluate a certain problem entity, in our case a metal forming process. This problem entity can be modelled by some sort of a simulation model. For metal forming, this simulation model is usually a nonlinear Finite Element code. The simulation model, and specifically these nonlinear FEM calculations, are still very time consuming to execute. Therefore, a *metamodel* or a *model from a model* [23] is made, which can be evaluated quickly. An accurate metamodel should be valid with respect to both the simulation model and the problem entity and if it is, it forms a very useful substitute for both the problem entity and the simulation model.

According to Kleijnen and Sargent [5], metamodels can serve four goals:

1. Understanding the problem entity
2. Predicting values of the output or response variable
3. Optimisation
4. Verification and Validation of prior qualitative knowledge or the simulation model with respect to the problem entity

Different goals require different types of metamodels and different levels of accuracy. For the optimisation of metal forming processes, the third of these goals is of course most important. Using metamodels for the other goals, however, comes additionally at low computational costs, which is seen as a major advantage of using metamodelling techniques for optimisation purposes.

In the next sections, two metamodelling techniques, Response Surface Methodology and Design and Analysis of Computer Experiments or Kriging, are introduced shortly. One should be aware that prior to fitting the metamodels, a Design Of Experiments (DOE) strategy carefully selects a number of design variable settings for which FEM simulations are being run. This results in a number of response measurements.

2.1 Response Surface Methodology (RSM)

Starting with RSM, the response measurements $\mathbf{y}$ are presented as the sum of a lower order polynomial metamodel and a random error term ε [13]:

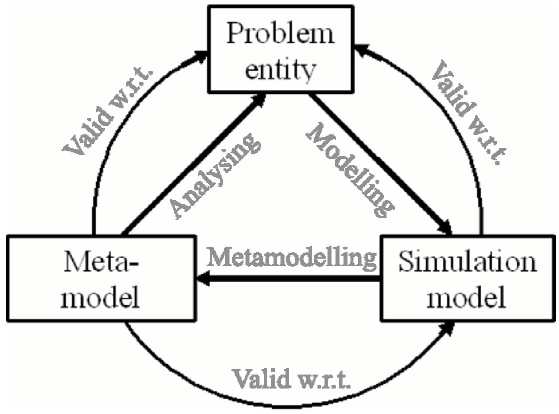

Figure 2: The principle of metamodelling

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \varepsilon \quad (1)$$

where $\mathbf{X}$ is the design matrix containing the experimental design points and $\boldsymbol{\beta}$ are the regression coefficients obtained by least squares regression:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} \quad (2)$$

Although Equation 1 seems to be a linear relation of the design variables, the design matrix $\mathbf{X}$ can also incorporate terms that are nonlinear with respect to the design variables. Equation 1 should, however, be linear with respect to the regression coefficients [13], which is clearly the case.

The metamodel is

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} \quad (3)$$

and can generally obtain four possible shapes, which are in ascending complexity:

- linear
- linear + interaction
- pure quadratic or elliptic
- (full) quadratic

A second order RSM metamodel dependent on one design variable is shown in Figure 3(a).

2.2 Design and Analysis of Computer Experiments (DACE)

DACE was proposed by Sacks et. al. [19, 20] to fit metamodels using deterministic computer experiments. *Kriging*, which was originally developed for locating possible spots to find gold, is used to interpolate between the response measurements. Using Kriging, the random error

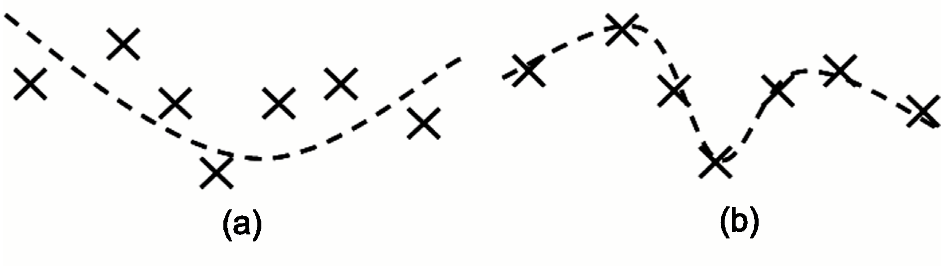

Figure 3: Metamodels based on (a) RSM and (b) DACE

term ε in Equation 1 is replaced by a Gaussian random function $Z(\mathbf{x})$, which forces the metamodel to go exactly through the measurement points:

$$\hat{\mathbf{y}} = \mathbf{F}^{\mathrm{T}}\boldsymbol{\beta} + Z(\mathbf{x}) \tag{4}$$

