Automotive Crash Simulation: A Personal Perspective *

J. Marczyk Ph.D.†

Abstract

This white paper provides a critical overview of current practice in computer simulation of automotive crash. It is argued that contemporary crash simulation techniques, while overlooking the essentially stochastic character of the physics of crash, lead to dangerous misconceptions. Once such misconception, that is gaining popularity, is that the optimization of vehicle crashworthiness is possible. However, phenomena that may be catalogued as chaotic and/or random in nature, are inherently deprived of any form of predictability. It is this lack of predictability that directly precludes optimization. Phenomena of this type, i.e. that are nonrepeatable, require a totally different approach, that is not based on optimization. This new approach is described in this paper.

1 Introduction

I performed my first full-car stochastic crash simulation in August of 1997 with a group of friends and colleagues from the European PROMENVIR research consortium ([1] and [2]). One of the most important findings of that project, which already then contained over seventy stochastic variables, was that the variable which dominated the crash response the most was the angle of impact with the barrier. The result is shown in figure 1, where the footwell intrusion versus angle of impact is depicted. Since then, our knowledge of crash has been rapidly increasing, our models have been getting closer to reality, impressive results have been accumulated. Enough to write a two-volume book. In 1998, the ST-ORM tool was born from PROMENVIR. Since those pioneering days, we have involved in stochastic crash many car manufacturers. However, quite a few still resist to recognize that crash is a stochastic and chaotic phenomenon and should therefore be treated as such. Our studies over the last three years have shown that much better crashworthiness designs can be obtained if the problem is approached with the Monte Carlo method, the natural method for crash.

In systems dominated and characterised by a mixture of chaotic and random behaviour, talking of optimization is simply absurd. Can we optimize a building for an unknown seismic load? Of course, we can make the building as heavy and stiff as possible, and cross fingers. Now suppose that the building has indeed resisted an earthquake. Is it because it was optimal? What is optimality in a similar case? Can you ever prove optimality for a system subjected to unknown and unpredictable outside actions? In car crash, where even in the test lab it is *impossible* to reproduce the same results, the situation is similar. You can't optimize a car for crash when you don't know what other car, or obstacle, it will hit, with which velocity, angle, mass, number of passengers, etc. Because of tolerances in manufacturing and assembly, it is even impossible to manufacture two identical cars. So what can we optimize? Of course, you can always build some deterministic function and then find some local minimum. But what is the practical use of such a result, when in reality, the *real* road conditions are unknown to a very large extent. Why do people ignore hard facts and evidence in the face of reality and physics? The resistance to innovation which pervades the industry is impregnated with a certain inquisitory flavor, orthodoxy and fundamentalism. The inquisition we have today is a high-tech one, but apart from that, I

†EASi Engineering, Vice President Advanced Technologies

find no differences with the one that devastated Europe's intellectual evolution for many centuries.

In economics you can get a Nobel Prize for coming up with an exotic protfolio or derivative pricing model.[1] However, these models will never be predictive since they lack the fundamental component that would make them realistic: they don't contemplate the socio-political sphere of life and society that drives the random fluctuations of the stock market. What is then the value of a "numerically optimal" financial operation stemming from an incomplete (but sophisticated) model?

2 Automotive Crash: From Bifurcations to Chaos

The key to the understanding of the stochastic nature of crash phenomena lies in its bifurcation-based character. Systems of physical interest are described by equations in which parameters appear. As these parameters are varied, changes may occur in the qualitative structure of the solutions for certain parameter values. These changes are called *bifurcations*. The term bifurcation was originally used by Poincare' to denote the *splitting* of solutions in a family of differential equations.

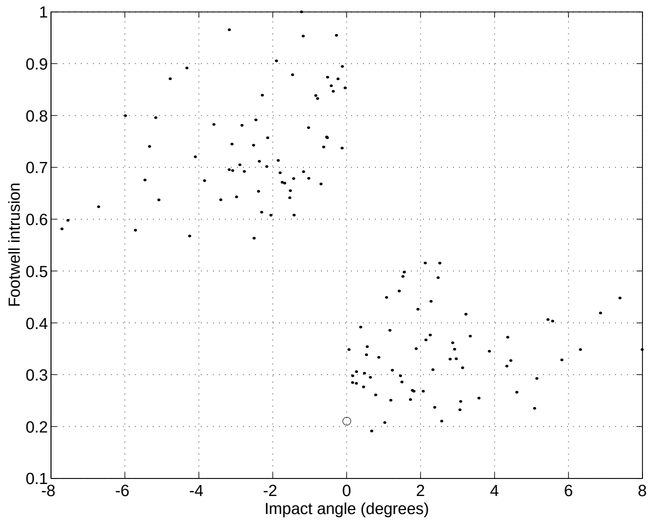

Figure 1: Normalized footwell intrusion as "function" of the impact angle. The principal bifurcation at 0^o is clearly visible in the form of two clusters. This kind of representation is known as the *ant-hill plot*. Ant-hill plots from crash simulations often exhibit clustering, especially if changes in angle of impact are considered. The circle lying on the 0^o axis represents the "optimized" deterministic solution. From [5].

