INTERNATIONAL DESIGN CONFERENCE - DESIGN 2006
Dubrovnik - Croatia, May 15 - 18, 2006

DESIGN 2006

VISUALIZATION OF VARIATION IN EARLY DESIGN PHASES: A CONVEX HULL APPROACH

J. Lööf, R. Söderberg and L. Lindkvist

Keywords: visualization, computer aided tolerancing, variation, convex hull

1. Introduction

As the demand on productivity in the manufacturing industry always increases, it's important with tools that shorten the time between concept and production phases. In the design phase, when initial styling is made and all parts that combine to a complete product have been initialised with nominal values, the tolerances are to be decided. The tolerance setting process is often done with the aid of standard tables and experience. Later on, calculations are made on how the tolerances accumulate to different critical measures (dimensions) on the product. Another method that is preferred is to define critical measures on the product and then allocate the tolerances (see [Hong and Chang 2002] for a comprehensive review). This implies an optimization problem with the constraints on the allowed variation in the critical measures. The allowed variation is collected from functional or aesthetic requirements. Of course, the latter method is more complex. Nonetheless, it results in a product that fulfils the initial demands. Computer aided tolerancing (CAT) software are often used to calculate the accumulated tolerance in defined critical measures. The software predicts how big the variation is in defined critical measures, depending on the choices of tolerances and positions of the locators. The locators define how parts are connected. This is described later in this paper (section 2.2). To calculate the variation, five methods have been used in 2D and 3D tolerance analysis: linearized or root sum square, method of system moments, Hasofer-Lind reliability index, direct integration and Monte Carlo simulation [Chase et al. 1996]. The CAT software used in section three for the collection of simulation data uses Monte Carlo simulation to calculate the variation. Even if the defined measure is fulfilled, the product may have visual defects in critical areas on the product. For example, critical areas on a car are the size of the gap and flush between a door and the fender. The purpose of this paper is to investigate earlier methods regarding the visualization of variation and suggest how to perform visualizations of possible displacements of parts or assemblies. This is done to investigate if there might be collisions between parts. Another application is to visualize measurement data in order to investigate the movement of a part during a given time. One example is how a motor has moved. The outline of this paper is as follows: section two is a brief review of earlier methods regarding the visualization of variation; section three presents three different methods for how to visualize worst case volumes of possible movements of an assembly which are based on simulation data from a CAT software; and, finally, section four includes a discussion and conclusions about the results of the three methods.

2. Visualization of variation

This section presents existing tools and modelling techniques for the visualization of variation. The focus is on capturing earlier techniques about visualizing the variation of an assembly. The variation

in critical measures/dimensions can be interpreted as part variation, assembly variation and the design of the concept (see figure 1). The part variation arises when the parts are produced, since there is always variation in a manufacturing process. Assembly variations arise in fixturing and locating systems. The design of the concept can assist on suppressing the variation.

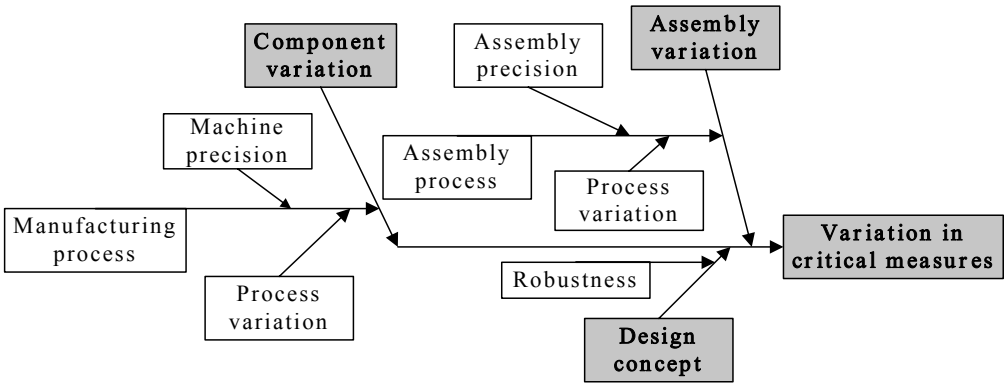

Figure 1. Variation sources [Söderberg 1998]

In early phases in the design process, there is no data available about the production processes. However, it can be simulated by statistical methods, assuming that the process has a statistical distribution. The normal distribution is often used to simulate the process.

