

An Engineer's Guide to Digital Twin Technology

Learn how digital twin technology works, how it can improve your IoT project, and see a detailed, real-world example.



What is Digital Twin Technology?

While the term sounds like something out of a bad science fiction novel, “digital twins” are revolutionizing the ways that businesses use artificial intelligence, data analytics, and the Internet of Things (IoT). So what is a digital twin exactly, and why should they be an essential part of your IoT projects?

Simply put, a digital twin is a virtual replica that is intended to imitate the real-world behavior of a physical object or system. Companies have used digital twins to model everything from aircraft engines to apartment buildings.

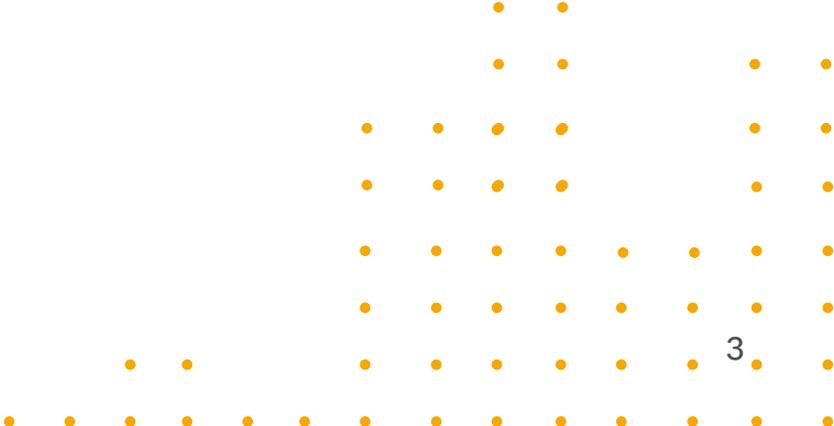
Digital twin technology has applications in many different domains: engineering, manufacturing, transportation, construction, energy, and more. Any time that you want to understand the real-world behavior of a device or product, building a digital twin can help.

When designing a car, for example, engineers have a set of preconceptions and expectations about how the vehicle will act under certain conditions. During the car’s operation, sensors and devices are used to collect measurements and data and send them back to the company for analysis. This information can then be compared with the output from the digital twin to determine whether the car is still behaving in line with the way it was designed.

In addition to diagnosing vehicle maintenance issues, digital twins can be applied to rocket ships, washing machines, electrical grids, and more. Creating a digital twin of an object or system is a good idea whenever you need to make predictions about its behavior or determine why it’s experiencing problems.

The “dirty little secret” about digital twins is that they aren’t exactly a revolutionary concept for engineers. While the term “digital twin” may be a

recent creation (largely popularized by marketers), organizations such as NASA, GE and race car teams have used virtual models to ensure the quality and safety of their products for decades.



How Does Digital Twin Technology Work?

Smart components that use sensors to gather data (about real-time status, working condition, or position) are integrated with a physical item. These components are connected to a cloud-based system that receives and processes all the data that the sensors monitor. This input is then analyzed against business and other contextual data.

Although data from these smart components is almost always sent to the cloud eventually, in certain situations you may need to perform additional monitoring on the device itself. If you commonly experience internet connectivity or latency issues, for example, then you might also create a simplified digital twin model that runs on the device for basic analyses.

To return to the example of cars, most automobile manufacturers are just beginning to start sending data from the vehicle into the cloud. Even with cloud connectivity, the car's onboard systems constantly run diagnostic checks that can detect when something is wrong. This allows the vehicle's computer to take immediate safety measures, instead of waiting for the data to be processed in the cloud.

How Digital Twins Can Improve Your IoT Project

Digital twins are an especially promising concept when used in combination with the Internet of Things. For one, a digital twin can provide you a better understanding of how your product is behaving and being used (especially when predicting and diagnosing failure).

Suppose that one of your products has a manufacturing defect that your company is unaware of. When it's first released, you experience a high volume of customer service requests about how a particular part in the device is failing. If you've been collecting data about this device through the IoT, you can compare it to the output of your digital twin model to find any deviations.