In the presence of bifurcation points it is impossible to assess system behaviour because bifurcations produce major topological changes in the character of the solutions (see the Hartman-Grobman theorem in [3]). It is known that even innocent and simple dynamical systems such as the Van der Pol, Duffing or Lorenz oscillators, possess an incredible variety of topologically different and exotic solutions. These are due essentially to the existence of numerous bifurcations (both local and global). In many cases, the behaviour of these systems degenerates to dramatically different classes of solutions even in the presence of minute changes of certain paramters (not to mention boundary or initial conditions). In the majority of the cases, these changes in parameters

[1] Normally, Nobel Prizes are "awarded in the tail regions" of the Gaussian curve. In these extreme regions, knowledge is so uncertain that no theory can be disproved, and therefore a Nobel Prize can be safely awarded.

are impossible to predict. A nice example is constituted by the discrete logistic map:

$$x_{n+1} = rx_n(1 - x_n) \tag{1}$$

which is frequently used to describe population dynamics. For the particular value of $r = 4$, the map may be written as $\sin {\theta_{n+1}}^2 = (\sin 2\theta_n)^2$. Since this is the square of the doubling of the sine function, the solution is seen to be $\theta_{n+1} = 2\theta_n$. In terms of initial conditions this comes down to $\theta_n = 2^n \theta_o$. This simple equation reveals, first of all, very high sensitivity to initial conditions. Secondly, considering that on a computer the initial condition is stored with a certain number of bits, at each iteration, due to round-off errors, the last bit is replaced by garbage. Therefore, supposing that the initial condition is known to 48 bits of precision, after 48 iterations of the logistic map, no information on the initial condition remains. This means that although the equation is perfectly deterministic, except for very short periods of time, information about the system's motion is contained in the initial conditions and not in the dynamics themselves. Something very similar happens in crash. For $r = 4$ the deterministic discrete logistic equation behaves like a random process.[2]

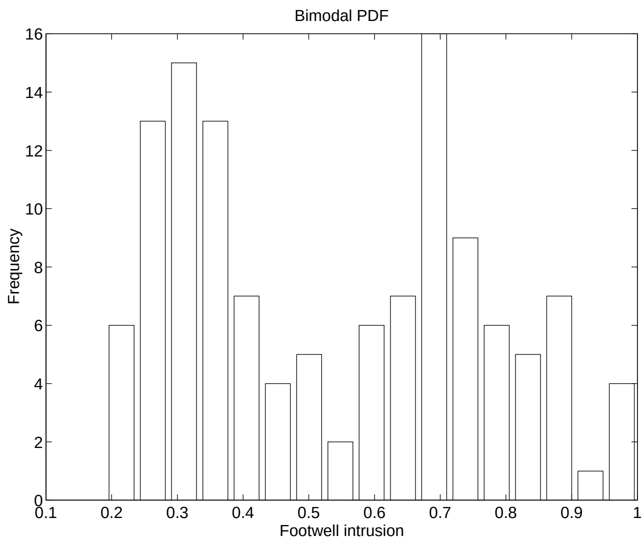

Figure 2: Histogram (PDF) of the footwell intrusion shown in Figure 1. The distribution is bi-modal given the existence of the two clusters. Bi-modal distributions cannot be obtained from second-order response surfaces due to their continuity.

Crash is a phenomenon that involves the interaction of hundreds of components and parts which come into contact violently and for very short periods of time. This interaction is highly nonstationary, , aperiodic and nonlinear and therefore the response of each and every component is strongly driven by its respective boundary and initial conditions. These, however, depend on the response of the parts that have previously come into contact fractions of miliseconds earlier. The response of the entire vehicle is therefore a cascade of hundreds of parallel bifurcation-driven responses, each of which starts from unknown initial and boundary conditions. Clearly, certain characteristic patterns in a similar scenario may still be identified and recognized. However, one can forget repeatability of details. Each crash phenomenon is unique. Just like every tornado, an earthquake, a stock market crash, the way a bottle breaks up when it is dropped. Global patterns are all we can ever hope to comprehend. This is clearly alien territory for optimization.

[2]The process described by the logistic map is equivalent to a bit shift operation, which is widely used in pseudo-random number generation in computers.