There are many computerized tolerancing tools that can simulate variations in different assemblies, but few of them can visualize the result virtually in a high-resolution model. To illustrate variations in assemblies, Lindkvist and Söderberg developed a tool that used colour coding to visualize the simulated variation [Lindkvist and Söderberg 2000]. Tools have appeared in relevant literature that use simulated variation data from commercial CAT software and visualize the result in a different tool. Wickman and others [Wickman et al. 2001] made a connection between a commercial VR (Virtual Reality) tool and CAT technologies to visualize variation in a high resolution model with respect to certain defined critical measures. The connection was made in real time so changes in the CAT software were immediately illustrated in the VR software. This connection illustrated different scenarios. Examples include min, mean and max of the variation taking into account defined critical measures. Another tool is VITAL (Visualizing the impact of tolerances), which comes from a project at the University of Leeds in England. The Vital project consisted of finding a tool to visualize the non-nominal variation of products. The software contained a deformation module that can visualize the effects of component deformation as a result of different geometric and assembly variations (see [Maxfield et al. 2002]). To simulate the deformation of compliant parts, the module uses a combination of FFD (Free Form Deformation) and FDM (Force Density Method). The combination of the CAT software and the visualization tool help designers examine critical areas in an assembly and support the decision-making regarding requirements, tolerances and positioning.

Another related area is the visualization of tolerance zones. These visualizations can be used to inspect the tolerance area interaction between parts in an assembly. They can also be employed to assist in the decision-making about tolerance settings in the design process. One of the earliest models for constructing tolerance zones in a computer environment was proposed by Requicha in [Requicha 1983]. Requicha constructed the tolerance zones by offsetting (expanding or shrinking) the object's nominal boundaries. For a comprehensive review about mathematical methods on construction and visualization of tolerance zones, please refer to [Pasupathy et al. 2003].

2.1 Point variation simulation

Later in this paper, we need to collect simulation data about the variation of each point a part consists of. To get the data we need, a point variation simulation is conducted that simulates the variation of each point of the part. There are two prerequisites: tolerances have been assigned to the part and a

positioning scheme is defined. The scheme used to position the parts is the 3-2-1 principle. It works as follows (see figure 2):

- Three primary points, A1, A2 and A3, represent a plane that locks two rotations and one translation.
- Two secondary points, B1 and B2, lock one rotation and one translation.
- A tertiary point, C1, locks the final translation.

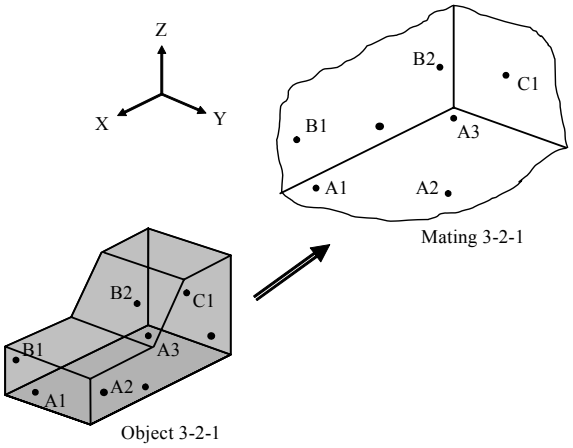

Figure 2. 3-2-1 positioning scheme

When parts are manufactured, the dimensions are centred on its nominal value. This value is often assumed, following a normal distribution with a certain standard deviation. Both the distribution and standard deviation can be defined when tolerances are set. The simulation is based on the Monte Carlo method. This means that, for each simulation, it picks random values for each tolerance using the distribution. In that way, the randomness follows the manufacturing process. After this, we can check the outcome for each point. By each point, we mean every triangle vertex from the VRML file that represents the part or assembly. The user can choose how many simulations will be performed, and all variation data is stored so it can be used later on. When data about the variation can be calculated, the only thing left to do is visualize the result.

3. Worst case volume

From simulation data or from real measurement data one could see how parts or assemblies have moved from different positions. In this section, we would like to capture the movement of a part based on this data. This will imply that we create a total volume the part has generated during its movement. Three different methods have been implemented and tested using the Visual c++.net environment. The first method is an approximation. The second and third are exact.

