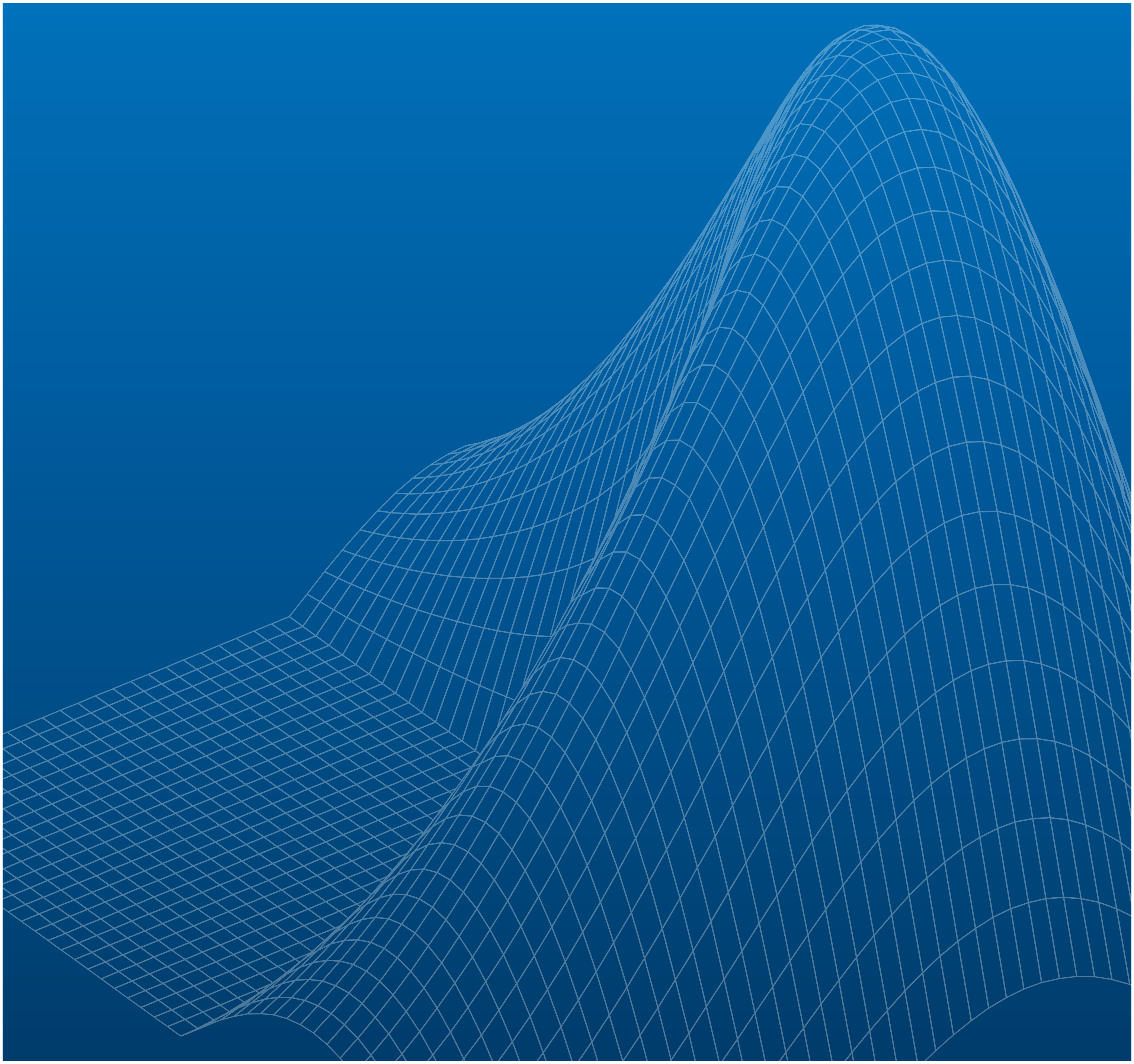
Improving Engine and Vehicle Design Using Big Engineering Data Analytics and MATLAB



WHITE PAPER

MathWorks®

Introduction

For many years, engineers across the automotive industry have been analyzing data in MATLAB® to improve performance and efficiency. Automotive engineers have learned to apply MATLAB to analyze data sets collected from the engine, transmission, chassis, and virtually all parts of the vehicle. But what happens when the data gathered mounts into the terabytes or larger, as it often does for vehicle data at the fleet level?

