

Application of Sampling Methods for statistical tolerance analysis

Andreas Stockinger^{1*}, Sandro Wartzack¹

¹ Chair of Engineering Design (KTmfk) of the University Erlangen-Nuremberg, Germany

Abstract

Manufactured parts differ from ideal shape. Therefore tolerances are used in product development in order to constrain the admissible deviations. The impact of tolerances is analyzed in different ways. It is common to employ Monte Carlo Sampling in order to obtain a statistical result by the iterative evaluation of the functional relationship. In contrast to this, Robust Design has established a bunch of sampling methods that are computationally more efficient.

The objective of the paper is therefore to outline the basics of two selected sampling methods. Implementing a case study, advantages and disadvantages of the compared methods are highlighted. A one way clutch assembly, which is known as benchmarking computer aided tolerancing, serves as case study. The functional relationship is nonlinear and therefore of special interest. A numeric comparison was performed between Monte Carlo Sampling (MCS) and Latin Hypercube Sampling (LHS) using MATLAB. It can be found from this study that Latin Hypercube Sampling is quite suitable for the use in tolerance calculation. Mean values and variances can be estimated using lesser samples compared to MCS reliably. Therefore LHS is proposed as suitable sampling strategy especially for time consuming numeric problems in tolerance analysis too.

Keywords: Tolerancing, computer aided tolerancing, sampling methods, Monte-Carlo-Simulation, Latin-Hypercube Sampling, numeric evaluation

* Contact: Dipl.-Ing. Andreas Stockinger, Chair of Engineering Design (KTmfk) of the University Erlangen-Nuremberg, Martensstr. 9, D-91052 Erlangen, E-Mail: stockinger [at] mfk.uni-erlangen.de

1 Introduction and Objective

Modern methodologies and tools enable the product developer to design products taking different aspects like Design to Reliability or Design to Quality into account. The process of virtual validation is one of the pivotal tasks during product development e.g. in automotive industry. Due to the demand for frontloading an intensive application of simulations in product development establishes and there is the need for precise and efficient analysis techniques.

One of the major tasks for engineers is to limit the influence of varying characteristics caused by manufacturing imperfections using tolerances. To support the product developers a plenty of computer aided methods and tools emerged. A high accuracy of simulation results obtained from these tools is demanded while considering the required calculation time. Therefore much effort for the optimization of analysis methods is spent in research (cf. Table 1). Among these techniques Monte Carlo Sampling based analysis is one of the most versatile methods for use in tolerances. It can be applied to linear and nonlinear relationships as well, but suffers from a high amount of samples which can be computationally expensive. Therefore in this paper Latin Hypercube Sampling (LHS) is introduced for use in tolerance analysis instead of Monte Carlo Sampling (MCS). Based on a comparison of input sampling accuracy, numeric effort (calculation time) and quality of the result due to the amount of samples chosen, the benefit from LHS is outlined. The distribution characteristics mean value and standard deviations are quantitatively evaluated performing simulations on a case study problem.

2 State of the art in Computer Aided Tolerancing (CAT)

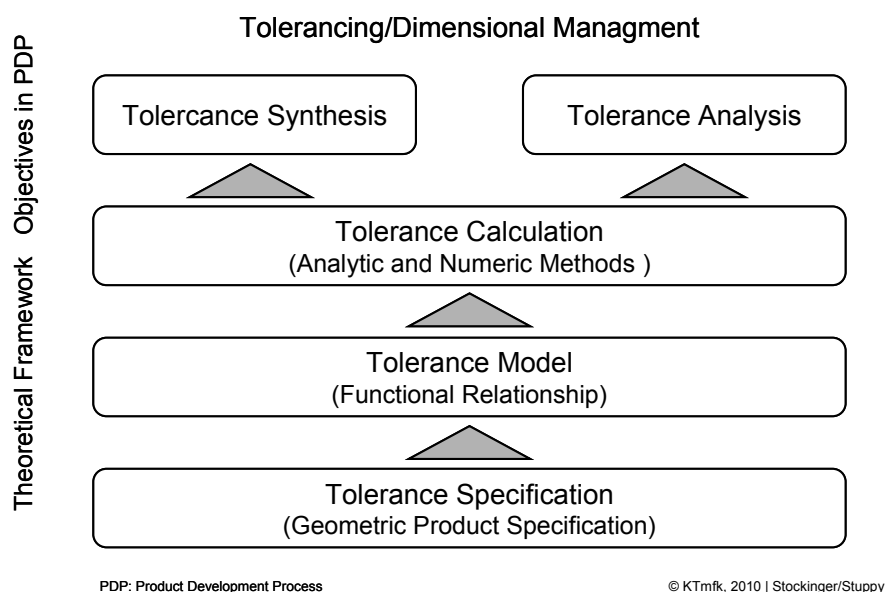


Figure 1: Objectives of and Theoretical Framework in Dimensional Management

Tolerancing is a major part of the so called Dimensional Management (DM). In DM two major objectives exist in the product development process: tolerance synthesis and tolerance analysis. They can only be performed if a consistent model of tolerance specification exists (such as parametric tolerance definitions). It allows specifying the functional behaviour of a system regarding deviations by setting up a tolerance model. (cf. Figure 1)

So a mathematical formulation of the tolerancing problem can be derived [NIGAM 1995]. The relationship of input variables and system responses in a mechanical assembly is often modelled as a vector loop and can consecutively be expressed by

$$y_j = f(x_1, x_2, \dots, x_m) \quad (1)$$

Where x_i ($i \in \{1, \dots, m\}$) are input parameters with known distributions/lower order statistical moments and y_j ($j \in \{1, \dots, k\}$) represent the assembly responses. The assembly responses are also the so called product key characteristics (PKC). They resemble a quality measure which is important for product function or for aesthetic purposes. A simple example of a tolerance chain is shown in the following figure (Figure 2).