3.1 Approximations on WC volume

The simulation data used in this application is based on point variation simulation presented in section 2.1. Variation data is collected for each point in the model. By each point, we mean every point in the triangle mesh from the VRML representation of the parts or assemblies. The procedure for creating the worst case volume is as follows: perform a point variation simulation, save all max and min variation for x,y,z, in each point, and draw six parts with each of them with respect to an extreme variation value. This method has been tested on the motor illustrated on the left in figure 3. Drawing the part in its extreme x,y,z variation values implies that a volume is reached. However, we get discontinuities in the corners as presented in the middle of figure 3. To avoid this, we add extra triangles to cover these discontinuities. The result is presented on the right in figure 3.

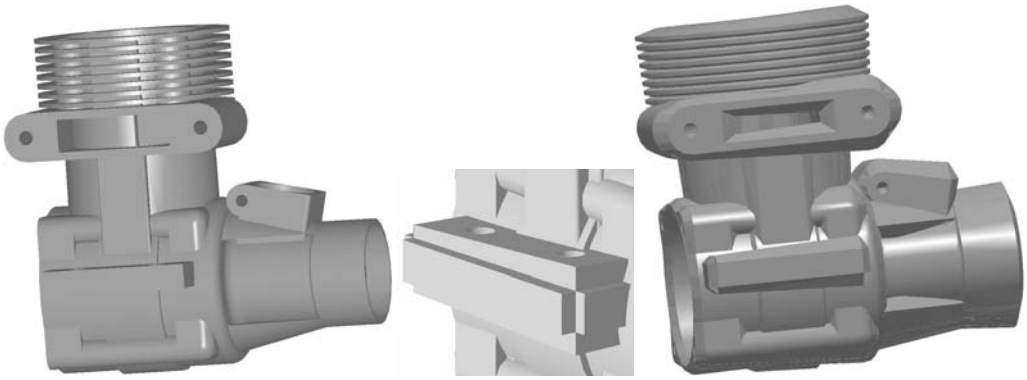

Figure 3. Left: Motor, Middle: Discontinuities, Right: Worst case volume

As mentioned earlier, this feature can be used to investigate how a part has moved when it's being affected by variation. A total volume is created from this movement. This can also be used to check whether there are any collisions between parts or assemblies and to calculate how big these collisions are. Such information provides a hint as to how much we have to lower our tolerances to avoid the collision. In the left of figure 4, another part has been added at the top of the cylinder. We shall check whether there will be a collision when simulating the variation of each point of the motor. The result is illustrated in the right part of figure 4. There is a collision with the other part with this choice of tolerances and positioning scheme.

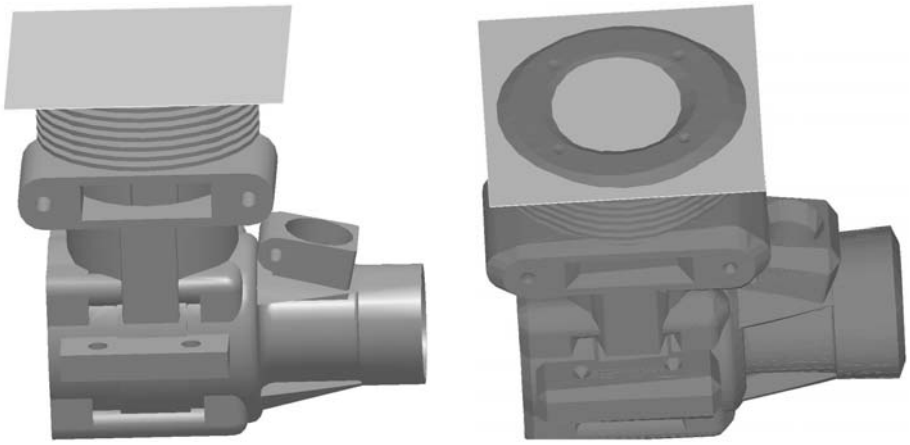

Figure 4. Collision detection

The motor consisted of 7,941 triangles in the VRML representation. This implies 47,646 triangles when the part is drawn in its six extreme positions. Earlier we pointed out that this would imply discontinuities. Since we fill in the edges, the number of triangles rapidly increases. To cover all possible discontinuities, combinations for each triangle have been drawn from the max of x,y,z to the min x,y,z. This results in a total of 762,336 triangles, 96 times more than the origin. Another suggested method (presented in the next section) that can be used to capture the total volume is based on the convex hull algorithm. That also creates the exact volume from the simulation data.