Note that the design matrix $\mathbf{X}$ is replaced by $\mathbf{F}^{\mathrm{T}}$, which is a standard notation in Kriging literature. The first part of Equation 4 covers the global trend of the metamodel. The Gaussian random function Z, which accounts for the local deviation of the data from the trend function, has zero mean, variance σ_{z}^2 and covariance

$$\mathrm{cov}\left(Z(x_1), Z(x_2)\right) = \sigma_{\mathrm{z}}^2 R\left(x_1 - x_2\right) \tag{5}$$

where R is the correlation function and x_1 and x_2 are two locations, which are determined by the design variable settings at these locations. For the proposed algorithm, a Gaussian exponential correlation function is adopted:

$$R(\vartheta, x_1, x_2) = \exp^{-\vartheta(x_1 - x_2)^2} \tag{6}$$

As opposed to other possibilities for the correlation function like e.g. cubic splines and ordinary exponential functions, see e.g. [8, 25, 24, 21], Gaussian exponential functions are intuitively attractive because they are infinitely differentiable. Moreover, Gaussian exponential functions are frequently used in literature [21] and have been found to give accurate results [24].

If more, say k, design variables are present, the total correlation function R is assumed to depend on the k one-dimensional correlation functions R_j as follows [19]:

$$R\left(x_1 - x_2\right) = \prod_{j=1}^{k} R_j\left(x_{1j} - x_{2j}\right) \tag{7}$$

That is, one assumes there is no relation between the different dimensions. Adopting the Gaussian correlation function introduced in Equation 6, the total correlation function becomes:

$$R\left(x_1 - x_2\right) = \prod_{j=1}^{k} \exp^{-\vartheta_j(x_{1j} - x_{2j})^2} \tag{8}$$

Thus, one ϑ is present for each design variable (each dimension).

Figure 3(b) presents a Kriging interpolation metamodel.

3 A metamodel based optimisation algorithm for metal forming

The proposed metamodel based optimisation algorithm for the optimisation of metal forming processes using time consuming FEM simulations is presented in Figure 4. Several steps mentioned in the figure are explained in the following Sections 3.1 through 3.4. Section 3.5 contains a few words on the implementation of the algorithm.