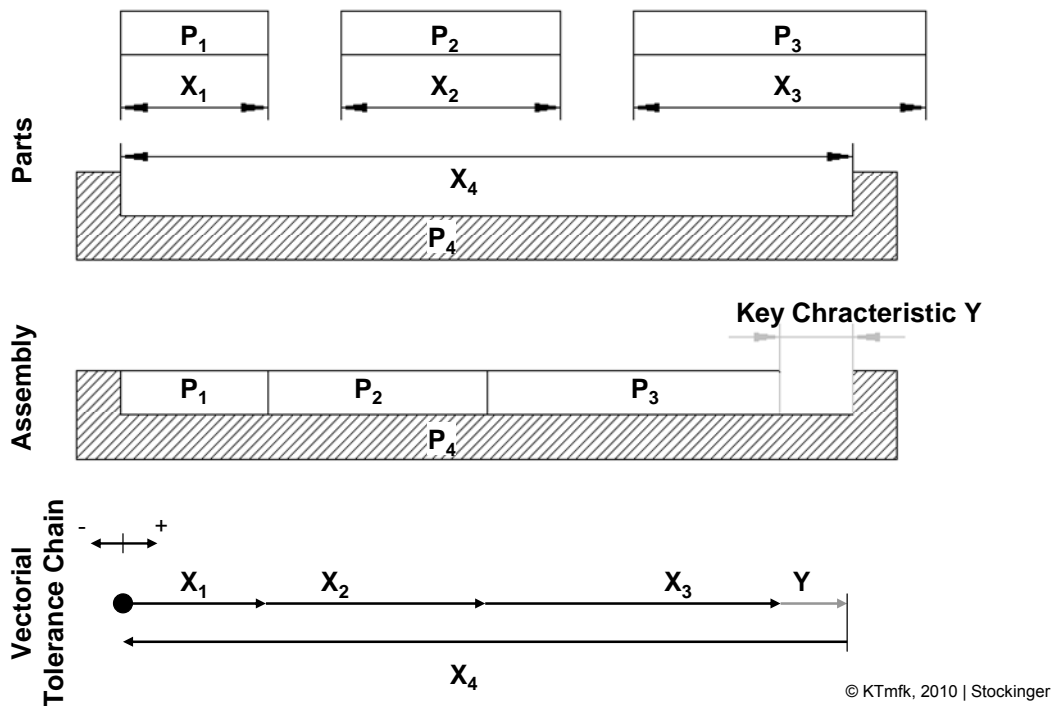


Figure 2: Linear tolerance chain on an assembly [LIU 1995]

Based on the (linear or nonlinear) functional behaviour (1) various calculation methods exist to determine the deterministic and stochastic behaviour of the assembly deviations. Analytic methods – which are limited in their application in some cases – as well as numeric approaches allow the estimation of key characteristic distributions (mean, standard deviation, skewness and kurtosis) based on this functional relationship. A selection of these methods is shown in the following overview:

Table 1: Analytic and numeric methods for use in geometrical tolerance analysis

Analytic methods	Numeric Methods
<ul style="list-style-type: none"> ▪ Worst Case Method [JORDEN 2001, KLEIN 2007], ▪ Root Sum Square [JORDEN 2001, KLEIN 2007, NIGAM 1995], ▪ Estimated Mean Shift Model [GREENWOOD 1988], ▪ Croft's Method [NIGAM 1995], ▪ Hasofer-Lind-Index [NIGAM 1995] 	<ul style="list-style-type: none"> ▪ Taguchi's Method [NIGAM 1995], ▪ Method of System Moments [GLANCY 1994], ▪ Second-Order Tolerance Analysis [GLANCY 1994], ▪ Monte Carlo Simulation [NIGAM 1995], McCATS [GAO 1996].

In addition to these methods also the sensitivity of a response is of interest – especially during the early stages and the optimization phases in product development. Therefore contributor analysis is performed to obtain information on the most important variables. This analysis is performed using local sensitivity methods. Again, analytic (MANNEWITZ 2005a, MANNEWITZ 2005) as well as numeric methods (e.g. High Low Median Analysis [WISNIEWSKI 1998]), can be found in literature and software applications.

This kind of analysis method is not in the focus of this paper. Please have a look at [STUPPY 2010] for further information and a complex mechanism analysed in a case study.

3 Sampling Methods Fundamentals and Implementation

There are different strategies to obtain a stochastic response from a system. A basic approach among these strategies is to iteratively evaluate the functional relationship of the system by different sets of scattering input parameter combinations. In order to obtain reliable stochastic results, sampling strategies were developed that cover the whole space of the multidimensional parameter space. Among them are Monte Carlo Sampling and Latin Hypercube Sampling which are introduced in detail in this paper.

3.1 Monte-Carlo-Simulation

According to [VDI 4008] the Monte Carlo Sampling is a method for modelling of random variables with the aim to calculate certain properties of probability density functions. The simulation process can be subdivided into the following steps [WISNIEWSKI 1998] (see Figure 3): First of all random variables have to be identified and the most important have to be selected. This is accompanied by the definition of the probability distributions of the input variables. In case of tolerance simulation the random variables are the part dimensions x_i (cf. Figure 2) and the distribution should resemble the tolerance value T_i of dimension x_i .

A simulation of deviating geometry and assembly is carried out by randomly selecting values for all dimensions x_i according to the tolerance specification T_i and the selected distribution type by using the plain Monte-Carlo technique.

The functional relationship is set up by defining the vector loop consisting of x_i and y_i . It allows the computation of the results for quality criteria y_i . The post-processing of the simulation results yields information on the performed measurements on $y_i = PKC_i$ (distributions, mean, standard deviation, process capability indices). The major advantage of this procedure is that the functional behaviour is described by analytical equations, which means evaluation using the Monte-Carlo method can be performed fast. The Monte-Carlo simulation process is outlined in Figure 3.

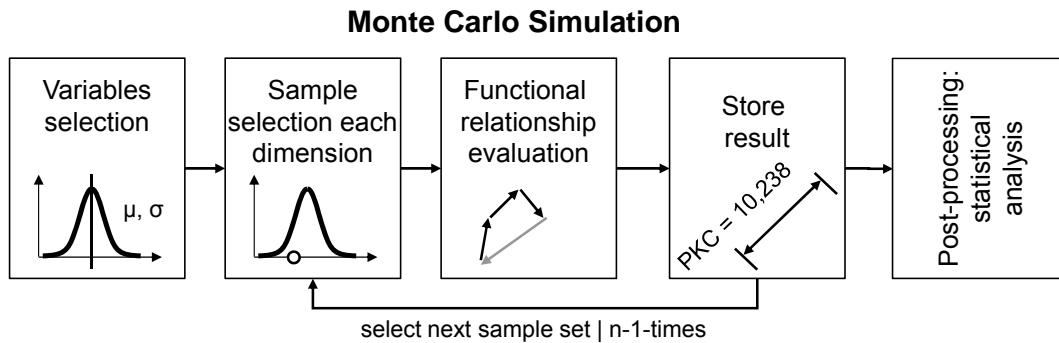


Figure 3: Application of Monte Carlo Method in tolerance simulation

3.2 Latin Hypercube Sampling

An alternative to sample a set of input values according to their probability distribution was proposed by [MCKAY 1979]. The procedure can be outlined as follows (cf. Figure 4): In analogy to MCS also the input variables x_i have to be determined. Moreover the suitable probability distributions of these parameters have to be selected.

