Datum Optimizer

A Genetic Algorithm Based Tool for 3DCS Advanced Analyzer and Optimizer AAO Add-on

Zesheng Zhang, 3DCS Developer DCS
Maria Harangus, Sr. Dimensional Engineer DCS
Gary Bell, Sr. Dimensional Engineer DCS
Paul Vickers, Sr. Dimensional Engineer DCS
Ben Reese, Director of Marketing DCS

Published October 30, 2020,

Executive Summary

3DCS Datum Optimizer (DO) is a tool that uses a genetic algorithm (GA) to determine the optimal datum features from a candidate set with the goal of minimizing the rigid part variation or compliant part deformation. Using DO during the design stage and prior to production gives users the ability to eliminate or minimize rework, reduce tuning mechanisms to select the proper datum features, and have a higher confidence in passing Gage Repeatability and Reproducibility (R&R) studies.

Improper selection of datum features during manufacturing and assembly may cause severe quality problems. Improper datum features risk turning all the quality parts, accurate dimensioning and tolerancing, and first-class operations into a defective assembly.

Datum Optimizer is an easy-to-use tool in [3DCS Advanced Analyzer and Optimizer Add-on], which does not require creating moves (Assembly Definition) or tolerances on the model. For a rigid part, it only requires a single CAD part with a set of candidate points. For a compliant part, besides the part and candidate points, it additionally requires the [3DCS FEA Compliant Modeler Add-on] and Finite Element Analysis (FEA) files to be ready as it involves FEA calculations.

This white paper discusses the background of Datum Optimizer in the following section. Two functionalities of DO are described in the third section, followed by a brief description of the genetic algorithm. To illustrate the process, an example model is included in the fourth section. Finally, the results are discussed in the conclusion as well as further planned development.

Table of contents

Executive Summary 2

Why Selecting Proper Datums is Critical 4

Understanding 3DCS Datum Optimizer Functions 5

- Primary datum locators for rigid parts 5
- Optimal clamp minimization for compliant parts 5
- Datum Optimizer Objectives 6
- Simulation type 6

Genetic algorithm (GA) 7

Example Case 9

Conclusion 12

Future Outlook 13

References 13

Why Selecting Proper Datums is Critical

As Xing et al. [1] stated, there is little room to reduce variation in manufacturing aspects due to the fact that the current manufacturing level is already close to the precision-control limit. However, many efforts can be done to improve the assembly quality, such as optimizing assembly sequences, fixture locators, and key control characteristics [2-3]. Camelio et al. [4] showed that the positions of fixture locators play a crucial role in reducing sheet metal assembly variation given part and fixture variation. Xing et al. [1] demonstrated that the fixture layout of an inner hood structure was optimized by finding optimal datum features using a non-domination sorting social radiation algorithm which reduced the overall part variation in the assembly.