Earlier this year we performed a Monte Carlo NVH analysis of a complete car. What was unique in the analysis was the fact that we eliminated randomly nearly 5% of all the welds, i.e. approximately 250.[3] One hundred Monte Carlo samples were run, each one having a different random set of welds, in addition to stochastic variations of the thicknesses of some structural components. The results, shown in figure 3, were indeed very surprising. A main cluster of results appeared, together with numerous outliers. What surprised was the distance of these outliers. [4] This pointed to a possible substantial loss of NVH performance of the car. The interesting thing was that we were able to identify in which locations should weld quality be high, precisely so as to avoid the formation of outliers. What surprised the most, however, was that weld failure is a known problem, and its impact on NVH, not to mention crash, is known to be considerable. Why is it then that this fact is not being modelled? Why is it that hundreds of papers on "robust" NVH and crash optimization are published, but the effects of failing welds are not taken into account? After all, the reason is pretty clear. Just look at figures 3 and 1. Try and imagine a response surface. Try to envisage optimization. Can any known DOE scheme ever hope to capture similar information? Having seen this result, can an NVH or crash calculation, that does not contemplate weld failure, result credible when claims of robustness and optimality are made?

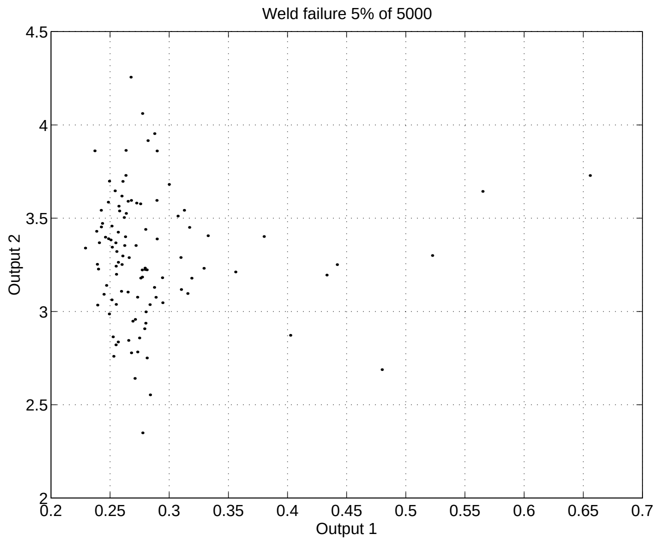

Figure 3: NVH response with weld failure. Numerous outliers are visible. In these cases, weld failure takes place in critical locations. In the case of crash, weld failure is even more important. DOE? RSM? Optimization?

3 Crash = Chaos

Some time ago, I asked a friend who works at the crash department in an automotive company, to give me a couple of acceleration time-histories from a typical crash test. I randomly selected one, depicted in the figure 3, and run a few typical tests for chaos. The idea behind the exercise was very simple: if I can verify that response is chaotic, then the problem of crash simulation should be viewed from a totally new point of view, certainly not as something that can be optimized. Here's what I obtained.

[3]It is known that in the welding process, a certain percentage of the welds can actually fail.
[4]Outliers can, in most cases, reflect potential recalls, warranty costs, or simply law suits.

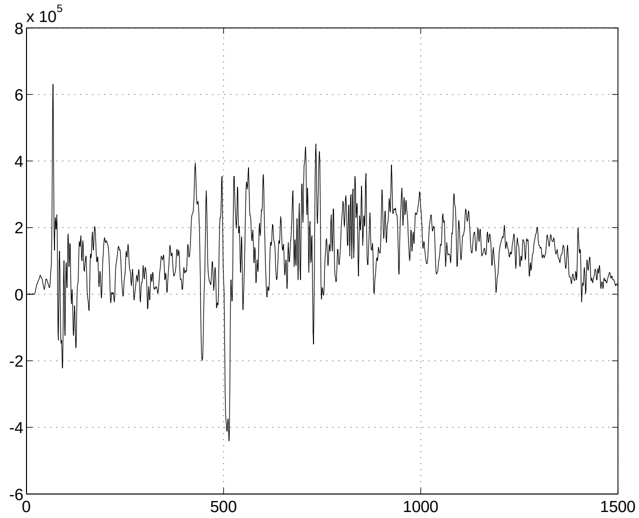

Figure 4: Measured acceleration time-history from a typical crash. The horizontal axis spans a duration of approximately 150 miliseconds. In many cases, similar curves cannot be distinguished from those of stock-market evolution.