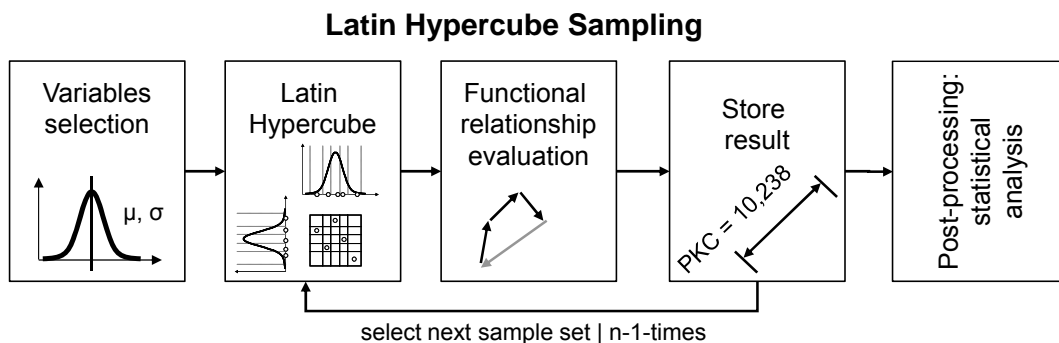


Figure 4: Application of Latin Hypercube Sampling in tolerance simulation
 This information is used to generate input parameter x_i combinations, which take account for the whole range of the parameter scatter and the involved distributions: This can be achieved by dividing the probability distribution into n sections of equal probability (n is the amount of simulations planned). Within these subsections an analysis point is selected randomly. Every input variable is mapped to a vector with n discrete input values resembling the specified distribution. All input parameter vectors are joined into an $n \times n$ matrix after permuting the vector components in such a way that a great area in the whole input parameter space is covered. The n rows of the matrix then contain n simulation input parameter sets

for computation of the functional relationship. This approach is called Latin Hypercube Sampling. Major advantage of this approach is that the admissible range of scatter is exhausted and no part of the probability distribution is left out. This sampling can be used to perform a tolerance analysis by the iterative process shown in Figure 4.

3.3 MATLAB Implementation

Both approaches can be easily implemented, configured and evaluated using MATLAB® Statistics Toolbox. For MCS a matrix of input variables mX is set up containing n samples of x_i per column. Each column is sampled under the assumption of normal distribution (*normrnd*) with a defined mean value (*vMean(i)*) and standard deviation (*vStandardDeviation(i)*).

MCS – Code Snippet

```
%generating random sampling for every variable x=[x1 x2 ...]
for i=1:size(vMean,2)
    mX(:,i) = normrnd(vMean(i),vStandardDeviation(i),nSampleSize,1);
end
```

For LHS a covariance matrix (*mCovariance*) is set up containing the standard deviations on its diagonal (*diag(vStandardDeviation.^2)*). Taking into account dependent variables is possible by filling in covariance values on non-diagonal elements of the symmetric matrix (e.g *mCovariance(2,3)*). Finally the samples are generated using (*lhsnorm*). This results in a matrix of input variables mX containing n samples of x_i per column.

LHS – Code Snippet

```
%generating covariance matrix
mCovariance=diag(vStandardDeviation.^2);
%for dependent variables:
mCovariance(2,3)=mCovariance(2,2);
mCovariance(3,2)=mCovariance(2,2);
%generating latin hypercube samples
mX=lhsnorm(vMean,mCovariance,nSampleSize);
```

4 Case Study

The objective of this paper is to quantify the differences of the two approaches using a case study with a nonlinear relationship based on this basic MATLAB implementation. As case study a well-known problem in computer aided tolerancing was chosen: a one way clutch assembly introduced and analysed by [GLANCY 1994, GLANCY 1999].

This mechanism is analysed here using MCS and LHS in order to be able to answer the following research questions:

1. Does the accuracy of the sampling of input parameters differ? What is a suitable amount of samples for the evaluation of the case study mechanism.
2. Does the computation time differ significantly for the two methods focused here?
3. Do the sampling strategies provide accurate results?

4.1 Problem statement and functional relationship

It is the task to analyse the function of a one way clutch assembly. It transmits torque in a single rotational direction. The assembly consists of a gear shaft (2) which provides torque to be transmitted to the bearing race (4) in case of counter-clockwise rotation. The force is transmitted by four roller wedges (1) which are constrained by four springs (3). “When the hub rotates in a counter-clockwise direction, the roller wedges between the hub and the ring, locking these two parts together. When the hub turns in a clockwise direction, the spring is compressed by the roller, the roller slips, and the hub is allowed to rotate freely.” [GLANCY 1999]. The assembly function can be described by a single vector loop. The function of the mechanism heavily depends on the pressure angle Φ_1 . It furthermore defines the position B of the ball (1) in horizontal direction. (cf. Figure 5)