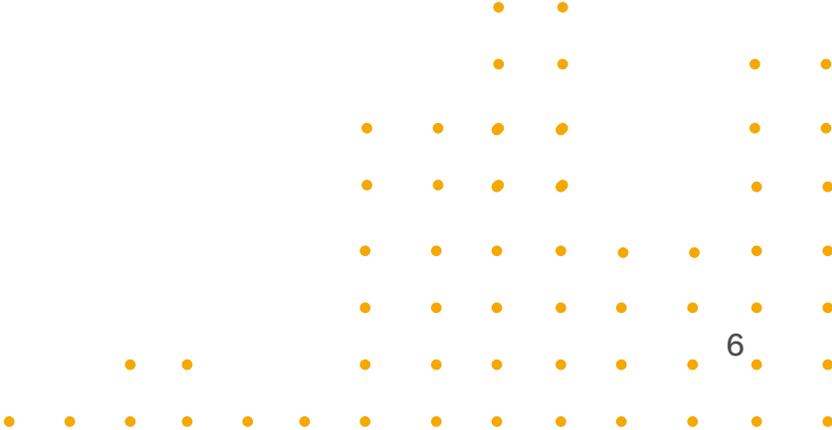
Even without discovering the root cause of such a failure, you may also be able to use this data to predict which devices are going to experience issues. Your company can then proactively alert certain customers that they may have a defective product and offer to fix it before it fails. The combination of digital twins and the IoT gives you a great deal more flexibility and proactivity in terms of support and maintenance.

Digital twins may also be able to reduce the overall cost of owning and supporting a device over its lifetime. If your engineers are rushing to get a product out the door, for example, they may not have enough time to create one of the components in-house. To save time, you decide to purchase a part from an external supplier, which costs you more in the short term but allows you to meet your project deadlines.

Once you've released the first version of your product, you can use digital twins and IoT to remake the component in-house after launch. By collecting data about the ideal behavior from this more expensive external component, you can drastically shorten the design process.

Similarly, your engineers may build a device with more material than

necessary in order to reinforce it, because you don't have time to calculate its breaking point. To save on the extra costs of this material, you can monitor the device in the real world and see what level of stress it's capable of withstanding. You can then determine how much material you can remove from the next production run, reducing costs over the lifetime of the product.



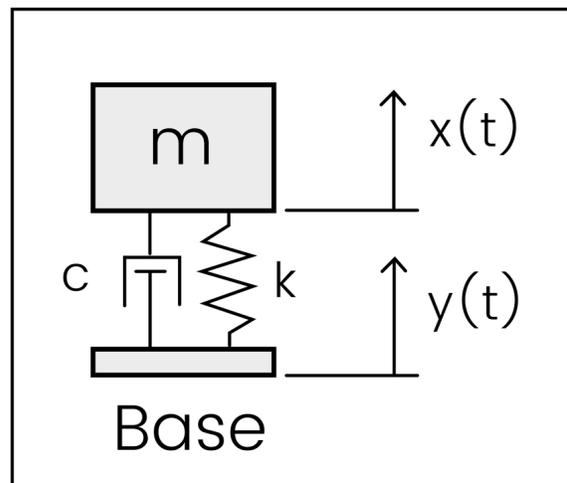
A Real-World Example

That's enough theory. Let's add some concrete math and code to the "Digital Twin" hype. We'll go through a simple example that's based around a standard undergraduate mechanical engineering concept. All units will be in metric.

Simplified Automotive Suspension (AKA Mass-Spring-Damper System)

The mass-spring-damper system above is governed by the following differential equation:

$$m\ddot{x} + c\dot{x} + kx = cy + ky$$



This result can be derived with some Physics 101 knowledge by building a free body diagram and substituting the forces into Newton's Second Law of Motion equation. Note that c represents the damper coefficient, and k represents the spring coefficient. If differential equations aren't your thing, the spring resistance changes with position of the mass/base while the damper resistance changes with velocity of the mass/base.

For this system, there are some other important equations and concepts that will become relevant. I won't go into their derivation, but you can find a fairly thorough example [here](#).

Undamped Natural Frequency, ω_n

This is the frequency at which the system will vibrate when disturbed if there is no damping. It can be calculated by the simple equation:

$$\omega_n = \sqrt{\frac{k}{m}}$$

Damped Natural Frequency, ω_d

This is the frequency at which the system will actually vibrate when disturbed.