Can MATLAB scale up and meet increasingly demanding fleet data analytics requirements? Is it possible to build big engineering data analytics algorithms and deploy them into a production environment for data scientists, product engineers, and business analysts? The short answer to both of these questions is "Yes."

This paper explains how MATLAB can be used to extract key insights from big engineering data for improving engine and vehicle design. Specifically, it describes a real-world system constructed to demonstrate big engineering data analytics with MATLAB. We built this system to collect our own data, so that we could further streamline and refine techniques for applying MATLAB analytics on big data. This system includes infrastructure for gathering actual fleet data, as well as a web-based fleet data analytics application built and deployed with MATLAB. The paper concludes with examples of valuable insights that can be gained from such a setup.

What is Big Engineering Data and Why Is It Challenging to Derive Insights from It?

Across industries, big data is commonly described in terms of the three Vs: volume, variety, and velocity. (A fourth V, veracity, is frequently included and will be covered later in this paper.) In engineering applications, we use the term "big engineering data" to characterize data that exhibits these qualities while presenting a subtly different set of problems. The data is typically better structured and more varied, and it has the potential to grow much faster than big data in other industries. For example, when dealing with fleet data:

- Volume refers to the scale of the data. For automotive OEMs, it is not uncommon to work with data sets of up to 20 TB at once and collect more than 30 GB of data per car per day. At these rates, the data can easily grow to sizes in the order of petabytes and sizes that cannot be analyzed at once in memory. Even the relatively modest demonstration system we constructed was capable of collecting 25 MB of sensor data per day, per car, and it produced almost 2 GB of data over several months with just a few drivers. These figures do not include video data, commonly used in advanced driver assistance systems, which can amass at a rate of several gigabytes per hour.
- Variety reflects the recognition among analysts that including data from many sources can lead to more valuable and unexpected insights. Today's vehicles are equipped with dozens of sensors—and instrumented fleet vehicles have many more—all generating a variety of signals including speed, fuel consumption, and temperature. Different vehicles produce different types of data. For example, hybrid and electric vehicles may report battery charge information rather than fuel flow. In our system, data from the vehicle is combined and time-aligned with data from other sensors, including GPS devices. Lastly, all of this time-series data can be complemented by simulation, audio, video, and CAN log data, among other types.

- Velocity, in big data terms, describes the speed at which data is accumulated. The MathWorks demonstration system collected data from several sensors every second. In practice, automotive sensors with sampling rates of 10 milliseconds generate a hundred data points per second. Data streaming in from multiple sensors across numerous vehicles quickly mounts into the gigabyte and terabyte range.

Organizations seeking to extract value from big engineering data must have systems in place to handle the sheer volume, variety, and velocity of that data. These systems must include flexible, scalable data analytics algorithms deployed in a production environment so data scientists, product engineers, and business analysts can derive insights from the data.

Building a Big Engineering Data Analytics System Using MATLAB

The MathWorks Consulting Services team, who have worked with many customers to build data analytics applications, wanted a framework for gathering and analyzing big engineering data that we could experiment with, explore, study, and share.

There are many architecture options for MATLAB data analytics applications. The system we built (Figure 1) begins with an adapter that plugs into a vehicle's ODB2 port and transmits automotive data wirelessly via Bluetooth to a smartphone. The phone relays the data to a cloud application running on a Linux-Apache-MongoDB-Rails (LAMR) stack on Amazon EC2. In this system, the vehicles act as edge nodes in an Internet of Things (IoT) solution, generating data that is gathered and sent to a data aggregator. The system uses the Java-based Hadoop software framework to simplify the distributed storage and processing of big engineering data across computer clusters.