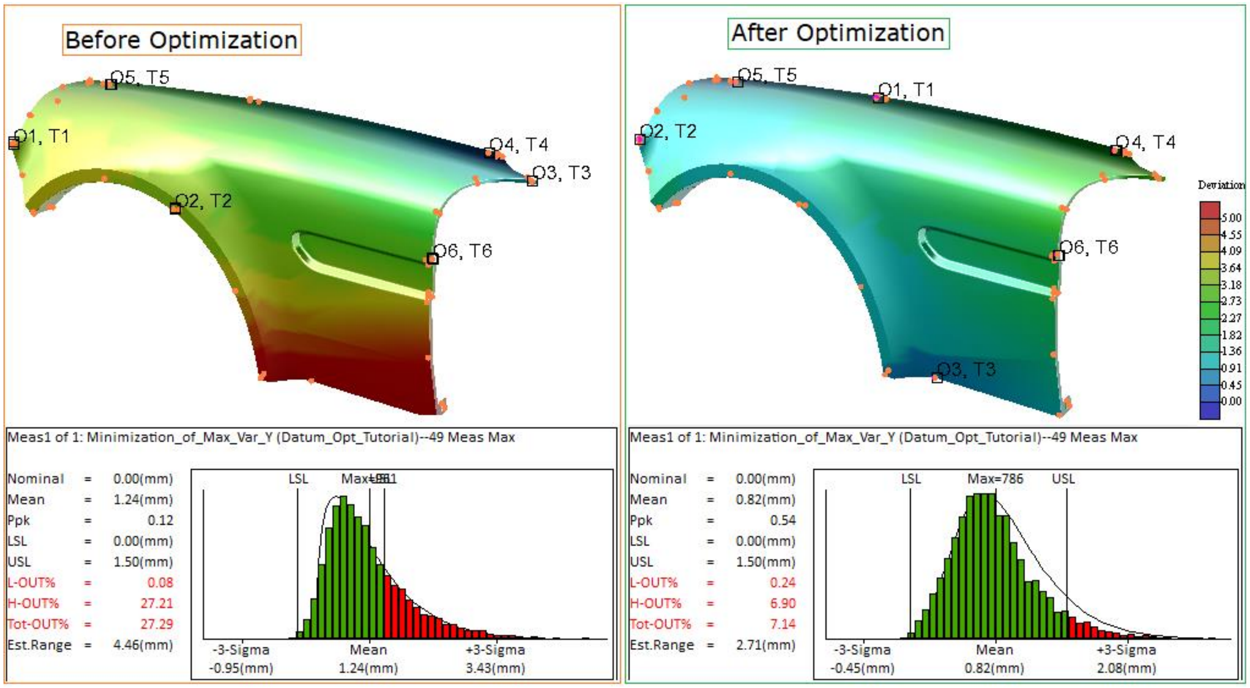
DCS has two optimizers for the assembly process; Sequence Optimizer to improve dimensional quality of assembly and throughput by reordering weld or clamp sequences, while also minimizing the travel distance of the weld gun or clamp operation, and Datum Optimizer, which is aimed at reducing rigid part variation or compliant part deformation caused by gravity and clamp operations by finding the optimal datum features from a candidate set.

Improper datum feature locations can cause large variation, unexpected mean shifts, and distort the part in certain conditions. Both mean shift and variation can cause a significant increase in the percentage of assemblies that are out-of-specification. Additionally, the Process Performance Index (e.g., Ppk) will be adversely impacted.

To demonstrate, the process of locating a fender to a fixture using a 3:2:1 locating scheme was simulated in 3DCS Variation Analyst Software. As shown in the left histogram in Figure 1, using initially selected primary datum features, the model produced a non-normal distribution with large out-of-specification range. Alternatively, as shown in the right histogram, the model using DO-optimized datum locators produced a quasi-normal distribution with much lower risk of out-of-specification conditions.

This example shows how important the proper datum features can be in the assembly process. The following section will discuss the two primary functions of DO.

Figure 1: Monte Carlo simulation results using randomly selected datum locators versus optimized datum locators



Understanding 3DCS Datum Optimizer Functions

Datum Optimizer has two functions in the current version. One is for rigid parts and the other is for compliant parts.

Primary datum locators for rigid parts

This function finds the three optimal primary datum features to locate a rigid part to a fixture. To achieve this, DO uses a step-plane move, applies a constant ± tolerance on the target locators, and measures point displacements over the part to minimize the variation.

To use this function, the user only needs to select the part, pick a set of points as primary datum candidates, and select desired secondary and tertiary constraint points. DO will find the optimal datum features based on the selected objective function, which is explained later in this paper.

Optimal clamp minimization for compliant parts

Flexible parts, especially large sheet metal such as automotive hoods or aircraft skins, may need more than a 3:2:1 locating scheme to control the large amount of deformation and pass Gage R&R. However, adding more locators than necessary drives costs up with no significant value add. DO reduces cost by finding the gain-cost relationship in terms of part deformation versus the number of optimal clamps, and provides the minimum number of clamps and locations for the optimal benefit.

To find the optimal clamp locations, DO requires the following: FEA data to be loaded, constrained degrees of freedom (DOFs) for secondary and tertiary points, applied clamps, applied gravity, and measured points for deformation over the part. Datum Optimizer will combine these inputs and provide the best options to minimize the deformation induced by gravity and clamp operation.

It is important to note that there are no tolerances considered in this function.

Datum Optimizer objectives

Datum Optimizer supports two objectives implemented for both rigid and compliant parts. One is minimization of maximum value, and the other one is minimization of average value. Value here represents variation for a rigid part or deformation for a compliant part.

The goal of minimization of maximum value is to guarantee the critical point to satisfy the specification, and the goal of minimization of average value is to satisfy the overall quality. Users should choose one based on their requirements.

