08MV-13

System Level RBDO for Military Ground Vehicles using High Performance Computing

David A. Lamb, David Gorsich, Dmitriy Krayterman
U.S. Army RDECOM-TARDEC

Kyung K. Choi, Ed Hardee, Liu Du
University of Iowa

Byeng D. Youn
University of Maryland

Bernie Bettig
West Virginia University

Dan Ghiocel
Ghiocel Predictive Technologies Inc.

ABSTRACT

The Army continues to improve its Reliability-based Design Optimization (RBDO) process, expanding from component optimization to system optimization. We are using the massively parallel computing power of the Department of Defense (DoD) High Performance Computing (HPC) systems to simultaneously optimize multiple components which interact with each other in a mechanical system. Specifically, we have a subsystem of a military ground vehicle, consisting of more than four components and are simultaneously optimizing five components of that subsystem using RBDO methods. We do not simply optimize one component at a time, sequentially, and iterate until convergence. We actually simultaneously optimize all components together. This can be done efficiently using the parallel computing environment. We will discuss the results of this optimization, and the advantages and disadvantages of using HPC systems for this work.

INTRODUCTION

To have Army ground vehicles play a better role in the Army's vision of rapid deployability, mobility, sustainability, and maintainability, the reliability of ground vehicles needs to be improved while reducing their weights. That is, better logistics (fuel efficiency) and unsurpassed mobility/maneuverability (enhanced strategic deployability and greater tactical mobility) require lighter vehicles. On the other hand, sustainability and maintainability require ultra-reliable and/or redundant components to remain operationally effective for a sustained mission period with minimal maintenance service or repair. As a result, it is necessary to reduce demand and minimize the maneuver sustainment burden on the ground vehicle effectiveness through balanced system reliability, redundancy, and repair, and to include embedded diagnostics and prognostics as well as modular component design. The challenge is that weight-optimized vehicles would be much more susceptible to uncertainty in order to maintain ultra-reliability. Furthermore, The FCS initiative is setting a challenging standard for reliability, which is calling for improvements even to current Army 'reliable' systems.

The objective of this project is to develop a modeling and simulation (M&S) software system that can be used to optimize for improvement/design of Army ground vehicles for reliability and durability while minimizing their weights. The envisioned M&S software system will demand major computational effort to obtain an optimized vehicle for system level reliability. To carry out system level reliability-based design optimization (RBDO) for durability with reduced vehicle weight on a single processor may take many months. This is where RDECOM-TARDEC's High Performance Computing (HPC) facility will offer significant advantages such that the whole ground vehicle system level RBDO could be achieved within a week of computation time.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **21 DEC 2007**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE
System Level RBDO for Military Ground Vehicles using High Performance Computing

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
David A. Lamb; David Gorsich; Dmitriy Krayterman; Kyung K. Choi; Ed Hardee; Liu Du; Byeng D. Youn; Bernie Bettig; Dan Ghiocel

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
US Army RDECOM-TARDEC 6501 E 11 Mile Rd Warren, MI 48397-5000, USA University of Iowa; University of Maryland; West Virginia University; Ghiocel Predictive Technologies Inc.

8. PERFORMING ORGANIZATION REPORT NUMBER
18569

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
US Army RDECOM-TARDEC 6501 E 11 Mile Rd Warren, MI 48397-5000, USA

10. SPONSOR/MONITOR'S ACRONYM(S)
TACOM/TARDEC

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
18569

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
Presented at the Society of Engineers (SAE) World Congress, Detroit,MI,, The original document contains color images.

