DIGITAL INDUSTRIES SOFTWARE

Leveraging AI in the vehicle design process

Using machine learning to complement the human mind in improving product engineering

Executive summary
Artificial intelligence (AI) is not a regression model of predicted values and measured results. However, it includes aspects of it. If the technology can be understood and applied to its full potential, it promises to multiply human capacity and accelerate the development and discovery of new technologies. To support this idea, this white paper proposes a framework for applying AI to the vehicle development process.

Our 3D framework: dull, data rich and decision support is based on combining the strengths of the human mind with a machine to turn the increasingly complex world of today's engineering into an advantage. We share specific examples for each case, illustrating where AI applies how it can be implemented.

SIEMENS

Contents

Introduction 3

1. AI is worth the bother 4

Enabler 1: big data 4

Enabler 2: hardware capacity and processing power 4

2. A short primer on AI terminology 5

Supervised learning 5

Unsupervised learning 5

3. The 3Ds framework: dull, data rich and decision support 6

Dull activities 6

Data-rich environments 7

Decision-support applications 9

Conclusion 11

References 11

Introduction

The 1997 victory of IBM's supercomputer, *Deep Blue*, over then-world chess champion Gary Kasparov, brought the power and maturity of AI into mainstream consciousness. This was the culmination of a century-long work on AI, including works by Alan Turing, such as, "Computing Machinery and Intelligence," published in 1950, a seminal paper that proposed a test for intelligent machines.

Since *Deep Blue's* victory, the use of AI has become pervasive but inconspicuous. The United States Postal Service uses AI to recognize hand-written postal codes to sort letters. Search engines use it to return data relevant to you.

But the application of AI has been sporadic in automotive companies. It has come down to the individual team's familiarity with the tools and concepts of AI.

We provide a practical framework for thinking about problems that should be solved with AI so we can leverage its capabilities to accelerate the discovery of new technologies.

Figure 1. A computer beat the ruling world chess champion Gary Kasparov in 1997.

1. AI is worth the bother

You might ask, *"Why bother with AI in the first place? The current process has worked well in designing world-class cars for several decades, and you already have some brilliant people doing the design work. The computers couldn't possibly replace them."*

That is true.

But we propose that AI can provide a head start in subsequent designs by leveraging historical data.

We propose that AI can catch mistakes that will likely go unnoticed until it is too late.

We propose that AI can show you a much quicker way of doing things, be it analysis, processing data or coming up with a robust answer.

Now is the perfect time to take advantage of AI because of two critical enablers in the field.

Enabler 1: big data
Big data – extremely large and complex data sets – has become prevalent worldwide. We generate terabytes of data weekly from computer-aided engineering (CAE) solvers crunching equations and data collected in test cells.

Ninety percent of this data is discarded mainly because we don't have the time to process data that we don't need immediately, even though it might contain valuable insights for the future.

AI is exponentially better at being used to process large amounts of data and identify and extract patterns efficiently, some of which our analog brains will never think of no matter how much time we spend poring through the data. So let AI tackle the big data, do the pattern recognition and develop unique connections that our analog brains can pick up, understand and solve.

Enabler 2: hardware capacity and processing power
You no longer need a computer science degree to run AI due to the rapid rise in the availability of inexpensive hardware computing capabilities and software tools.

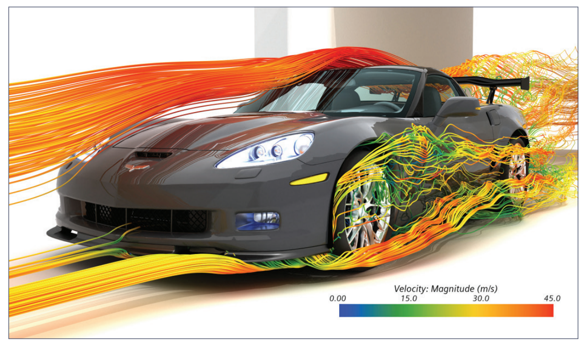
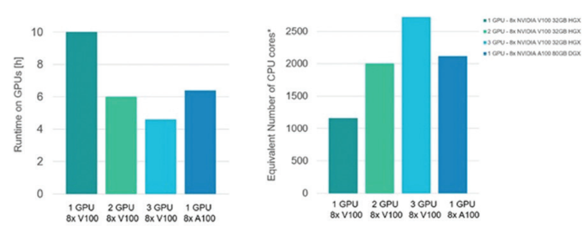


Figure 2. Using Simcenter STAR-CCM+ delivers faster turn-around times at lower hardware investment costs.