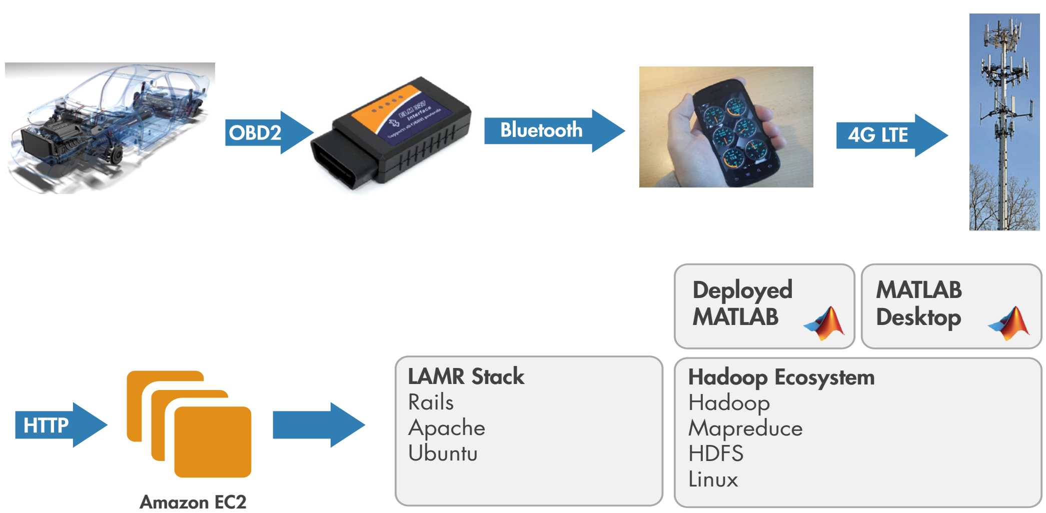


Figure 1. Hardware and software infrastructure for collecting fleet data and performing data analytics with MATLAB and Hadoop.

With this data collection infrastructure in place, we employed a MATLAB based workflow that starts with standard data analysis techniques on a single computer and culminates with the deployment of a production web application. We used MATLAB on the desktop to explore the data we gathered and to develop, test, and visualize ideas for processing it. Building upon our work on the desktop, we developed and deployed the Fleet Data Analysis web application, which combines packaged MATLAB analytics with Hadoop and other web technologies to enable the visualization, optimization, and analysis of vehicle fleet performance characteristics (Figure 2).

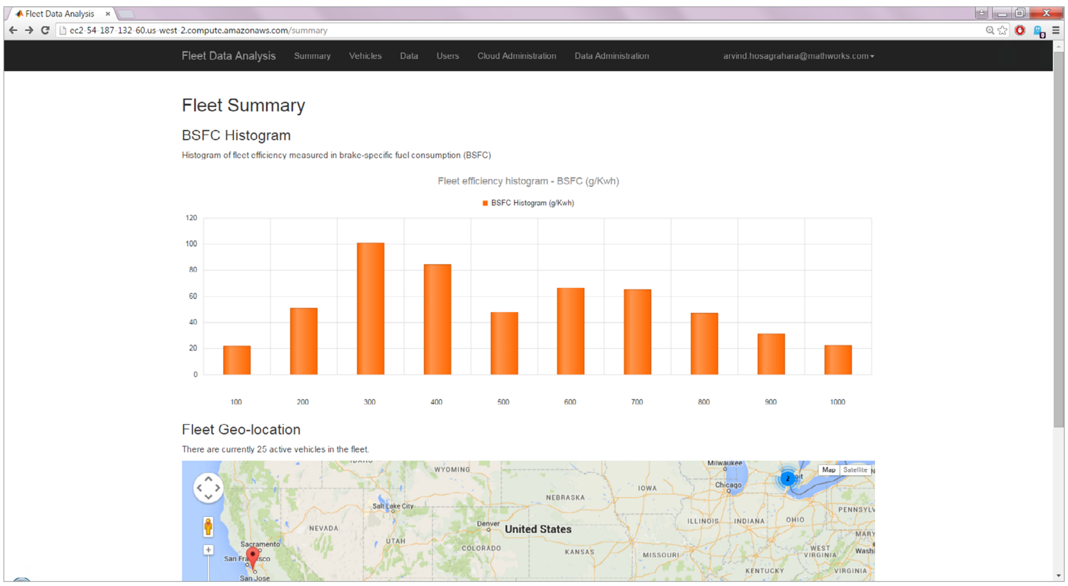


Figure 2. The MATLAB based Fleet Data Analytics web application.