14. ABSTRACT

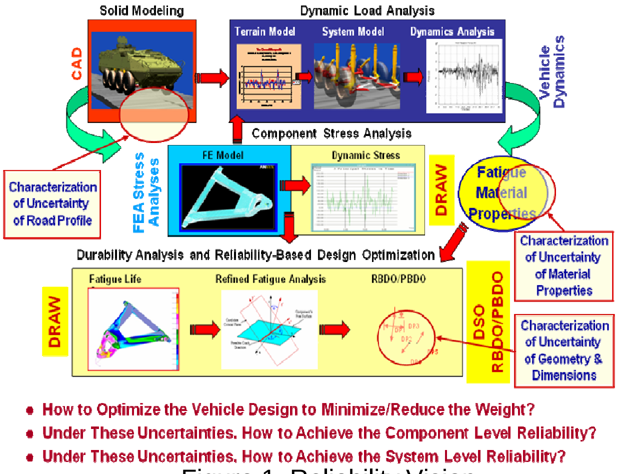
The Army continues to improve its Reliability-based Design Optimization (RBDO) process, expanding from component optimization to system optimization. We are using the massively parallel computing power of the Department of Defense (DoD) High Performance Computing (HPC) systems to simultaneously optimize multiple components which interact with each other in a mechanical system. Specifically, we have a subsystem of a military ground vehicle, consisting of more than four components and are simultaneously optimizing five components of that subsystem using RBDO methods. We do not simply optimize one component at a time, sequentially, and iterate until convergence. We actually simultaneously optimize all components together. This can be done efficiently using the parallel computing environment. We will discuss the results of this optimization, and the advantages and disadvantages of using HPC systems for this work.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**SAR**	**6**	

THE TARDEC RELIABILITY VISION

The long-term planned vision of Army's vehicle durability optimization & reliability process is shown in Fig. 1. Tying together the different analysis software used to calculate multibody dynamics modeling and simulation, finite element analysis (FEA), fatigue calculation, and the optimization provided by the RBDO method, the U.S. Army will improve the design of the ground vehicle fleet by getting more reliability while taking into account expected variability. This is going to require that many different disciplines work together, making a significant software system out of diverse parts. In the end, a methodology will be produced for how to get a tool vehicle designers will use to optimize their designs in the face of stochastic uncertainty. That is the plan, and this project is part of the solution to get to this methodology.


Solid Modeling
Dynamic Load Analysis
Terrain Model
System Model
Dynamics Analysis
CAD
Vehicle Dynamics
Component Stress Analysis
FE Model
Dynamic Stress
FEA Stress Analyses
DRAW
Fatigue Material Properties
Characterization of Uncertainty of Road Profile
Characterization of Uncertainty of Material Properties
Durability Analysis and Reliability-Based Design Optimization
Fatigue Life
Refined Fatigue Analysis
RBDO/PBDO
DRAW
DSO RBDO/PBDO
Characterization of Uncertainty of Geometry & Dimensions

- How to Optimize the Vehicle Design to Minimize/Reduce the Weight?
- Under These Uncertainties, How to Achieve the Component Level Reliability?
- Under These Uncertainties, How to Achieve the System Level Reliability?

Figure 1. Reliability Vision

OUR GOAL

We are planning for something very ambitious, using four or five physics and many sources of uncertainty requiring Monte-Carlo techniques. Estimates climb into the tens of millions of FEA runs of small-sized models, and hundreds of years of clock time if done in serial. Fortunately, there is no need to do this in serial, since most of the FE analyses are independent, and we can parallelize. Utilizing 10,000 processors to parallelize the FEA runs will keep the turn-around time below two weeks. To be useful in influencing the acquisition process, turn-around times longer than week are not helpful. Unfortunately, we cannot immediately jump to using 10,000 processors, but will have start out more modestly and grow to that level.

THE METHOD

To realize this vision, the University of Iowa has developed an integrated software system that includes multibody dynamics of vehicle system (DADS), finite element analysis for stress influence coefficient calculation (MSC/Nastran), dynamic stress computation and durability analysis (DRAW), design sensitivity analysis of the fatigue life (DSO), and reliability/possibility-based design optimization (RBDO/PBDO/MVDO) as shown in Fig. 2.