The prevalence of high-performance computing and graphics processing units (GPUs) in gaming and consumer industries has decreased their price. Recent performance comparison of running industrial-grade external aerodynamics (see figure 2 for an example of computation) on GPUs has shown a 160X improvement in speed and a 90 percent reduction in power consumption (see figure 3 for GPU performance comparison).



**CPU count to match respective GPU runtime.*
Compute-optimized CPU instance: dual-socket Xeon Gold (40 cores per node)

Figure 3. GPU performance comparison.

2. A short primer on AI terminology

In our discussion, we will use machine learning (ML) methods, a subset of AI.

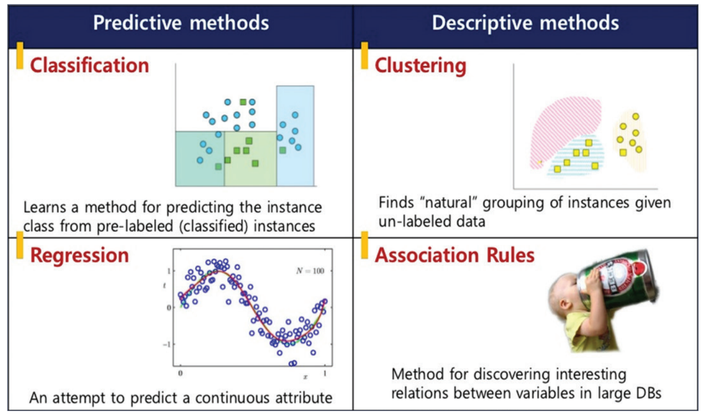


Figure 5. Common methods of unsupervised learning.

Supervised learning
Supervised learning is a method where the correct answer is given to the AI algorithm. The algorithm is told what the inputs are and what the correct answer is supposed to be, as shown in figure 4. The algorithm is then trained on a set of data with the inputs and outputs before it makes predictions on its own based on never-seen-before input channels.

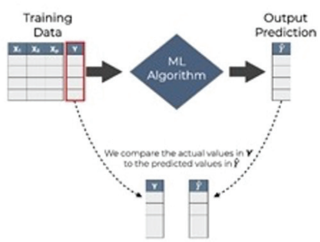


Figure 4. In supervised learning, the target variable Y supervises the modeling process.

An example of supervised learning is a vehicle identifying stop signs. The raw data from the cameras are labeled as inputs, and the final correct answer (a stop sign) is marked as an output.

Therefore, supervised learning is said to use labeled data.

Unsupervised learning
Unsupervised learning is a method where no final answer is given to the algorithm. Instead, we ask the algorithm to process the data and find interesting trends and patterns. Because the data is not labeled, human intervention is significantly lower.

According to IBM Cloud Education,[1] there are three ideal applications for unsupervised learning – clustering, association and dimensionality reduction.

Clustering algorithms determine the highlighting patterns and commonality in the data set, which might not have been apparent to the human mind or determined from classical statistical methods.

Association algorithms use the data of two variables to find a relationship. A Netflix recommendation engine such as, "Customers who watched this might also like…" is an example of association analysis.

Dimensionality reduction cuts down a large set of data to a manageable size without affecting the trends and nuances contained in the data. This is achieved by reducing the number of random variables in the problem while maintaining most of the meaningful properties in the original data.

The earlier example of chess illustrates the power of unsupervised learning. The moves that beat Kasparov in that seminal match two decades ago are now considered simple, intermediate moves by today's standards. The chess game has come a long way since the advent of AI by identifying moves and patterns experienced grandmasters had not thought of. The chess grandmasters of today stand on the shoulders of millions of chess games that AI computers play to discover new moves.

