SAE TECHNICAL PAPER SERIES

2008-01-0288

Designing Automotive Subsystems Using Virtual Manufacturing and Distributed Computing

William Goodwin and Amar Bhatti
General Motors Corporation

Michael Jensen
Synopsys®, Inc.

ISBN 978-0-7680-1633-8

2008 World Congress
Detroit, Michigan
April 14-17, 2008

400 Commonwealth Drive, Warrendale, PA 15096-0001 U.S.A. Tel: (724) 776-4841 Fax: (724) 776-0790 Web: www.sae.org

By mandate of the Engineering Meetings Board, this paper has been approved for SAE publication upon completion of a peer review process by a minimum of three (3) industry experts under the supervision of the session organizer.

ISSN 0148-7191

Printed in USA

2008-01-0288

Designing Automotive Subsystems Using Virtual Manufacturing and Distributed Computing

William Goodwin and Amar Bhatti
General Motors Corporation

Michael Jensen
Synopsys®, Inc.

ABSTRACT

Adopting robust design principles is a proven methodology for increasing design reliability. General Motors Powertrain (GMPT) has incorporated robust design principles into their Signal Delivery Subsystem (SDSS) development process by moving traditional prototype manufacturing and test functions from hardware to software. This virtual manufacturing technique, where subsystems are built and tested using simulation software, increases the number of possible prototype iterations while simultaneously decreasing the time required to gather statistically meaningful test results. This paper describes how virtual manufacturing was developed using distributed computing.

INTRODUCTION

The design of an automotive signal delivery subsystem requires the analysis of a number of components and their associated interactions. A typical automotive subsystem incorporates electrical, mechanical, and hydraulic components and includes multiple transformations along the signal path. These transformations introduce noise, error, and non-linearity into the subsystem. There are a number of traditional methods used to analyze the performance of these subsystems. One method involves analyzing a large sample size of selected components within the subsystem. While this approach is effective at reducing warranty costs associated with individual components, it is ineffective at addressing subsystem design issues. Another analysis approach is to test a small sample of subsystems to study both components and their interactions. The problem with this approach is the sample size used is statistically insignificant compared to the volume of subsystems that are eventually produced. This approach, therefore, provides very little information on the full range of problems typically encountered in production. A structured development methodology, based on the principles of robust design and incorporating simulation, is needed to account for multiple design variations without building hundreds or thousands of physical subsystem prototypes. In support of this structured methodology, GMPT has developed a new approach to subsystem analysis using statistical methods, simulation tools, and distributed computing. Known as virtual manufacturing, this subsystem design methodology can accurately characterize post-production performance in less time and cost then traditional methods.

ROBUST DESIGN

Robust design is a proven development methodology for improving system reliability. Popular methodologies include Taguchi, Design for Six Sigma (DFSS), and Total Quality Management (TQM). As applied to the automotive design market, the objective is to make subsystem performance immune to variations that inevitably affect it over its service lifetime. The result is a subsystem that is optimized for performance, reliability, and cost.

As shown in Figure 1, general robust design theory[1] looks at an automotive subsystem with respect to four factors: signal factors, response factors, noise factors, and control factors. The signal and response factors describe the nominal performance of the subsystem. Noise factors are the sources of variation that adversely affect a subsystem's performance. Control factors are the techniques used to compensate for the variations.

A robust design methodology requires testing of multiple variations to ensure the subsystem operates reliably under a broad range of conditions. These conditions include variations in component and manufacturing tolerances, environmental conditions, use patterns, and subsystem aging. While a robust design methodology does not require simulation, accounting for these variations using traditional hardware prototype test methods is nearly impossible with typical development

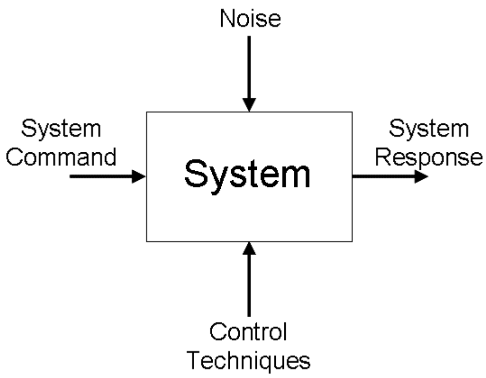

Figure 1: Factors that determine, and influence, subsystem performance

schedules and budgets. The only practical test method is to use subsystem modeling and simulation. Models should be chosen or developed that accurately model the components and allow changes in parameter values in order to simulate variations. In most cases, a full-featured hardware description language like the IEEE standard VHDL-AMS language or the MAST® modeling language from Synopsys are needed to achieve the best model accuracy and flexibility. Simulation tools should be chosen that allow critical design parameters to be identified, their values parameterized, and the entire system to be exercised in an advanced statistical analysis.