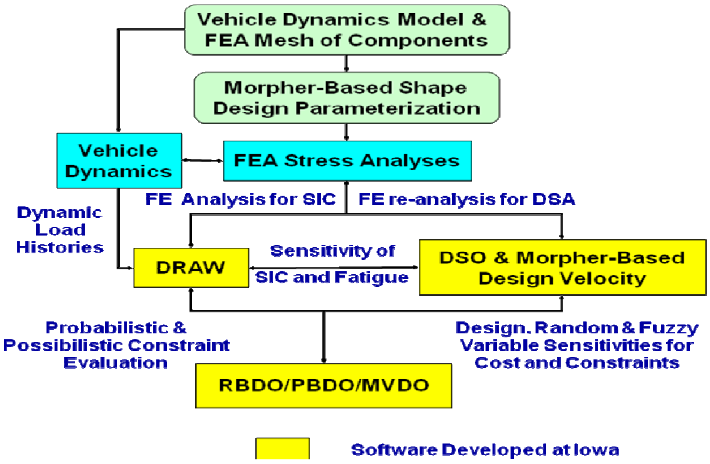

Vehicle Dynamics Model & FEA Mesh of Components
Morpher-Based Shape Design Parameterization
Vehicle Dynamics
FEA Stress Analyses
FE Analysis for SIC
FE re-analysis for DSA
Dynamic Load Histories
DRAW
Sensitivity of SIC and Fatigue
DSO & Morpher-Based Design Velocity
Probabilistic & Possibilistic Constraint Evaluation
Design. Random & Fuzzy Variable Sensitivities for Cost and Constraints
RBDO/PBDO/MVDO
Software Developed at Iowa

Figure 2. Integrated Durability RBDO/PBDO/MVDO Process Installed on TARDEC HPC

Application of the integrated computing process shown in Fig. 2 to all critical structural components of Army vehicle systems such as HMMWV is very much compute intensive with multibody dynamic analysis, durability analysis, design sensitivity analysis, and reliability-based design optimization. To speed up the computational process and realize RBDO of the vehicle system level for improved durability and minimized weight in meaningful time (i.e., within a week of computation time), it is necessary to take advantage of multiple processors at the TARDEC's High Performance Computing (HPC) facility.

THE PROJECT

We made the runs in September-October 2007 on the High Performance Computers located at U.S. Army RDECOM-TARDEC. We describe here the results seen in these runs.

We analyzed the lower driver's side A-arm from the M-1097 HMMWV. (See Figure 2.) This was analyzed to improve the design for fatigue life. We chose this part because it was very similar to another study done earlier using serial processing. In addition, there was thought to be a lot of data available for this vehicle and this part.

We wanted to do a multi-scale, multi-physics analysis of a subsystem, but as the saying goes, you have to walk before you can run. We were limited on resources we could bring to the pilot project and found that the only way to get anything run with the limitation on our resources was to be more modest in our immediate goals. This caused us to restrict ourselves for the pilot project. We only did a single component and a single physics-of-failure.

THE TARDEC HIGH PERFORMANCE COMPUTERS

The vast majority of the FE analyses were run on the Origin 3900 platform. Only the analyses that required more than 24 processors were conducted on the Onyx 350 due to the limited number of processors on the Origin 3900. (See figure 3.) Local disk space was used for all files (e.g. input, scratch, output) which helped speed up analysis run times. Specialized queues were created to handle the execution of the analyses in which the number of processors, finite element analysis code licenses, and optimization constraints varied. The queues set the number of processors and number of finite element code licenses available to the analyses.

sgi ONYX 3900: ***unix***

24 MIPS R16000 PROCESSORS
4 IR2 GRAPHICS PIPES
4 IR3 GRAPHICS PIPES
24 GBYTES MEMORY
36 GBYTES LOCAL DISK SPACE

sgi ONYX 350: ***unix***

32 MIPS R16000 PROCESSORS
4 IP GRAPHICS PIPES
32 GBYTES MEMORY
36 GBYTES LOCAL DISK SPACE

Figure 3. TARDEC HPC Assets Used in the Project

By utilizing TARDEC's HPC, a coarse grained parallelization of the computational process can be developed as shown in Fig. 4. In the formulation of RBDO to minimize the vehicle weight and improve durability, the fatigue life at the selected critical points becomes performance functions that define probabilistic constraints. Evaluations of these probabilistic constraints require a Most Probable Point (MPP) search using the First Order Reliability Method (FORM) based inverse reliability analysis. The FORM-based inverse reliability analysis for MPP search requires an optimization process, which by itself is a compute intensive process. For a typical RBDO formulation for durability with weight being the cost function, there could be a number of probabilistic constraints that depends on the critical regions of HMMWV where fatigue life is low. These probabilistic constraint evaluations could be distributed over a number of processors as shown in Fig. 4 to have coarse grained parallelization.