3.2 WC volume using convex hull

This section will present an exact method that covers the total variation volume. This is important, since the simulation data is an assumption of the possible outcome and there might be errors in measurement data. So the exact method avoids the errors accumulating even more. The suggested method uses convex hull, a property that has been used in a wide range of applications. Examples

include pattern recognition, image processing, collision avoidance and shape analysis [O'Rourke 1998]. Instead of going into the mathematical definition of convex hull, it is more important to get an intuitive appreciation of how the method works. Letting all points in a set of points in two dimensions stick out of the plane and then placing a rubber band around all points illustrates the convex hull of that set. Finally, the convex hull is the boundary that the rubber band forms around the extreme points. Generally, the convex hull of a set of points is a closed region, including all the points inside. However, in this paper we interpret the convex hull as the boundary that contains all the extreme points. The convex hull of the point cloud illustrated in the left in figure 5 is presented right beside the point cloud.

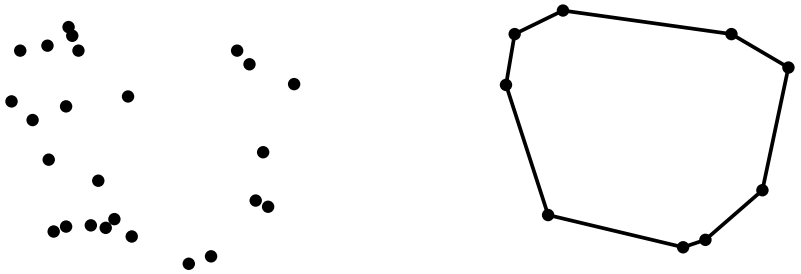

Figure 5. Left: Point cloud, Right: Convex hull in two dimensions

In three dimensions the convex hull can be interpreted as wrapping the point cloud with a paper. The computation of the convex hull in three dimensions can be performed by a number of different algorithms. Some examples are Divide and Conquer, Gift wrapping, Incremental Algorithm and Quickhull [O'Rourke 1998]. The Quickhull algorithm [Barber et al. 1996] is used in the method presented later in this section. The Quickhull algorithm is implemented in the mathematical software Matlab. The complexity of the Quickhull algorithm is O(n log n) in the best case, where n is the number of points that occur when the points are randomly distributed. The Quickhull algorithm has been downloaded as an executable file from the webpage "http://www.qhull.org/". The Quickhull program qhull.exe includes options for the input of files that consist of data that shall be processed and calculates the convex hull. It also consists of options for output data about triangulation and normals for each triangle of the calculated convex hull.
The algorithm developed and implemented in the Visual C++.net environment calculates the convex hull of each triangle that a part or an assembly consists of. This must be done to each triangle. Otherwise, all holes and detailed shapes in the part would be smoothed out. Sending text files back and forward does the communication with qhull. While very time consuming, this works well on general problems. The algorithm can be described as follows:
For each triangle

- Write measurement or simulation data about the three points to a text file.
- Send the text file to the Quickhull algorithm, which calculates the convex hull for the points.
- Qhull writes a text file about the triangle mesh and normals to each triangle.
- Read and store the triangulated output and data about normals.

The prerequisite for the algorithm is the availability of software that computes the variation for every point included in the specific part; or real measurement data. This implies that if there are 1,000 simulations performed, then 1,000 points exist for each nominal point. To calculate the total volume, we also need a list of all triangles that the part consists of. Since a triangle is convex, the algorithm will produce exact variation volumes for each triangle. The algorithm has been tested on the same motor as in the previous section. The motor is illustrated in the left in figure 3. Data is collected from a point variation simulation that simulates 500 possible variations for each point. Since the origin part consisted of 8,409 points, we have a total of 4,204,500 simulated points. The motor was built up with 7,941 triangles. This means the algorithm is called 7,941 times. In each case it calls the Quickhull algorithm with 1,500 points. Since the triangles have common points, this will imply smooth connections between all the calculated convex hulls (as can be seen in figure 6 in the left).