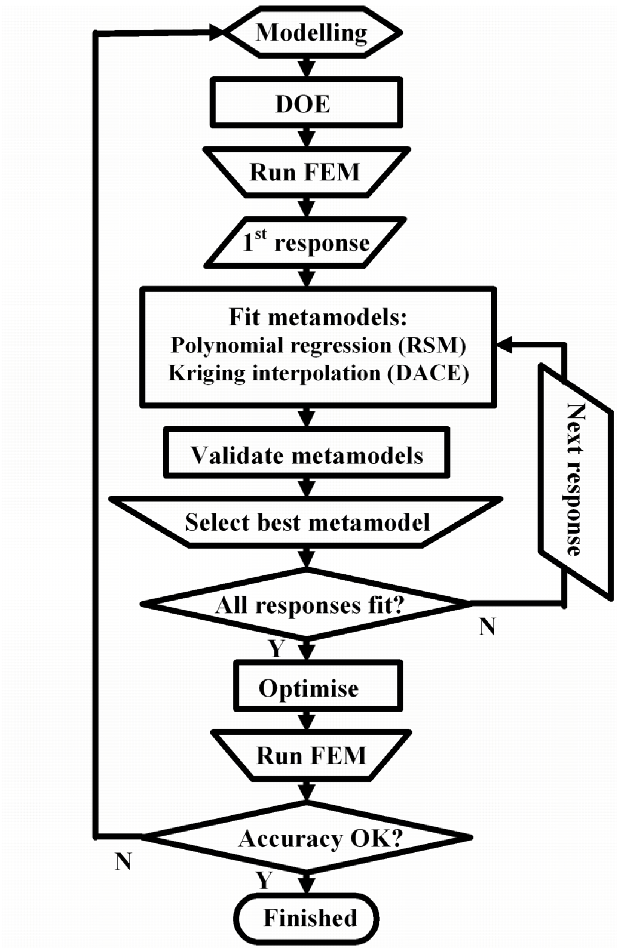

Figure 4: A metamodel based optimisation algorithm for metal forming processes

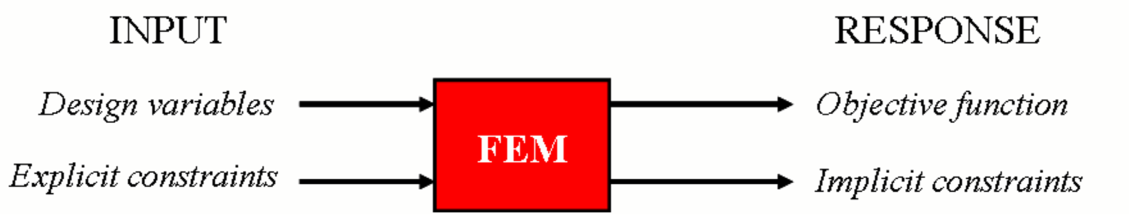

Figure 5: The difference between explicit and implicit constraints

3.1 Modelling

The first step is to start with modelling the optimisation problem, i.e. quantifying objective function and constraints and selecting the design variables. Regarding the constraints, a distinction is made between explicit and implicit constraints. To explain the difference, running a FEM simulation can be seen as an input-throughput-response model such as the one depicted in Figure 5. Certain quantities are known beforehand: there is no necessity to run a FEM calculation for evaluating them. The design variables are clear examples of these quantities and there can also be constraints that explicitly depend on the design variables. These constraints are called explicit constraints. In case of metal forming explicit constraints are related to the undeformed product, e.g. constraints on the initial shape of a blank.

Quantities that depend on the response, in other words, one needs to run a FEM simulation for evaluating them, depend implicitly on the design variables. The objective function is generally such an implicit quantity and it is also possible to have implicit constraints. For metal forming, implicit constraints are related to the deformed product, e.g. necking should not occur.

It is stressed again that the modelling of an optimisation problem is officially not part of an optimisation algorithm, which is solely a mean for solving the optimisation model. Clever modelling and solving are both crucial for mathematically optimising an optimisation problem as was already emphasised in the introduction.

3.2 Design Of Experiments (DOE)

When the optimisation problem is modelled, Figure 4 shows that the first step of the algorithm is to carefully select a number of design sites by a Design Of Experiments (DOE) strategy. The selection of a suitable DOE strategy depends on the type of metamodel that is fitted. Desirable properties of DOE strategies are quite different for RSM than for DACE metamodels [21] and it is required to fit both RSM and DACE from the same response measurements to ensure time efficiency. Hence, a choice should be made for a DOE strategy suitable for RSM or one more applicable to DACE. Because of the deterministic nature of computer experiments, DACE is slightly more applicable to simulation models such as FEM than RSM and it was chosen to implement a good and popular DOE strategy for DACE rather than a suitable strategy for RSM: a spacefilling Latin Hypercube Design (LHD) [12, 21].

When a metamodel is used for optimisation, it is important that the metamodel gives accurate results in the neighbourhood of the optimum. Often, this optimum will be constrained, i.e. lies on the boundary of the design space. Therefore, an accurate prediction is needed on the boundary, which implies performing measurements on the boundary. An LHD will generally provide design points in the interior of the design space and not on the boundary. To compensate for this lack of points on the boundary, the LHD is combined with a full factorial design, which puts DOE points right in the corners of the design space. This method was also proposed by Van Beers et. al. [26] and Kleijnen et. al. [6]. Figure 6(a) presents the LHD modified with a full factorial design for a two dimensional rectangular design space.

