

INTEGRATING OPERATIONS SIMULATION RESULTS WITH AN IMMERSIVE VIRTUAL REALITY ENVIRONMENT

Gordon D. Rehn
Marco Lemessi

Industrial Engineering, Simulation Group
Deere & Company
One John Deere Place
Moline, IL 61265, U.S.A.

Judy M. Vance
Denis V. Dorozhkin

Department of Mechanical Engineering
Iowa State University
Virtual Reality Applications Center
Ames, IA 50011, U.S.A.

ABSTRACT

This paper demonstrates the utilization of immersive virtual reality environments in the investigation of operations simulation results. The authors outline the benefits offered by virtual reality over the traditional two-dimensional computer interfaces, such as the monitor, keyboard and mouse. The paper describes the state of the art in the area of operations simulation and the implementation details and functionality of the program developed as a result of this research. The experience of using the developed application for the analysis of a manufacturing operation is presented.

1 INTRODUCTION

The use of simulation models for the analysis of manufacturing operations is growing worldwide. However, despite the growth of simulation and the growing awareness of simulation's value, few companies are fully utilizing simulation for management considerations.

Simulation is the only tool that can determine the impact of one system component on another, and, as a result, point out problems associated with the interaction of complex components, which would otherwise go undetected until capital was invested, or great operating expense incurred.

Most simulation models are linked to animation tools, either concurrent or post-processed. Animation tools permit the analysis and evaluation of system performance and simulation results in either a two- or a three-dimensional environment. Those tools, however, rarely allow users to interact with the objects being simulated, and those that do, only permit very limited interactions.

The possibility to directly link operations simulation results to an immersive virtual reality (VR) environment opens exciting avenues for complex interactions between

model users and the objects being simulated. In particular, a simulation-driven immersive virtual environment will:

- Pave new avenues for concurrent product and process design;
- Set a first step towards creating a virtual assembly training process and a virtual manufacturing laboratory;
- Increase, in the long term, the 'speed to market' of new products.

2 STATE OF THE ART

Virtual reality is defined as a computer-generated three-dimensional environment created by virtual environment (VE) systems, which can be interactively experienced and manipulated by the participants (Barfield and Furness 1995). According to Stuart (2001), a VE system is a human-computer interface capable of providing "interactive immersive multisensory 3-D synthetic environments." In these systems position sensors are used to track the user's motions and to update the visual and auditory displays in real-time, creating the illusion for the participants of being inside of the environment.

Several simulation packages that claim to utilize the VR approach to operations simulation currently exist (Whitman et al. 2002; Kibria and McLean 2002). However, the majority of these programs rely on the traditional two-dimensional (2D) computer interfaces, such as the monitor, keyboard and mouse, to view the 3D models of the simulated environment. Interaction with the environment and the data-interrogation methods remain essentially unchanged from the standard interaction with 2D schematic representation of the simulated manufacturing operations.

One of the few research projects aimed at the investigation of the benefits of a VR environment in the context

of operations simulation was undertaken by Kesavadas and Ernzer (1999) at the University of Buffalo. They have developed the VR-Fact program – a virtual environment for modeling and designing factories and shop floors. VR-Fact was created for quick implementation of factory design algorithms ranging from plant layout to factory flow analysis. However, the program's features were primarily focused on the design of shop floor arrangements using the Cellular Manufacturing (CM) System method. Furthermore, it could only be used with a head-mounted display or stereo glasses with a computer monitor.

These issues were addressed in the research work done by Kelsick and Vance (Kelsick et al. 2003) at Iowa State University. Instead of incorporating the discrete event simulation logic within the virtual environment application, they chose to develop an interface to existing simulation software. The developed program, VRFactory, combined results from a commercial discrete event simulation program, SLAM II, with a virtual environment, implemented in a projection-based VR system with multiple screens. Three-dimensional computer models were used to allow investigation of how various changes to the manufacturing cell affect part production. This was performed while the user was immersed in a computer-generated stereoscopic representation of the cell.

Operation of the VRFactory proved that immersion in the virtual factory facilitated the exploration of design changes and their effect on the simulation. The users were also successful at investigation of the operations simulation results and communication of the implications associated with the design changes to participants who did not have extensive theoretical simulation background. However, the VRFactory program suffered from the fact that throughout the development process it was tailored to a single manufacturing operation. As a result, it lacked the ability to perform analysis of any other simulation scenarios of interest.

