Quality Technology & Quantitative Management
Vol. 6, No. 4, pp. 409-421, 2009

QTQM

Springback of Stamping Process Optimization Using Response Surface Methodology and Interval Computation

Daniel Lepadatu[1], Abdessamad Kobi[2], Xavier Baguenard[2] and Luc Jaulin[3]

[1]Civil Engineering Department, Technical University "Gh. Asachi", Iasi, Romania
[2]ISTIA – LASQUO, Angers, France
[3]DTN –ENSIETA, Brest Cédex, France
(*Received* September 2006, *accepted* June 2008)

Abstract: In manufacturing process, the quality of final products is significantly affected by both product design and process variables. However, historically tolerance research primarily focused on allocating tolerances based on product design characteristics of each component. This work proposes to expand the current tolerancing practices, and presents a new optimization method of tolerancing mechanical systems using interval computation for the prediction of system response. The proposed methodology is based on the development and integration of three concepts in process optimization: mechanical tolerancing, response surface methodology, and interval computation method. An industry case study is used to illustrate the proposed approach.

Keywords: Design of experiments, interval computation method, mechanical tolerancing, optimization, response surface methodology, springback.

1. Introduction

Manufacturing parts is a process whose specific nature makes it imperfect, and this can perhaps have a negative impact on the final product. Modern mass production was among the first to develop the concept of interchangeability. For this reason, it was necessary to introduce tolerances, which guarantee interchangeability. Many researchers have studied the different applications of tolerances [4, 28, 33, 34] in the context of ever more varied production requiring interchangeability [13]. In order to ensure better quality, parts production by different participants in the market, have perfected the necessary functional conditions that certainly use mechanical tolerances. This practice is a necessary tool for the correct functioning of a very important market in the world economy, ever more demanding when it comes to respect for optimal manufacturing conditions with sometimes uncertain variables. In this paper, we propose an extension of the practice of tolerances in order to give for scientific computation an efficient means, capable of factoring in different degrees of process variability, ensuring the same quality of the final product.

It is therefore the purpose of this study to provide a design method combining Mechanical Tolerances (*MT*), Response Surface Methodology (*RSM*) and Interval Computation (*IC*) in process optimization. This method consist provides a powerful tool to be used for minimizing the variability of the manufacturing process. Our approach introduces a new concept for process optimization called Interval Response Surface. For a given response, the target corresponds to the set of parameters that are consistent with some given functional tolerances for various factors. This method will allow a new way of process optimization approach by introducing non-targeted responses. The resulting response surface will enclose a part of the non-corresponding products having failed to fulfill the standard quality, which was until now targeted on a certain value. The multiresponse optimization will undoubtedly be a much more significant application. In this case the products are conditioned to fulfill several quality standards simultaneously. Here the method will facilitate this task by "tolerated" but always functional responses.

This new method, which exploits the power of the three approaches, recalled in this work, is an effective method, perfectly adapted to multiresponse optimization and will bring a robust solution by giving certain flexibility for the variables in process optimization.

2. Response Surface Methodology (*RSM*)

Response Surface Methodology (*RSM*) consists of a group of empirical techniques devoted to the evaluation of relations existing between a cluster of controlled experimental factors and the measured responses, according to one or more selected criteria [5, 8, 24]. *RSM* provides an approximate relationship between a true response y and p design variables, which is based on the observed data from the process or system. The response is generally obtained from real experiments or computer simulations. Thus, computer simulations are performed in this paper. In this case the true response y is the same as expected response. We suppose that the response y, can be written as follows:

$$y = F(x_1, x_2, \cdots, x_p), \tag{1}$$

where the variables $x_1, x_2, \cdots, x_p$ are expressed in natural units of measurement, called "natural variables".

Once the variables having the greatest influence on the responses were identified, a special design was developed to optimize the levels of these variables. This design is a Box-Wilson Central Composite Design, commonly called 'Central Composite Design (*CCD*)', which contains an imbedded factorial or fractional factorial design having center points and being augmented by a group of 'star points' that allow estimation of curvature (Figure 1). If the distance from the center of the design space to a factorial point is ± 1 unit for each factor then the distance from the center of the design space to a star point is $\pm\delta$ with $|\delta| > 1$. The *CCD* is the most popular class of designs used to fit a second-order model. In this case the model is defined as follows:

$$y = \beta_0 + \sum_{j=1}^{3} \beta_j x_j + \sum_{j=1}^{3} \beta_{jj} x_j^2 + \sum_{i=1}^{3} \sum_{i<j} \beta_{ij} x_i x_j + \varepsilon. \tag{2}$$

If the phenomenon is strongly nonlinear (our case) the *CCD* is a very efficient design for fitting the second-order model.