Unfortunately, the design space will often not be rectangular when explicit constraints are present. In this case, the proposed algorithm will:

1. check which points of the LHD + full factorial design are non-feasible
2. skip the non-feasible points
3. replace the non-feasible points with new points
4. repeat the above procedure until all points are feasible

Replacing the non-feasible points is also done in a spacefilling way by selecting a large number of sets of additional design points. The new set of points is the one for which the minimum point to point distance is maximised. This so-called *maximin* criterion is used for both the initial DOE and for the case when the user wants to generate additional experimental design points, for example for improving the accuracy of the metamodels. The final DOE strategy incorporated in the proposed optimisation algorithm is presented in Figure 6(b) for two design variables (x_1 and x_2) and two explicit constraints (g_1 and g_2).

3.3 Running the FEM simulations and fitting the metamodels

Subsequently, using the settings indicated by the DOE strategy, a number of FEM calculations is run on paral-

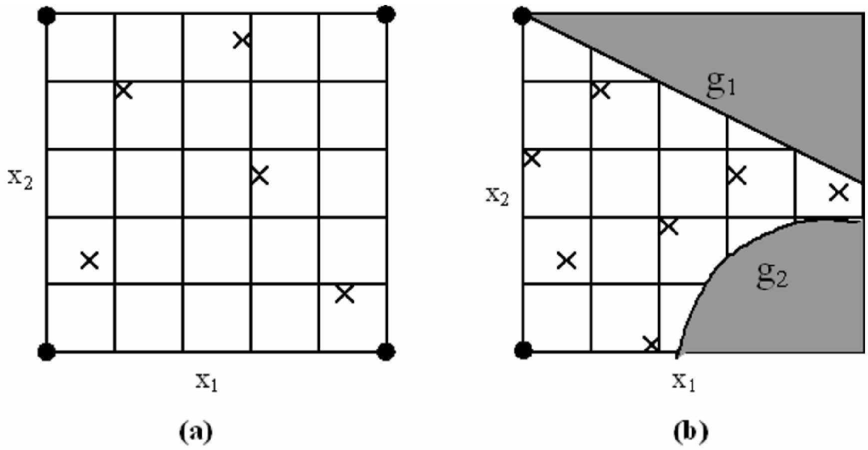

Figure 6: (a) LHD + full factorial design (b) LHD + full factorial design including explicit constraints

lel processors and the response points (objective function and implicit constraint values) are obtained. Following Figure 4, the next step is to fit for each response seven metamodels:

1. A linear polynomial using RSM
2. A linear + interaction polynomial using RSM
3. A pure quadratic or elliptic polynomial using RSM
4. A full quadratic polynomial using RSM
5. A Kriging interpolation metamodel with a 0^{th} order polynomial as a trend function
6. A Kriging interpolation metamodel with a 1^{st} order polynomial as a trend function
7. A Kriging interpolation metamodel with a 2^{nd} order polynomial as a trend function

3.4 Validation and optimisation

Metamodel validation based on leave-1-out cross validation (see e.g. [11]) is used to select the best metamodel for the observed response. Using cross validation, one leaves out one, say the i^{th}, of the response measurements and fits the metamodel through the remaining response measurements. The difference between the real value y_i and the value predicted by the metamodel at this location $\hat{y}_{-i}$ is a measure for the accuracy of the metamodel. One can repeat this procedure for all say n measurement points and calculate the cross validation Root Mean Squared Error ($RMSE_{\text{CV}}$):

$$RMSE_{\text{CV}} = \sqrt{\sum_{i=1}^{n} \frac{(y_i - \hat{y}_{-i})^2}{n}} \qquad (9)$$

As $RMSE_{\text{CV}}$ approaches 0, the metamodel becomes more and more accurate. Cross validation can also be visualised in a cross validation plot. An example of such a plot is presented in Figure 7. If the measurements follow the line $x = y$, the metamodel fits the data well.