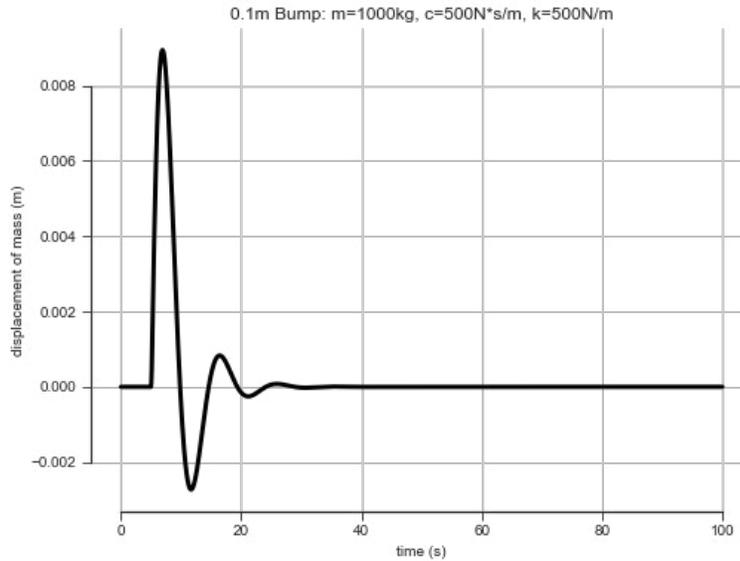
$$\omega_d = \omega_n \sqrt{1 - \zeta^2}$$

where ζ is defined as the damping ratio, and is defined by the equation:

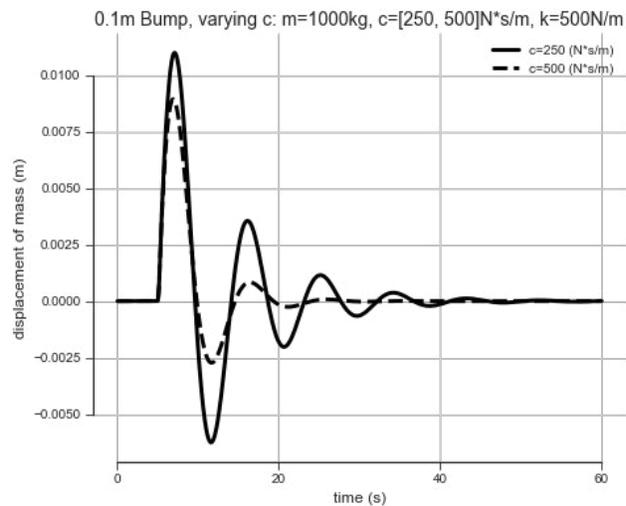
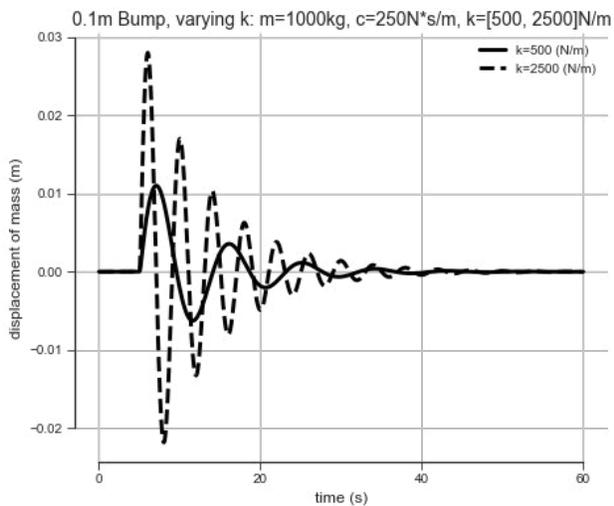
$$\zeta = \frac{c}{2m\omega_n}$$

Solving These Equations

There are many computational solvers for Ordinary Differential Equations, in this case, it was easy to reach for SciPy. Using the SciPy solver, for a single tuple of m , c , k , the position of the mass over time when a car goes over a bump with a height $y = 0.1\text{m}$, looks like:



