

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.				
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.				
1. REPORT DATE (DD-MM-YYYY) 5/3/2012		2. REPORT TYPE Final Technical		3. DATES COVERED (From - To) 01-Mar-07 - 30-Sep-11
4. TITLE AND SUBTITLE Infusion of Gaming Paradigm Into Computer-aided Engineering Design Tools			5a. CONTRACT NUMBER	
			5b. GRANT NUMBER N00014-07-1-0686	
			5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Roger A. Dougal			5d. PROJECT NUMBER	
			5e. TASK NUMBER	
			5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Sponsored Awards Management University of South Carolina 901 Sumter Street, Columbia, SC 29208			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Research ONR Code 331 875 North Randolph Street Arlington, VA, 22203-1995			10. SPONSOR/MONITOR'S ACRONYM(S) ONR	
			11. SPONSORING/MONITORING AGENCY REPORT NUMBER	
12. DISTRIBUTION AVAILABILITY STATEMENT Approved for Public Release; distribution is Unlimited				
13. SUPPLEMENTARY NOTES				
14. ABSTRACT This project introduced into the VTB interdisciplinary system design/simulation environment several gaming concepts such as network-centric multiplayer interactions, detection of differences between independent client-side simulation models, and simplified human interfaces to physics-based models. It also developed new computing methods that facilitate exploitation of the gaming environment for design purposes. Technical progress can be separated into two broad categories; realization of a hybrid physics-based gaming environment as a surrogate for a highly-interactive design environment, and improvement of simulation speed so that the physics-based gaming environment can operate fast enough for effective human interaction.				
15. SUBJECT TERMS Computer-aided design, gaming				
16. SECURITY CLASSIFICATION OF: a. REPORT b. ABSTRACT c. THIS PAGE		17. LIMITATION OF ABSTRACT		18. NUMBER OF PAGES
19a. NAME OF RESPONSIBLE PERSON			19b. TELEPHONE NUMBER (Include area code)	

20120509077

FINAL REPORT

PROJECT N00014-07-1-0686

Attn: Terry Ericson
Office of Naval Research
ONR Code 331
875 North Randolph St,
Arlington, VA 22203-1995

**INFUSION OF A GAMING PARADIGM INTO COMPUTER-AIDED
ENGINEERING DESIGN TOOLS**

3 May 2012

Submitted by:
Dr. Roger A. Dougal, PI

with contributions by
Dr. Jijun Tang
Dr. Jian Shi
JJ Shepherd
Tianlei Zhang
Yiwei Zhang

Technical POC

Dr. Roger Dougal
Dept. of Electrical Engineering
University of South Carolina
Columbia, SC 29208
Telephone: (803) 777-7890
Fax: (803) 777-8045
email: dougal@engr.sc.edu

Administrative POC

Danielle McElwain
Sponsored Awards Management
901 Sumter Street
Columbia, SC 29208
Telephone: (803) 777-1119
Fax: (803) 777-4136
email: dmcelwai@mailbox.sc.edu

EXECUTIVE SUMMARY

This project introduced into the VTB interdisciplinary system design/simulation environment several gaming concepts such as network-centric multiplayer interactions, detection of interferences between independent client-side simulation models, and simplified human interfaces to physics-based models. It also developed new computing methods that facilitate exploitation of the gaming environment for design purposes.

Technical progress can be separated into two broad categories; realization of a hybrid physics-based gaming environment as a surrogate for a highly-interactive design environment, and improvement of simulation speed so that the physics-based gaming environment can operate fast enough for effective human interaction. The multiplayer gaming environment developed here couples a multi-disciplinary engineering simulation software (VTB) with a gaming engine to produce an environment that allows participants to define vehicles from component parts, operate the virtual vehicles in realistic environments, and race them against each other in a multiplayer gaming environment. The computational intensiveness of the detailed simulation models would slow game play, so we also developed algorithms to partition large simulation models into several parts that could be processed on multiple cores of a CPU. We also demonstrated that graphics processing units (GPUs) can be used to execute system models at speeds significantly greater than can be accomplished with off-the-shelf CPUs.