We can do the same with our product development process and the terabytes of data we generate.

We can learn new but counterintuitive ways of doing things.

3. The 3Ds framework: dull, data rich and decision support

We propose the following as a framework to apply machine learning:

Dull, data rich and decision support identify specific scenarios where the machine is superior to the human mind at the given task and can complement the human mind to achieve far greater output.

Dull activities
Repetitive and tedious work is an unfortunate reality in today's engineering design process. These repetitive activities are dull, require little engagement with the engineer's mind and take a long time to execute. Examples include reviewing a large dataset for coherence before using it or preparing computer-aided design (CAD) geometry for analysis. A manual procedure delays the design process and causes high costs as expensive labor is wasted on unskilled tasks. Moreover, errors creep in due to the repetitive nature of the work.

Here is an example where we applied supervised learning to automate and reduce four-day CAD geometry cleanup work to half a day.

Using CAD geometry cleanup for CAE analysis

The raw CAD geometry created by the designer is rarely ready to be meshed and analyzed immediately. There are gaps where the surfaces meet, hollow 2D surfaces instead of 3D solid objects and the presence of many small parts, such as bolts and fasteners, must be removed for better mesh generation.

We applied ML to identify nuts and bolts automatically. Geometry data from past vehicle programs were used to train the ML algorithm on the same components that needed to be identified, tagged and removed in the geometry cleanup process.

Once trained, the ML algorithm could be used to process vehicle CAD geometry that it had never seen before and automatically identify the components for deletion. Figure 6 shows the outcome of the ML algorithm that identified and tagged the nuts and bolts for deletion.

In this instance, the engineer still reviewed the tags and verified the identification was correct and complete. This expert-in-the-loop process ensures a high quality of work while automating the tedious, laborious process of CAD geometry cleanup.

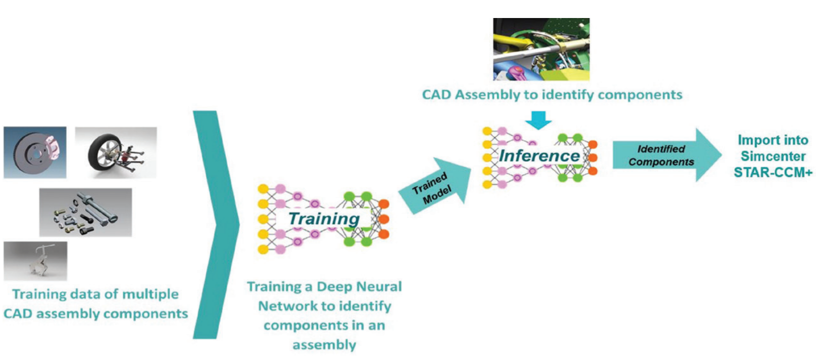


Figure 6. The CAD part recognition workflow for identifying components. Note that for the current example the dataset came from two cars.

Figure 7. The ML algorithm automatically identifies and tags the nuts and bolts for deletion, even if it has not seen this geometry before.

Similar supervised learning can also identify and populate the missing material properties in CAD.

Data-rich environments

Data rich might sound like an odd criterion.

Data rich is not an application but rather a necessary condition for ML. Applying ML in data-rich environments is akin to looking for a life where there is water. Here we don't start with a problem to solve but rather reverse the thinking and ask, "What can I do or learn with all this data?"

Reversing the thinking this way uncovers numerous applications.

We can leverage supervised and unsupervised learning methods to extract insights from these large datasets generated from CAE and testing.

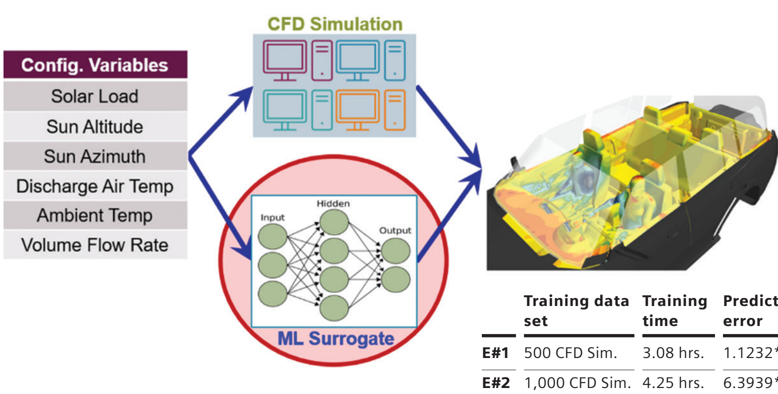
Leveraging large sets of CAE data: reduced order models