3. Interval Computation

3.1. Introduction

Interval computation was introduced to compute all solutions of nonlinear problems in a guaranteed way. The basic idea is to represent uncertain real numbers by intervals containing them - see [25] or [1] for more details. Arithmetic laws for interval calculation [26, 27] make it possible to compute an interval which encloses the range of any function over a box (*i.e.*, a Cartesian product of intervals). Given two intervals $[x_1] = [a, b]$, $[x_2] =$

$[c, d]$, basic arithmetic operations are defined as follows:

$$\begin{aligned} &[x_1]+[x_2]=[a,b] + [c,d] = [a+c,\ b+d]; \\ &[x_1]-[x_2]=[a,b]+ (-[c,d]) = [a-d,\ b-c]; \\ &[x_1]*[x_2]=[\min\{ac,\ ad,\ bc,\ bd\}, \max\{ac,\ ad,\ bc,\ bd\}]; \\ &1/[x_1]=[1/b,\ 1/a],\ 0 \notin [a,b]. \end{aligned} \tag{3}$$

These operations are interval extensions of classical operations over real numbers. Many properties that were available for real numbers cannot be transposed to interval numbers. For example, we have $x - x = 0$ when x is a real. This property is no longer true when x is an interval. For instance, if $[x]=[-1,1]$, we will have $[x]-[x]=[-1,1]-[-1,1]=[-2,2]$. This phenomenon is known as the *dependency problem*. The size of the interval results of an expression grows with multi-occurrences of variables. Interval analysis is used in many domains of science such as: Global optimization [18], Solving nonlinear equations [22], Differential computation [17], Robotics [21], Bounded error estimation [6].

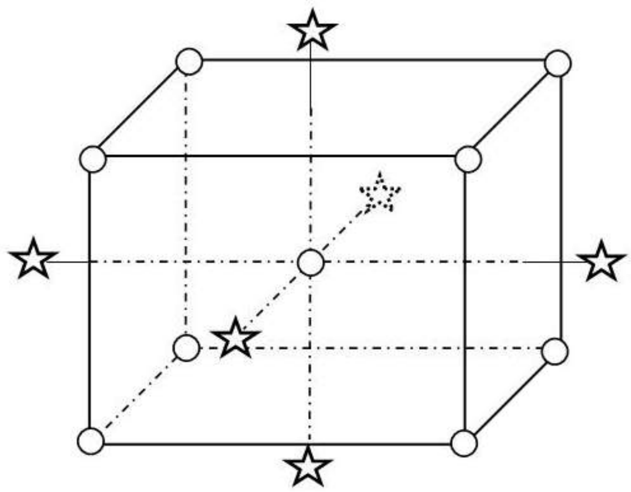

Figure 1. Central composite design for three factors.

Recent developments use constraint propagation to get more accurate results in faster way [7, 11]. Indeed, interval constraint propagation makes is possible to limit the dependency problem and provides a new way of considering problems.

A Constraint Satisfaction Problem *CSP* is defined by

a set V of n variables $x_1,\ldots, x_n$ of R;
a set D of n intervals $[x_1],\ldots,[x_n]$ of R, called domains; (4)
a set C of m constraints relating variables $c_1,\ldots, c_m$.

The solution set of this *CSP* is defined by

$$S = \{\boldsymbol{x} \text{ in } [\boldsymbol{x}], c_1(\boldsymbol{x}), c_2(\boldsymbol{x}),\ldots c_m(\boldsymbol{x})\}. \tag{5}$$

Constraint propagation aims at contracting domains by taking sequentially all constraints into account. The main objective is to compute the smallest box which encloses *S*.

An example of constraints propagation is given in the next section. Many free solvers are available to characterize the solution set of a *CSP* - see [2, 10, 15].

3.2. Constraint Propagation

Consider the following *CSP*:

$$c_1 : x_1 + x_2 = x_3, \tag{6}$$

$$x_1 \text{ in } [1, 3], \quad x_2 \text{ in } [0, 2], \quad x_3 \text{ in } [0, 2].$$

For some given pairs (x_1, x_2), we cannot find a consistent value x_3 inside the interval $[x_3]$. Such values are said to be inconsistent.