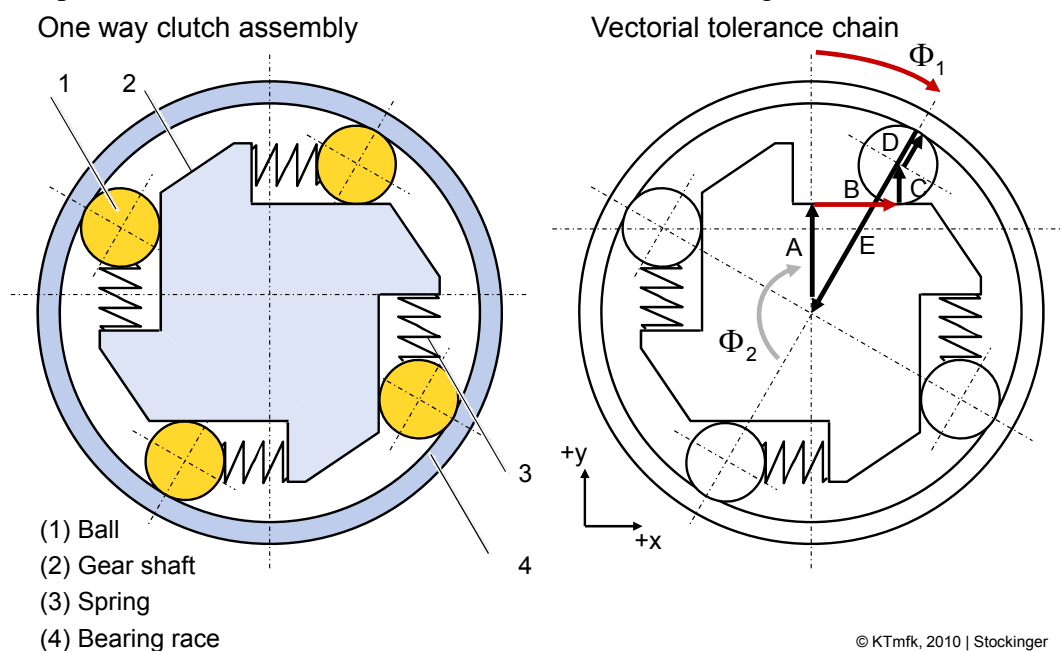


Figure 5: One way clutch assembly geometry and vectorial tolerance chain
The tolerance loop reveals that the pressure angle is influenced by the four dimensions A, C, D, and E; whereas in general all variables are treated independently the variable C is dependent on D and vice versa, because the two dimensions are manufactured during the same process step (cf. Figure 5). The statistical moments of the dimensions as well as the key characteristic specification can be found in Table 2. It is assumed that the type of the distribution is normal: A normal distribution with 6σ -range compared to tolerance $\sigma = \frac{T_i}{6}$. In general any other ratio or distribution type could be used for simulation.

Table 2: Input variable properties and assembly specification [GLANCY 1999]

Input Variables			
Dimension	Mean	Standard Deviation	
A in mm	27.645	0.01666	
C in mm	11.43	0.00333	
D in mm	11.43	0.00333	
E in mm	50.8	0.00416	
Assembly Specification			
Dimension	Nominal	Lower limit	Upper limit
Φ_1 in $^\circ$	7.0184	6.4184	7.6184
B in mm	4.8105	4.4011	5.2195

The pressure angle Φ_1 can be described by two linear dimension chains in x- and y-direction. The position of the ball (1) in x-direction can also be calculated easily using the following nonlinear equations:

$$\begin{aligned}
 X &= E \sin \Phi_1 \\
 Y &= A + C + D \cos \Phi_1 \\
 \Phi_1 &= \arctan\left(\frac{X}{Y}\right) = \arccos\left(\frac{A + C}{E - D}\right) \\
 B &= E \sin \Phi_1 - D \sin \Phi_1
 \end{aligned} \tag{2}$$

4.2 Sampling Setup

The system of equations was solved by sampling with MCS and LHS. To obtain a series of results the example problem was analysed at thirteen different sample sizes: 10; 20; 30; 40; 50; 100; 200; 500; 1000; 5000; 10000; 100000; 1000000. The blocks 2-4 in Figure 3 and Figure 4 were fully implemented in MATLAB in order to obtain automated results generation. The postprocessing was performed in MATLAB and Microsoft Excel.

All calculations were performed on a personal computer with an Intel® Core™i7 CPU Q720 @ 1.6 GHz and 6.0 GB RAM.

4.3 Results and discussion

4.3.1 Sampling of the input variables

The application of the sampling strategies results in the following table, exemplarily shown for input A. Mean values and standard deviations can be compared in dependence of the amount of samples generated. It can be observed that LHS converges more rapidly in mean values than MCS. Furthermore can be observed that both strategies finally meet the desired mean value and standard deviation (at least at 100 samples, cf. Table 2 for given values of A). This is valid for variables C, D, E too.

Table 3: Mean value and standard deviation comparison for input A

Samples	A in mm	A in mm	A in mm	A in mm
	MCS Mean Value A in mm	LHS Mean Value A in mm	MCS Standard Deviation A in mm	LHS Standard Deviation A in mm
10	27.643728	27.644317	0.006606	0.015981
20	27.643205	27.645086	0.016693	0.015959
30	27.642065	27.644839	0.016899	0.016693
40	27.642723	27.644695	0.018300	0.017235
50	27.644847	27.644883	0.015788	0.016366
100	27.643572	27.645034	0.017817	0.016797
200	27.644987	27.644996	0.017396	0.016652
500	27.646914	27.644989	0.015601	0.016646
1000	27.645165	27.645000	0.017550	0.016690
5000	27.645050	27.645001	0.016569	0.016658
10000	27.644806	27.644999	0.016685	0.016660
100000	27.644997	27.645000	0.016688	0.016660
1000000	27.645012	27.645000	0.016682	0.016660

This result is reflected too, considering and calculating the errors of mean and standard deviation in relation to the simulation result with 1 000 000 MC-samples. The result is shown in Figure 6 and Figure 7.