Simulation type

DO has two simulation types - Genetic Algorithm-based optimization or full-factorial simulation. Full-factorial simulation guarantees that users can find global optimal (Best) datum features, but requires more computational resources. The number of possible combinations of n candidates taken r at a time is Cnr =n!/(r!(n-r)!). The number of simulations grows dramatically as the number of candidates and number of datums grows. As shown in Table 1, finding 3 optimal primary datums out of 80 candidates, full-factorial simulation requires 82,160 runs. The Genetic Algorithm-based optimization limits the maximum number of simulations to the population size times the number of generations. With default parameters set, DO only requires 13,144 runs, which is 16% of the full-factorial runs, while having more than a 90% possibility to find the global optimum. This means that you can use only 16% of the computational power to get a 90% chance of success in finding the global optimum. Even though the Genetic Algorithm-based optimization method has an excellent gain-cost ratio, full-factorial simulation is recommended if computational resources are not a concern for you.

Table 1: Comparison of number of runs of full-factorial and Genetic Algorithm-based optimization

Number of Candidates	Number of Primary Datums	Number of Full-factorial Runs	Number of Opt Runs	Run Time Ratio
20	3	1140	480	42.11%
20	6	38760	6200	16.00%
40	3	9880	1576	15.95%
80	3	82160	13144	16.00%

Genetic Algorithm

Genetic Algorithm has been used for solving difficult problems (such as non-deterministic polynomial time hard (NP-hard problems)) and machine learning. Finding the optimal datum features among many candidates is a NP-hard problem, meaning that you are guaranteed to find the global optimum only if you search all combinations of candidates. Therefore, a Genetic Algorithm is a perfect method to use in such problems. The current version of DO utilizes a general Genetic Algorithm [5]. Figure 2 shows the optimization process.

As the name implies, Genetic Algorithm is a class of evolutionary algorithms inspired by natural selection and survival of the fittest. There are a few key parameters users can set up in DO, as follows:

- Population - The number of samples in one generation. It is related to the number of candidates and the number of primary datums.
- Generation - The maximum number of generations that the optimization can evolve
- Crossover rate - The percentage of population that will be generated from parents' mating for the next generation.
- Mutation rate - The percentage of the population that will perform gene mutation to introduce randomness.
- Seed - The beginning seed of the random number generator. The same beginning seed will keep the simulation repeatable. It can be any integer.

The optimization process is described in the following steps:

- Given the candidate points and the number of locators selected, an initial population of solutions is randomly generated. The initial population is a set of different potential solutions.
- Each member of the population is evaluated and ranked using the objective function discussed above.
- Members of the population are randomly selected to mate and create offspring using fitness selection. The best performing members are more likely to be selected to create

the next generation. Meanwhile, a very small portion of the population is selected to perform gene mutation to introduce randomness.

- Second generation is created and follows the same evolutionary process in the initial generation and creates the third generation.
- Evolution will stop once the algorithm satisfies one of the stop criteria, such as reaching maximum generation or convergence. The best result in the last generation will be the optimized results.

As mentioned before, the Genetic Algorithm-based optimization method cannot guarantee to find the global optimum. To improve the possibility, a systematic research process was conducted. It concluded that if the population parameter is related to the number of candidate datums, a higher possibility would be achieved. As shown in Table 2, the averaged possibility of the selected scenarios is more than 90%. However, the possibility is heavily dependent on many factors, such as model, other Genetic Algorithm parameters (crossover rate and mutation rate etc.), and objective function.