For each response (objective function and implicit constraints) the metamodel outperforming the other six metamodels is selected. These best metamodels for objective function and implicit constraints are added to the explicit constraints in the optimisation model, which is subsequently optimised using a standard Sequential Quadratic Programming (SQP) algorithm, see for example [18]. In case implicit constraints or Kriging metamodels are present in the final optimisation problem, there is a risk of ending up in a local optimum. This problem is overcome by initialising the SQP algorithm at multiple locations. This implies performing many function evaluations, but this is hardly a problem since both RSM and DACE metamodels, being explicit mathematical functions, can be evaluated thousands of times within a fraction of a second. The DOE points are used as initial locations for the SQP algorithm.

The obtained approximate optimum is finally checked by running one last FEM calculation with the approximated

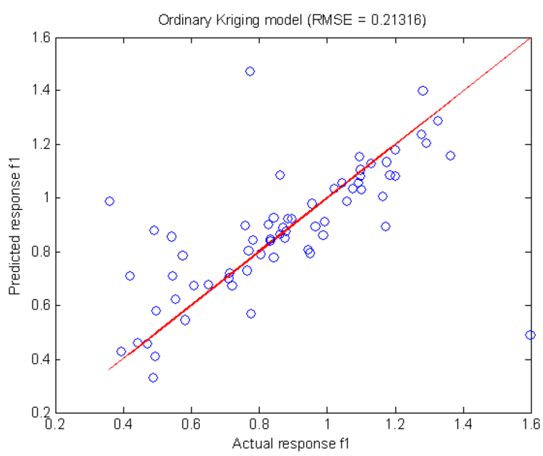

Figure 7: A cross validation plot

optimal settings of the design variables. The difference between the approximate objective function value and the real value of the objective function calculated by the last FEM run is a measure for the accuracy of the obtained optimum. If the user is not satisfied with this accuracy, the algorithm allows for remodelling the optimisation problem (e.g. zooming near the optimum) and repeating the procedure presented above until one is satisfied with the accuracy. Hence the proposed algorithm incorporates all the advantages of sequential approximate optimisation algorithms.

3.5 Implementation

The optimisation algorithm was implemented in MATLAB and can be used in combination with any Finite Element code. For the fitting of the DACE/Kriging metamodels, use was made of the MATLAB Kriging toolbox implemented by Lophaven, Nielsen and Søndergaard [16, 25, 24].

4 A metal forming application

The optimisation algorithm introduced in the previous section is applied to a simple hydroforming process. The product to be hydroformed is presented in Figure 8(a). Making use of symmetry, the axisymmetric product can be modelled in 2D as shown in Figure 8(b).

For metal forming, several groups of design variables can be distinguished:

1. Geometrical parameters:
   (a) Product geometry
   (b) Tool geometry
2. Material parameters
3. Process parameters

The group of geometrical parameters is divided further into variables belonging to the product, e.g. product radii, thicknesses, etc., and variables related to the die geometry (drawbeads, blank size and so on). Examples of material parameters are strain hardening coefficients, the initial yield stress or simply several discrete materials in itself. The group of process parameters includes process forces, pressures, tool displacements, friction coefficient, process temperature, etc.

For the simple metal forming example considered here, we are interested in optimising the time variation of the internal pressure p and axial feeding u. These are typically process parameters for the hydroforming process. A typical time dependent load path for hydroforming is shown in Figure 8(c). Assuming a strain rate independent material (α is irrelevant), three design variables are remaining: the time when axial feeding starts t_1, the time

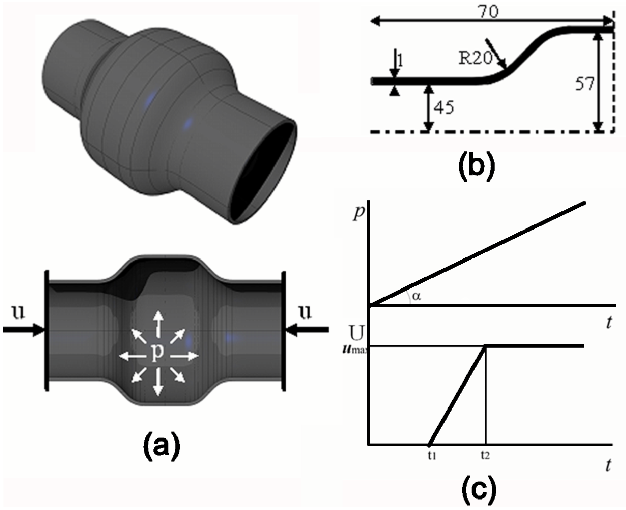

Figure 8: Hydroformed product that is used as a simple test case to show the applicability of the metamodel based optimisation algorithm in the field of metal forming

when axial feeding stops t_2 and the total amount of axial feeding $u_{\max}$.