MapReduce with MATLAB and Hadoop

Among the challenges of processing big engineering data sets is that they are often too large to fit into available memory and take too long to analyze on a single processor. The MapReduce programming technique addresses these challenges by processing data in small chunks that individually fit into memory to produce intermediate results, and then aggregating these intermediate results to produce a final result. Hadoop MapReduce is a popular implementation that works with the Hadoop Distributed File System (HDFS), and MATLAB provides its own implementation of the MapReduce technique.

Analysts can interactively develop and debug algorithms with MapReduce in MATLAB on the desktop. They then have two options for running their algorithms—without any modifications to the code—using Hadoop. They can use MATLAB Distributed Computing Server™ to execute MATLAB MapReduce based algorithms within Hadoop MapReduce for data that is stored and managed on Hadoop. Alternatively, they can use MATLAB Compiler™ to create applications based upon MATLAB MapReduce for deployment within production instances of Hadoop.

Traditional data analysis techniques typically involve moving data into a computational environment to be analyzed. This approach—bringing the data to the analytics—may not be feasible for big engineering data. Support for Hadoop in MATLAB enables teams to bring analytics to the data, and leave big data where it is stored.

Gaining Insights from Big Engineering Data Analytics

Organizations seeking to use big data to drive real improvements in engine and vehicle design should enable anyone in the organization to extract insights from the data that has been accumulated and stored. This section describes three examples of insights gained through data analytics with MATLAB.

Understanding Real-World Brake Specific Fuel Consumption

Automotive engineers know that even a well-designed engine cannot operate at maximum efficiency when attached to a transmission that is not tuned to operate with that engine. Brake-specific fuel consumption (BSFC) measures, which reflect not only energy efficiency but also how well an engine is tuned, are used to optimize gearboxes, shift schedules, and other transmission parameters. Although BSFC is often addressed as a simulation-optimization problem, a data-driven approach provides a different perspective, giving a view of fuel economy across real-world driving patterns.

We used MATLAB data analytics to study BSFC for the approximately 25 vehicles in the sample fleet, operated by real drivers going about their daily commute and other trips. In MATLAB, we plotted torque as a function of engine speed and added hyperbolas of constant power to create the BSFC map shown in Figure 3.

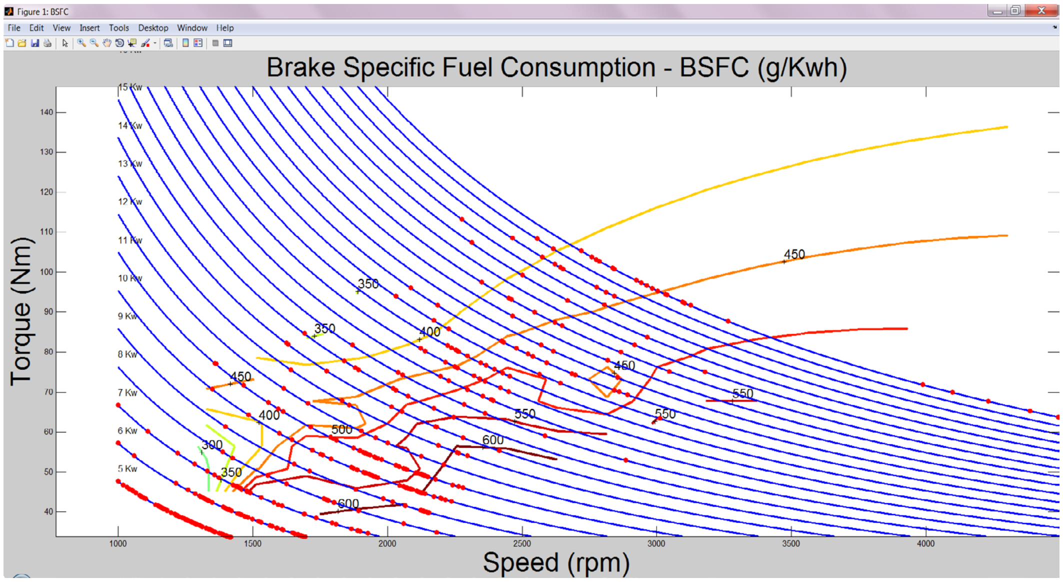


Figure 3. BSFC map created in MATLAB based on real-world fleet data.