Constraint propagation techniques contact domains for the variables by removing values that have been proven to be inconsistent. For our example, domains obtained after contractions are :

$$[x_1] = [1, 2],\ [x_2] = [0, 1],\ [x_3] = [1, 2]. \tag{7}$$

The corresponding interval computation is as follows:

$$\begin{aligned} &[x_3] := [x_3] \cap ([x_1] + [x_2]), \\ &[x_1] := [x_1] \cap ([x_3] - [x_2]), \\ &[x_2] := [x_2] \cap ([x_3] - [x_1]). \end{aligned} \tag{8}$$

To implement an interval constraint propagation algorithm, we should first implement some contracting operator for primitive constraints (*i.e.*, non decomposable) such as:

$$\begin{aligned} &c_2 : x_1 * x_2 = x_3, \\ &c_3 : \sin(x_1) = x_2, \\ &c_4 : \exp(x_1) = x_2. \end{aligned} \tag{9}$$

Most of the constraints to be met in our context involve functions made by compositions of the operators $+ - * /$ and elementary functions such as *sin*, *cos*, *exp*. As a consequence, complex constraints can often be decomposed into primitive constraints. Constraints propagation calls some basic operators to contract all primitive constraints up to equilibrium [21].

3.3. An Example of Constraint Propagation

To illustrate the principle of constraint propagation, consider two variables x and y related by the three following constraints.

$$\begin{array}{ll} \text{(i)} & x \cdot y = 1 \\ \text{(ii)} & x + y = 1 \\ \text{(iii)} & y = x^2 \end{array}$$

The initial domains for the solutions are taken as $[x] = [y] = (-\infty, \infty)$. From the last constraint (iii), we get that y should be positive and thus, the domain $[y]$ for y should be contracted to $[y]=[0,\infty)$. From the constraint (i), we get that x is also positive and thus its domain is contracted to $[x]=[0,\infty)$. By taking (ii) into account, we are able to contract the domains for x and y to $[x]=[y]=[0,1]$. By considering again the first constraint, we get points for the domains of x and $y:[x]=[1,1]$ and $[y]=[1,1]$. Then (ii) contracts all domains to the empty set. We thus have shown that the *CSP* had no solution. This reasoning can be made automatic by resorting to interval computation. The resulting calculus is as follows:

(i) $[y]=[y]\cap(-\infty,\infty)=[0,\infty)$.
(ii) $[x]=[x]\cap([0,\infty)-1)=[0,\infty)$.
(iii) $[x]=[x]\cap(1-[0,\infty))=[0,1]$,
$[y]=[y]\cap(1-[0,1])=[0,1]$.
(vi) $[x]=[x]\cap([0,1]-1)=[1,1]$,
$[y]=[y]\cap([0,1]-1)=[1,1]$.
(v) $[x]=[x]\cap(1-[1,1])=\varnothing$,
$[y]=[y]\cap(1-[1,1])=\varnothing$.

3.4. Estimation Problem

To illustrate the estimation problem, consider the model:

$$f(x)=a_1\cdot\exp(a_2\cdot x), \tag{10}$$

where some interval data $[y_i]$ are available for some x_i inside $[x_i]$. The feasible set for the parameters a_1 and a_2 is defined by

$$S_1=\{(a_1,a_2) \text{ such that exists } x_i \text{ in } [x_i],\ y_i \text{ in } [y_i],\ f(x_i,a_1,a_2) \text{ in } [y_i]\}. \tag{11}$$

In Figure 2 gray boxes represent the interval's domains for x_1 and y_1. A solution included in the set S_1 is given by the dark curve. This curve corresponds to a couple of points (a_1,a_2) which represents a solution for our estimation problem. In Figure 2 the dotted line is a non-solution. In this case $f(x_2,a)$ cannot belong to $[y_2]$. In our problem, we need to obtain an interval value for (a_1,a_2) which will help us to choose a reliable value for the parameter vector.

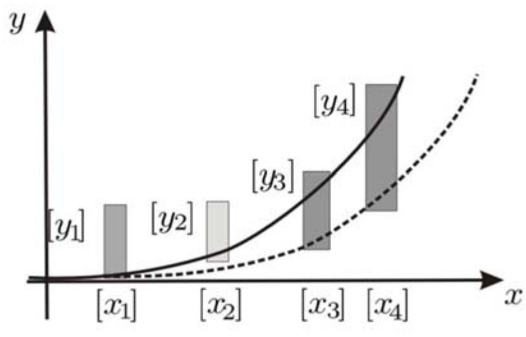

Figure 2. Parameter estimation for an exponential function.

You can also try to find a more specific set where constraints are satisfied for all the values inside the $[x_1]$'s [3]. The *robust feasible set* for the parameters is defined as follows:

$$S_2 = \{(a_1, a_2), \text{ such that } \forall\ x_i \text{ in } [x_i], \text{ there exists } y_i \text{ in } [y_i],\ f[x_i, a_1, a_2] \text{ in } [y_i]\}. \quad (12)$$

An infeasible value is drawn in Figure 3. In the circle there exists some values for x_i inside $[x_i]$ such that $f[x_i, a_1, a_2]$ is outside $[y_i]$ (Figure 3). The robust feasible set S_2 is included inside the feasible set S_1. In our application, we are more interested in the robust version.