As an optimisation objective, it was chosen to minimise variations in the wall thickness of the final product with respect to the initial tube thickness. One implicit and two explicit constraints were formulated. The implicit constraint ensures that the final product fills out the die nicely, one explicit constraint makes sure that the time when axial feeding stops is larger than the time when it starts and the last constraint was formulated to overcome convergence problems of the FEM calculations when t_2 approaches t_1 and the amount of axial feeding is high (large $u_{\max}$). Methods to handle non converged simulations are lacking and is a field of open research. The total optimisation problem is modelled as follows:

$$\begin{aligned} \min f\left(t_1, t_2, u_{\max}\right) &= \left\| \frac{h - h_0}{h_0} \right\|_2 \\ \text{s.t.} \qquad g_{\text{impl}} &= V \leq 0 \\ g_{\text{expl1}} &= t_1 - t_2 \leq 0 \\ g_{\text{expl2}} &= u_{\max} - 9\left(t_1 - t_2\right) \leq 0 \\ 0\,\text{s} &\leq t_1 \leq 5\,\text{s} \\ 2.5\,\text{s} &\leq t_2 \leq 10\,\text{s} \\ 0\,\text{mm} &\leq u_{\max} \leq 9\,\text{mm} \end{aligned} \tag{10}$$

where h is the final wall thickness at a certain location in the hydroformed product, h_0 is the wall thickness of the initial tube and V is the volume between the final product and the die. If this volume is larger than zero, there is a gap between the final product and the die and the final shape of the product is not satisfactory.

The optimisation problem given by Equation 10 is now optimised by the optimisation algorithm proposed in Section 3. Figure 9(a) shows the 2D axisymmetric FEM

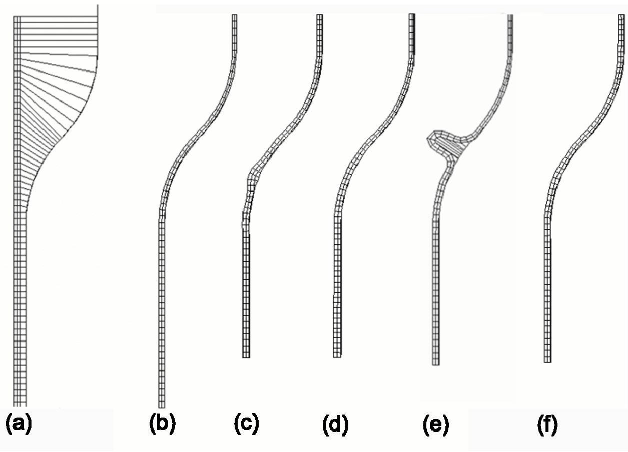

Figure 9: (a) FE model of the initial tube; (b-e) Final product formed with several arbitrary selected load paths; (f) Final product formed with optimised load paths

model used during the optimisation. Note that the model is rotated $90°$ with respect to Figure 8(b). The contact between the product and the die is modelled by contact elements, which have no stiffness when there is no contact and a very high stiffness when contact between the product and the die is established. The FEM calculations were run in batches of 16 parallel calculations using the FE code DiekA.

Figures 9(b) through (e) present some final products deformed with arbitrary load paths. The design variable settings for t_1, t_2 and $u_{\max}$ and the response values for the objective function f and the implicit constraint g_{impl} are presented in Table 1. Note that product (a) is the initial undeformed product, which is seen as the product with the perfect wall thickness distribution by the objective function quantified in Equation 10. For the perfect product, the objective function equals 0. Also note that products (c) and (e) do not satisfy the implicit constraint g_{impl}, which can also clearly be seen from Figures 9(c) and (e).