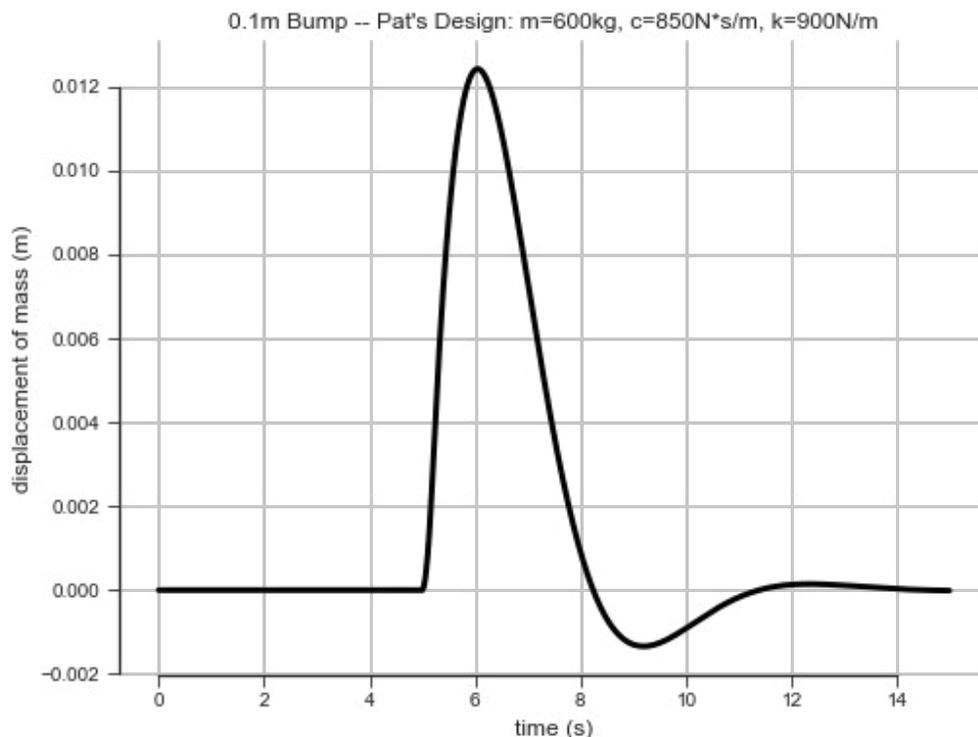
In addition, as k/c are varied, you can see that the frequency of oscillation and decay rate change respectively.



Designing an Automotive Suspension

Automotive engineer, Pat, works for a hot new automotive company, ALSET. Pat is helping to design a new luxury sedan that will turn the industry upside down because it will be built with “Digital Twin Technology.” Pat is in charge of the suspension, and is under a time crunch, choosing the simplified model shown above.

Because Pat is working on a luxury sedan, Pat will design the suspension to be comfortable. This means that the design will target roughly 1Hz oscillation frequency, which will decay to less than 5% of max displacement after 5 seconds. The target mass of the car is fixed at 2400kg and is evenly distributed between all four wheels. This means each mass-spring-damper system will have a mass of $\frac{2400\text{kg}}{4} = 600\text{kg}$. Using the given constraints and the modeling equations above, Pat can find a k and c that will theoretically yield the following curve when the car drives over a 0.1m bump:



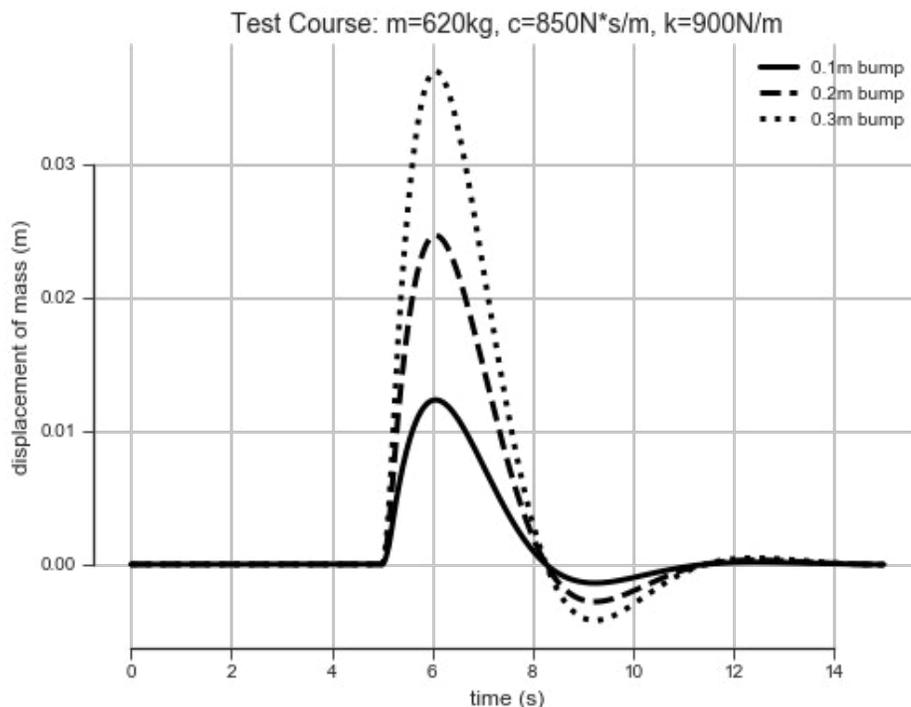
Pat's designs are put into the prototype, and everyone loves the way the car rides, but this is just step one of the digital twin process.