A reduced order model (ROM) is a simplification of a high-fidelity static or dynamical model that preserves essential behavior and dominant effects for the purpose of reducing solution time or storage capacity required for the more complex computational model (for example, a 3D finite element model). Compared to data-rich environments, ROMs reduce the number of variables in the systems while maintaining the accuracy of the original complex computational model. As such, ROMs are much faster and more straightforward. A popular approach to generating a ROM is using ML trained on results from a high-fidelity CAE analysis.

The case discussed here applied ROM to predict the surface temperature distribution inside a vehicle cabin. A simulation results database was generated by running several high-fidelity computational fluid dynamics (CFD) simulations in a batch run for various operating conditions and inputs.

The team tested the accuracy of the ROM trained on 500, 1,000, 1,500 and 2,000 data samples. The ROM model accurately predicted even the smallest dataset, as shown in figure 8.

The ML model only takes a few seconds to predict temperatures compared to several hours for a high-fidelity CFD. This ROM can be used for predictions if the cabin geometry does not change significantly. The figure below compares the various ML models with measurement data.



	Training data set	Training time	Prediction error
E#1	500 CFD Sim.	3.08 hrs.	$1.1232*10^{-4}$
E#2	1,000 CFD Sim.	4.25 hrs.	$6.3939*10^{-5}$
E#3	1,500 CFD Sim.	5.40 hrs.	$4.9629*10^{-5}$
E#4	2,000 CFD Sim.	6.46 hrs.	$4.8784*10^{-5}$

Figure 8. Configuration variables and errors in reduced order model prediction for different sizes of training datasets.

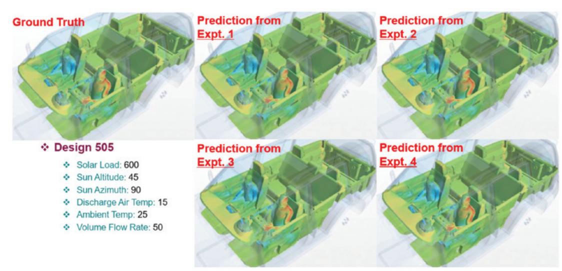


Figure 9. Temperature distribution comparison in the cabin from each of the trained ROM models.

Once a ROM has been generated, it can then be integrated into a system model (for example, 1D A/C and heat pump model) to replicate accurate 3D predictions of a CFD model at the speed of a 1D model. A complete drive-cycle analysis, multi-attribute balancing or controls development can now be performed in a fraction of the time.

Leveraging test data. Part I: models of nonlinear subsystems

Let's look at a scenario where test data is the source.

When ML is applied to test data it can be used to solve difficult simulation modeling problems, such as components with complex nonlinear characteristics.

Work from Nakatsugawa[2] is on a self-switchable hydro mount component that has a nonlinear characteristic depending on amplitude and frequency. The hydro mount was excited by various constant-amplitude frequency sweeps, where the input displacement and output hydraulic force were measured. An ML model was then trained to predict the hydraulic force time series given the input displacement.

The ML model showed accuracies above 90 percent, which is comparable to or better than a physics-based model. Moreover, the data-driven hydro mount model was time-stable and could be directly integrated into a larger transient 1D CAE system model.

A similar approach from Gorgoretti[3] on full-vehicle, test-track data was used to obtain a tire model that can be integrated into a 15 degrees-of-freedom (DOF) vehicle dynamics simulation model. The obtained tire models were similar (between 9 to 17 percent mean absolute percentage error) to high-fidelity, physics-based tire models, which are obtained by more cumbersome conventional means; for instance, where specialized test infrastructure (a dedicated tire test rig) and a specific test procedure are required.