The optimised settings found by the proposed optimisation algorithm are also presented in Table 1 as product (f). The final shape of this product is shown in Figure 9(f). Figure 10 shows the wall thickness throughout the final product for all load paths. It can be concluded from the Figures 9 and 10 and Table 1 that the product deformed with the optimised load paths outperforms the other products formed with arbitrary settings, which demonstrates the good applicability of the proposed algorithm to metal forming.

Product	t_1(s)	t_2(s)	$u_{\max}$(mm)	f	g_{impl}
(a)	–	–	–	0	–
(b)	0	0	0	1.39	-0.29
(c)	0	3	9	0.52	1.79
(d)	0	10	9	1.42	-0.34
(e)	4.8	6.2	7.7	1.37	32.64
(f)	0	2.5	8.3	0.32	-0.47

Table 1: Design variable settings and response values

5 Conclusions

An optimisation algorithm based on metamodelling techniques is proposed for the optimisation of metal forming using time consuming FEM calculations. It uses both Response Surface Methodology and DACE (or Kriging) as metamodelling techniques. As a Design Of Experiments strategy, a combination of a maximin spacefilling Latin Hypercubes Design with a full factorial design was implemented, which takes into account explicit constraints. Additionally, the algorithm incorporates cross validation as a metamodel validation technique and uses a Sequential Quadratic Programming algorithm for metamodel optimisation. To overcome the problem of ending up in a local optimum, the SQP algorithm is initialised from every DOE point, which is very time efficient since evaluating the metamodels can be done within a fraction of a second. The proposed algorithm allows for sequential improvement of the metamodels to obtain a more accurate optimum.

As an example case, the optimisation algorithm was applied to obtain the optimised internal pressure and axial feeding load paths to minimise wall thickness variations in a simple hydroformed product. The results are satisfactory, which shows the good applicability of metamodelling techniques to optimise metal forming processes.

Acknowledgements

This work is conducted within the framework of project MC1.03162, Optimisation of Forming Processes, which

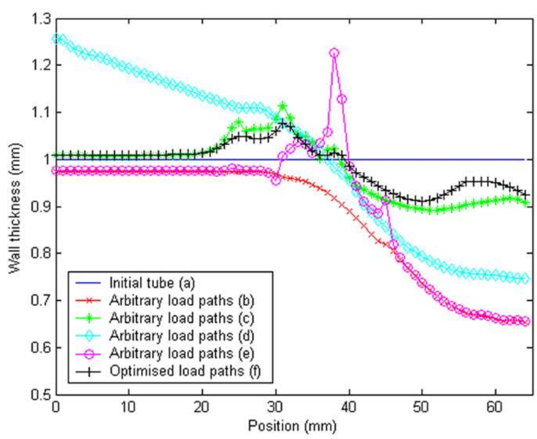

Figure 10: Wall thickness distribution of several hydroformed products

is part of the research programme of the Netherlands Institute for Metals Research (NIMR). The NIMR and its industrial partners are gratefully acknowledged for their support and useful input.

References

[1] C.F. Castro, C.A.C. António, and L.C. Sousa. Optimisation of shape and process parameters in metal forging using genetic algorithms. *International Journal of Materials Processing Technology*, 146:356–364, 2004.

[2] L. Fourment, T.T. Do, A. Habbal, and M. Bouzaiane. Gradient, non-gradient and hybrid algorithms for optimizing 2D and 3D forging sequences. In *Proceedings of ESAFORM*, Cluj-Napoca, Romenia, 2005.

[3] T. Jansson. *Optimization of Sheet Metal Forming Processes*. Licentiate thesis, Universitet Linköping, Linköping, Sweden, 2002.

[4] T. Jansson, A. Andersson, and L. Nilsson. Optimization of draw-in for an automotive sheet metal part – an evaluation using surrogate models and response surfaces. *Journal of Materials Processing Technology*, 159:426–234, 2005.

[5] J.P.C. Kleijnen and R.G. Sargent. A methodology for fitting and validating metamodels in simulation. *European Journal of Operational Research*, 120:14–29, 2000.

[6] J.P.C. Kleijnen and W.C.M. van Beers. Application-driven sequential designs for simulation experiments: Kriging metamodelling. *European Journal of Operational Research*, submitted in 2004.