A detailed statistical analysis is the foundation of a robust design methodology. Tolerances and statistical distributions are assigned to the parameters that most affect subsystem performance. The analysis then runs a series of simulations based on random combinations of these parameters. During the statistical analysis, values are assigned to parameters based on the nominal value, its assigned tolerance and statistical distribution, and the random number seed of the analysis. Once the statistical setup is complete, the number of simulation runs required to produce statistically meaningful results must be determined. The number of runs depends on the complexity and characteristics of the design. Simulation runs on the order of several hundred to a few thousand are typically required.

GMPT'S DEVELOPMENT SYSTEM

The robust design and development of electronic controls at GMPT is accomplished using their SDSS development process. As shown in Figure 2, this SDSS process accounts for two distinct signal paths: from the sensor to the software, and from the software back to the actuator. In the first case, signal delivery is defined as the means of translating a physical parameter from a sensor measurement to the application software in the electronic controller. In the latter case, it means translating a digital word from the application software into the desired actuator response. The SDSS process, therefore, is intended to analyze the behavior of both the feed-forward and feedback paths of an automotive control subsystem.

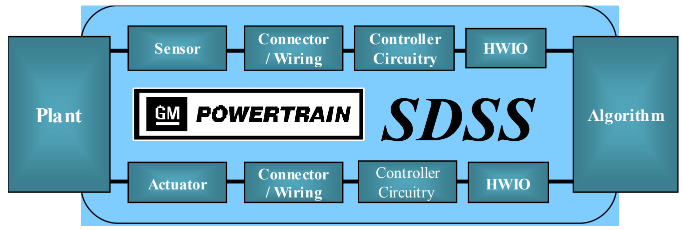

Figure 2: GMPT's Signal Delivery Subsystem (SDSS) process flow

Being based on robust design principles, a primary objective of the SDSS process is to quantify how all known sources of variation may affect a specific set of subsystem responses. These sources of variation can include component tolerances, environmental conditions, and effects due to aging. An automotive subsystem with even a small number of components can have numerous sources of variation. Quantifying any subsystem performance metric, therefore, requires determining how these sources of variation affect the response.

Traditional methods used to develop automotive electronic controls rely on the use of prototyping and testing. This approach to design, however, has been falling out of favor due to shortened development cycles and pressures to reduce development costs. The introduction of statistical quality processes, such as Design for Six Sigma, has shown that prototyping is ineffective at predicting post-production performance.

The SDSS process has addressed these shortcomings by incorporating subsystem modeling and simulation into the design of automotive electronics controls. Physics based component and subsystem models developed using Synopsys' Saber® simulator minimize the traditional prototype build-then-test process by creating and testing virtual subsystem prototypes. Both GMPT and its suppliers who participate in the SDSS process report reductions in both development time and costs.

VIRTUAL MANUFACTURING

Understanding the effects of changes in subsystem performance variables can require hundreds or even thousands of samples to be statistically significant. Building and testing this many subsystem prototypes is economically impractical. GMPT solves this problem by combining the subsystem virtual prototype with statistical simulation to completely test the design. To support this process, GMPT has developed a virtual assembly line that allows them to effectively manufacture and test thousands of subsystem prototypes in a fraction of the time required for traditional hardware prototyping. This approach helps design teams understand and characterize how variations affect product performance.

This virtual approach to manufacturing gives GMPT the ability to build and test hundreds or even thousands of subsystem virtual prototypes in a fraction of the time required to build a relative handful of hardware prototypes. Statistical simulation for even a moderately complex system, however, can be compute intensive. Depending on how the subsystem and statistical analyses are setup, the compute load can be beyond the practical limits of a single CPU. The solution to this problem is distributed computing.

Using distributed computing technology, GMPT and Synopsys have employed a cluster of computers to develop an efficient virtual assembly line. This assembly line can manufacture and test hundreds of subsystem prototypes in hours. With various automation techniques, simulation, analysis, and report generation activities can take place unmanned 24 hours a day, 7 days a week.

The statistical results generated from these thousands of virtual build and test cycles can be used to quantify the affects of variation. Design changes or modifications can then be made to these subsystems to improve performance and product quality while simultaneously reducing development costs.