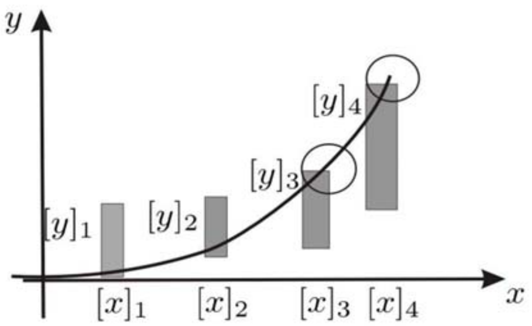

Figure 3. Example of estimation with an infeasible solution.

3.5. Proposed Algorithm

First a local method is applied, in order to find a good candidate for the parameter vector (a_1, a_2) From this candidate, we get an initial domain $[a] = ([a_1], [a_2])$. The set of all equations can be interpreted as a *CSP*. A constraint propagation procedure is then performed in order to contract all domains for the variables. This step is called *CS* $([a], [x], [y])$ (Figure 4).

```
Algorithm RSNP
Input: [a] [x], [y].
Output: [a].

If (width([a]) > epsilon)
    [x_old] := [x]; [y_old] := [y];
    C_S([a], [x], [y]);
        [Y_0] := Evaluations([x],[a])
        if (Criterion([y_0] [y_old])< 0
        [a] solution;
        else bisection ([a], [a_1], [a_2]);
            [y_1]:= Evaluations([x],[a])
            [y_2]:= Evaluations([x],[a])
            if (Criterion([y_1] [y_old])< Criterion([y_2] [y_old])
              RSNP ([a_1], [x], [y_old]);
              else RSNP ([a_2], [x], [y_old]);
```

Figure 4. The proposed algorithm.

Secondly we bisect the box $[a]$ into two subboxes. Using an interval evaluation over the box $[a]$ (Figure 5), we compare the two boxes in order to choose the most promising.

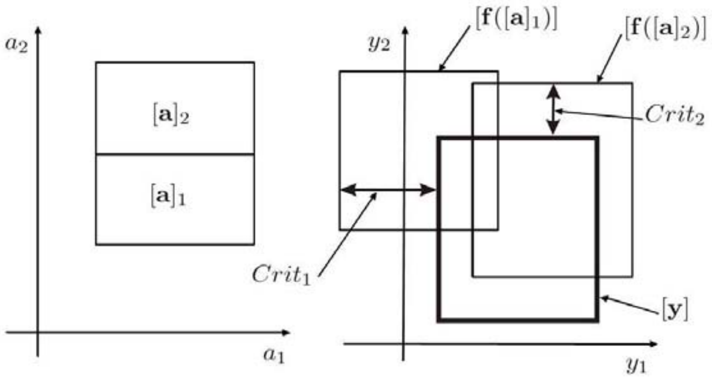

Figure 5. Criterion to select a_i box.

If the evaluation falls within in $[y_i]$ for all i, the criterion is negative. We thus have a solution box $[a]$.

In the example that we have chosen, we did not have multi-occurrences for the variables. In practice, it is not always the case and an overestimation may appear. It is the role of contractors to limit this pessimism as much as possible [3].

The proposed algorithm *RSNP* (Figure 4) is a branch and bound algorithm, which does not explore all the parts of the searching domain. If the entire searching domain would have been explored, the algorithm complexity would become exponential. In this case the optimal solution would certainly be found but not in a polynomial time. The present approach is different: its goal is to find quickly an interval solution which satisfies all constraints of the problem. The evaluation of the criterion allows us to choose the part of the searching domain where the solution has a much higher probability to belong. In some cases some solution points could be lost and thus the algorithm could converge to an approximate solution which does not satisfy all constraints. A backtracking could be made in order to avoid such a problem.

4. Mechanical Tolerancing

The inherent imperfections of manufacturing process cause a degradation of product characteristics, and therefore of product quality [9]. “Tolerance” is a method used to describe variability in a product or production process. It defines the acceptable ranges in the actual performance of a system or its components, across one or more parameters of interest, under the conditions considered during design, for which the system or components are fit for purpose, i.e., meet the specifications and/or customer expectations. Tolerances historically provide the means for communication between product and process designers [23]. Higher precision would mean lower tolerance and better machines are needed to manufacture the parts and thus, this will increase the cost to manufacture the parts. Tolerance is a key factor in determining the cost of a part. As mentioned earlier narrower tolerance will results in a higher cost of producing the parts. The relationship between tolerance and manufacturing cost is shown in the Figure 6.