One of the main goals of the project described in this paper was the development of a general-purpose VR application, capable of the analysis of arbitrary manufacturing scenarios. The application users were to have the ability to easily modify the parameters associated with the simulation, including the geometric layout of the assembly line and the geometry models that describe the assembly environment and the assembly components, and to alter the simulation results dataset. The interaction and data interrogation methods were to utilize the unique capabilities of the immersive virtual reality systems to the maximum extent. The program developed as the result of this project complies with these requirements.

3 OPERATIONS SIMULATION

Despite the fact that operations simulation has existed since the early 1960s, industry, in general, has yet to take full advantage of the potential of simulation analysis. In most companies, simulation analysis is relegated to ob-

scure Operation Research projects, with very few companies utilizing simulation in their ongoing manufacturing process planning and/or operation.

Reasons for this lack of simulation use in industry include difficulties in:

- timely development of simulation models;
- inability to access real and current production data;
- lack of cost effective means to deploy useful simulation models.

In the early 2000s, Deere & Company gave birth to ALiSS, Assembly Line Solution Set, in order to answer the high-demanding requests from Deere factories for detailed simulation models of their assembly lines. ALiSS was designed to allow production-support personnel to use simulation applications in their daily planning and operating responsibilities without the burden of building a model.

ALiSS is an integrated software package custom designed for Deere by Deere that links an Excel User Interface with a discrete event simulation code and animation software. Through a series of Excel worksheets users specify their scenarios to ALiSS, execute the simulation and animation, and import simulation results.

The objectives behind the development of ALiSS were:

- Reduction in model development time;
- Extension of simulation resources by enabling non-simulationists to perform analysis;
- Transfer of simulation technology from design phases into actual operation;
- Consistency in the design and operation of assembly lines.

Ideal applications of ALiSS have a dual purpose. ALiSS is used in the design of assembly lines accounting for detailed work measurements, line balance, model and option mix, staff levels and deployment, assembly line configurations and equipment, and hours of production. Once the line is designed, the same application can be used in the actual production to identify future production problems and diagnose solutions.

Since the creation of ALiSS, over 80 assembly lines have been modeled throughout the corporation, in North America, Latin America, Asia and Europe. Over 100 engineers, supervisors, or wage employees have been trained in the use of ALiSS. From a Deere & Company perspective, ALiSS represents an approximate 10:1 increase in productivity in development type. From a unit perspective, ALiSS has provided feedback, education, and insight into the issues surrounding a good line design.

ALiSS incorporates two commercial simulation software packages from Wolverine Software Corporation: SLX and Proof Animation.

SLX (Henriksen 2000a) stands for Simulation Language with eXtensibility, and is a classical simulation stand-alone tool that includes a programming language with a C-like syntax. SLX's ultra-fast compiler translates models at nearly 90,000 lines per second, probably making it the fastest simulation language on the market. Being based on SLX, ALiSS outperforms any other simulation package on the market as far as execution speed is concerned. Moreover, the speed of SLX opens the possibility of future concurrent simulation/animation execution.

Proof Animation (Henriksen 2000b) is a standalone simulation animation package. Since the display of moving objects in Proof Animation is proportional to time, the movement in a Proof animation is very smooth and continuous, unlike other animation packages that resort to a 'paint-repaint' method resulting in a motion that has incremental jumps in object positioning (see Figure 1).

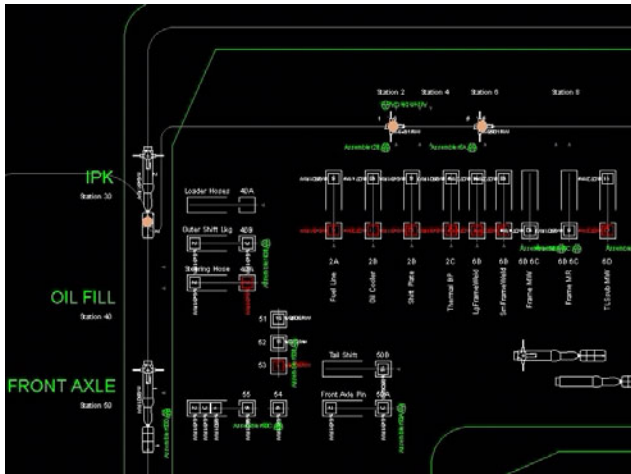


Figure 1: Sample of a Proof Animation

Furthermore, most commercial packages merge the simulation engine with the animation engine. This means the simulation must run for the animation to be seen (or the animation can be turned off entirely). The overhead of running the simulation engine (which are much slower than SLX already) in order to animate further slows the animation display that exasperates the discontinuous paint and repaint motion. In ALiSS the simulation and animation functions are separate and can be run independent of one another.