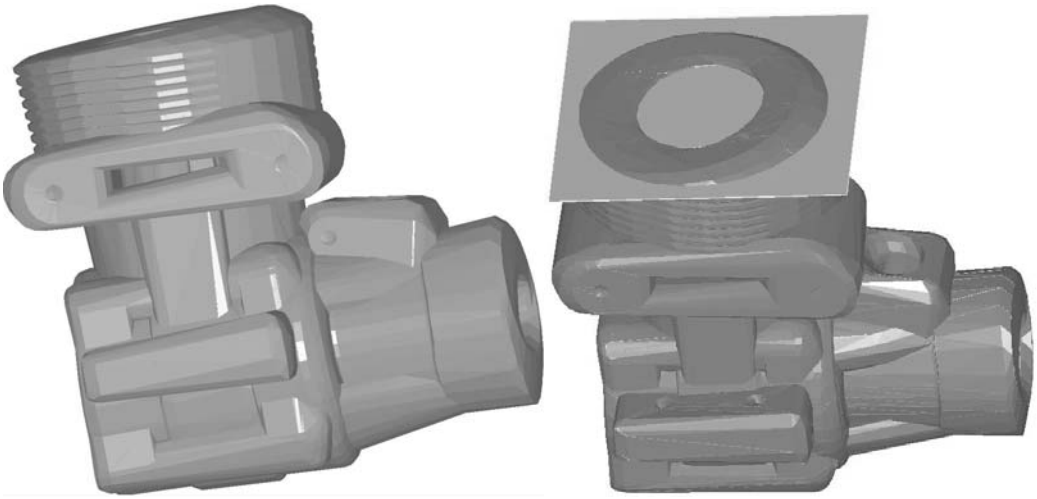

Figure 6. Left: Worst case volume, Right: collision

A similarity with the earlier method presented is that it also produces a lot of triangles - namely 691,006, about 87 times the original size. Many of these triangles can be removed since a lot of them lie inside the outer boundary. The method introduced in this section can also be used to check whether there are collisions between parts (illustrated on the right in figure 6). Another method that does not produce the same amount of triangles is based on convex partitioning.

3.3 WC volume using convex Partitioning

The method presented in this section will partition a part into subparts that are convex. When this partition is made, each convex subpart has a list of points. That list can be used later on to find corresponding points from simulation or measurement data. The convex partitioning is an important preprocessing step for many geometric algorithms, because most geometric problems are simpler and faster on convex objects than on non-convex ones. We are better off whenever we can partition a non-convex object into a small number of convex pieces. This is because it is easier to work with the pieces independently than with the original object. So one approach to an algorithm is the following: first partition the part in convex subparts, then calculate the convex hull for each subpart based on variation data, and finally combine all subparts to one part.

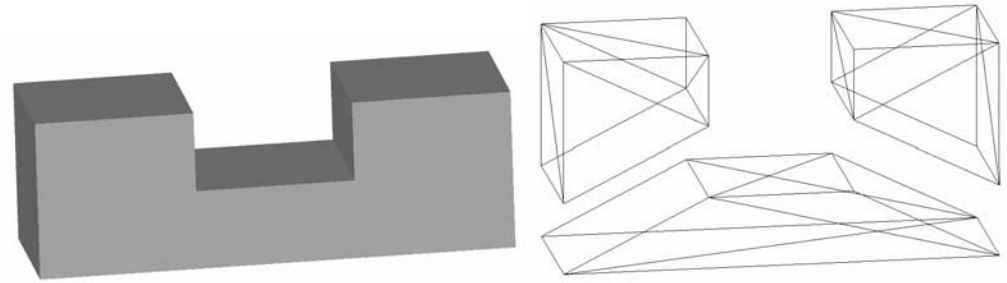

Figure 7. Part partitioning

Unfortunately, we do not have an algorithm for partitioning general parts. Instead, we illustrate the idea of the method by manually partitioning a simple object into convex parts. The part used in this method is presented on the left in figure 7. The part consists of 16 points and 28 triangles. This would imply 28 calls in the algorithm presented in section 3.2. But in this case the part can be divided into three convex subparts, as described in the right of figure 7.

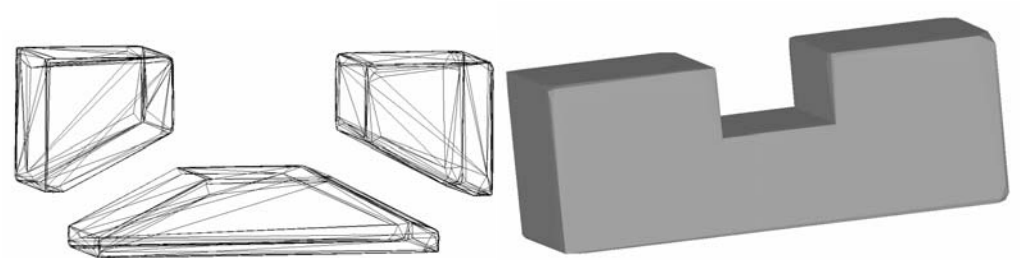

Figure 8. Result from convex partition

To test this method, point variation simulations have been conducted to collect variation points. The triangulated result of the three convex subparts, produced by calling qhull three times with point variation data for each subpart, is illustrated at the left in figure 8. The three parts consisted of 210, 186 and 208 triangles each. This gives a final result of 604 triangles, which is about 22 times the origin. The final result is presented on the right in figure 8.