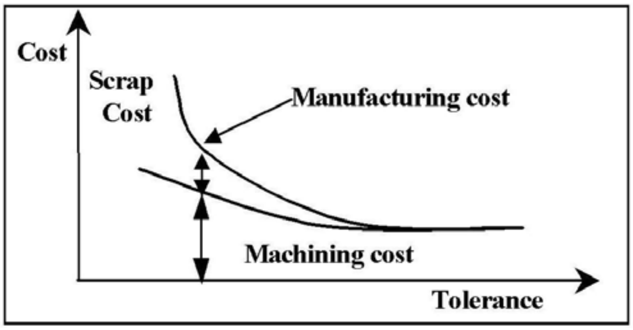

Figure 6. Manufacturing cost versus mechanical tolerance.

The manufacturing cost is divided into machining and scraps cost (Figure 6).

- The machining cost is the cost of first producing the part.
- The scrap cost is the cost encountered due to rejecting some parts that fall outside the specified tolerance range.

Generally, product or process is considered in conformity when it is in one acceptation interval (tolerance) [31]. Tolerance analysis views component-related tolerances as a range of values in terms of variation from a nominal value. Tolerance analysis takes a given set of component tolerances, usually based on designer experience or standards, and calculates the resultant variation in the assembly. Through iteration, component tolerances are tightened to meet assembly tolerances, establishing both the product and process design requirements. In contrast, tolerance allocation looks at a range of component designs around a functional or assembly description to absorb the variability. Tolerance allocation is used to maximize quality, minimize production cost, or both. The result can be looser component tolerances and better matching of product and process [14, 32]. In order to minimize the scraps cost, we propose a new method, which increases the acceptation interval of the assembly parts in manufacturing process of mechanical pieces.

5. Proposed Approach

This work aims at defining a new method of optimization that will use three concepts:

1. Response Surface Methodology;
2. Interval Computation Method;
3. Mechanical Tolerancing.

Actually, we will tolerate every level of parameters $X_{i_{\max}^{\min}}$ with specific bilateral tolerances $\pm\Delta_i$, which will later allow the usage of the proposed Interval Computation algorithm (Figure 4) in order to obtain what one may call "Interval Response Surface" (*IRS*). The obtained equation of the *IRS* will allow us to choose several sets of "parameter games" so as to make the system more flexible. It is very important to mention the fact that for all the sets of "parameter games" the response to be optimized will always remain "admissible". That means that in an acceptance interval of the response established by experts or by engineers a priori while respecting specifications, the response will no longer represent a single value "target", but an interval. Specifications often take the shape of a target value (the nominal value) m with the bilateral tolerance Δ_i. It is an error to think that such a specification means that all values included between $m-\Delta_i$ and $m+\Delta_i$ will also have the same low quality. Therefore the "engineering of the target" doesn't eliminate the need of tolerances. The existence of tolerances will also confer certain flexibility to the manufacturing process and therefore will increase the chances of products' acceptance within the bearable limits so as to be functional. This new method will bring flexibility in adjusting parameters to find the optimum of a manufacturing process, specifically for multiresponse optimization where the probability to "play" on the sets of parameters to find an acceptable optimum is not as high.

6. Application

In order to illustrate the proposed approach we present a stamping process optimization problem. Stamping process (Figure 7) is one of the most commonly used manufacturing processes in modern manufacturing. Aiming at meeting the ever-increasing demands for product quality and productivity, many approaches are proposed to simulate the metal stamping process [12, 14, 15, 17, 18, 19]. Nevertheless, the stamping process is a complicated process affected by many material and process parameters like the strain hardening exponent, elastic and plastic deformation of the sheet metals, the sheet thickness, friction and lubrication, punch velocity, blankholder force, *etc*. [20].

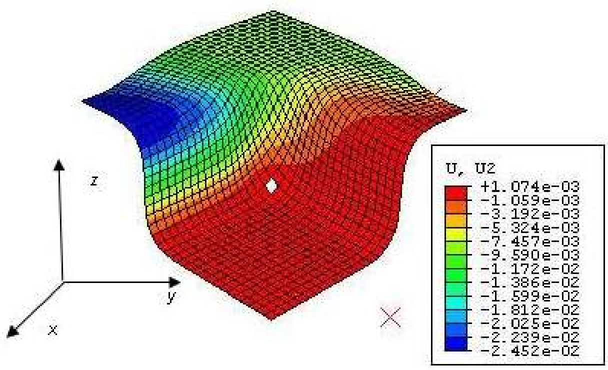

Figure 7. FEM of metal stamping process.