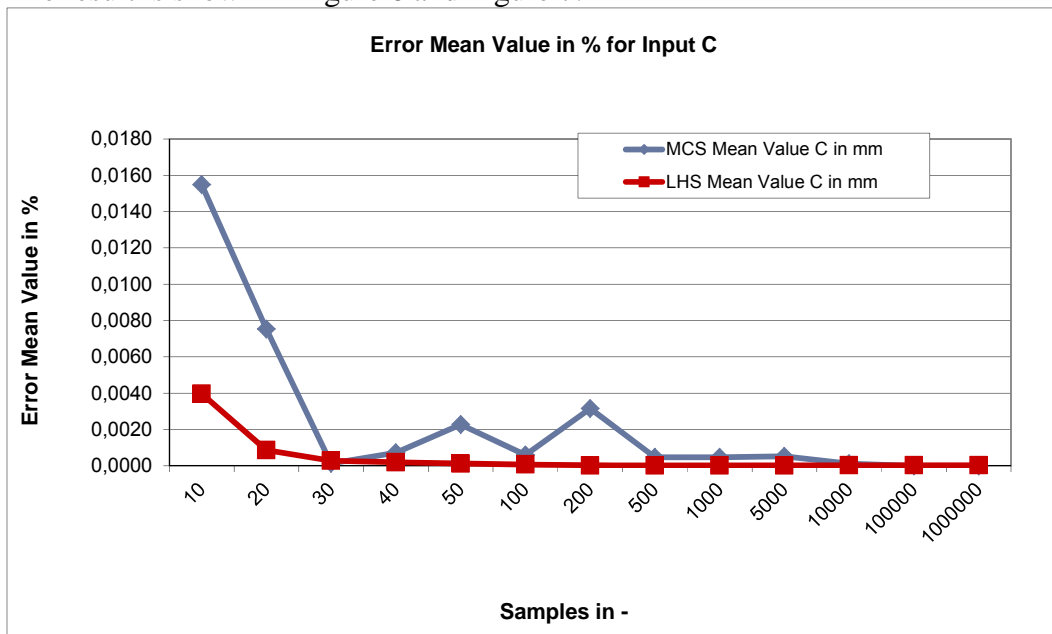


Figure 6: Error of mean values for input variable C

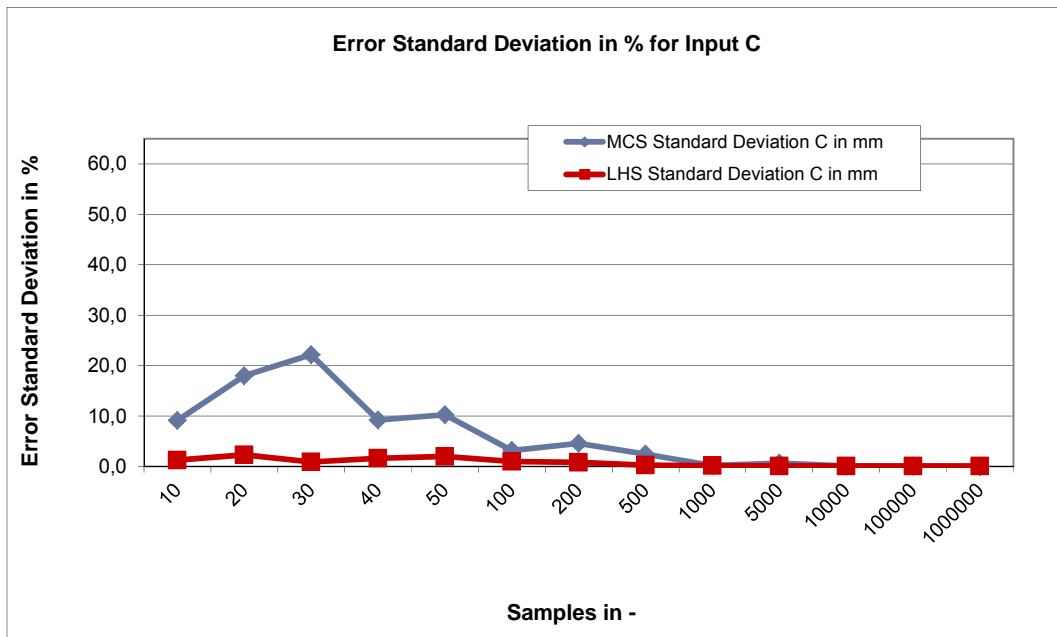


Figure 7: Error of standard deviation for input variable C

4.3.2 Computation time

During the performance of the calculations the computational effort was measured in form of time intervals which are required for sampling and the evaluation of the objective function. CPU-, RAM and file I/O operations were included in the time measurement. In order to avoid deviations caused by initialization issues the time measurement was repeated several times. The last run was used for the evaluation presented here. The following code shows the basic principle of time measurement.

Calculation time – Code Snippet

```

for iCount = 1:length(vSampleConfig)
    %MCM Result and Calculation Time Measurement
    tic
    fNumeric(0,vSampleConfig(iCount),vMean,vStdDev,sFilenameMCM,1)
    vTime(1) = vTime(1)+toc;
    %LHS Result and Calculation Time Measurement
    tic
    fNumeric(1,vSampleConfig(iCount),vMean,vStdDev,sFilenameLHS,1)
    vTime(2) = vTime(2)+toc;
    fWriteTime(vTime);
end

```

Table 4 and Figure 8 show the results on computation time in MATLAB for tolerance simulation using MCS and LHS.

Table 4: Comparison of computation time for MCS and LHS

Samples	MCS Calculation Time in s	LHS Calculation Time in s
10	0,017475	0,019924
20	0,010789	0,011478
30	0,017348	0,012995
40	0,021800	0,016701
50	0,027425	0,025100
100	0,040457	0,038246
200	0,079197	0,072978
500	0,219715	0,178774
1000	0,395009	0,346138
5000	1,701156	1,709271
10000	3,418460	3,326363
100000	33,226791	33,224203
1000000	335,333669	337,462984