In addition to saving modeling time, this approach removes the need for these additional specialized tests.

Leveraging test data. Part II: data-driven virtual sensors

In automotive proving ground testing, the vehicle is traditionally fitted with the many types of sensors needed to characterize the dynamic behavior of a new vehicle variant. The physical instruments used in these tests include some expensive and hard-to-install sensors.

We used measurement data from cheaper and easier-to-install sensors to estimate the expected output from expensive and cumbersome-to-install physical sensors' signals. This is achieved by training the ML algorithm on previous measurement data, where the prediction targets are the difficult sensors (for instance, wheel force transducers) and the inputs to the model are cheaper sensors (for instance, accelerometers and strain gauges).

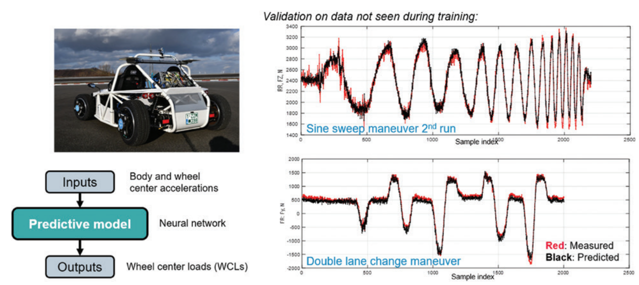


Figure 10. Demonstrating the accuracy of a similar approach applied to "Simrod," the Siemens demonstrator vehicle, to predict wheel center loads.

Automating benchmarking with unsupervised clustering of historical data

Many automotive original equipment manufacturers (OEMs) possess large historical databases of test measurements. Although these databases potentially contain useful knowledge, in practice the engineers only analyze at most up to 30 percent of the total measured data due to lack of time, likely missing some essential insights in the data.

Moreover, these expert analyses are often reactive (for instance, responding after an issue has occurred).

Unsupervised clustering approaches can help by automatically identifying the most relevant clusters in the database, which can then be passed to the experts for more thorough analysis.

In noise, vibration and harshness (NVH) benchmarking, a hierarchical clustering approach was used to compare a vehicle fleet's road noise sound quality performance. The vehicles were driven on an automotive proving ground in coast-down test drives, and acoustic recordings were made inside the passenger cabin of each car.

The goal of the benchmarking was to understand the relative performance of the vehicles' sound quality.

Manually assessing the data was difficult and time-consuming – it was possible if looking at just one metric, but it was near impossible when looking at 10 metrics simultaneously, even with conventional statistical methods.

Applying an unsupervised clustering method to the problem was a natural choice.

By applying an agglomerative hierarchical clustering approach to the extracted sound quality metrics for each car and then visualizing the dendrogram, the engineer can get a global overview of trends in the acoustic data.

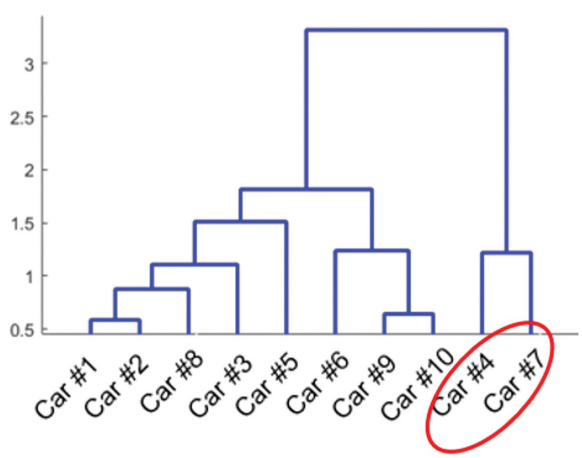


Figure 11. A dendrogram showing hierarchical clustering results for sounds recorded inside a fleet of 10 cars.

You notice that cars #4 and #7 were consistently clustered together in different measurements, indicating they have a comparable sound quality performance. However, different vehicle segments highlight a possible issue with one of them.

The research and development (R&D) engineer can then perform a more focused drill down into the individual sound quality metrics (see figure 11) for more detailed analyses to understand why these cars are different from the rest of the fleet and which sound quality aspect was the main cause of this difference.