RELIABILITY/FATIGUE ANALYSIS SOFTWARE

We used several pieces of propriety code from the University of Iowa for this project. These included a fatigue analysis software called DRAW, a design sensitivity software called DSO and a reliability-based design optimization software, called RBDO. All three were ported from the University of Iowa to TARDEC's HPC center and installed for run.

In addition to these, we made use of some numerical analysis software called DOT from Vanderplaats. This was used primarily to perform the optimization in the loop.

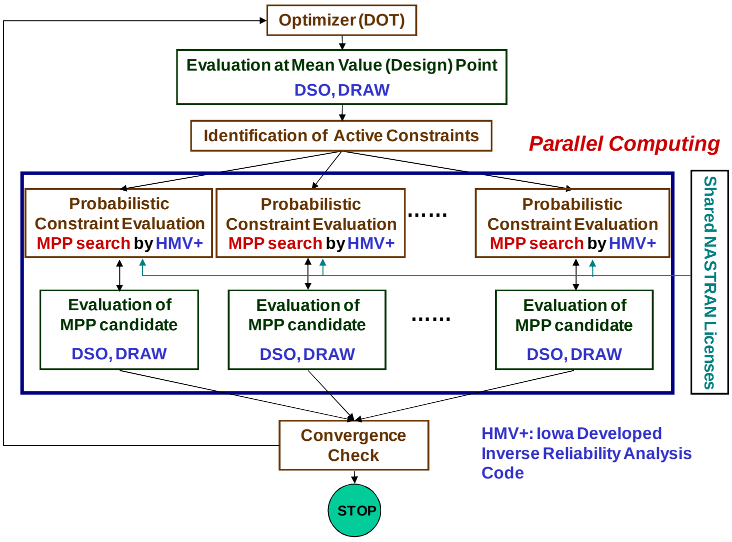

Optimizer (DOT)
Evaluation at Mean Value (Design) Point
DSO, DRAW
Identification of Active Constraints
Parallel Computing
Probabilistic Constraint Evaluation MPP search by HMV+
Probabilistic Constraint Evaluation MPP search by HMV+
……
Probabilistic Constraint Evaluation MPP search by HMV+
Evaluation of MPP candidate DSO, DRAW
Evaluation of MPP candidate DSO, DRAW
……
Evaluation of MPP candidate DSO, DRAW
Shared NASTRAN Licenses
Convergence Check
HMV+: Iowa Developed Inverse Reliability Analysis Code
STOP

Figure 4. Parallel Computing for RBDO using HPC

FINITE ELEMENT ANALYSIS SOLVER

We needed extensive use of a finite element analysis solver. For this, we choose to use NASTRAN from MSC. This turns out to be a significant roadblock and challenge for projects of this type. To accomplish significant parallelization of the method, we required that multiple copies of an FEA solver be running on different processors, solving variations of the same analysis, in parallel. Unfortunately, we found that most vendors of FEA code treat this situation as requiring a license for each solver we run. So, to run on sixteen processors required having sixteen licenses, and to run on a hundred processors would have required a hundred licenses.

So we find that this becomes a very costly hurdle for expanding this project. We are not likely to make the progress we want, if we must purchase several hundred licenses for an FEA solver to parallelize across hundreds of processors. A better way of handling this must be found to facilitate further progress.

For our pilot project, we negotiated with MSC to obtain a limited time window where we could use sixteen NASTRAN licenses for this project, but only on an experimental basis to demonstrate the method we are developing. We will then need to start buying licenses for future work.

It will be very advantageous for future work in this area to find a vendor of FEA software that will offer a better pricing scheme. What would seem best would be for the vendor to allow for multiple (hundreds?) runs of their software to be made in parallel, across hundreds of processors, on variations of the same problem, for some fixed price. Perhaps some control could be imposed to insure that all the runs are variations of the same base problem, as a way to prevent fraud. While it is not clear how to adequately protect the software vendor's interest while keeping costs reasonable, still it is obvious that without something like this, the potential for this method is very limited. We cannot easily see how to expand the current method to a hundred or more processors if we must effectively buy a license for the FEA solver for each processor utilized.