DISTRIBUTED COMPUTING

In a distributed computing environment, compute intensive loads that have traditionally been executed on a single computer can be spread across a grid of multiple CPUs. This is key for performing statistical analyses on complex automotive subsystems.

By its very nature, a statistical analysis is a series of individual simulation runs. Each run differs from the previous only by a change in parameter values. This is analogous to building prototypes using parts with different values. On a single CPU, these individual runs are executed in series. In a distributed computing environment, individual runs are spread across the compute grid and are executed in parallel. This parallel processing capability significantly improves simulation performance and is limited only by the number of computers on the grid.

Traffic on the grid is controlled by a grid manager program, which manages the grid resources and monitors grid traffic to know when to submit additional simulation tasks. Once a CPU on the grid is assigned a simulation task, the CPU is "locked" while the simulation runs. When the simulation is complete, results are assembled into a common data file, the compute resource is released, and the grid manager repeats the process until the overall statistical analysis is complete.

SIMULATION EXAMPLE

To better understand the benefits of distributed computing, consider a variable valve timing (VVT) subsystem for an internal combustion engine. A typical engine has intake and exhaust valves. The operation of these valves is controlled by one or more camshafts, which have a cam lobe for each valve. These lobes open and close each valve in-synch with the engine's crankshaft. The lobe is designed to open the valve for a specific length of time and is optimized for a particular engine power band, trading-off between low-end torque and high-end power. The objective of a VVT subsystem is to adjust the timing between the cam and crankshafts in order to improve the performance of the engine across the range of its power band. While there are different ways to create the VVT effect, one of the most popular is the cam phaser subsystem. An example of a cam phaser subsystem for a four-cylinder overhead cam engine is shown in Figure 3. This subsystem interprets information from cam and crankshaft sensors to control the operation of cam phasers. These phasers ultimately control the timing between the crankshaft and the intake and exhaust valves.

The performance of the cam phaser subsystem can be affected by various design parameters. Because the subsystem is fairly complex, a statistical analysis of even moderate size executed on a single CPU may take several hours or days to complete, depending on the level of complexity, the number of statistical simulation runs, the analysis setup, and the configuration of the CPU. With a properly configured distributed computing environment, this same statistical analysis can be performed in a fraction of the single-CPU time.

A common simulation test setup verifies cam phaser subsystem operation as it responds to a step input. For the subsystem in Figure 3, the output of interest is the position of the cam. The cam phaser was first set to 8 cam degrees. Once the cam position settled, a step input was applied to achieve 24 degrees. As shown in Figure 4, the risetime of the cam position for this test is equal to 0.276 seconds for the nominal analysis.

A second simulation on this same subsystem was performed to understand how variations in just the mechanical components affect performance. Once again, the stimulus to the subsystem was a step input from 8 to 24 degrees. The results from a Monte Carlo analysis showed that variations due to just the mechanical components resulted in a risetime variation from 0.198 to 0.327 seconds. The results from this Monte Carlo simulation are shown in Figure 5.