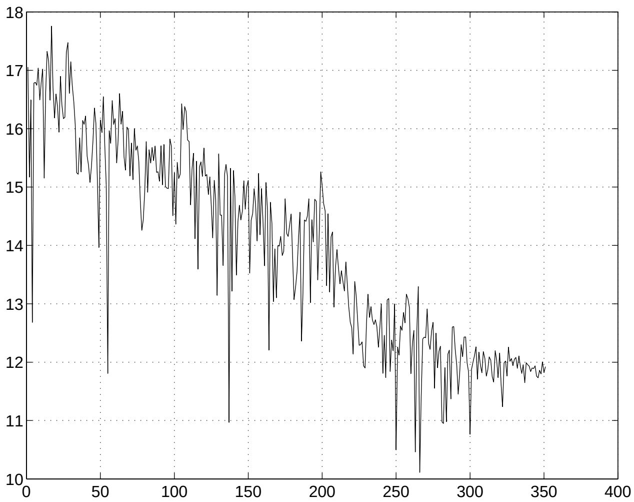

Figure 5: Log-linear plot of the Power Spectrum of the acceleration time-history. The linear shape suggests a power-law, typical of chaotic systems.

1. Power Spectrum. Chaotic phenomena are known to have power spectra that resemble a line with a negative slope when reported on a log-linear scale. This kind of power-law, very common in Nature, constitutes strong evidence of presence of chaos. In this case, the power spectrum indeed resembles a (noisy) line with a negative slope.

2. Lyapunov Characteristic Exponents. Lyapunov exponents reflect how phase-plane trajectories diverge from perturbed initial conditions. If at least one Lyapunov exponent is positive, the system is chaotic. Close to a bifurcation point, the exponents have a value of zero. In this case I found $\lambda_1 = 0.4$. The response is clearly coming from a chaotic system.[5] This means,

[5] The Lyapunov Dimension, computed as $D_L = 1 - \frac{\lambda_1}{\lambda_2}$ characterizes the inherent predictability of the system, is another useful measure of chaos. In the present case it possible to compute only one characteristic exponent.

essentially, that the system is extremely sensitive to changes in initial conditions. What is strange, is that in many cases, a single measured acceleration curve is used to trigger airbag deployment in the case of an accident. Clearly, knowing that crash is a chaos-flavoured type of phenomenon, stochastic triggering algorithms should be employed and, most importantly, these should be based on measurements from multiple sensors.

3. Hausdorff Dimension (also called Capacity Dimension). I obtained a dimension of 1.8. This fractal (non-integer) dimension is a characteristic of chaotic phenomena. Dimensions higher than 5 normally reflect random nature. [6]

4. Correlation Matrix. This matrix is built by placing on the diagonal values of the correlation function with delay $\tau = 0$, while values with increasing τ are placed on both sides of the diagonal. The number of significant Singular Values of this matrix is a measure of the complexity of the system. I found 5 dominant Singular Values. This indicates high complexity. Similar analysis can be performed building first the corresponding time-shifted Hankel matrix and extracting its Singular Values. [7]

5. Return (Poincare') Maps. Lack of structure in the return map reflects a random response. In this case, I obtained a map with a clear structure. Again, this pointed to a chaotic nature of the time-history.

But let's define chaos (I am sure there are many misconceptions here too). Chaos is a *deterministic* phenomenon. In fact, chaos, and don't de surprised, is *hidden determinism*! Chaos implies unpredictability, but it does not imply randomness. A phenomenon is random if it cannot be described via an equation. Chaos can be described by deterministic closed form equations. Look at the Lorenz equations, or the logistic map. What characterizes chaos is extreme sensibility to initial conditions (like the butterfly effect). This dependence on boundary conditions is so strong that predictability is impossible to achieve. Therefore, since initial conditions are *never* known exactly, it makes no sense to speak of optimization of chaotic systems. In fact, as the logistic map example illustrates, information on the initial conditions is quickly lost. Normally, however, neither pure chaos nor randomness exist, they tend to come together. The point, in any case, is that phenomena of this class are unpredictable, nonrepeatable, and must be treated with appropriate stochastic tools. Application of deterministic techniques, not to speak of optimization, has purely numerical value and significance. A slap in the face of physics. In theory, you can pretend that you can predict the stock market a hundred years ahead, or even the climate. Numerically, you can do anything. Even if your model misses a lot of the essential physics. [8]

What is surprising is that this type of signal processing and analysis is not at all popular in the crash community. Normally, before employing a mathematical tool to a particular problem, one should first establish into which class does the problem fall, and then, only then, select the adequate tool. We have seen crash to possess a clear chaotic flavor. Therefore, specific tools should be used to analyze it. Similarly, certain other tools should *not* be used. This brings us to the next section.

Different techniques for the computation of Lyapunov exponents are discussed in [8].