PARALLELIZATION AND WORK FLOW CONTROL

As stated before, the overall objective of the project is to be able to carry out RBDO for durability of the vehicle system (like HMMWV) while reducing the vehicle weight in a meaningful computational time period. For this purpose, reduction of real execution time using parallel processors is critical. Parallelization on the TARDEC High Performance Computing facility made it possible to execute the runs in a reasonable amount of time. A durability reliability analysis run that would normally take 1397 minutes as a serial process was performed in 206 minutes with parallelization for 15 constraints, which is more than an 85% time reduction.

As was shown in Fig. 4, the parallelization was centered on evaluating multiple fatigue life constraints simultaneously to perform inverse reliability analysis. Each simultaneous run involved the University of Iowa's RBDO code for inverse reliability analysis, the University of Iowa's DRAW code for durability analysis, the University of Iowa's DSO code for sensitivity analysis, and two MSC/Nastran Finite Element structural analyses. The code was prepared for parallelization by extracting the probabilistic constraint evaluation subroutine from the RBDO code to be a standalone executable. Then the RBDO code made parallel calls to this executable for each constraint requiring evaluation. This constraint evaluation executable performed an MPP search by HMV+ calling DSO to calculate function evaluations and sensitivities. The DRAW code and MSC/Nastran were in turn called from DSO. LSF from Platform Computing Corp. was used to implement the parallelization. LSF "bsub" commands were generated directly in the Fortran code in order to queue the execution of the constraint evaluation executable for the different constraints. The main program, the RBDO code, could be started directly from the Unix command prompt, but to obtain timing and resource usage information it was started from the Unix command prompt using a bsub command. MSC/Nastran runs were started from a Unix script file launched from the DSO code through a system call that waited for the MSC/Nastran run to finish before continuing.

SCALABILITY STUDY

To better understand the factors affecting the efficiency of our parallelization of the RBDO code, a scalability study was carried out. A series of test runs was performed on an SGI Origin 3900 with 24 MIPS R16000 processors, 24 Gigabytes of Random Access Memory and 72 Gigabytes of local disk storage which was restricted from being used by other users. Each test run involved 1 inverse reliability analysis for a given number of design constraints. The inverse reliability analysis, as shown in Fig. 4, involves the probabilistic constraint evaluation by carrying out inverse reliability analysis (MPP search) using the University of Iowa developed HMV+ code. As stated above, the parallelization was centered on evaluating multiple fatigue life constraints simultaneously where each simultaneous run involved the University of Iowa's RBDO code for inverse reliability analysis, Iowa's DSO code for sensitivity analysis, the University of Iowa's DRAW code for durability analysis, and two MSC/Nastran Finite Element structural analyses.

For the scalability study, 22 experiments (20 training runs and 2 test runs) were designed with different numbers of MSC/Nastran licenses, processors, and constraints, as shown in Table 1. Note that a dependence of the parallel runtime (PR) on the number of MSC/Nastran licenses occurs when the number of licenses is less than the number of processors and individual constraint runs are forced to wait for a license to become available. For the MPP search, finite element analysis by MSC/Nastran accounts for about 22% of computational time in a serial run. So the number of MSC/Nastran licenses has a large effect on the parallelization of the process, but does not completely control the degree of parallelization. For the 20 training runs, 1, 2, 4, 8, and 16 licenses, 1, 8, 15, and 30 processors, and 15 and 30 constraints were used. Not all possible combinations made sense for a run. In particular the number of processors should be greater or equal to the number of licenses, else there are unused licenses. We had a slight violation of this rule for runs 8 and 16, since configuring those runs for all the available 16 licenses was more natural. Also the number of constraints should be greater or equal to the number of licenses and the number of processors; else there are unused licenses or processors. Again a slight violation of this rule is present in runs 8 and 16. Finally two test runs were performed (Runs 21 and 22) as a "sanity check" on using the training runs in a predictive way.