Figure 2: Optimization process

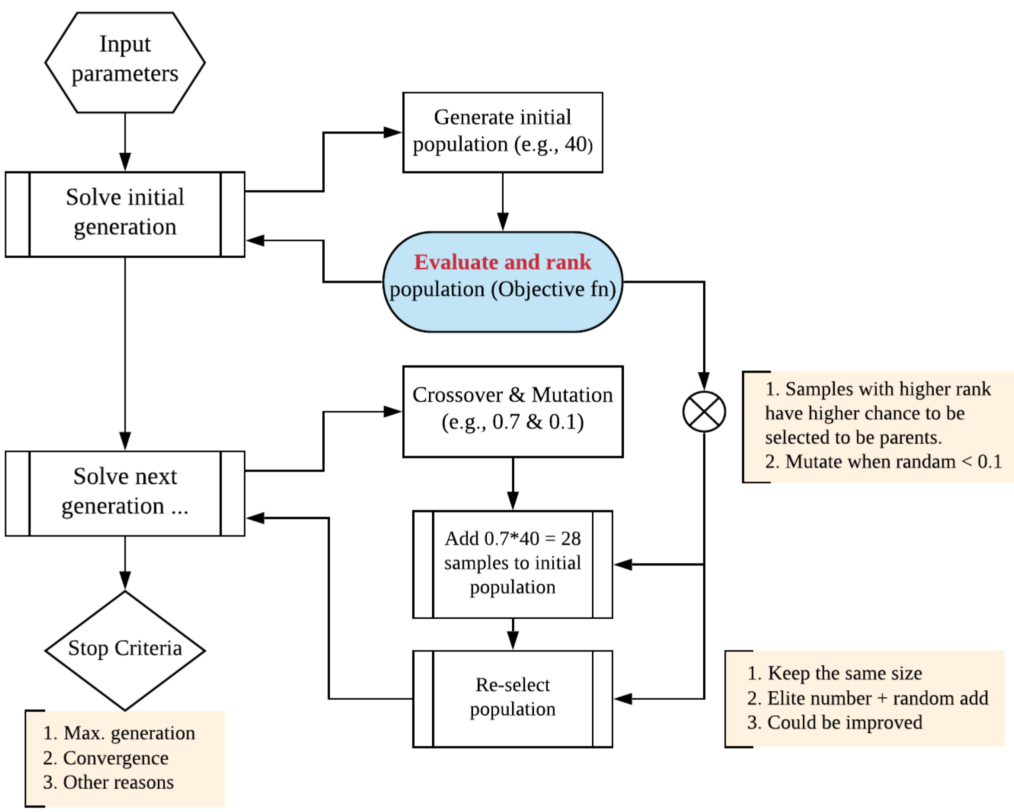


Table 2: Possibility to find global optimum for different scenarios

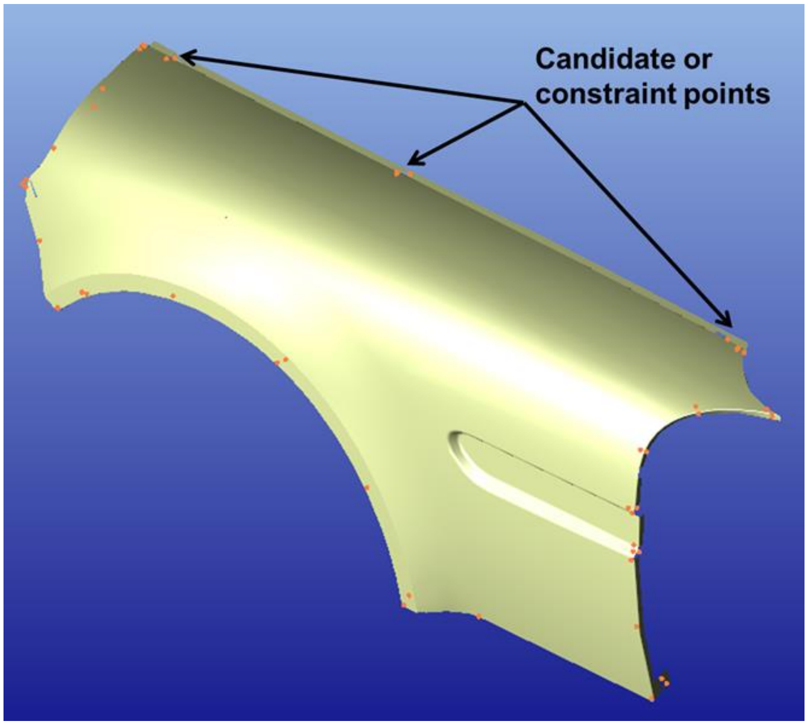
Number of Candidates	Number of Primary Datums	Possibility to find global optimum
10	3	85%
10	4	100%
10	5	95%
10	6	100%
20	3	60%
20	4	90%
20	5	100%
20	6	100%
30	3	85%
30	4	85%
40	3	90%
40	4	80%
50	3	85%
60	3	100%
70	3	100%
80	3	100%
Average		91%

Example Case

As an example of Datum Optimizer, a fender model as shown below was analyzed for both rigid and compliant values. Since DO does not require any assembly moves or tolerances, the fender model is a good example with only a single part and a set of points created as candidate and constraint points, as shown in Figure 3. However, to validate the optimized results, users have to add moves and tolerances to see the difference between using initial datum features and optimal datum features. This is illustrated in detail in the 3DCS Variation Analyst Tutorial and Help Manual [6]. The process of validating is not included herein, only the results are analyzed in this paper.