Typical for complex sheet metal forming parts is an inhomogeneous deformation, which leads together with elastic plastic material proprieties and the form drag, caused by the geometry of the part, to deviations from the desired shape. This phenomenon is called springback [30]. Unknown geometrical deviation due to springback is a serious problem of sheet metal forming process, especially in the automotive industry. In this work a Finite Element Model (*FEM*) was developed using ABAQUS [16] code program (Figure 7) to obtain the numerical simulations. The proposed method is applied in order to reduce the process variability involving for the springback of sheet metal stamping operations. This example used a Central Composite Design (*CCD*) with 3 variables, that is, X_1 (Thickness), X_2 (Friction), X_3 (Blankholder Force) (Table 1). This design provides five levels for each design variable ($\pm\delta$, ± 1, and 0 - Table 1). Each level for each corresponding parameter is written in interval form in term of real values (Table 1). It is important to mention that the two interval limits are bilateral mechanical tolerances ($\pm\Delta_i$) for each parameter. The response for this work is Y_1 = Springback (Table 2).

Table 1. Coding of the parameters – interval form.

Parameter	Levels				
	$-\delta \pm \Delta_i$	$-1 \pm \Delta_i$	$0 \pm \Delta_i$	$1 \pm \Delta_i$	$\delta \pm \Delta_i$
X_1	[0,00073; 0,00080]	[0,00086; 0,00095]	[0,00105; 0,00116]	[0,0012; 0,0014]	[0,00136; 0,00151]
X_2	[0,057; 0,063]	[0,086; 0,095]	[0,0124; 0,0137]	[0,162; 0,179]	[0,19; 0,21]
X_3	[16906; 18685]	[18525; 20475]	[20900; 23100]	[23275; 25725]	[24894; 27515]

Table 2. Interval Design Matrix of CCD.

Runs	X_1	X_2	X_3	Y_1
1	[0,00086; 0,00095]	[0,086; 0,095]	[18525; 20475]	[0,000106; 0,000118]
2	[0,00086; 0,00095]	[0,086; 0,095]	[23275; 25725]	[0,000101; 0,000112]
3	[0,00086; 0,00095]	[0,162; 0,179]	[18525; 20475]	[0,000102; 0,000113]
4	[0,00086; 0,00095]	[0,162; 0,179]	[23275; 25725]	[0,000096; 0,000106]
5	[0,0012; 0,0014]	[0,086; 0,095]	[18525; 20475]	[0,000056; 0,000062]
6	[0,0012; 0,0014]	[0,086; 0,095]	[23275; 25725]	[0,000050; 0,000056]
7	[0,0012; 0,0014]	[0,162; 0,179]	[18525; 20475]	[0,000053; 0,000058]
8	[0,0012; 0,0014]	[0,162; 0,179]	[23275; 25725]	[0,000047; 0,000052]
9	[0,00073; 0,00080]	[0,0124; 0,0137]	[20900; 23100]	[0,000124; 0,000137]
10	[0,00136; 0,00151]	[0,0124; 0,0137]	[20900; 23100]	[0,000040; 0,000045]
11	[0,00105; 0,00116]	[0,057; 0,063]	[20900; 23100]	[0,000080; 0,000088]
12	[0,00105; 0,00116]	[0,19; 0,21]	[20900; 23100]	[0,000072; 0,000080]
13	[0,00105; 0,00116]	[0,0124; 0,0137]	[16906; 18685]	[0,000078; 0,000086]
14	[0,00105; 0,00116]	[0,0124; 0,0137]	[24894; 27515]	[0,000068; 0,000075]
15	[0,00105; 0,00116]	[0,0124; 0,0137]	[20900; 23100]	[0,000074; 0,000081]

Ordinary Least Squared (*OLS*) estimation technique was first applied to the initial data to develop the Ordinary Response Surface Model (*ORSM*) for the Sprinback response Y_1. The equation for generated model (in terms of coded factors) is represented in Table 3. Using the proposed algorithm (Figure 4) for the data in Table 2, the Interval Response Surface Model (*IRSM*) is developed for the response Y_1. The equation for generated model (in terms of coded factors) is as follows (Table 3).

Table 3. Equations for *ORSM* and *IRSM*.