During the parallel run, a processor is either busy with computation or idle because there are no more constraints to evaluate and it is waiting for the other processors to finish. (For simplicity, we consider time waiting for a MSC/Nastran license as part of the computational runtime and not as part of processor idle time.) Therefore we have the following relations.
For:

PR = parallel runtime in real time
CR = total computational runtime, summed up over the processors
I = total idle time, summed up over the processors
np = number of processors
nc = number of constraints

we have:

$$PR = (CR + I) / np$$

Table 1. Scalability Study Results

	Run #	No of constr.	No of licenses	No of proc.	Ave. runtime (per constraint)	Ave. idle time (per processor)	Time (PR)
Training runs	1	15	1	1	93.1	0.0	1397
	2		2	8	136.4	35.3 (282)	291
	3		4	8	125.1	23.6 (189)	259
	4		8	8	121.1	16.5 (132)	244
	5		2	15	179.1	57.6 (864)	237
	6		4	15	187.7	28.5 (428)	217
	7		8	15	191.8	13.6 (204)	206
	8		*16*	*15*	*184.9*	*17.3 (259)*	*203*
	9	30	1	1	94.1	0.0	2822
	10		2	8	126.5	53.8 (430)	529
	11		4	8	123.9	37.3 (298)	502
	12		8	8	122.4	32.3 (258)	492
	13		2	15	176.7	65.3 (979)	419
	14		4	15	170.9	33.2 (498)	376
	15		8	15	168.6	15.9 (239)	354
	16		*16*	*15*	*165.7*	*14.0 (210)*	*346*
	17	30	2	30	324.2	122.8 (3684)	448
	18		4	30	330.1	63.6 (1909)	395
	19		8	30	339.9	41.2 (1236)	382
	20		16	30	340.8	30.0 (901)	372
Test runs	21	15	7	10	125.7	53.2 (532)	242
	22	30	15	20	190.9	64.5 (1289)	352

or:

PR = (CR / nc) * (nc / np) + I / np

That is,

parallel runtime in real time = (ave. computational runtime)*(ratio of constraints to processors)
+ ave. processor idle time

For example for Run 2, parallel runtime 291 is approximately equal to 136.4 * 15/8 + 35.3. From this formula we can see that ideally it is desirable to minimize the average computational runtime, the ratio of constraints to processors, and the average processor idle time. From the experiments it appears that the average computational runtime (CR/nc) varies over the number of constraints (nc) and also varies based on the number of licenses and the number of processors (np). Similarly, the average processor idle time (I/np) is a function of the number of processors, the number of licenses, and the number of constraints. The results of the scalability study were analyzed to get an idea of what these three factors depend on, what trade-off there are between them, and which factor is the most effective and efficient to minimize. The 30 processor runs, Runs 17-20, were cautiously used in the following analysis since they were performed on a different machine from the other runs. The timings suggested that runs on the 30 processor machine were unexpectedly taking significantly longer other factors being equal. This was confirmed with subsequent testing.

THE PAYOFF

When talking about reliability, it is important to consider ‘total lifecycle cost’ as the relevant measure. This is because adding reliability often costs extra at the front end (during research, development, design and manufacturing) but realizes savings during the Operations and Sustainment phase of the life cycle due to reduced costs to keep the vehicle available. To understand the value added by the increased reliability, the key is to balance the added up front costs against the savings later on, in other words, to look at total cost across the entire life cycle of the vehicle.

Also, the projected savings from improved reliability is often based on the current level of reliability we start with (based on the law of diminishing returns). If a fleet is showing low reliability before efforts begin, then a large cost savings due to improved reliability is possible, but it is hard to realize great savings when starting from a fleet of very reliable vehicles. Based on current data from Army fleets, it appears that improved reliability in Army ground vehicles has a potential for very respectable cost savings.