INTRODUCTION/OBJECTIVES

Young people are fascinated with electronic gaming environments. Many of these environments now incorporate basic physics engines for the purpose of animating game objects and characters with realistic-looking motions. Some of these environments (e.g. Delta Force Land Warrior) have been released both as popular games and as training tools for the armed forces. In fact, physics-based simulations have a long history of success in training; they are so successful that no commercial or combat pilot ever begins service without first spending hours in a flight simulator to develop the reflexive skills that are essential to a long and safe career. We thus see comparable opportunities to improve several specific engineering design tools, to develop a person's design skills, and ultimately to improve the future supply of engineers (ship designers in particular) by capitalizing on advanced gaming concepts.

A main objective of this work was to introduce the concepts of massively multiplayer online gaming environments, MMOGs, to the process of engineering design. In MMOGs, many participants interact with each other through a distributed computing infrastructure, which we believe represents the next frontier in design tools for large, complex, multidisciplinary systems. While requirements, work decomposition and allocation, interface standards, and design reviews will continue to be critical elements of the process, one cannot deny the immense promise held by new processes that enable very rapid iteration of the design cycle across traditional disciplinary lines. Hence, in this work, we sought to implement and investigate the effectiveness of engineering analogs of the massively multiplayer gaming environment that support these design processes.

TECHNICAL ACHIEVEMENTS

Technical progress was achieved in two broad categories; 1) realization of a hybrid physics-based gaming environment, and 2) improvement of simulation speed so that the physics-based gaming environment can operate fast enough for effective human interaction. Specific accomplishments include:

- o Physics based gaming environment
 - Prototyping of a gaming environment for system design and analysis, following an auto-racing theme
 - Development of strategies for real time path-finding subject to motion constraints
 - Development of methods for visualization and testing of physical interferences
 - Development of methods for optimizing system (vehicle) design, considering controls
- o Improving simulation speed
 - Method for automatically dividing system models for parallel execution using Latency Insertion
 - Use of GPUs for high speed parallel computing

Each of these are briefly summarized in this report, with references to more detailed information that is available in publications.

Physics based gaming environment

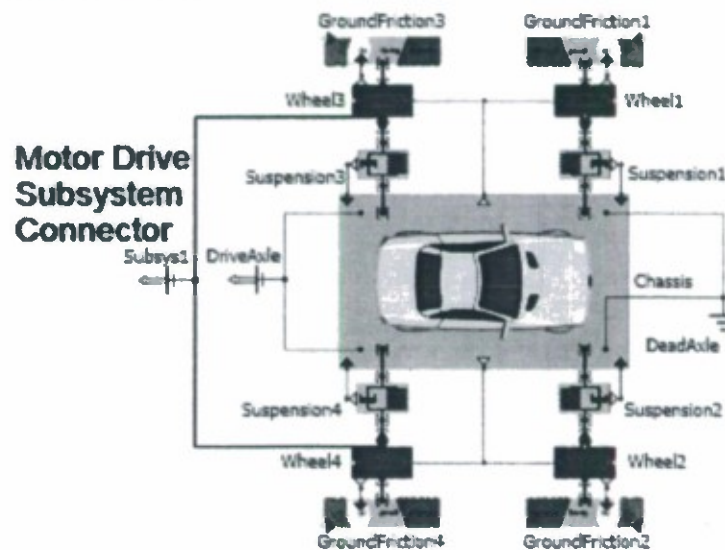
Our initial study showed how multi-player online gaming environments using client-server technology could be used to enhance engineering collaboration while executing a simulation-based design process. To this end, we developed an automobile racing game that permitted simultaneous users running client-side simulations to interact with each other while racing self-defined vehicles (power trains, suspensions, power sources) across various terrains. Of course this type of interaction is not limited to such an automobile racing game, and represents interactions that must occur during a complex system design process. Using client-server interactions between multiple simulations can open up new types of collaborations between engineers, as well as the whole process for designing large systems. During this work, we identified interference (collision) detection as an important representative process in multi-user environments wherein physics-based models run independently on each user's computer, yet collisions between the independently-simulated vehicles must be detected and managed globally. We found that it was most effective to implement collision detection on the server side, rather than the client side, to minimize the number and speed of data communications among all players. The system had the ability to detect collisions between the bodies of different vehicles, or between the tires of a vehicle and the terrain. This latter effect was used to apply appropriate traction forces.