Despite its smooth animation and independency from any simulation engine, Proof Animation's geometric limitations became soon evident to Deere & Company Simulation Group. Such limitations include:

- Two-dimensional, schematic layouts;
- Simple object geometry;
- Limited object interrogation, mainly related to object ID, position and rotation, and speed;

- No direct human-model interaction, apart for pausing, zooming, changing viewing speed, and interrogating objects.

As a consequence, Deere & Company and Iowa State University started exploring methods to link ALiSS simulations to an immersive virtual reality environment, as an alternative but not a substitute to two-dimensional Proof animations, as well as a preliminary but necessary step towards a future virtual laboratory for assemblers' training. The first phase of the research aimed at creating a flexible but post-processed linkage between the outputs of an ALiSS simulation and the immersive virtual reality environment.

To this aim, a brand new SLX module was developed and integrated into the ALiSS existing code. By exploiting SLX's unique extensibility features (Henriksen 2000a), the new module is able to extract relevant information from ALiSS, and pass it to an output standalone ASCII file (later referenced to as VRF file). The new module does not interfere with the ALiSS simulation itself, and can easily be deactivated if the specific ALiSS simulation does not require the collection of relevant information for a virtual reality visualization.

Relevant information is extracted by interrogating ALiSS variables and main active objects (namely, parts assembled, tasks performed, and operators working) during any simulation run. Such information includes: when each operator arrives and leaves a given working location, or starts and ends a given task, when each part arrives and leaves a given station, or is loaded on or unloaded from a given material handling system, what is the status (e.g., busy, idle) of a given operator, a given part, or a given material handling system, etc.

A procedure has been specifically developed to allow users to choose whether to pass to the output ASCII file all the information relative to a single ALiSS run, or to limit it to a selected area of the simulated assembly line. Such a procedure reduces memory requirements and improves system performances in the VR system when the VR visualization is not extended to the whole assembly line.

4 THE IMMERSIVE VR ENVIRONMENT

The virtual reality simulation program is written in the C++ programming language. Creation of the computer graphics objects is achieved with the SGI OpenGL Performer™, a software development environment that supports implementation of high performance graphics applications and is built atop the industry standard OpenGL® graphics library (Woo et al., 1999). The immersive VR environment is managed with the VRJuggler virtual reality software library (Bierbaum et al. 2001). The application was designed to be used in the rear-projection four- and six-screen fully-immersive VR systems at John Deere and Iowa State University. It takes advantage of the hardware

configurations of those systems, including the wireless wands and tracking systems, for interaction with the generated virtual environment.

The developed program is comprised of four primary components:

- Graphics module, which is used for generation and display of the three-dimensional objects in the VR environment such as the models of the assembly parts and vehicles and the program interface;
- Data processing module, responsible for interpretation of the simulation results contained in the dedicated ASCII input data files, generated by ALiSS; it also processes the standard Proof Animation layout files, which contain the motion path information;
- Logical module, which determines the behavior of the objects in the environment on the frame-by-frame basis, including motion of the parts and vehicles along the paths, animation of the assembly operations, update of the objects' statuses, etc.;
- Interaction module, which supplies the ability to control the program's functionality with a dedicated VR interface, interrogate the simulation objects, and modify the simulation parameters.

One of the criteria specified for the application was the ability to recreate the assembly environment that is being simulated with the highest level of realism possible. Therefore, actual CAD models of the parts, vehicles and assembly fixtures, provided by John Deere, are used in the VR environment by the graphics module of the program. The models are provided in the Direct Model format and then converted into the Performer binary format before being utilized by the application.