The Model	Y_1 *ORSM*	Y_1 *IRSM*
C	0,00007755	[0,00007655 ; 0,00008071]
X_1	-0,00002619	[-0,000029809; -0,000023571]
X_1X_1	0,00000301	[0,000002509; 0,000003311]
X_2	-0,00000228	[-0,000002708; -0,000002.052]
X_2X_2	0,00000064	[0,000000556; 0,000000704]
X_3	-0,00000302	[-0,000003522; -0,000002718]
X_3X_3	-0,00000038	[-0,000000438; -0,000000342]
X_1X_2	0,00000031	[0,000000259; 0,000000341]
X_1X_3	-0,00000009	[-0,000000109; -0,000000081]
X_2X_3	-0,00000004	[-0,000000047; -0,000000036]

For example the equation 13 (in interval form) of the Interval Response Surface for the Springback is:

$$
\begin{aligned}
Y_1 = {} & [0{,}00007655;\ 0{,}00008071] + [-0{,}000029809; -0{,}000023571]\ X_1 \\
& + [0{,}000002509; 0{,}000003311]\ X_1X_1 + [-0{,}000002708; -0{,}000002.052]\ X_2 \\
& + [0{,}000000556; 0{,}000000704]\ X_2X_2 + [-0{,}000003522; -0{,}000002718]\ X_3 \\
& + [-0{,}000000438; -0{,}000000342]\ X_3X_3 + [0{,}000000259;\ 0;000000341]\ X_1X_2 \\
& + [-0{,}000000109; -0.000000081]\ X_1X_3 + [-0{,}000000047; -0{,}000000036]\ X_2X_3,
\end{aligned} \tag{13}
$$

where $Y_i = [Y_{i\min}, Y_{i\max}]$ and $X_i = [X_{i\min}, X_{i\max}]$.

This equation (Table 3) allow the obtaining of a Tolerancing Response Surface, called *Interval Response Surface*, and represents a new manner for making many products accepted in the manufacturing process optimization. In order to better understand the new method, I would like to say that, the imperfect tolerances control involves important costs during manufacture process of the parts: important rejection rate and contributes to product quality deterioration. Our aim is to reduce the cost of rejected products by a better consideration of the mechanical tolerances. The intervals solutions give statistically more chances for certain products which, can be reject because of certain limitations of quality. This method makes it possible to better exploit the various combinations of the parameters so that the products are acceptable.

7. Conclusions

This paper has described a manufacturing process optimization method that combined the Response Surface Methodology, Interval Computation Method and Mechanical Tolerancing. In this work we proposed a new method to obtain a new Response Surface Methodology called Interval Response Surface used in the process optimization. Using this method more final pieces are produced and accepted in the manufacturing process optimization. Our aim for the future work is to introduce an Interval Statistical Analysis to evaluate the effect of the standard error of the parameter on its interval result, calculated with the Interval Computation Method.

References

1. Alefeld, G. and Herzberger, J. (1983). *Introduction to Interval Computations*. Academic Press.
2. Baguenard, X., Dao, M. and Jaulin, L. (2004). *Solver IntervalPeeler*. Available at, http://www.istia.univ-angers/~baguenar.
3. Benhamou, F., McAllester, D. and Hentenryck, P. Van. (1994). CLP Intervals Revisited. *Proceedings of the International Logic Programming Symposium*, 124-138, Ithaca, NY.
4. Bjorke, O. (1989). *Computer Aided Tolerancing*. Tapir Publishers, Trondheim, Norway.
5. Box, G. E. P. and Wilson, K. B. (1951). On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society*, *Series B*, 13, 1-45.
6. Braems, I., Jaulin, L., Kieffer, M. and Walter, E. (2001). Set computation, computation of volumes and data safety, *Scientific Computing, Validated Numerics*, *Interval Methods*, eds. W. Kraemer and J. W. Gudenberg, Kluwer Academic Publishers, Boston/Dordrecht/London, 267-280.

7. Cleary, J. G. (1987). Logical arithmetic. *Future Computing Systems*, 2, 125-149.

8. Cornell, J. A. (1990). *How to Apply Response Surface Methodology*, Revised edition. American Society of Quality Control, Wisconsin.

9. Dantan, J. Y., Bruyere, J., Bigot, R. and Martin, P. (2003). Spécification géométrique fonctionnelle des engrenages par frontière virtuelle. *Proceedings of International Conference of Integrated Design and Production CPI'2003* – Mekhnès Maroc.

10. Dao, M., Baguenard, X. and Jaulin, L. (2001). *Solver Proj2D*. Available at http://www. istia.univ-angers/~dao/Proj2DV2.zip/

11. Davis, E. (1987). Constraint propagation with interval labels. *Artificial Intelligence*. 32(3), 281-331.

12. De Magalhães Correia, J. P. and Ferron, G. (2004). Wrinkling of anisotropic metal sheets under deep-drawing: analytical and numerical study. *Journal of Materials Processing Technology*, 155-156, 1604-1610.

13. Ding, Y., Jin, J. and Ceglarec, D. (2005). *Process-oriented tolerancing for multi-station assembly systems*. *IIE Transactions*, 37, 493-508.