Total savings will also be a function of the number of similar vehicles in the fleet based on the improved design. It is obviously easier to realize large cost savings from improving the reliability of a design with 10,000 fielded vehicles that improving the design that only fields 50 vehicles. Still, once methods are developed to improve the reliability of a design, and the cost to develop the methods is recouped from improving the design of a few vehicles, the same methods will still be available to use on all other vehicle designs with little added cost. The key, therefore, is to apply the new methods to a few systems where the development costs of the new methods can be quickly recouped, and then deliver to the Army a 'paid for' tool to improve the reliability for other platforms.

It is reasonable to assume that tens of millions of dollars in total life cycle cost savings might be realized for a fleet of a single ground vehicle design due to improved reliability designed in from the beginning. (Savings will be spread across the whole life cycle and across the fleet of similar vehicles.) If this method can be used to improve the design of just ten future vehicles, with various sizes of fleets and various results of reliability improvement for each, the method could potentially lead to savings of hundreds of millions or even billions of dollars. Even just one vehicle design will more than repay the costs of developing and implementing the method, based on modest reliability improvements to the design from the use of this tool.

CONCLUSION

While the Army strives to improve the reliability of its current and future fleets of ground vehicles, there is a great need for a tool of this sort. We want to make it a good tool, one based on physics and not heuristics, and one that considers system level reliability with interactions between components and between failure modes captured. This requires the massively parallel environment of High Performance Computing to be realized quickly enough to impact the design loop. We are working to build this technique, make it multi-physics and multi-scale and non-heuristic. As this project progresses, we will add additional complexity to the models and generate predictions that encompass more of the true range that reliability should include.

The most significant hurdle still to be made is how to obtain, at a reasonable cost, sufficient licenses for FEA solving software to parallelize across hundreds of processors as desired.

ACKNOWLEDGMENTS

We would like to acknowledge the support of all our institutions. We also want to acknowledge the kind donation of software from MSC Software Corporation and Vanderplaats Research and Development, Inc. to this effort.

REFERENCES

1. Davis, T. P., 2004: "Science engineering and statistics"; Henry Ford technical Fellow for Quality Engineering; Ford Motor Co.; http://www.timdavis.co.uk/research.htm.
2. Kokkolaras, M., Mourelatos, Z. P., and Papalambros, P. Y., 2004: "Design Optimization of Hierarchically Decomposed Multilevel System under Uncertainty", Proceedings of the ASME 2004 Design Engineering Technical Conferences, Salt Lake City, Utah, September 28 - October 2, 2004, DETC2004/DAC-57357.
3. Hartmanis, Juris, Stearns, Richard, 1965: "On the computational complexity of algorithms", Trans. Amer. Math. Soc. 117 (1965), 285-306.
4. Youn, B.D. and Choi, K.K., 2004: "Selecting Probabilistic Approaches for Reliability-Based Design Optimization," AIAA Journal, Vol. 42, No. 1, 2004, pp. 124-131.
5. Choi, K.K. and Youn, B.D., 2001: "Hybrid Analysis Method for Reliability-Based Design Optimization," 27th ASME Design Automation Conference, September 9-12, 2001, Pittsburgh, PA.
6. Socie, Darrell F., 2005: Probabilistic Aspects of Fatigue, *http://www.fatiguecalculator.com/pdfs/ProbabilisticFatigue.pdf#search=%22SAE%20950X%20deviation%22*
7. Youn, B. D., Choi, K. K., and Tang, J., "Structural Durability Design Optimization and Its Reliability Assessment," *International Journal of Product Development*, Vol. 1, Nos. 3/4, 2005, pp. 383-401.
8. Choi, K. K., Tang, J., Hardee, E., and Youn, B. D., "Application of Reliability-Based Design Optimization to Durability of Military Vehicles," *2005 SAE Transactions Journal of Commercial Vehicles*, 2005-01-0530.

CONTACT

Prof. KK Choi can be reached at kkchoi@engineering.uiowa.edu.

ADDITIONAL SOURCES

DEFINITIONS, ACRONYMS, ABBREVIATIONS

HPC: High Performance Computing

FEA: Finite Element Analysis

RBDO: Reliability Based Design Optimization

TARDEC: Tank-automotive Research, Development and Engineering Center in Warren, MI. Part of RDECOM

RDECOM: U.S. Army Research, Development and Engineering Command