Sensing Suspension Behavior

Though Pat has done some nice digital modeling, the boss, Nole, is not yet satisfied. A core part of the digital twin concept is tracking the real-world item through its lifetime, and making constant comparisons between it and the digital model.

To make this possible, Pat mounts an accelerometer on each corner of the chassis, and hooks each one into the on-board control module. This module periodically streams data from the suspension and other critical systems to ALSET servers. Though Pat will be tracking acceleration, as opposed to position, it is possible to calculate position as the second integral of acceleration. Though this can get complicated when there are multiple axes of motion, it is relatively simple and reliable in one dimension. It should also be noted that the vibrational characteristics of the system are identical whether you are observing an acceleration graph or a position graph.

Before the design is set for production, the prototype car drives on a test track while streaming data from all the on-board sensors. Pat pays special attention to the “bumps” portion of the course. This portion sends the car over 0.1m, 0.2m, and 0.3m bumps, which yields the following graph:

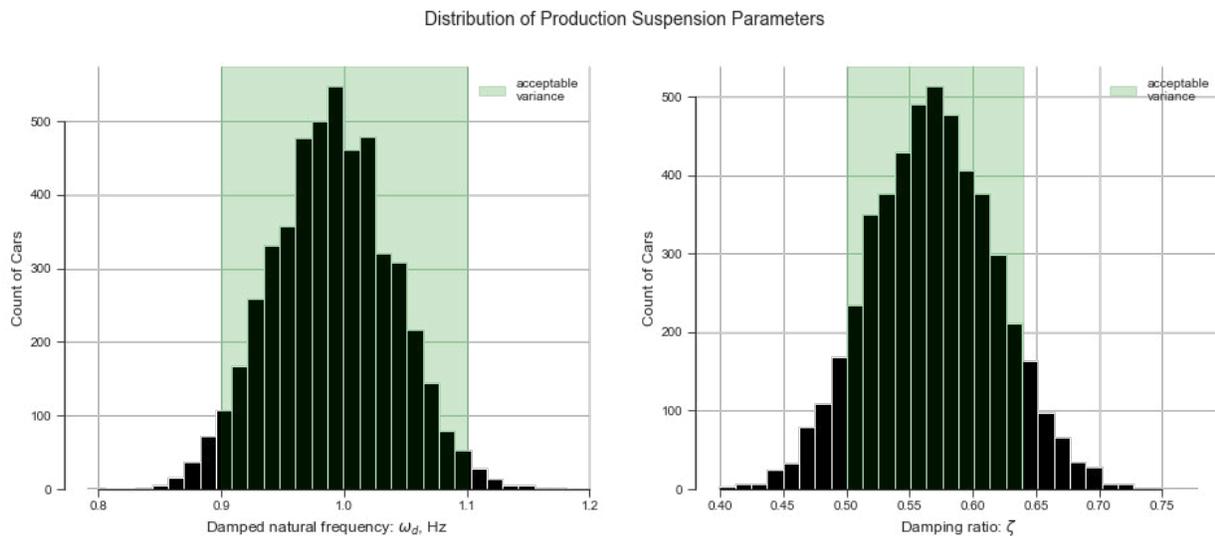


Because of design changes in other parts of the sedan, the results don't match Pat's model exactly. However, it is easy to use these data to infer a new model, so Pat does just that.

After a test production run, 10 cars are randomly sampled to drive on the test track, and their results are all nearly identical to the prototype.