[6] The capacity dimension is calculated by successively dividing the phase space with embedding dimension D into equal hypercubes and plotting the log of the fraction of hypercubes that are occupied with data points versus the log of the normalized linear dimension of the hypercubes. The average slope of the resulting line is taken as the capacity dimension.

[7] I recently performed exactly the same exercise for a publically available s&P500 curve and I obtained 8 dominant singular values.

[8] If you put tomfoolery into a computer, nothing comes out but tomfoolery. But this tomfoolery, having passed through a very expensive machine, is somehow ennobled and no one dares criticize it. *Pierre Galois.* The more complex the numerical game, the more appealing and credible it looks.

4 The Fallacy of Crash Optimization

How is crash optimization attempted nowadays? Does it work? The process is more or less as follows:

1. Certain variables of the problem are defined as design variables.
2. Design Of Experiments is used to sample the design space. Often the Latin Hypercube Sampling technique is used.
3. A surrogate second-order Response Surface-type model is constructed that relates the design variables to certain performace variables, such as mass, injury criteria, deformations, stiffness, etc. The stepwise regression method may be used for the purpose. Due to cost considerations, and not reasons pertaining to physics, very often the cross-terms are neglected in the response surface definition. This fact, together with the choice of the order of the surface, is a totally arbitrary choice.
4. Numerical optimization is performed with the response surface.
5. Confirmation runs are executed.
6. If the accuracy/convergence of the process is insufficient, additional points are added and the response surface is reconstructed.
7. Upon convergence, robustness assessment is performed. In practice, it is assumed that since the design variables are uncertain, i.e. subjected to tolerances, the corresponding density functions are introduced into the response surface by means of Monte Carlo techniques. Since the response surface is very simple, compared to the full FE model, it can be called thousands of times, leading to accurate representations of the probability densities of the responses.

Now, let us now see what is wrong with this approach. Actually it falls apart no matter which way you look at it. The main point, as we have seen, is that crash is a chaotic phenomenon, therefore similar techniques *cannot* be used. They could prove useful in problems that are less dependant on initial and boundary conditions. Crash certainly does, see figure 1. But there is more to come.

Contemporary crash models are excessively simplistic, almost aseptic. No weld failure, no thickness fluctuations resulting, for example, from stamping, no angle of impact variations, no material property scatter. All is made artificially clean so as to prepare the response surface for a smooth ride. The point is that if these forms of uncertainty are introduced into crash models, DOE and response surface-type techniques simply fail. Again, please look at figures 1 and 3. The minute angle variations alone, that appear even in crash tests, are sufficient to disqualify the response surface method. How is then possible to claim that a design is robust, if only the ideal impact condition has been examined?

Violation of the $f(x+y) = f(x) + f(y)$ rule. This one is brutal. The solver should be called *while* uncertainty is introduced, not later after the surface has been built. In practice, one should introduce the uncertainties into the model while he is doing his DOE stuff, only then build the response surface. You can't invert the sequence of operations when operations are nonlinear.[9] However, respecting the correct sequence of operations, i.e. introducing the uncertainty into the crash model, and not into the response surface, leads to responses so scattered that response surfaces are not possible to define or to construct. Therefore, to avoid the problem, reality is bent to suit the tools. In practice, scatter is added on top of a more "clean" response to make things easier. The equation I show here symbolizes the linear superposition of operations that

[9]Think of finite rotations for example.

should not take place. Here, $f(x)$ represents the generation of the response surface, while $f(y)$ the *a-posteriori* addition of scatter. Clearly, the equation holds only when f is a linear operator.

Response surfaces cannot reflect bifurcations. Due to the fact that response surfaces are polynomial expressions, they are unable to "split", i.e. exhibit sudden jumps. Therefore, if any of the design variables are modelled as stochastic, with, say, Gaussian distributions of tolerances, the performance variables, will only exhibit unimodal probability distributions. However, as we have seen in *numerous* occasions, crash-type problems do indeed exhibit clustering in ant-hill plots (i.e. multi-modal PDFs). This is consequence of the simple but unquestionable fact that is crash bifurcation-dominated. Again, please look at figure 1.[10] This deficiency of the response surface is not surprising. A model can only deliver (unwrap, in software terms) what was programmed into it. A second order polynomial, no matter of how many variables, cannot be expected to represent complex crash physics. Mapping crash physics onto a second order numerical-flatland is a miserable way of concluding a series of 100 expensive crash code executions. [11]

The response surface is said to yield small errors (around 5%, but sometimes more) versus confirmation runs. This is not surprising, since "clean" models are used to construct the surface. In fact, uncertainty is cautiously introduced only *after* the surface has been built! Not very ethical is it? I invite all those doing the RSM to introduce small angle of impact variations at each DOE sample. See the previous paragraph.