[7] J. P. Kleinermann and J. P. Ponthot. Parameter identification and shape/process optimization in metal forming simulation. *Journal of Materials Processing Technology*, 139(1-3):521–526, 2003.

[8] J.R. Koehler and A.B. Owen. *Handbook of Statistics*, chapter Computer Experiments, pages 261–308. Elsevier Science, New York, USA, 1996.

[9] Z. Lin, X. Juchen, W. Xinyun, and H. Guoan. Optimization of die profile for improving die life in the hot extrusion process. *Journal of Materials Processing Technology*, 142(3):659–664, 2003.

[10] M. Emmerich, A. Giotis, M. Özdemir, T. Bäck and K. Giannakoglou. Metamodel-assisted evolution strategies. In *Proceedings of the International Conference on Parallel Problem Solving from Nature*, 2002.

[11] J.D. Martin and T.W. Simpson. A study on the use of kriging models to approximate deterministic computer models. In *Proceedings of the ASME Design Engineering Technical Conferences DETC*, 2003.

[12] M.D. McKay, R.J. Beckman, and W.J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21:239–245, 1979.

[13] R.H. Myers and D.C. Montgomery. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. John Wiley and Sons, Inc., New York, USA, 2^{nd} edition, 2002. ISBN 0-471-41255-4.

[14] H. Naceur, S. Ben-Elechi, C. Knopf-Lenoir, and J.L. Batoz. Response surface methodology for the design of sheet metal forming parameters to control springback effects using the inverse approach. In *Proceedings of NUMIFORM*, Columbus OH, USA, 2004.

[15] H. Naceur, Y. Q. Guo, J. L. Batoz, and C. Knopf-Lenoir. Optimization of drawbead restraining forces and drawbead design in sheet metal forming process. *International Journal of Mechanical Sciences*, 43(10):2407–2434, 2001.

[16] H.B. Nielsen. DACE, A MATLAB Kriging toolbox. http://www.imm.dtu.dk/ hbn/dace/.

[17] P. Y. Papalambros and D. J. Wilde. *Principles of optimal design*. Cambridge University Press, New York, USA, 2000. ISBN 0-521-62727.

[18] R. T. Haftka and Z. Gürdal. *Elements of structural optimization*. Kluwer academic publishers, Dordrecht, Netherlands, 3^{rd} edition, 1992. ISBN 0-7923-1504-9.

[19] J. Sacks, S.B. Schiller, and W.J. Welch. Design for computer experiments. *Technometrics*, 31:41–47, 1989.

[20] J. Sacks, W.J. Welch, T.J. Mitchell, and H.P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4:409–423, 1989.

[21] T.J. Santner, B.J. Williams, and W.I. Notz. *The Design and Analysis of Computer Experiments*. Springer-Verlag, New York, USA, 2003. ISBN 0-387-95420-1.

[22] O. Schenk and M. Hillmann. Optimal design of metal forming die surfaces with evolution strategies. *Computers and Structures*, 82:1695–1705, 2004.

[23] T.W. Simpson, J.D. Peplinski, P.N. Koch, and J.K. Allen. Metamodels for computer-based engineering design: Survey and recommendations. *Engineering with Computers*, 17:129–150, 2001.

[24] S.N. Lophaven, H.B. Nielsen and J. Søndergaard. Aspects of the MATLAB Toolbox DACE. Technical Report IMM-REP-2002-13, Technical University of Denmark - Department of Informatics and Mathematical Modelling, Lyngby, Denmark, 2002.

[25] S.N. Lophaven, H.B. Nielsen and J. Søndergaard. DACE - A MATLAB Kriging Toolbox. Technical Report IMM-TR-2002-12, Technical University of Denmark - Department of Informatics and Mathematical Modelling, Lyngby, Denmark, 2002.

[26] W.C.M. van Beers and J.P.C. Kleijnen. Kriging interpolation in simulation: A survey. In R.G. Ingalls, M.D. Rossetti, J.S. Smith, and B.A. Peters, editors, *Proceedings of the 2004 Winter Simulation Conference*, 2004.