14. Gerth, R. J. (1997). Tolerance analysis: a tutorial of current practice, *Advanced Tolerancing Techniques*, eds. H. C. Zhang, Wiely-Interscience, New York.

15. Granvilliers, L. (2002). *Realpaver*. Available at http://www.sciences.univ-nantes.fr/ info/perso/permanents/granvil/realpaver/

16. Habbit, Karlsson and Sorensen, Inc. (2001). *ABAQUS/Standard Manual version 6.2.4*.

17. Hammer, R., Hocks, M., Kulisch, U. and Ratz, D. (1995). *C++ Toolbox for Verified Computing*. Springer-Verlag, Berlin.

18. Hansen, E. (1992). *Global Optimization Using Interval Analysis*, Marcel Dekker, Inc., New York.

19. Incandela, O., Tabourot, L., Porret, P., Balland, P., Arrieux, R. and Ducher, F. (2004). Modelling and analysis of a deep-drawing operation: key factors for successful comparisons between experimental and simulated results. *Journal of Materials Processing Technology*. 155-156, 1105-1110.

20. Jaisingh, A., Narasimhan, K., Date, P. P., Maiti, S. K. and Singh, U. P. (2001). Sensitivity analysis of deep draving process. *9th International Conference on Sheet Metal*, Leuven, Belgium.

21. Jaulin, L., Kieffer, M., Didrit, O. and Walter, E. (2001). *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics.* Springer-Verlag, London.

22. Kearfott, R. B. (1997). Rigorous global search: continuous problems. *Kluwer Academic Publishers*, Dordchet.

23. Milberg, C., Tommelein, I. D. and Alves, T. (2002). Improving design fitness through tolerance analysis and tolerance allocation, *3rd International Conference on Concurrent Engineering in Construction,* 1-2 July 2002, University of California, Berkeley.

24. Montgomery, Douglas C. (2001). *Design and Analysis of Experiments*, 5th edition. Wiley & Sons, New York.

25. Moore, R. E. (1966). *Interval Analysis*. Prentice Hall, Englewood Cliffs, NJ.

26. Moore, R. E. (1979). Methods and applications of interval analysis. *SIAM Studies in Applied Mathematics*, Philadelphia.

27. Neumaier, A. (1990). *Interval Methods for Systems of Equations*, Cambridge University Press.

28. Ngoi, B. K. A. and Ong, C. T. (1998). Product and process dimensioning and tolerancing techniques: a state-of-the-art review. *International Journal of Advanced Manufacturing Technology*, 14, 910-917.

29. Ratschan, S. (2000). *Approximate Quantified Constraint Solving* (AQCS), 861-880, Available at: http://www.risc.unilinz.ac.at/research/software/AQCS.

30. Rohleder, M. and Roll, K. (2001). Springback prediction on a ring taken from a deep drawn cup. *9th International Conference on Sheet Metal*, Leuven, Belgium.

31. Sergent, A., Daniel, D. and Pillet, M. (2003). La Tolérance Type. *Proceedings of International Conference of Integrated Design and Production CPI'2003* – Mekhnès Maroc.

32. Trabelsi, A. and Delchambre, A. (2000). Assessment on tolerance representation and tolerance analysis in assemblies. *Concurrent Engineering: Research and Applications*, 8(4), 244-262.

33. Voelcker, H. B. (1998). The current state of affairs in dimensional tolerancing. *Integrated Manufacturing Systems*, 9, 205-217.

34. Zhang, G. (1996). Simultaneous tolerancing for design and manufacturing. *International Journal of Production Research*, 34, 3361-3382.

Authors' Biographies:

Daniel Lepadatu is a lecturer (Ph.D.) at Technical University "Gheorghe Asachi" of Iasi Romania, he teaches a course on Topograpy. His mainly interest consist in robust design for reduction the variability of the process design for processes optimization. Response Surface Methodology is the most privileged domain in the interested research field with its applications in the process optimization for structural mechanics in civil engineering.

Abdessamad Kobi is a Professor at University of Angers, Frence, he teaches a course on Statistical Process Control, Automatic, Statistic. His interest lies mainly in robust design for product, process improvement, non normal process control and multivariate process.

Xavier Baguenard received the Ph.D. degree in automatic from the University of Angers, France in 2005 at LISA, on using interval methods and constraint propagation for the calibration of robots.

Luc Jaulin was born in Nevers, France in 1967. He received the Ph.D. degree in automatic control from the University of Orsay, France in 1993. He is currently Professor of Robotics at the ENSIETA engineering school Brest, France since 2004. He does his research on underwater robotics using interval methods and constraint propagation.