Once the response surface is available, a Monte Carlo analysis of tens of thousands of samples is run. Why so many? First of all because it is cheap! Secondly, to get precision? Precision of what? We already drag along the response surface error, but the PDFs and CDFs of the performance variables must be precise! Precisely approximate or approximately precise. What is not clear is why Monte Carlo is used to compute the densities of the quantities represented by the response surface. Why not use analytical expressions? Consider, for example, the following second-order response surface:

$$y(x) = x^2$$

assume also that $f_X(x) = \lambda e^{-\lambda x}$. The the density of y has the following expression:

$$f_Y(y) = \frac{\lambda e^{-\lambda x}}{2x} = \frac{\lambda e^{-\lambda\sqrt{y}}}{2\sqrt{y}}$$

Use of Monte Carlo simulation is unjustified, since output probability densities can be obtained analytically. In the majority of cases the arguments of the response surface are Gaussian variables and this makes the whole problem very simple and solvable with infinite precision. Surprising that Monte Carlo is used over an approximate model, but with high precision! But, after all, Western culture has always been more attracted by precision than by knowledge. Precision, supposedly, conveys a message of progress and sophistication that the uneducated public is quickly ready to believe.

While sampling the design space with a DOE scheme, one does not know a-priori what the response cloud (ant-hill) looks like. If the ant-hill were known beforehand, then it would be useless to run DOE. Clearly, an ant-hill can only originate from a Monte Carlo Simulation (MCS), but once you've done MCS, you don't need anything beyond it. MCS is the "last thing" that you can do with a computer. The spirit of MCS is precisely to simulate, i.e. to imitate nature. MCS "contains" all other numerical procedures given that it is the most complete and generic

[10] The number of bifurcations is equal to the number of clusters minus 1.

[11] Mapping a problem from one domain to another is a critical operation both in mathematics and in physics. A good mapping preserves information content while making the problem appear simpler. One of the best mappings I have ever seen is the one used by Kurt Goedel in the proof of his famous incompleteness theorem.

procedure known to numerical analysis. Once you've invested in MCS, you don't need parametric or sensitivity studies, or optimization. You have it all, right there, staring at you. DOE is a stick-in-the-fog approach. Hoping that the fog is nicely shaped.

The response surface method requires a number of solver calls between 3 to 4 times the number of design variables. With MCS, the cost is fixed, and depends only on the accuracy one wants. In the case of 100 samples, the error in the solution is less than 10%. But who wants more in a highly noisy and uncertain problem, where the uncertainty of the inputs often greatly surpasses 10%? With 25 design variables, the cost of DOE and MCS is the same. Beyond 25 variables, the cost of DOE quickly exceeds that of MCS. We have run, with many of our customers, cases with hundreds of stochastic variables, and tens of design variables, with a total cost of around 100 solver calls. MCS was specifically designed to be immune to the curse of dimension.

Why is the order to the response surface restriced to be two? It is basically due to cost reasons, not physics! Again, if the uncertainties that are known to exist, were introduced while calling the crash code, the construction of a second-order RS would not be possible. Higher order surfaces would be needed, but this would increase the cost greatly since many more DOE samples would be necessary in order to compute the large number of coefficients. See, again we speak about numerical issues, not physics! However, there is also another point. Imagine you find a minimum (an optimal design) with a second-order surface. Suppose now that you incerase the order to three and you obtain the minimum again. Will it be the same as the previous one? Most likely not. So, the result depends on the order of the function. It also depends on the starting point for the search algorithm, on the search algorithm itself. In summary, the results depend on the tool that is being used, not on the physics. Strange, don't you agree?

The last point is that of robust design. This is another major conceptual deficiency of the method Since the arguments of the response surface are design variables only, the robustness of the design (of the model, in reality) is checked with respect to uncertainties in these variables. But what about the rest? When we run Monte Carlo crash simulations with our customers, we include uncertainties in *all* the materials, in hundreds of thicknesses, in angles and velocities of impact, stiffnesses of engine mounts, weld failure in thousands of welds, etc. Isn't robustness supposed to be a characteristic that guarantees acceptable responses for the anticipated and expected uncertainties in certain variables? Therefore, as many of these uncertainties should be modelled as possible, not just those of the design variables. How can someone speak of a robust design under crash conditions when angle of impact uncertainties are not considered? Aren't models supposed to be realistic to be of any use?