For the rigid datum feature case, to validate the performance of DO, a step-plane (3:2:1 locating scheme) move was used to locate the fender to a fixture, and a normal distributed tolerance was applied on selected datum targets. As shown in Figure 1, the model with initially selected datum features produced a risk of 26% out-of-specification and a poor process performance value (Ppk = 0.13). After datum optimization, three optimal primary datum features were found. The model with optimal datum features produced an excellent result distribution with only 0.8% out-of-specification, meanwhile increasing the process performance value (Ppk) to 0.81.

Figure 3: A fender model with candidate and constraint points



For the compliant datum feature case, lock DOF (boundary condition), clamp, and gravity moves were used to clamp the fender to a fixture. Part deformation caused by the clamping operation and gravity was measured. Five scenarios were then evaluated under the compliant case. First, DO found the best 3 of the 20 candidate datums. Then DO found the best 4 datums. Because the model is compliant, the part was overconstrained using 4:2:1 locating in the scenario. Best 5:2:1, 6:2:1, and 7:2:1 datum schemes were founded sequentially. As shown in Figure 4, DO not only finds the optimal datum features such as optimal 3, optimal 4 etc., but also reveals the gain-cost relationship when users consider adding clamps to reduce the part deformation. For example, the maximum deformation decreases significantly from 2.44 mm to 0.74 mm (70% drop) when increasing the clamp number from 3 to 5. However, there is only an 18% drop when increasing the number of clamps from 5 to 7. The user should balance the gain and cost to make a decision, even though an 18% drop is still a great improvement.