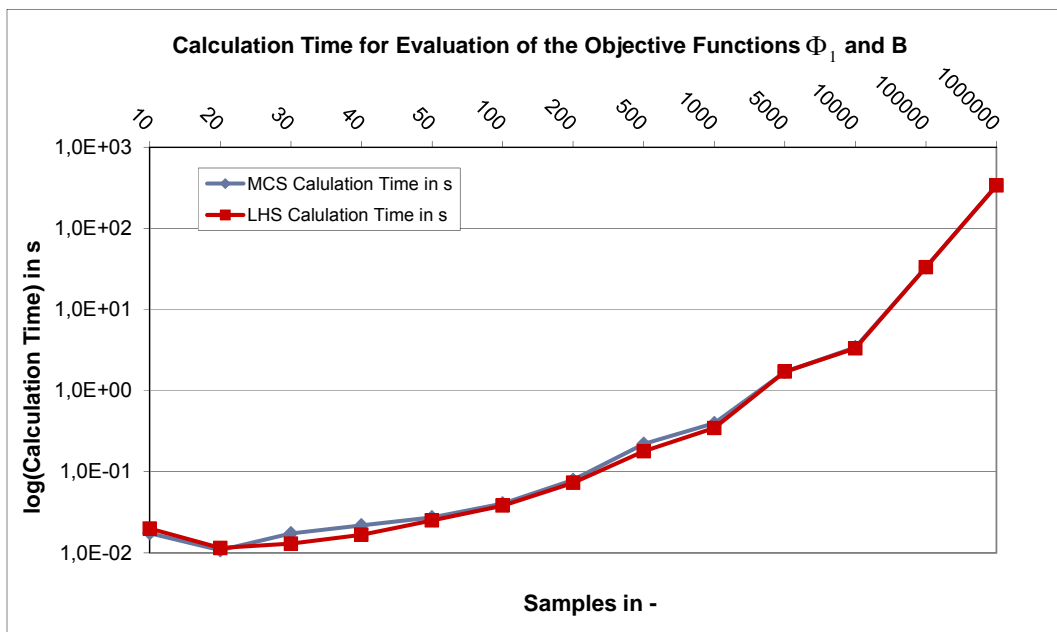


Figure 8: Calculation time comparison

4.3.3 Accuracy of the results from MCS and LHS

The calculation results for the functional relationship are based on the evaluation of equation 2. The MATLAB code for the relationship is an external function which is called by MCS and LHS likewise.

Functional Relationship – Code Snippet

```

%vRatio = ((A+C)/(E-D))
vRatio = (mX(:,1)+mX(:,2))./(mX(:,4)-mX(:,3));
PHI1 = zeros(size(vRatio));
for iCount = 1:length(vRatio)

```

```

%PHI1=acos((A+C)/(E-D))
PHI1(iCount)=acos(vRatio(iCount));
end
%B=E*sin(PHI1)-D*sin(PHI1)
B=mX(:,4).*sin(PHI1)-mX(:,3).*sin(PHI1);

```

Table 5 shows the output for pressure angle Φ_1 which is comparable to the results gained by [GLANCY 1999] (cf. Table 6). Thus the basic calculations are valid. Using LHS sampling 200 runs are sufficient to fit the reference value already.

Table 5: Mean value and standard deviation comparison for output Φ_1

Samples	MCS Mean Value Φ_1 in $^\circ$	LHS Mean Value Φ_1 in $^\circ$	MCS Standard Deviation Φ_1 in $^\circ$	LHS Standard Deviation Φ_1 in $^\circ$
10	7,029528	7,016808	0,094011	0,248979
20	7,040783	7,014244	0,194486	0,193276
30	7,069954	7,017303	0,197701	0,205584
40	7,028787	7,016215	0,220048	0,227451
50	7,014058	7,016044	0,196949	0,218138
100	7,023337	7,013887	0,224252	0,235550
200	7,001659	7,014966	0,224821	0,224955
500	6,991726	7,014906	0,202431	0,225054
1000	7,011048	7,014910	0,221809	0,220332
5000	7,014911	7,015014	0,211620	0,217673
10000	7,014916	7,014966	0,211054	0,219663
100000	7,016755	7,014959	0,212312	0,219597
1000000	7,015027	7,014962	0,212224	0,219483

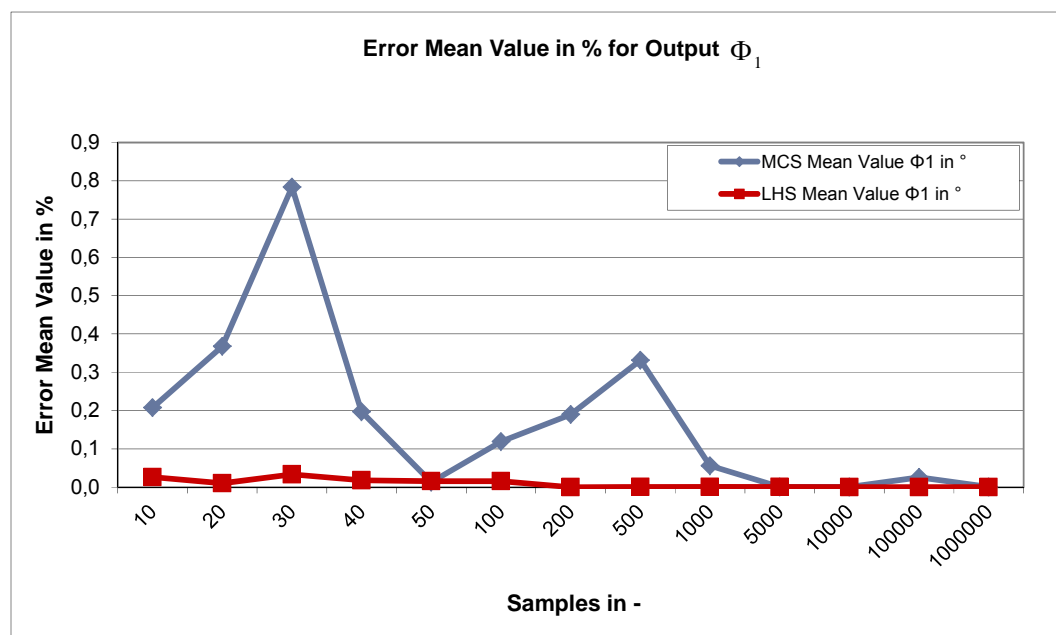


Figure 9: Error of mean value for output variable Φ_1

Figure 9 and Figure 10 reflect the trend for pressure angle Φ_1 too, considering and calculating the errors of mean and standard deviation based on the simulation compared to the particular reference result [GLANCY 1999] (cf. Table 6).