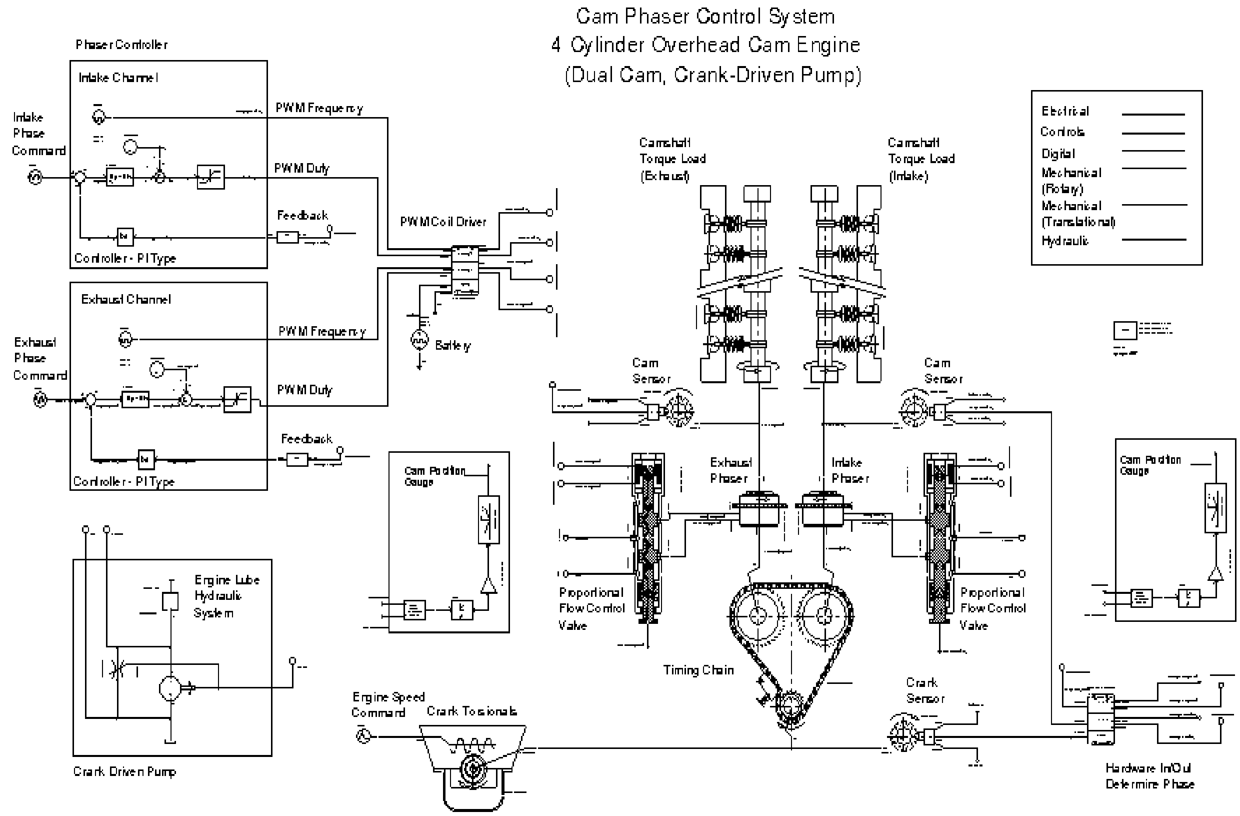

Figure 3: Cam phaser control system schematic

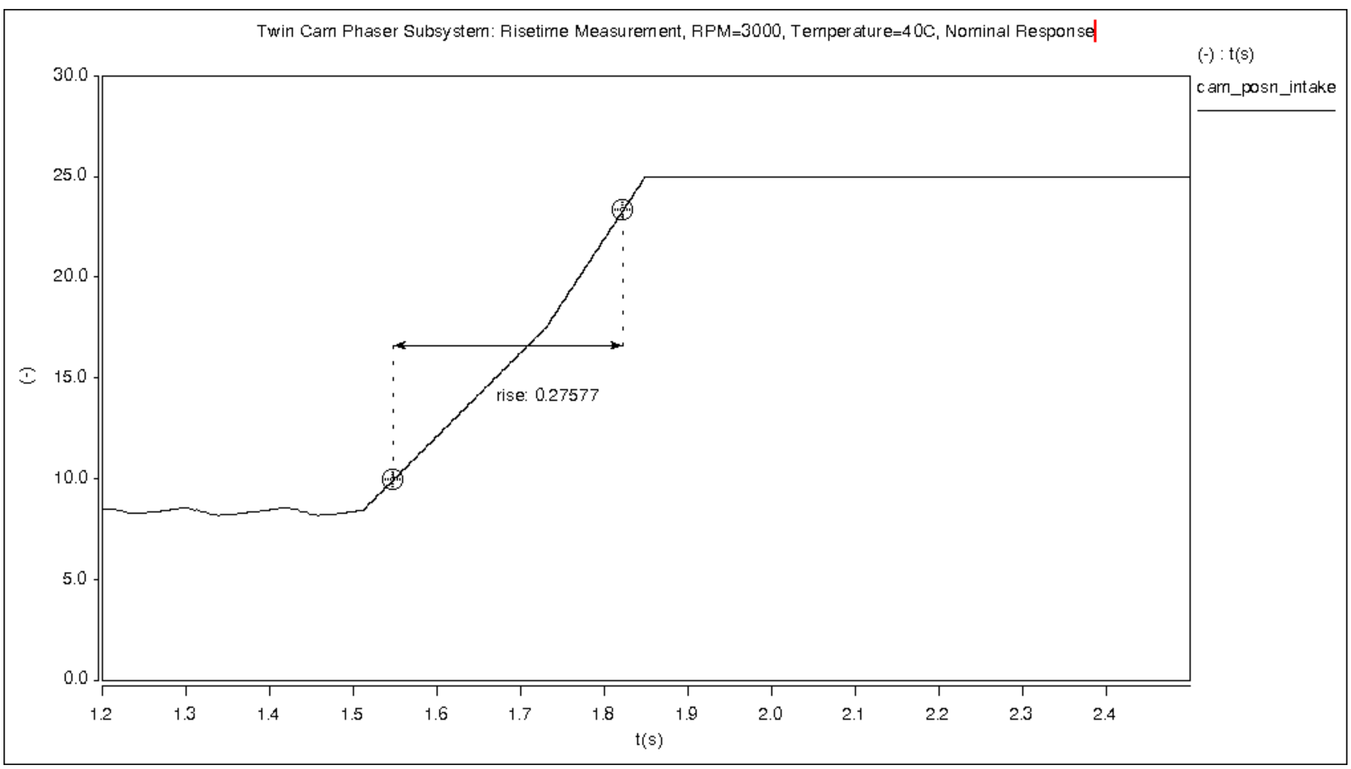

Figure 4: Plots of cam phaser nominal simulation results

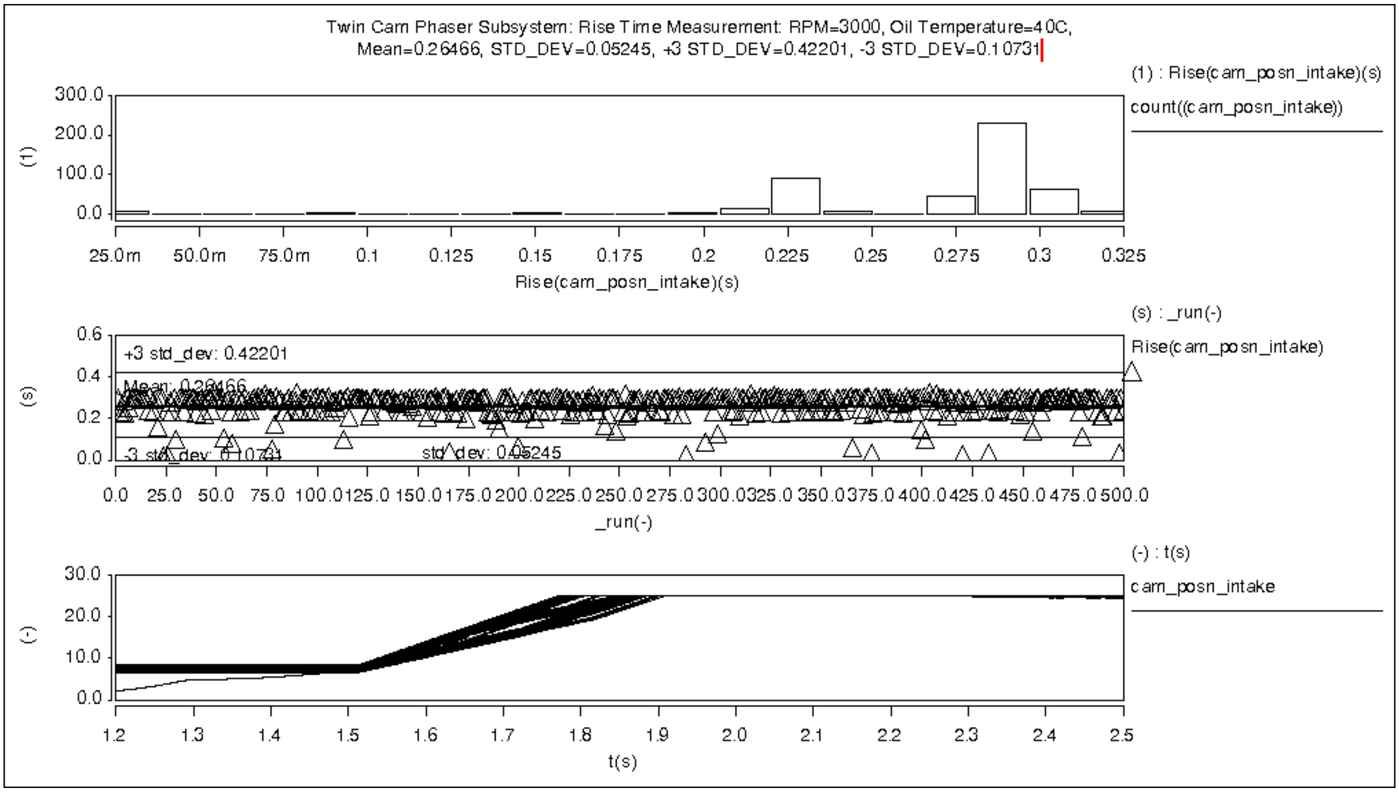

Figure 5: Plots of cam phaser subsystem Monte Carlo analysis results