The multiplayer gaming environment was a hybrid of the multi-disciplinary engineering simulation software, Virtual Test Bed (VTB), and the gaming tool, Unity3D. This hybrid gaming environment coupled a three-dimensional (3D) multibody vehicle system model - defined as a composition of VTB model objects, each at a proper level of complexity for system analysis - with actual road information exported from Google Earth to the 3D visual front-end fabricated around Unity3D.

The hybrid environment was sufficiently developed to support analyses of the energy use of vehicles. Several types of powertrains with energy storage systems (ESS) could be defined, using standard models in VTB. Novice users could choose among partially-pre-configured chassis assemblies while more advanced users could construct one-off designs by pulling models from a library of components that covered various technical disciplines – mechanical linkages (suspension systems, gears, tires), electric power (batteries, power converters, motors), control systems (battery charge control, motor control). The vehicle system model so constructed could then be “driven” interactively using a gaming controller (rather than controlled from a predefined script, which is more common in the engineering simulation world). This permitted the gaming environment to be used in diverse ways -- for example, to study the performance of an energy storage system as it operated through realistic and complicated driving scenarios, or to help an individual vehicle owner plan where and when to recharge an electric vehicle during a trip, given the distances and terrains to be covered, and the locations of recharging stations. This is a good analog to the process of evaluating suitability of a conceptual ship design for some mission.

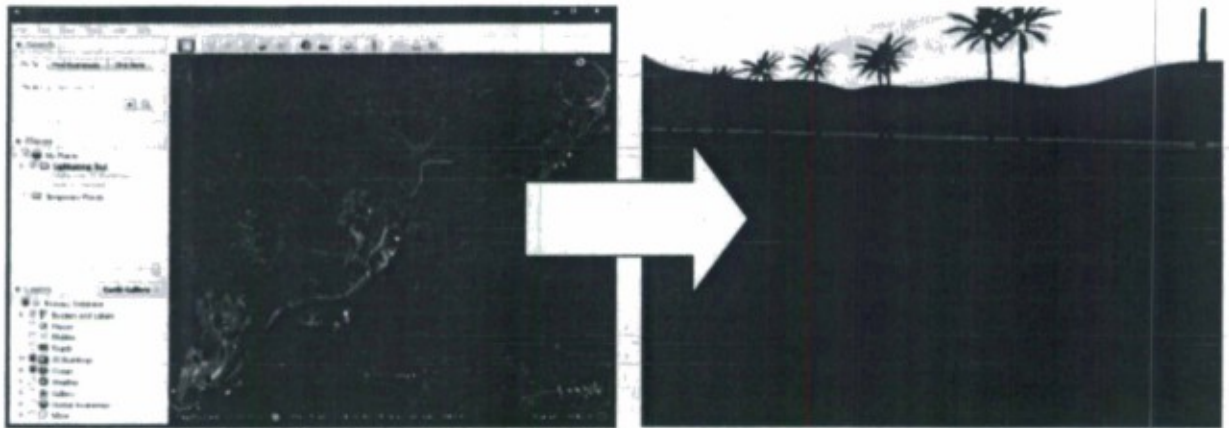
A vehicle model that simulates the 3D kinetics and dynamics, as composed in the VTB environment, is shown in the figure below. The components are connected via natural (acausal) coupling ports which enforce energy conservation laws. Such ports are indicated in the figure by red circular icons, for the case of a 1-dimensional power interaction. A square bracket around a red circle indicates an array of such 1-D power interactions and thus can represent, for example, the power interactions in 3 dimensions.

We demonstrated the environment by constructing a battery-powered vehicle having parameters similar to those of a Chevy Volt. The control parameters of electronic power converters were tuned appropriately and the operating quadrants of the converters were controlled by the throttle control signals from the gaming controller to enable full electrical control of acceleration and deceleration.