Leveraging Twin Data Right Away

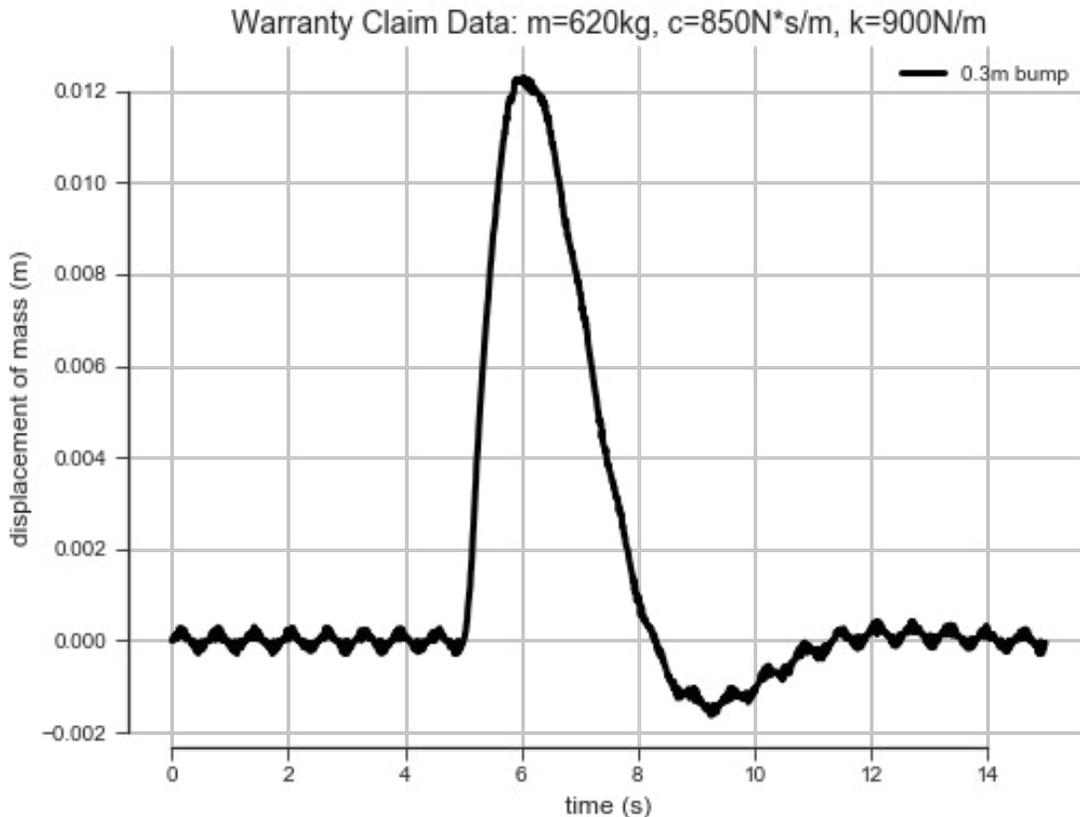
Five thousand cars are sold the first week after launch, and Pat receives suspension data from each of them. By running a large map-reduce job, Pat obtains the frequencies and damping ratios from many "bump" events for each of the 5,000 cars and plots the distributions:



At this scale, it can be seen that there is more variance than desired for some of the key performance constraints. There have been no customer complaints, but Pat notifies the production team, and the team puts measures in place to assemble the suspension with greater consistency.

Big Paybacks in the Long Run

A year later, some warranty claims begin trickling in. The main complaint is unwanted shaking/vibration. Pat looks at the data from some of the claims, and sees something like this:



As a seasoned suspension expert, Pat knows this extra vibration is probably due to a worn bushing or failure of some other rubber component, and alerts the repair technicians. Most repair technicians find that the upper spring seat is severely worn to the point of failure. The production team reports that the OEM supplier for this part has been flaky, and the due to high production demands, they have had to work with several backup suppliers and were unable to track which version of the part went into each car.

Total failure of this part could cause unpredictable vehicle handling, and Nole is on the verge of issuing a massive recall. This has huge financial implications, because replacing the part requires removing most of the

suspension, and is very time-consuming. Before the recall, Nole puts all hands toward finding a better solution.

While combing through the data of all the warranty claims, Pat realizes that there were signs of the imminent failure long before the customers reported a problem. Though the team can't directly identify cars with faulty parts, they can monitor the entire fleet and determine if something has gone awry.

Nole is relieved, and Pat is a hero. Pat works with an ALSET data scientist to finalize a predictive model, which works by comparing the real-world data with the theoretical suspension model. This leads to car owners being selectively notified of the pending part failure as needed, saving the company millions.

Wrapping Up

Pat's story has shown us a (simplified) real-world example of digital twin technology. In this example, a failure prediction algorithm was implemented reactively. But anomaly detection for failure prediction can be put in place preemptively if the use case warrants it. The key is that neither are possible if you aren't collecting data or do not have a computational model of the system (whether physics-based or learned via machine learning).

Though digital twin technology does have origins in rocket science, it's not that hard to understand, and even a simple model can be valuable. Of course, like anything, as the system gets more complex, or the requirements for model precision get higher, it can become a very difficult problem.

At Very, our data scientists and engineers are familiar with these problems and can work independently or with your engineers to develop a robust, scalable, digital twin solution.



**Bring Your
Product
to Life**

SEE WHAT WE DO