The real-world data points used to create the BSFC map carry much more information than just engine speed and torque. Extra data may include latitude, longitude, and altitude, so we can determine where any vehicle in the fleet is while its engine is operating in its efficiency sweet spot. Additional data may also include noise and vibration metrics, which can yield insights into how measures to improve fuel efficiency may run up against noise, vibration, and harshness (NVH) constraints.

Assessing Infrastructure Changes

Traffic engineers can use MATLAB data analytics with the same fleet data to identify problematic traffic patterns and evaluate ways to resolve them. For example, we developed analytics to identify areas in which fleet vehicles were consuming the most fuel. We integrated the MATLAB analytics with third-party business intelligence analytics software to rapidly construct an interactive web dashboard (Figure 4). User interactions with the dashboard trigger calls to MATLAB code to recompute results, which are then used to update the dashboard’s visual components.

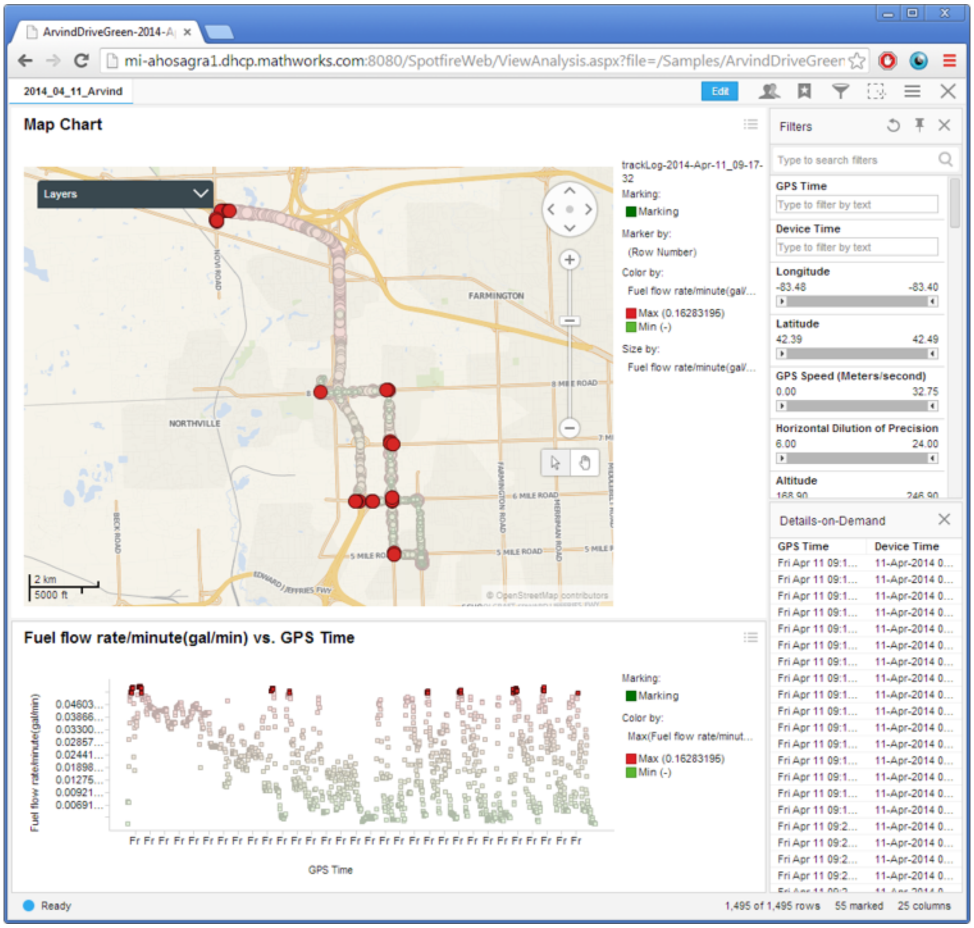


Figure 4. Interactive web dashboard created and powered by MATLAB and business intelligence analytics software.