4. Discussion and Conclusion

This paper has presented three different methods for how to capture the total volume a part or an assembly creates when it is affected by variation. Simulation data has been collected from Computer aided tolerancing software. Nonetheless, the methods also work on real measurement data if the initial part structure is available. The main purposes for capturing the total volume are to visually investigate whether there might be collisions between parts or assemblies and to investigate movements due to measurement data.

There are some drawbacks to the three methods. The first method is only approximate, and it produces too many triangles. However, it works fast on general problems. The second method is time consuming and also produces a lot of triangles. An algorithm that calculates the convex hulls directly, instead of reading and writing to text files is needed. Also an algorithm that reduces the number of triangles is needed to overcome these problems. The volume produced by the second method has been tested on a program that reduces triangles inside the outer boundary. This resulted in a decrease in the number of triangles from 691,006 to 113,211. That corresponds to approximately 14 times the origin number of triangles. But the reduction is most important inside the loop in the second method. There it helps avoid storing unnecessary triangles. The last method is the most promising in the number of produced triangles. However, a general convex partitioning algorithm needs to be implemented and tested on other more general problems (like the motor used in section 3.1 and 3.2). Another interesting question that future work may solve is automatically calculating the total volume of collision. Finally, an investigation shall be conducted on methods that work directly on point clouds. Alfa shapes are one example of such method.

Acknowledgement

The authors would like to acknowledge the Swedish Foundation for Strategic Research, through the research program ProViking, for their financial support, as well as Complex and assembled products, VINNOVA, Swedish Agency for Innovation Systems.

References

Barber, C. B., Dobkin, D. P. and Huhdanpaa, H., "The Quickhull Algorithm for Convex Hulls", ACM Transactions on Mathematical Software, 22(4),1996, 469-483.

Chase, K. W., Gao, J., Magleby, S. P. and Sorensen, C. D., "Including Geometric Feature Variations in Tolerance Analysis of Mechanical Assemblies", IIE Transactions, 28,1996, 795-807.

Hong, Y. S. and Chang, T.-C., "A Comprehensive review of tolerancing research", International Journal of Production Research, 40(11),2002, 2425-2459.

Lindkvist, L. and Söderberg, R., "Tool For Assembly Locating Scheme Definition", ASME 2000 Design Engineering Technical Conferences and Computers and Information in Engineering Conferences, Baltimore, Maryland, September 10-13, 2000,2000,

Maxfield, J., Dew, P. M., Zhao, J., Juster, N. and Fitchie, M., "A Virtual Environment for Aesthetic Quality Assessment of Flexible Assemblies in the Automotive Design Process", SAE 2002 World Congress,, March 4-7, Detroit, Michigan,2002,

O'Rourke, J., "Computatational geometry in C (2nd ed)", "Cambridge University Press", Cambridge, 1998.

Pasupathy, T. M. K., Morse, E. P. and Wilhelm, R. G., "A survey of mathematical methods for the construction of geometric tolerance zones", Journal of Computing and Information Science in Engineering, 3(1),2003, 64-75.

Requicha, A. A. G., "Toward a Theory of Geometric Tolerancing", The International Journal of Robotics Research, 2(4),1983, 45-60.

Söderberg, R., "Robust Design by Support of CAT Tools", 1998 ASME Design Engineering Technical Conferences, September 13-16,1998, Atlanta, Georigia,1998,

Wickman, C., Söderberg, R. and Lindkvist, L., "Toward Non-Nominal Virtual Geometric Verification By Combining VR and CAT Technologies", "Geometric Product Specification and Verification: Integration of Functionality, Edited by P. Bourdet and L. Mathieu, Kluwer Academic Publishers", Dordrecht, 2001.

Johan Lööf, Master of Science
PhD candidate
Division of Product Development
Department of Product and Production Development
Chalmers University of Technology
SE- 412 96 Göteborg, Sweden.
Tel.: +46 31 772 5855
Fax.: +46 31 772 1375
Email: johan.loof@chalmers.se
URL: http://www.ppd.chalmers.se