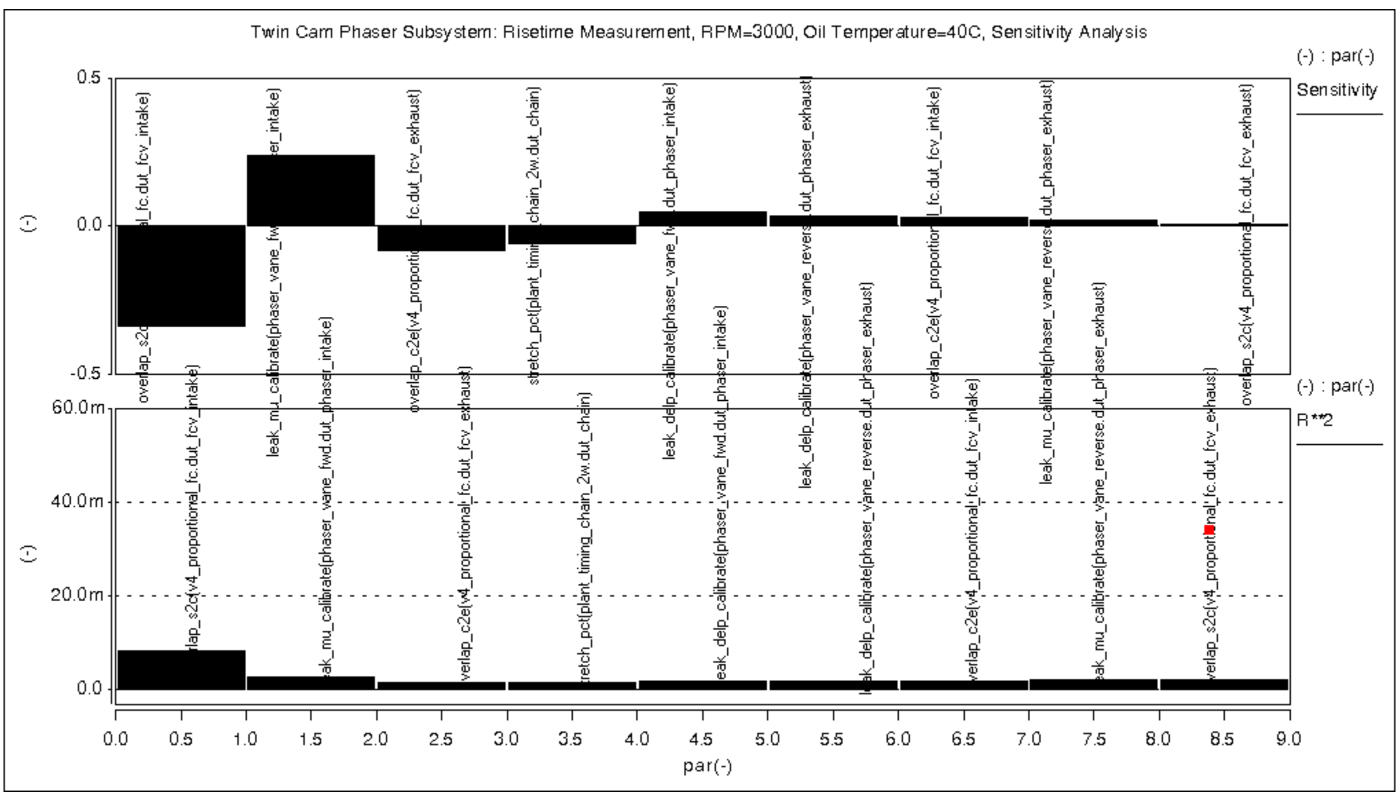

Figure 6: Plots of cam phaser subsystem sensitivity analysis results

These Monte Carlo results were then used to perform a sensitivity analysis on the subsystem. The results from this analysis were used to determine the sources of variation that have the greatest effect on cam position. Sensitivity analysis results are shown in Figure 6.

These results indicate that the largest sources of variation are the overlap in the control valve and leak in the phaser. As this plot illustrates, sensitivity results easily identify the sources of variation that need to be managed as well as those that can be ignored.