This dashboard highlighted several intersections at which lengthy idle times led to particularly inefficient fuel consumption. Exploring the data further in MATLAB, we computed how much fuel was consumed by a vehicle waiting for nearly 90 seconds at one of the most problematic intersections. Based on the result—0.035 gallons per vehicle—we estimated that replacing the intersection with a roundabout would save more than 120 gallons of fuel per day for local commuters and reduce CO2 emissions by about 4.5 million pounds per year. For an automotive OEM, such an analysis could be used by engineers to optimize car features, such as a start-stop system, using real-life traffic patterns.

Understanding Veracity

Volume, variety, and velocity are central to any discussion of big data. Organizations have recognized that understanding the veracity of their data is vital to avoid drawing the wrong conclusions from data analytics. Consider, for example, an automotive engineering team that is creating prognostics models by training them with fleet data. If the data used to train the models is erroneous, then business decisions based on those models will be unfounded. Similarly, a team training networks for machine learning applications would face a comparable challenge, because a classifier model that is trained on faulty data will not perform as anticipated.

One of the drivers for our fleet had applied an aftermarket software modification that promised higher peak torque as well as increased boost pressure and optimized ignition timing. When driving the car it was easy to feel the increase in torque that the modification provided compared to the stock setting. When we examined the data, however, we found almost no difference between the settings. We realized that the aftermarket software was conditioning the data—essentially our data was lying to us. The discovery that certain data collected in the field may be untrue is in itself a valuable insight, because it can help organizations avoid making faulty design decisions based on that data.

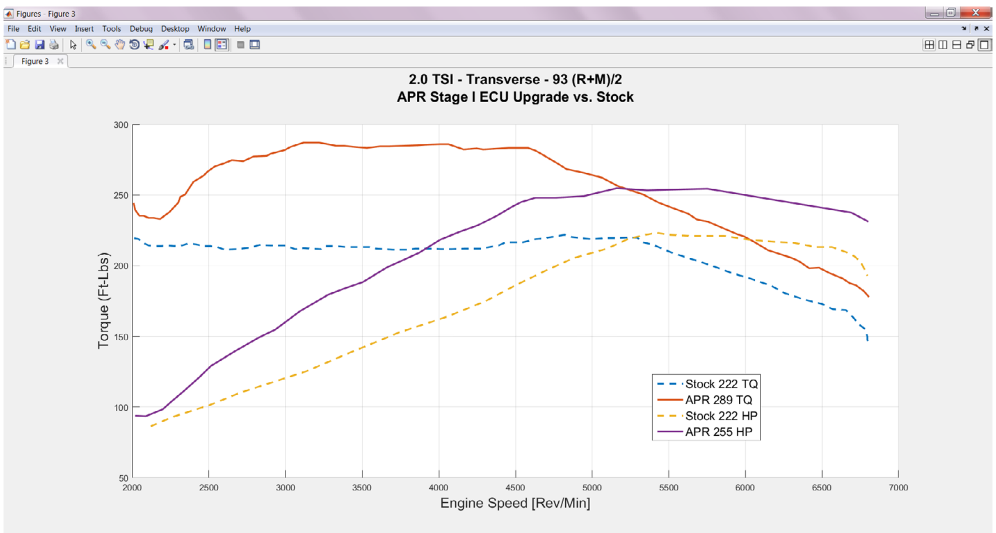


Figure 5. Estimated performance of stock vs upgraded ECU

Conclusion

Across a wide range of automotive use cases, MATLAB provides an open and extensible data analytics platform that supports a proven workflow for the rapid development, refinement, and deployment of data analytics applications. MathWorks is continuing to invest in big data analytics and build upon existing MATLAB support for datastores, MapReduce, Hadoop, and related technologies. For those who seek a quick start and expert guidance, MathWorks Consulting Services is available to help organizations improve their ability to extract valuable insights from big engineering analytics.

Watch MathWorks consultant Arvind Hosagrahara summarize the result of the yearlong research project to implement a vehicle fleet test data analytics system, including interesting insights into automotive engineering data for different use cases.

MathWorks®

92951v00 10/15