Figure 4: Datum optimization for compliant fender

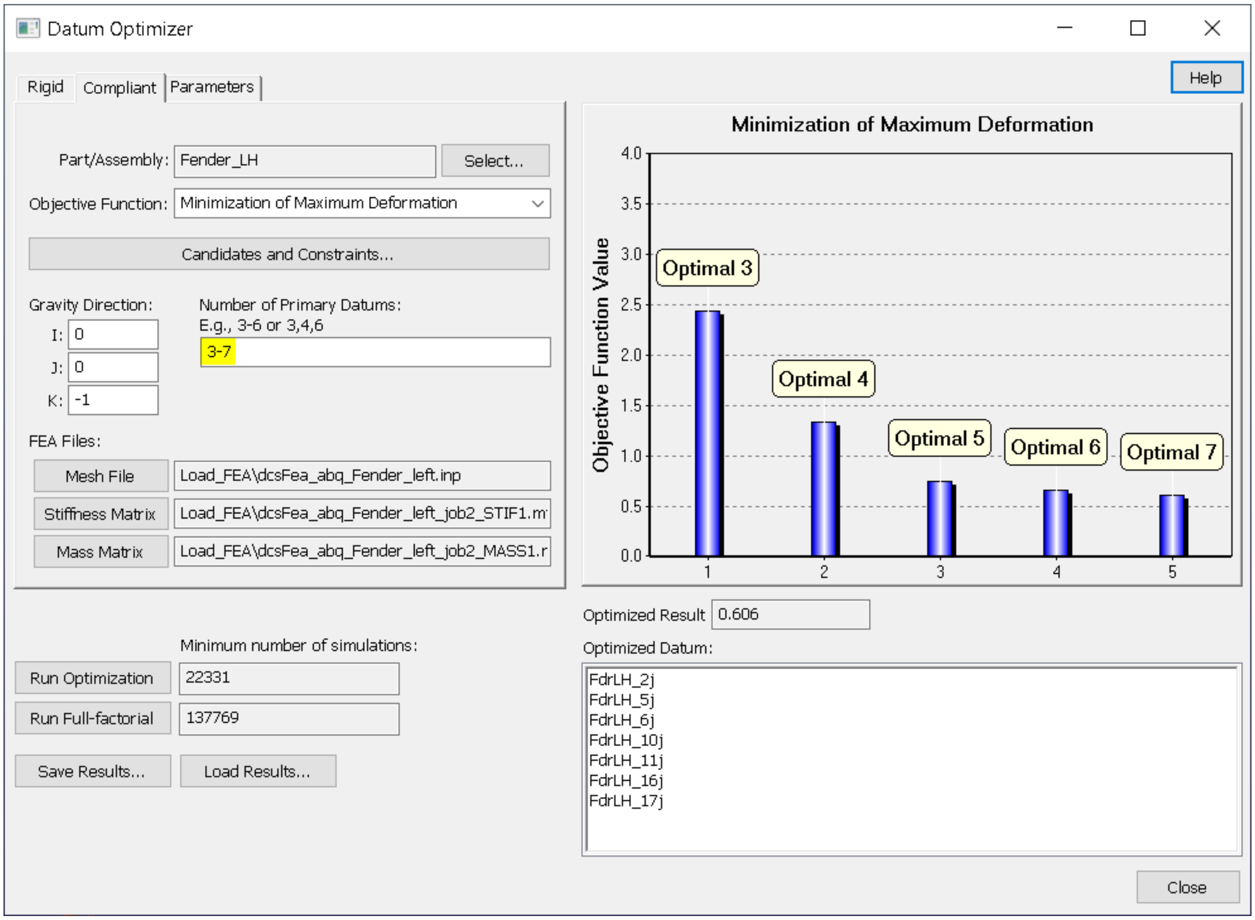
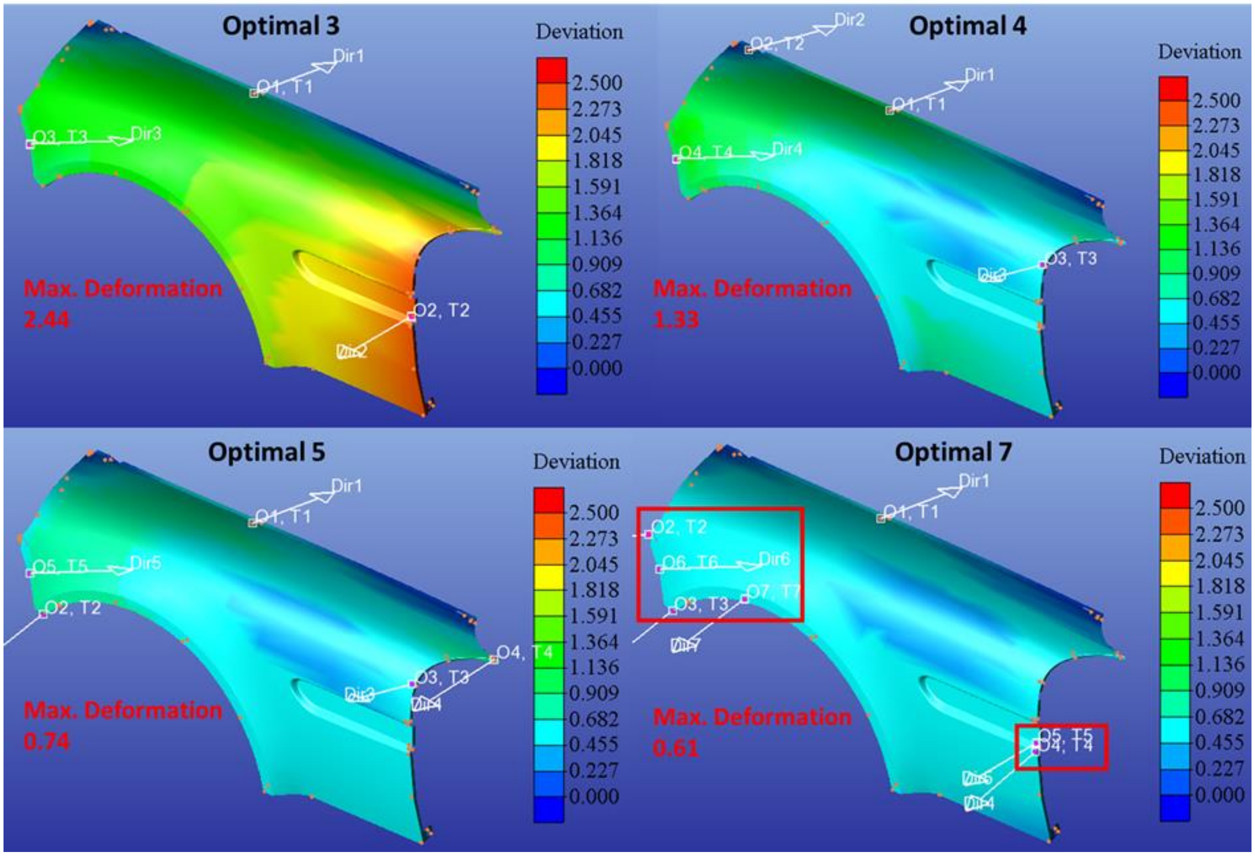