SIMULATION PERFORMANCE

Single CPU versus grid simulation execution times were investigated at GMPT as a student intern project during the summer of 2006. Students worked with Mr. Bhatti and Mr. Goodwin to quantify the effect of grid computing on simulation performance for statistical analysis in general, and for a cam phaser subsystem in specific. The cam phaser subsystem used in this investigation was similar to that used for the simulation example presented in this paper. Each node in their 24 CPU compute grid was configured as shown in Table 1.

CHARACTERISTIC	DETAILS
Platform	HP XW 8000 or XW 8200
Processor	2 Intel® Xeon® processors 3 GHz or 3.3 GHz
Operating System	Windows® Server 2003 SP2
RAM	2-3 GB
Hard Disk Space	34 GB
Simulation Software	Saber simulator 2006.06 SP2
Grid Management Program	Platform LSF® 6.1 from Platform™ Computing, Inc.

Table 1: CPU configuration

The compute grid was configured as shown in Figure 7. User nodes were connected to the grid CPUs through the LSF® grid manager program from Platform Computing, which managed the compute load for the grid of 24 CPUs. A file server was also added to the grid to handle program and datafile requirements.

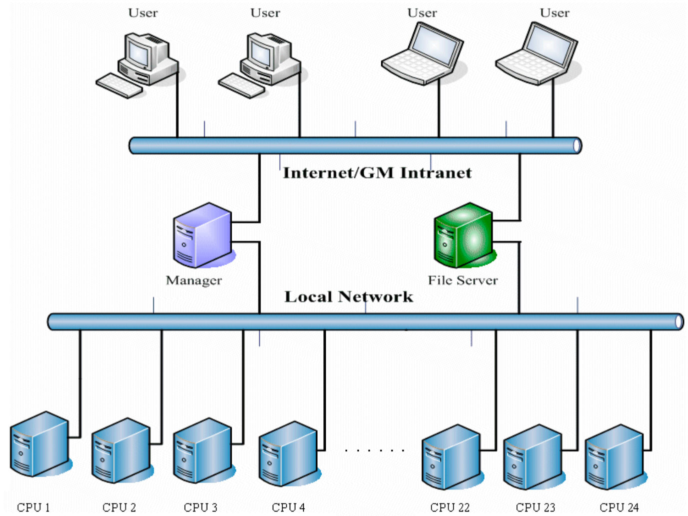

Figure 7: Compute grid configuration used to quantify the effects of grid computing on simulation performance

Simulation performance results, as shown in Figure 8, provide case study evidence of the advantages of grid computing. The graph compares single and grid-based CPU execution times for the individual simulation runs of a Monte Carlo analysis.

A Monte Carlo analysis is an algorithmic method for analyzing the statistical performance of a subsystem. The analysis is divided into a series of individual simulation runs, where the number of runs is user-specified at the beginning of the analysis. One-thousand runs was used as the benchmark for this simulation performance study.

As shown in Figure 8, the average execution time for a single run of the statistical analysis is 220.5 seconds on a single CPU, and 17.8 seconds for a grid of 24 CPUs. Based on these averages, a single CPU running 24 hours a day would require just over 2.5 (2.55) days to complete the 1000 run Monte Carlo analysis. The compute grid is able to finish the same 1000 run analysis in just under 5 (4.94) hours. The compute grid, therefore, is able to complete the analysis over 12 times faster than with a single CPU. This represents a significant decrease in the time it takes to complete the subsystem design. The result is a considerable savings in development resources and a faster time to market.

To further validate the benefits of distributed computing in a virtual manufacturing methodology, a single engineer at GMPT recently completed a backlog of 60,000 simulation runs in 2.5 weeks.