The VTB simulation of the vehicle dynamics ran concurrently with and interacted with the gaming engine, Unity3D which provided the visualization environment. The gaming space communicated the vehicle control signal from the player's gaming controller to the vehicle system model as simulated in VTB. The terrain of the gaming environment, including altitude, was exported from Google Earth into Unity3D. Unity3D then passed appropriate terrain slope

information to the VTB simulation model so that tractive forces were appropriately computed. The figure below shows the terrain information in Google Earth and the representation of the vehicle on the road in the gaming display.



Additional details of this work are given in [5], and [9].

Real time path-finding subject to motion constraints

In a collaborative engineering design tool or in a video game, one person may wish to interact with “the system” instead of another individual. For instance, a game player may wish to race a car against the computer. Or an engineer may wish to evaluate a design concept in the absence of her colleague who should be working the collaborative design tool contemporaneously in a different domain. In these cases, it is often necessary to move objects along dynamically-determined paths, but subject to physical constraints. Most games use tricks to avoid rigorous enforcement of physical and path constraints in order to avoid computationally intensive solutions. In this task we developed a new algorithm for quickly determining paths that are subject to differential constraints. The method uses Dubins curves to precalculate several path estimates, then stores these estimates in a data structure until they are needed. This method facilitates single-player action in the vehicle racing game, and also can be applied during evaluation of mission-oriented system models.

Details of this work were presented at 2009 Computer Games, Multimedia and Allied Technology [8, 10].

Methods for visualization, including solid-body motion

Video game engines provide many tools for visualization. Some game engines offer functionality that can be harnessed in the context of engineering simulations. In this effort, we investigated whether Microsoft XNA can expedite visualization of nontrivial engineering constructs. We first created simple visualization objects to become familiar with the environment, and then incrementally increased the difficulty of the simulation/visualization problems.

In the process of vetting XNA we investigated two critical areas -- the maturity of the framework from the software developer’s point of view, and the overall performance of the environment with respect to rendering complex geometries in real-time. Much like other game engines, XNA provides a framework to quickly develop visual applications, and it also provides some foundation for supporting physics-based simulations. We found that simple Newtonian

physics such as gravitational affects, collision detection, and basic object movement could be computed quite easily within the visualization environment. In order to determine if the FPS (frames per second) rate was adequate we exercised the XNA visualization engine by rendering data that was generated from an off-line simulation. The off-line simulation was a typical escape and evade mission where two agents attempt to intercept one agent that is escaping. The simulation data was recorded in a data file and then loaded and executed by the visualization tool. For our purposes, a frame rate of anything greater than 30 FPS was deemed sufficient to create the illusion of continuity to the user. The result of this experiment was the creation of an environment that allowed the user to interact with and modify the viewing perspective while the data was played back in real-time as defined above.

As part of an engineering design tool, a gaming visualization environment needs to not only render data-based views efficiently and quickly, but it must also support fast data interactions with the simulation tool that generates the data. All of these processes must occur at faster than real-time rates -- the simulation tool must provide results faster than real-time, the visualization tool needs to render the results faster than real-time, and the communication between the tools needs to be fast and efficient. A test scenario was developed to exercise the various pieces. The test scenario permitted the user to inject changes into a simulation via direct manipulation of objects in the 3D environment. In this test scenario a user could drive a simulated vehicle on a race track and, at will, modify the radius and tread width of a tire. This information was propagated to the simulation engine which would make the necessary modifications to parameters of the appropriate models. The simulation would then compute the next position and orientation of the vehicle in 3D space. This data would then be sent to the visualization engine which would render the new view to the user. Several iterations of the software were required in order to produce a communications library that could relay the data between both tools quickly enough. Ultimately the end product was built using socket communications; this was the most reliable and had the least overhead. A generalized solution to the problem was difficult to develop because of the complexity of the visualization, the complexity of the simulation, and the amount of data that needed to be exchanged between tools. Nonetheless, we successfully demonstrated that our implementation of the visualization and simulation environment could render the results to the user in real-time.