5 A Natural Approach to Crash Analysis

In life, things that are *natural*, are easier. Going against nature is always a lost battle. With the advent of digital computers, progressivley larger and more complex (not necessarily more realistic) models have been treated. As computers became faster and cheaper, it was no longer necessary to adopt archaic and elaborate numerical techniques, since the computer could take on the burden of solving a larger but simpler problem. A good example of this philosophy is the Lanczos technique for eigen-analysis. Therefore, many elegant methods have been lost, or simply forgotten. This is a pity, but then computers are supposed to simplify life, not to complicate it. This important point, however, is not being shared by everyone in the engineering community. Some people still insist on resorting to excessively complex procedures, such as the DOE-RSM one outlined in the previous section, whereby the computer is confined to a numerics-oriented exploitation (a form of slavery if you prefer) instead of assuming its natural role of a Physics Simulation Engine. As I have already mentioned, Monte Carlo Simulation is the most natural and simplest way of using a computer. To imitate nature. To simulate. Isn't this what we all want? To have a synthetic physics laboratory? One reason why the business of computer hardware companies is so small (in terms of High Performance Computing) is because with the unnatural usage of computers it is sometimes difficult to justify the large investments for the results they deliver. CAE did not reduce dramatically the number of prototype tests. Virtual Prototyping is still a dream because model validity is unknown. Confidence in the computation does not enter the equation. Therefore, why buy large computers? In fact, most of the business of HW companies comes from non-computing sources, such as internet, broad-band, storage, transmission, etc. All non-computing stuff. If computers were used for what they were originally meant, that is for *simulation*, HW companies could see their businesses boosted by at least an order of magnitude. With unnatural and contrived usage of computers, it is really hard to unleash their true and still hidden power.

Monte Carlo Simulation has already enjoyed many successes in the auto industry since its introduction in 1997. In some of the projects that we have recently run with our customers, we reached mass reductions, while maintaining desired performance in terms of multiple condition crash scenarios *and* NVH, ranging from 15 to 25 kilograms! And here we're talking of realistic models, not synthetic and physically poor surrogates: angle of impact variations, uncertainties in hundreds of materials, thicknesses, velocity of impact, weld failure, in addition to tens of design variables. Sounds like pretty realistic models, doesn't it? It is because of physically poor models, that are uncapable of accounting for all these sources of uncertainty, that auto companies have to face recalls, warranty costs, and, of course, law suits.

A "popular" issue in crash simulation is that of non-repeatability of results in multi-processors environments. What happens is quite amusing. Running the same crash model on one processor yields different results when the same model is run on four or eight or more processors. Why does this happen? Recall the logistic map. Information on initial conditions is lost very quickly in chaotic systems. Running a model on multiple processors introduces minute differences, somewhere in the last bit, in the initial conditions. This is enough to lead to macroscopically different, although correct, collapse patterns. Beautiful! For almost a couple of decades now, crash has been treated as a deterministic, and therefore repeatable phenomenon, even though evidence has always pointed in the direction of stochasticity. Consequently, the non-repeatability of crash simulations has been seen as a major problem, as a menace. Strangely enough, this fact has been pointing in the right direction, i.e. hinting that crash *is* a non-deterministic problem, but, due to its political incorrectness, numerics claimed another victory over physics and common sense. Crash codes were forced to deliver *identical* results, regardless of the number of processors used. In 1997, when the first full-car stochastic crash simulations were run, it was quickly realized that statistically identical answers were obtained, regardless of the number of processor used. As it should be. The single responses may be different, but the statistics is the same. In practice, the ant-hill plot in figure 1 does not depend on the number of processors used, the element or node numbering, even the solver. Again, facing a problem from an unnatural perspective-crash is stochastic and *must* be treated as such-makes things difficult. The inexistent predictability that many seek in crash gives rise to the *false problem* of non-repeatability. Natural techniques, such as MCS, on the other hand, help to concentrate on physics rather than on numerical hairsplitting.

6 Conclusions

Current practice in crash analysis, and in particular the attempt to arrive at optimal solutions, imposes on the analyst to construct models that necessarily miss a lot of the essentials. In fact, many important sources of uncertainty are neglected, mainly in order to enable the use of tools, such as DOE and the response surface method, and, finally, optimization. These physically poor models are then claimed to deliver not only optimal, but also robust performance. This claim has been shown to be totally unfounded.

Classical time-domain analysis techniques from non-linear dynamics reveal that crash is indeed a chaotic phenomenon. Of course, there is a certain underlying dose of determinism in crash, but the amount of chaos and randomness that "modulates" the responses of *all* impact-type bifurcation-driven phenomena, is sufficient enough to, first, deprive these problems of any form of predictability and, secondly, to invalidate the use of classical DOE + RSM approaches. DOE and the response surface techniques may thrive in other fields of CAE. Certainly, crash and NVH is alien territory for these methods.