Figure 5 shows the deformation color contour and positions of optimal datum features for different numbers of clamps. As can be shown in the color contour, with the increasing number of clamps, the maximum and overall deformation was reduced significantly. However, in the bottom-right figure, even though the deformation is minimum, multiple clamps are positioned close to each other (within red boxes). In this example, the user should stop adding clamps after finding 5 optimal datum features, since adding two more clamps cannot significantly reduce the deformation, and overcrowding clamps increases the cost. Therefore, DO tells users not only where the optimal locations are to put clamps but also when to stop adding more clamps.

Figure 5: Deformation color contour for different number of optimal clamps



Conclusion

Datum Optimizer is a tool to find the optimal datum features for manufacturing and assembly processes, which does not require any assembly moves or part tolerances on the model. It only requires a single part with a set of candidate points and FEA files if conducting compliant deformation. Using DO during the design stage and prior to production, users can eliminate or minimize the rework and tuning of mechanisms to select the proper datum features while retaining a high confidence in passing Gage R&R.

To achieve the goal of minimizing rigid part variation or compliant part deformation caused by gravity and clamp operations, DO uses a state-of-the-art genetic algorithm to efficiently find optimal datum features from a candidate set.

As shown in the example, DO not only shows users where the optimal locations are but also suggests when to stop adding more locators due to diminishing returns.

Future Outlook

In the current version, Datum Optimizer is designed to deal with primary datum feature optimization at the component level. As such, DCS is developing this tool further. This entails the addition of optimization of coordinated datum features between parts at the assembly level as well as the optimization for combination of primary, secondary, and tertiary datum features. These advancements are expected in the future releases of 3DCS, and will further expand Datum Optimizer to cover more scenarios for optimization.

References

[1] Yanfeng Xing, Min Hu, He Zeng & Yansong Wang (2015) Fixture layout optimisation based on a non-domination sorting social radiation algorithm for auto-body parts, International Journal of Production Research, 53:11, 3475-3490

[2] Hu, S. J. 1997. “Stream of Variation Theory for Automotive Body Assembly.” CIRP Annals – Manufacturing Technology 46 (1): 1–6

[3] Liao, X., and G. G. Wang. 2005. “Employing Fractals and FEM for Detailed Variation Analysis of Non-rigid Assemblies.” International Journal of Machine Tools and Manufacture 45 (4–5): 445–454

[4] Camelio, J., S. J. Hu, and D. Ceglarek. 2004. “Impact of Fixture Design on Sheet Metal Assembly Variation.” Journal of Manufacturing Systems 23 (3): 182–193

[5] Mohammadi, Arash, et al. "openGA, a C++ Genetic Algorithm library." Systems, Man, and Cybernetics (SMC), 2017 IEEE International Conference on. IEEE, 2017

[6]https://www.3dcs.com/tolerance-analysis-software-and-spc-systems/add-ons/3dcs-advanced-analyzer-optimizer

About DCS

DCS is a software developer providing tolerance analysis and quality SPC solutions to the automotive, aerospace, medical device, electronics, and energy industries. With more than 20 years' experience, DCS has grown to include clients from every region of the globe including companies like Airbus, BMW, GM, LG, Jaguar Land Rover, Phillips, Sony, Textron Aviation, and Volkswagen. As a quality solution provider, DCS prides itself on providing clients with not just software, but services, staffing, and dedicated support to guarantee the success of their quality initiatives.

To learn more about DCS and its products, visit 3dcs.com.