A further experiment focused on whether it was possible to use a physics engine plug-in to XNA (the *JigLib* library) as part of a larger physics simulation engine, in a co-simulation process with VTB. This provided a convenient means to account for solid-body interactions such as ball-to-ball contact forces, gravitational effects, touch forces, collisions, and inertially-governed movement in three spatial dimensions. A test scenario was developed which involved the interaction of spherical objects piled into an inverted cone, with one of the buried spheres being lifted by a force computed within the VTB engine, while opposing forces generated by the gravitational pull on the overburden of spheres were computed by the *JigLib* engine within the visualization environment. The experiment was successful, showing that the balance of physical forces could be computed in the two separate environments. When a sufficient lifting force was applied within VTB, the ball was lifted through the overburden of balls and freed from the container. Movement of the one ball caused additional interactions as positions of other balls in the container shifted, and these were realistically rendered in the XNA visualization environment.

Details of this work were presented at 2010 Grand Challenges in Modeling Simulation [6].

System Design Optimization considering Control Strategies

Different control strategies generally produce different optimum physical configurations of systems, yet the tools for optimal design of systems generally do not consider the effect of control strategies on the optimal design outcome. Because of this we investigated how a design engineer working in a game-like environment can optimize a system design by considering control strategies. The method involves use of high-level simulation models to abstract the designer from the discipline-specific nuances of individual component behaviors. The first simple study applied the process to design of just an electronic power converter, as preparation for a more sophisticated study applying the technique to the design of the energy storage system for an electric vehicle.

In a first study, we defined a cost function for a buck converter as a combination of hardware cost and the system performance cost, and used these costs to obtain an optimum design point. We compared the realized optimal design points that were obtained by following two different processes: In the first, process (the traditional way) we determined the optimal hardware design by finding the local minimum point of the hardware cost, and then we designed a control strategy to achieve the lowest performance cost. In the second process, (the new way) we identified both the hardware design and the control strategy simultaneously in order to find the global minimum system cost. The results were enlightening, and can be generalized into a preferred design process. Process 1 always found the local minimum of hardware cost because it only considered meeting the minimum design criteria under investment budget. This produced a hardware configuration having cost 14% lower than the hardware obtained by Process 2, but the system performance cost was 16% higher for the system obtained following process 1. Process 2 improved the overall design in a holistic way by partially compromising on hardware costs to improving performance cost. These results show the importance of considering control strategies during the hardware design.

Future work will demonstrate the co-optimization method applied to energy storage design for an electric vehicle, using the VTB/Gaming system previously developed. Different driving route will be used to represent the mission requirements for the system. Different control strategies will be substituted in the power source subsystem. Considering the battery SOC and velocity values, the performance of different system structures will be compared for the same driving route; or a selected control strategy will be evaluated for different driving routes.

Details of this work were presented at 2011 Grand Challenges in Modeling Simulation [2].

Improving simulation speed

Automatic division of system models for parallel execution using latency methods

The Latency Insertion Method (LIM) can be applied to divide a large system model into a multiplicity of smaller models to reduce simulation execution time. We developed an algorithm for optimal placement of LIM connectors into a large system model so that the large model can be partitioned into many smaller models to achieve maximum speed up. Our new method automatically exploits actual latencies in a large network model rather than inserting synthetic latencies. First we employ a method that automatically checks the system to see where LIM can be applied to partition the system based on a Breadth-First-Search algorithm. This method transverses the whole system and checks all paths of connectivity between adjacent nodes. If two nodes are connected only by components that have LIM equivalents, then a LIM connector is

applied to substitute the original component and these two nodes now belong to different sub-systems, linked through the LIM algorithm. Otherwise they are grouped together. Then we introduced a method to estimate the speed up that results from each possible partitioning and to choose the partition that offers the maximum speed up. We applied this method in the VTB (Virtual Test Bed) environment for the simulation of several large-sized systems containing both linear and non-linear component models. Results showed that for reference system models having structures similar to electric power distribution systems, simulation speed was increased by a factor of more than three while keeping the simulation error within 1%.

The results of this work were documented in [1, 3, 7].