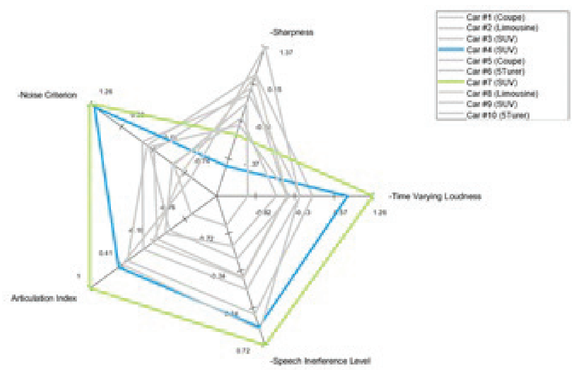


Figure 12. Radar plot showing further drill down into sound quality metrics, with the trends of cars #4 and #7 highlighted.

Decision-support applications

The use of AI in decision support and diagnostic applications is a natural choice for two reasons.

First, human judgment is noisy. Daniel Kahneman highlights this in his book, *Noise: A flaw in human judgment*,[5] that not only do people make different judgment calls on the same data, but the same person could also make different judgments in the morning and the afternoon on the same case. For example, a test subject assessing cabin comfort for an identical setting might make foreign judgments on comfort based on many factors unrelated to the vehicle cabin comfort (for example, sleep, health and whether the local sports team won or lost the previous night).[5] A reduction in such variability will lead to more robust engineering.

Secondly, humans are not good at detecting anomalies for many variables or if the trends take a long time to evolve – for example, a slow degradation in the performance of a part over several days operating in the field. A manual inspection of the telemetry data might only detect the issue upon failure, but an AI algorithm can detect it much earlier.

Let's look at two examples where AI can help significantly in decision-support applications.

Detecting faulty equipment

Work from Hendrickx[6] provides a good example of applying machine learning for anomaly detection. Several electrical machines in a fleet are tested in a shared warehouse space. Electrical and vibration signatures are measured at various locations and used to train machine learning models to identify unsafe operational behavior of a specific component.

Using a similar approach, ML models can detect the slow degradation of batteries or critical components in vehicles in the field by analyzing the telemetry data.

Mitigating subjective evaluations

ML models can be suitable alternatives to many subjective evaluations of design performance.

Work by Lopes[7] showed how trained neural networks predicted the passenger response to in-cabin noise. The goal of the neural net model was to establish a relation between objective and subjective psychoacoustic attributes. Such a capability enables complex models of human comfort levels and preferences to be built into the virtual prototyping framework.

Figure 13. Detecting faulty components using ML.