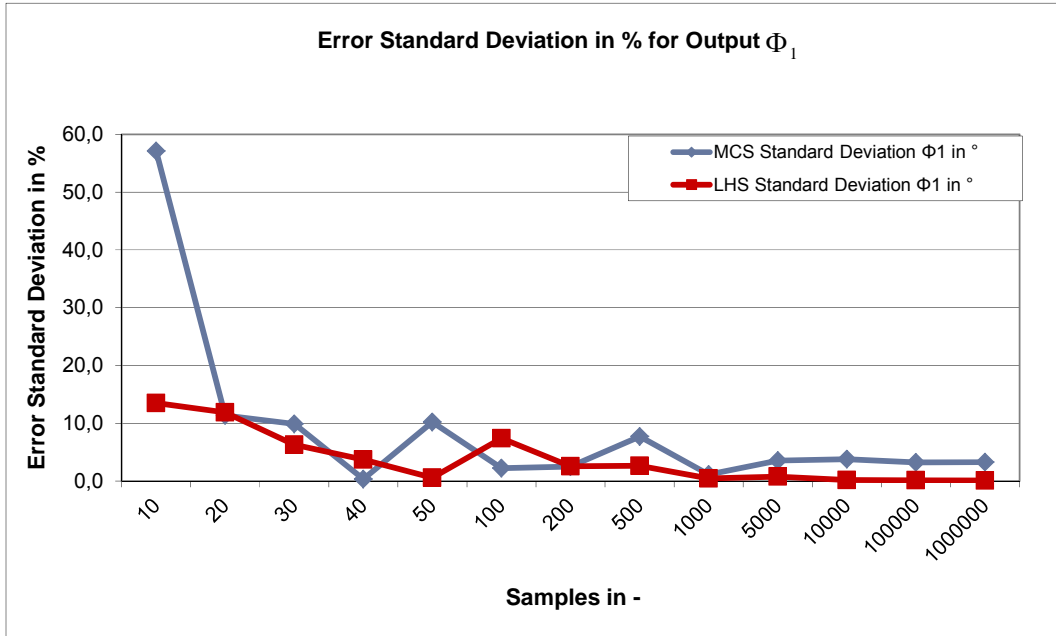


Figure 10: Error of standard deviation for output variable Φ_1

Thus B resembles a derived value, Figure 11 and Figure 12 reveal the identical behaviour for B that can be observed for pressure angle Φ_1 . Again the errors of mean and standard deviation obtained from MCS- and LHS-simulation are calculated related to the particular reference result [GLANCY 1999] (cf. Table 6).

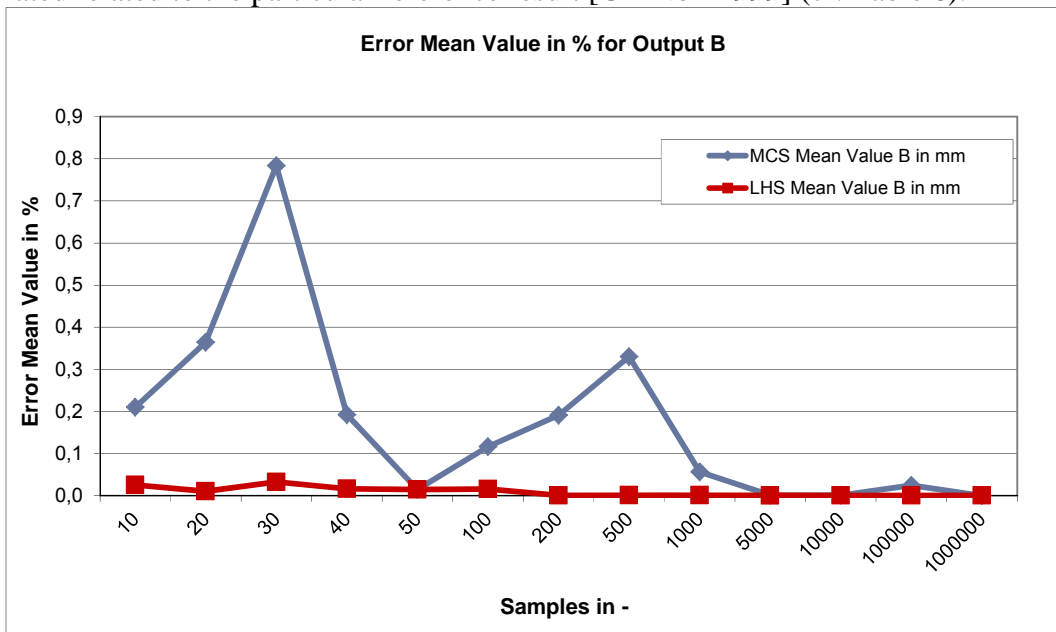


Figure 11: Error of mean value for output variable B

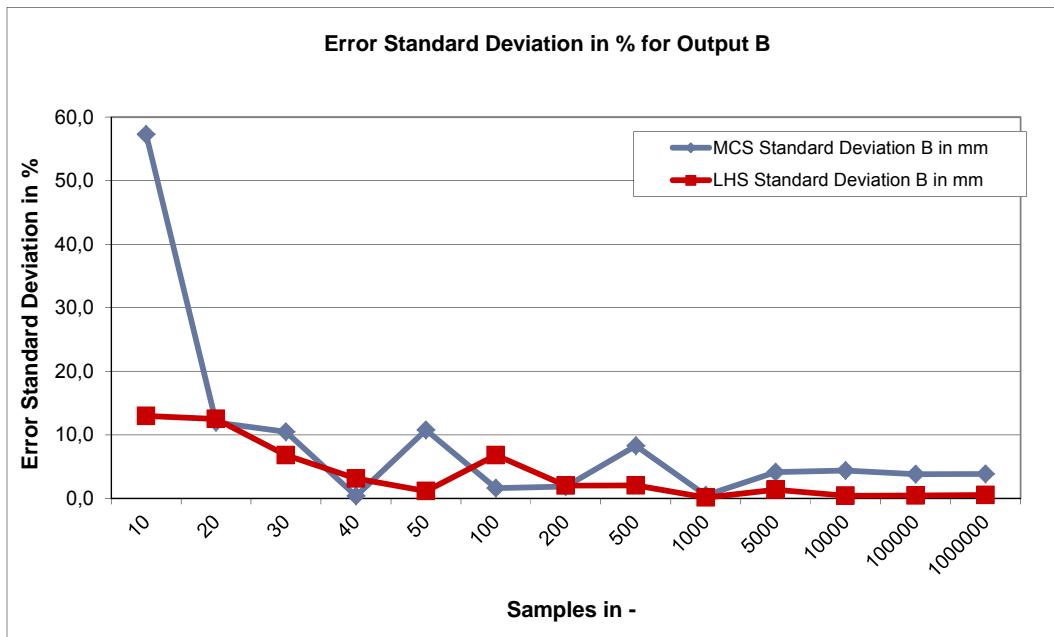


Figure 12: Error of standard deviation for output variable B

Table 6: Second-Order-Tolerance-Analysis (SOTA) Reference Result [Glancy 1999].