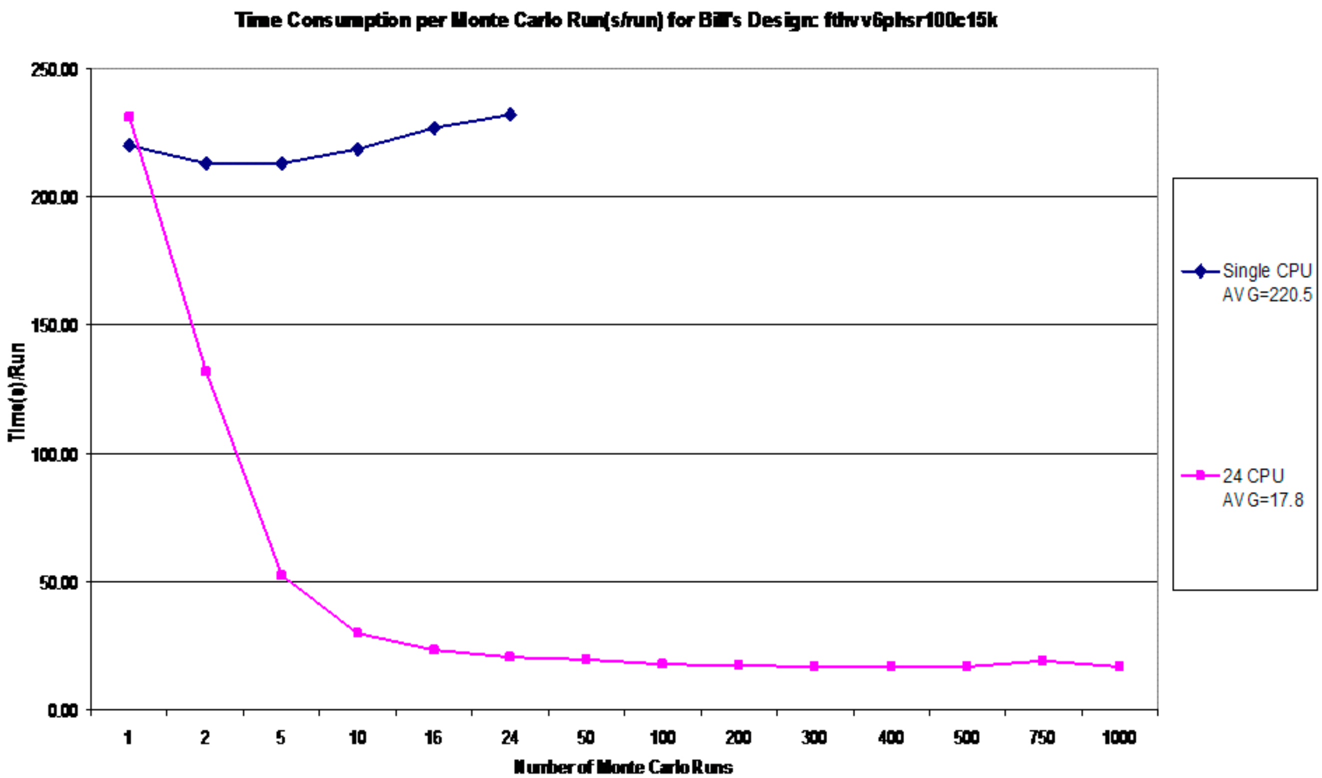

Figure 8: Single CPU vs. compute grid simulation performance for the statistical analysis of a cam phaser subsystem

CONCLUSION

Advances in automotive technology require increasingly complex subsystem designs. As subsystems become more complex, issues of quality and reliability become more prevalent. Robust design methodologies, where the effect of variations on subsystem performance are an integral part of the design process, provide an organized approach for verifying subsystem quality and reliability. Implementing a robust design methodology for complex subsystems requires the use of virtual prototyping in a virtual manufacturing environment. GMPT uses their SDSS development process, along with the Saber simulator from Synopsys, to implement a virtual manufacturing process to build and test complete subsystems. In this way, GMPT is able to build and test hundreds, or even thousands, of subsystem virtual prototypes in the same time that it would take to build and test a handful of physical prototypes. Key to this process is a detailed statistical analysis performed on a compute grid. The statistical analysis implements the virtual manufacturing environment while grid computing reduces the time required to gather complete simulation data. This paper describes GMPT's SDSS development process, explains the importance of the virtual manufacturing concept, illustrates key principles by simulating a cam phase subsystem, and shows the performance gains possible when using a distributed computing environment to perform the required statistical analyses.

REFERENCES

1. Madhav S. Phadke: *Quality Engineering Using Robust Design*, Prentice Hall, ©1989 AT&T Bell Laboratories, Chaps. 1, 2.
2. Vamshi Gantla, Suilin Wang: *AITE Program at GM Proving Grounds, Milford* summer intern project and presentation.

CONTACT

William Goodwin
Senior Project Engineer
Electronic Integration & Software
General Motors Corporation
Email: william.goodwin@gm.com

Amar Bhatti
IT Manager
GMPT IS&S Global Product Development Electrical, Controls & Software
General Motors Corporation
Email: amar.bhatti@gm.com

Michael Jensen
Corporate Applications Engineer, Saber Product Line
Synopsys, Inc.
Email: mikej@synopsys.com
Web: www.synopsys.com/saber