Use of GPUs for high speed parallel computing

Graphics Processing Units can provide significant computing power at low cost, so we explored the application of these devices for solving the types of equations typically encountered in dynamic system simulators. To best understand where speed ups could be realized, we performed a detailed profiling of the time required to execute each sub-process within a system simulation in terms of the system model characteristics and the data characteristics of the system conductance matrices. According to the profiling process, we identified three methods to improve simulation performance: 1) high performance algorithm engineering, 2) develop a new linear equation solving method, and 3) develop a well-designed parallelism model.

Through the principle of cache-aware programming, we realized about 6x speedup on serial level without substantial algorithmic change and parallelization. Compared to commercial matrix solvers that run on a CPU, we realized speedups ranging from 5 (for system size ≈ 700) to 460 (for system size ≈ 5800). While calculation time for a commercial matrix solver increased with matrix size at the rate $\approx O(N)^{2.3}$, our new GPU-based Preconditioned Generalized Minimal Residual (PGMRES) technique yielded scaling at the rate $\approx O(N)^{1.2}$. A significant component of this performance was achieved by development of new Basic Linear Algebra routines for the NVIDIA Tesla GPU that directly address characteristics typical of matrices that describe the time domain response of naturally-coupled dynamic systems.

The results of this work were documented in [4, 10].

Publications:

Journal:

1. Optimal Activation of Latency Decouplers to Partition Large Network Models, Jijun Tang, Roger A. Dougal, R. Leonard, Y. Zhang, Journal of Computational Science (under review)

Conference:

2. The Simulation Tool for Mission-Optimized System Design, J.J. Shepherd, J. Tang, R.A. Dougal, T. Zhang, Grand Challenges in Modeling & Simulation 2011 (GCMS 2011), the Hague, Netherlands, June 2011.
3. Optimally applying latency insertion method in large system models, Jijun Tang, R. Leonard, Y. Zhang, R. Dougal, Grand Challenges in Modeling & Simulation 2011 (GCMS 2011), the Hague, Netherlands, June 2011.
4. New Basic Linear Algebra Methods for Simulation on GPUs, J. Tang, R.A. Dougal, B. Langland, Y. Zhang, J. Shi, Grand Challenges in Modeling & Simulation 11, The Hague, June 2011
5. A gaming environment approach to analysis of energy storage for electric/hybrid vehicle, T. Zhang, J.J. Shepherd, J. Tang and R. A. Dougal, International Conference on Clean Electrical Power 2011 (ICCEP 2011), Ischia, Italy, June 2011
6. The Simulation and Data Visualization Potential of Microsoft's XNA, J.J. Shepherd, R.A. Dougal, J. Tang, 2010 Grand Challenges in Modeling Simulation, pp. 306-310, Ottawa, Canada, July 2010.
7. Method for Partitioning Large System Models when using Latency Insertion Method to Speed Network Solution, Y. Zhang, R.A. Dougal, J. Tang, B. Langland, J. Shi, Grand Challenges in Modeling and Simulation 09, pp. 259-263, Istanbul, Turkey, July 2009.
8. Path-Finding with Motion Constraints in Real Time Strategies, J.J. Shepherd, R.A. Dougal, J. Tang, Computer Games, Multimedia and Allied Technology 09 Conference, pp. 83-88, Amara Singapore, May 2009.
9. Application of a Multiplayer Computer Gaming Paradigm to Engineering Design Tools. JJ Shepherd, J. Shi, T. Zhang, J. Tang, R.A. Dougal, Grand Challenges in Modeling and Simulation 08, pp. 383-388, Edinburgh, Scotland, June 2008.

Thesis/dissertation:

Master:

10. Path-Finding with Differential Constraints Amongst Obstacles in Real Time Strategy Games, J. Shepherd, Master's Thesis, University of South Carolina, December 2009.

Phd:

11. Improving simulation performance with GPUs, J. Shi, Dissertation, University of South Carolina, August 2011.
12. Graph based Optimization for Scientific Computing, Y. Zhang, University of South Carolina, August 2012 (ANTICIPATED)