Method	Φ_1 in °	Standard Deviation Φ_1 in °	B in mm	Standard Deviation B in mm
SOTA	7,014968	0,219346	4,808204	0,150720193

5 Conclusion

The objective of the paper was to outline the basics of two selected sampling methods and to perform a quantitative comparison. In a case study of a nonlinear tolerance analysis problem, advantages and disadvantages of Monte-Carlo Simulation and Latin Hypercube Sampling are identified.

The sampling principles and the nonlinear relationship were implemented in MATLAB in order to obtain tolerance analysis results for a one way clutch assembly, which is known as benchmark in computer aided tolerancing. The results were validated also by comparing them to published results from [GLANCY 1999]. For this paper some research questions were derived (cf. chapter 4, p. 6), which can be answered here finally:

Question: Does the accuracy of the sampling of input parameters differ?

Answer: Using a small amount of samples, the accuracy of mean / standard deviation differs up to 0.012 % / 60%. This highlights, that there is need for caution setting up samplings. The sampling of the input has to be tested statistically already. Moreover it can be found, that LHS performs better, especially regarding mean values.

Question: What is a suitable amount of samples for the evaluation of the case study mechanism?

Answer: The suitable amount of samples depends on a) the problem (not proved here) b) the criteria to be evaluated (which means: Are the observations focussed on the mean values or the standard deviations or even on higher statistical moments?) and c) on the admissible error. For this one way clutch tolerancing problem about 50 samples will be suitable for predicting the mean using LHS. In order to predict the standard deviation of the quality criteria observed (Φ_1 , B) about 200 simulations have to be performed.

Question: Does the computation time differ significantly for the two methods focused here?

Answer: Using modern tools like optiSLang and MATLAB no significant disparity concerning the setup time between the sampling strategies can be found. Thus the effort for generation and evaluation of samples is the same and there are no numeric disadvantages using LHS compared to MATLAB. Moreover the effort to determine reliable and accurate input distribution properties (mean, standard deviation, skewness, and kurtosis) is more time consuming than the numeric process in many cases in the opinion of the authors.

Question: Do the sampling strategies provide accurate results?

It can be derived from the data in chapter 4.3.3 that the quality criteria (Φ_1 , B) can be predicted safely by choosing a sufficient amount of samples. The values obtained from this study perfectly fit the values from the benchmarking simulations performed by [GLANCY 1999]. The necessary amount of samples is significantly smaller for LHS than the amount of samples using MCS.

Finally it can be found from this study that Latin Hypercube Sampling is **quite suitable** for the use in tolerance calculation. Mean values and variances can be estimated using lesser samples compared to MCS reliably. Therefore LHS is proposed as suitable sampling strategy especially for time consuming numeric problems in tolerance analysis too.

This work served and serves as a basis for further stochastic simulations of manufacturing processes, assembly processes and operating conditions of assemblies which are much more time consuming than the work presented here [STOCKINGER 2010].

Aknowledgements

The authors acknowledge support from the German Research Foundation DFG, grants ME 1029/15-1.

6 References

Literatur

GAO, J.; CHASE, K. W.; MAGLEBY, S. P.: A New Monte Carlo Simulation Method for Tolerance Analysis of Kinematically Constrained Assemblies, Mechanical Engineering Department, Brigham Young University, 1996.

- GLANCY, C. G.: A Second-Order Method for Assembly Tolerance Analysis, Department of Mechanical Engineering, Brigham Young University, 1994.
- GLANCY, C., CHASE, K.: A Second-Order Method for Assembly Tolerance Analysis. In: *Proceedings of the 1999 ASME Design Engineering Technical Conferences*. September 12-15, 1999, Las Vegas, Nevada.
- GREENWOOD, W. H.; CHASE, K. W.: Design Issues in Mechanical Tolerance Analysis, ADCATS Report No. 87-5, Reprinted from *Manufacturing Review*, ASME, Band 1, Heft 1, pp. 50-59, 1988.
- JORDEN, W.: Form- und Lagetoleranzen. 2. Auflage. München: Carl Hanser Verlag, 2001.
- KLEIN, B.: Prozessorientierte statistische Tolerierung. Renningen: expert-Verlag, 2007.
- LIU, S.: Variation Simulation for Deformable Sheet Metal Assembly. 1. Auflage. Michigan: UMI Dissertation Services, University of Michigan, 1995.
- MANNEWITZ, F.: Baugruppenfunktions- und prozessorientierte Toleranzaufweitung, Teil 1: Die richtige Toleranzfestlegung. In: *Konstruktion 10/2005*, Düsseldorf: Springer-VDI-Verlag GmbH & Co. KG, 2005.
- MANNEWITZ, F.: Baugruppenfunktions- und prozessorientierte Toleranzaufweitung, Teil 2: Die richtige Toleranzfestlegung. In: *Konstruktion 11-12/2005*, Düsseldorf: Springer-VDI-Verlag GmbH & Co. KG, 2005.
- MCKAY, M. D.; BECKMAN, R. J.; CONOVER, W. J.: A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, Vol. 21, No. 2 (May, 1979), pp. 239-245. Published: American Statistical Association and American Society for Quality.
- N.N.: VDI 4008. Monte-Carlo-Simulation. Düsseldorf: VDI, 1999.
- NIGAM, S.D.; TURNER, J. U.: Review of statistical approaches to tolerance analysis. In: *Computer-Aided Design*, Volume 27, Number 1, January 1995, pp. 6-15.
- STOCKINGER, A.; WITTMANN, S.; MARTINEK, M.; MEERKAMM, H.; WARTZACK, S.: Virtual Assembly Analysis: Standard Tolerance Analysis Compared to Manufacturing Simulation and Relative Positioning. In: MARJANOVIC D., ŠTORGA M., PAVKOVIC N., BOJCETIC N. (HRSG.): *Proceedings of the 11th International Design Conference DESIGN 2010*, Dubrovnik: 2010, pp. 1421 - 1430.
- STUPPY, J.; MEERKAMM, H.; WARTZACK, S.: Statistical tolerance analysis and result visualisation for systems in motion. In: MARJANOVIC D., ŠTORGA M.,

PAVKOVIC N., BOJCETIC N. (HRSG.): *Proceedings of the 11th International Design Conference DESIGN 2010*, Dubrovnik: 2010, pp. 1431 / 1440.

WISNIEWSKI, D. M.; GOMER, P.: Tolerance Analysis Using VSA-3D® for Engine Applications, *Geometric design tolerancing*, pp. 453-464, 1998.