Monte Carlo Simulation prvides a *natural* and sound platform for crash analysis. Its cost does not depend on the dimension of the problem since the method is immune to the curse of dimension. Due to its inherent limitations, the response surface cannot today go beyond 30-40 design variables without becoming prohibitively expensive. Therefore, in order to satisfy this cost constraint, reality is "adjusted" to suit the tool. Models, therefore, intentionally miss a lot of the essential physics. And all this with very expensive computers!

However, maybe the biggest problem lies in optimization itself. Optimization is one of those unnatural things that you can push a computer to do. Optimization is synonymous to excessive specialization, to fragility, to stiffness. Actually, optimal is the exact opposite of robustness (see [9]). Consider the equation $y = 1/x$. Here, x and y are, optimality and robustness, respectively. Optimality and robustness are mutually exclusive. A system is robust when it performs in an acceptable manner in a *wide* range of operational conditions. Think of a car designed to withstand crash over a large range of impact angles. At the same time, think of a car that has been optimized for an ideal impact condition at zero degrees.[12] Which would you prefer? Exhaustive treatment of the problems associated with optimization may be found in [10].

Strong predictability is not going to be a dominant characteristic of realistic and complex models. Rather, their function will be to evaluate possible scenarios, to perform what-if studies, to seek acceptable and robust rather than optimal designs. Models should always be built with this goal in mind. Only such an approach enables true learning and accumulation of knowledge and experience. However, in order to satisfy this requirement, models must first be representative. Today the technology to build realistic crash models exists and relies on the most powerful and versatile numerical technique, Monte Carlo Simulation. In any case, we must keep one thing in mind. Due to the inherently chaotic nature of crash, the optimization of crashworthiness is a futile exercise. It resembles strongly an attempt to square a circle.

Note

I feel the need to clarify one important point that is often raised when I speak to people in the industry. I always get the question as to how ST-ORM, our MCS tool, relates to numerous commercial optimization-type tools. Actually, the overlap is negligible. First of all, ST-ORM is the first commercial Simulation Environment that has been made available to the CAE community. Therefore, from this point of view only, ST-ORM belongs to a category of its own. ST-ORM's mission is not to optimize, but to simulate physical systems in a *realistic* fashion. To this end, ST-ORM supports unique capabilities, such as random fields, for example. Since ST-ORM is based on only *one* numerical technique, namely MCS, the analyst is free from having to choose a specific algorithm. In ST-ORM, results do not depend on the technique, or the starting point, but rather on physics. ST-ORM emphasizes physical relevance and not numerical accuracy. Moreover, ST-ORM provides specific techniques to *stochastically validate* models versus experimental data. Confidence in the simulation is ST-ORM's main concern. Do software houses that commercialize optimization tools ever mention the embarassing issue of model validity? Finally, ST-ORM has been conceived

[12] As it has been shown, it is impossible to optimize a system that has unpredictability as its main underlying characteristic. Optimization, providing it is possible, is always a very fragile victory.

to help study *stochastic systems*, i.e. systems which are non-deterministic. Optimization tools that neglect uncertainty can, therefore, only address the very limited class of repeatable phenomena.

References

[1] Marczyk, J., Holzner, M., et.al., *Stochastic Automotive Crash: A new Frontier in Virtual Prototyping*, PAM 97 User Conference, October 16-17, 1997, Prague, Czech Republic.

[2] Marczyk, J., Editor *Computational Stochastic Mechanics in a Meta-Computing Perspective*, CIMNE, Barcelona, 1997.

[3] Guckenheimer, J. and Holmes, P., *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, Springer Verlag, New York, 1986.

[4] Thompson, M. and Stewart, H.B., *Nonlinear Dynamcics and Chaos*, John Wiley and Sons Ltd., 1986.

[5] Marczyk, J., *Principles of Simulation-Based Computer-Aided Engineering*, FIM Publications, Madrid, 1999.

[6] Marczyk, J., et.al. *Uncertainty Management in Automotive Crash: From Analysis To Simulationm*, ASME 2000 Conference, Baltimore, USA, September 2000.

[7] Doltsinis, I., editior, *Stochastic Analysis of Multivariate Systems In Computational Mechanics And Engineering*, International Center For Numerical Methods In Engineering, Barcelona, 1999.

[8] Marczyk, J., *Theory of Eigenvalue Orbits*, International Center For Numerical Methods In Engineering, Barcelona, 1998.

[9] Marczyk, J., *Stochastic Design Improvement: Beyond Optimization*, AIAA/NASA/USAF Conference on Multi Disciplinary Optimization, Long Beach, USA, September 2000.

[10] Marczyk, J., *Beyond Optimization in CAE*, FIM publications, to appear, Spring 2001.