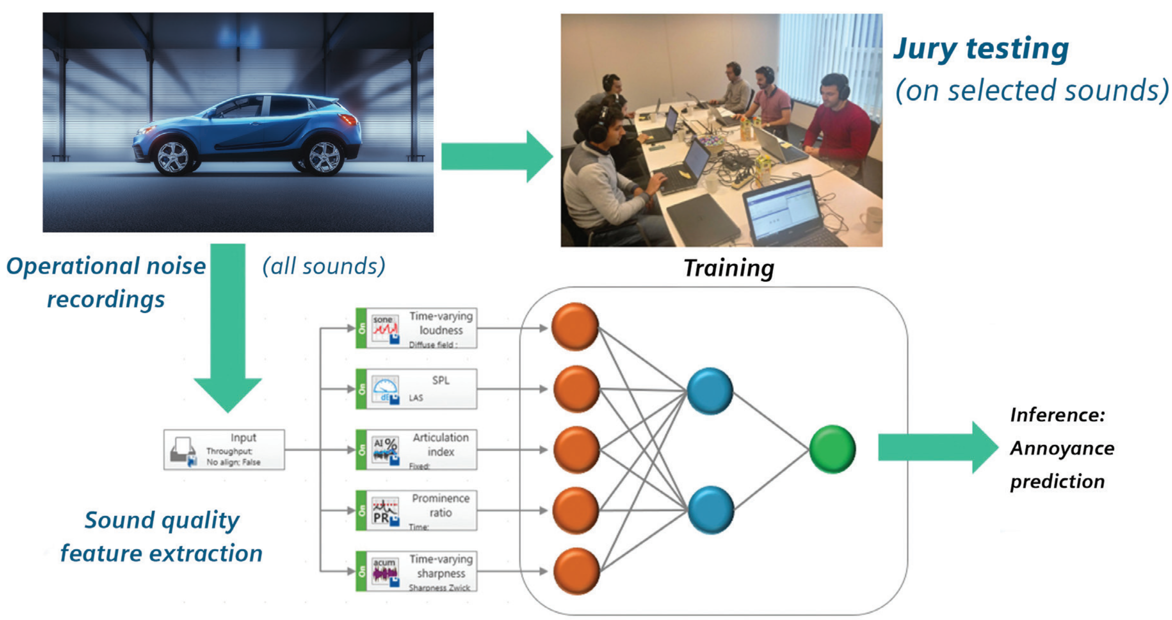


Figure 14. Process for developing a data and analysis pipeline for training a neural network for sound annoyance.

Conclusion

With the rapid rise in the ability to generate data and the availability of artificial intelligence tools, the time is right to apply AI to the vehicle development process.

Our framework for applying machine learning: dull, data rich and decision support identifies specific scenarios where the machine can complement the human mind to achieve speed and efficiency that is an order of magnitude higher than today.

Applying ML to tedious processes such as CAD preparation for external aerodynamics analysis shows an 8X acceleration, decreasing the time from four days to one-half day.

Generating reduced order models brings the high fidelity of a 3D CAE analysis to a fast-running 1D system model, enabling multi-attribute balancing, robust engineering and complex controls development.

When applied to test data, ML can create accurate subsystem models of complex nonlinear systems in a fraction of time without having to model the physics and fine-tune the model.

Finally, unsupervised learning brings insights into view that were previously missed, reducing the number of late design changes and failures during real-world operations.

References

1. https://www.ibm.com/cloud/learn/unsupervised-learning
2. Nakatsugawa, Eiji, et al. "Development of modeling method for the self switchable hydromount by machine learning." Transactions of Society of Automotive Engineers of Japan 52.2, (2021).
3. Gorgoretti Davide et al. "From track to digital twin: a machine learning and sensing approach to tire and steering system modeling." Virtual Vehicle Symposium, Graz, (2022).
4. Elkafafy, Mahmoud, et al. "Machine learning and system identification for the estimation of data-driven models: an experimental case study illustrated on a tire-suspension system." Proceedings of ISMA. (2020).
5. Kahneman, Daniel; Sibony, Olivier; Sunstein, Cass (16 May 2021). Noise: A Flaw in Human Judgment. New York: Little, Brown Spark. ISBN 978-0-00-830899-5
6. Hendrickx, Kilian, et al. "A general anomaly detection framework for fleet-based condition monitoring of machines." Mechanical Systems and Signal Processing 139 (2020): 106585.
7. Lopes, Bernardo, et al. "Neural network models for the subjective and objective assessment of a propeller aircraft interior sound quality." INTER-NOISE and NOISE-CON congress and conference proceedings. Vol. 259. No. 5. Institute of Noise Control Engineering, 2019.

Siemens Digital Industries Software

Americas: 1 800 498 5351

EMEA: 00 800 70002222

Asia-Pacific: 001 800 03061910

For additional numbers, click here.

About Siemens Digital Industries Software

Siemens Digital Industries Software is driving transformation to enable a digital enterprise where engineering, manufacturing and electronics design meet tomorrow. Siemens Xcelerator, the comprehensive and integrated portfolio of software, hardware and services, helps companies of all sizes create and leverage a comprehensive digital twin that provides organizations with new insights, opportunities and levels of automation to drive innovation. For more information on Siemens Digital Industries Software products and services, visit siemens.com/software or follow us on LinkedIn, Twitter, Facebook and Instagram.

siemens.com/software

© 2023 Siemens. A list of relevant Siemens trademarks can be found here. Other trademarks belong to their respective owners.

85027-D4 1/23 H