A dedicated model-part association input file is used to identify the particular 3D model that should be used to represent a simulation part or a vehicle in the VR environment. The file also provides the information about the scale of the model and its initial translation and rotation, which could be used to correct for improperly positioned models. This approach ensures that the differences between the physical assembly environment and its virtual representation are minimal. Furthermore, the realistic representation of the assembly workspace makes it well-suited for implementation into the future virtual laboratory for assemblers' training.

The data processing module contains routines for interpretation of the VRF and layout files - the primary simulation data files. The layout file contains the description of the paths that will constrain the motion of the parts and the carrier vehicles during the simulation sequence as well as the definitions of the individual segments comprising each path. Motion characteristics associated with individual paths, such as the time it takes for the part or the vehicle to

complete the path or the initial position of the object on the path are provided in the VRF file.

The VRF file, produced by ALiSS, contains descriptions of all the major events that take place during the simulation time span. Every event or combination of events is preceded by a time stamp, identifying the time elapsed from the beginning of the simulation. During the program's operation, the VRF file is sequentially accessed and each time entry is processed for any events that are to take place. Figure 2 depicts the main execution sequence during the program's operation.

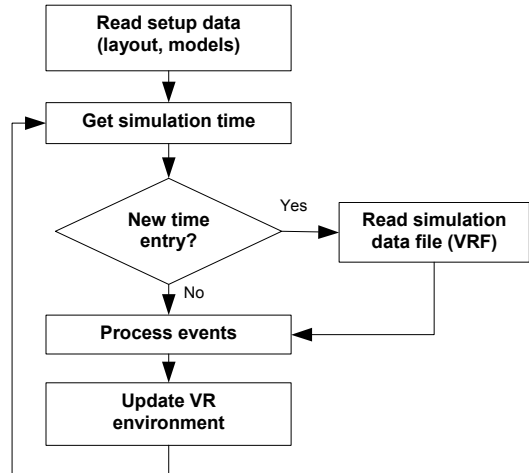


Figure 2: Operational Sequence of the Program

The update of the VR environment refers to the time spent rendering the scene during each frame of the program's operation. This time is closely related to the graphical state of the environment. The realistic geometry models of the assembly parts, assembly vehicles, assemblers, and the assembly lines increase the immersion factor of the application considerably. However, introduction of the additional graphical entities into the environment inevitably leads to a significant increase in the complexity of the scene that is being rendered by the application every frame, resulting in the reduction of performance.

During initial development stages of the application we have experienced frame rates as low as one per second. This problem was especially evident with multiple highly complex models, such as a tractor transaxle assembly, being displayed in the VR environment at the same time.

In order to boost the application's performance to an acceptable level the total number of polygons simultaneously rendered by the system had to be decreased. The intention of this project was to continue using the standard CAD geometry models, provided by the John Deere engineering services. Therefore, an additional set of criteria used by the application to determine what level-of-detail (LOD) configuration of a geometry model should be displayed in the environment corresponding to the viewer's

current position has been developed and implemented. A low resolution model of an object (e.g. carrier vehicle) that has relatively few polygons is normally used. As the distance between the user and the model is reduced, the resolution of the model is gradually increased up to the maximum available level. This approach has resulted in a significant reduction in rendering time and the corresponding improvement of the application's performance.

The amount of time spent rendering each frame is determined precisely using the system clock. That value is then used to compute the current simulation time, which, in turn, is used as the reference value while processing the VRF simulation data file. Before the next frame is drawn, successive time entries and the associated simulation events in the VRF file are processed up to the current simulation time value. The events could include creation and destruction of the parts and vehicles, changes in status of the objects and assemblers, positional placement commands, etc.

The logical module of the application is responsible for keeping the behavior of the simulated assembly objects consistent between the discrete events contained in the VRF file. For instance, it handles the motion of the parts and vehicles along the paths.

The motion of an object in the environment is specified by assigning the object to one of the existing paths, indicating its initial location on the path, and identifying the time it takes for the object to reach the end of the path. From this point in time, the location of the object is updated by the logical module every frame according to the current simulation time, and the object will proceed to the end of the path and remain there until further instructions are provided. This normal procedure is aborted if any additional events associated with the object occur while the object is moving along the path. Such events can include attachment of an assembly part to an assembly vehicle, in which case the motion of the part will be controlled by that of the vehicle, or an encounter of another object on the given path, in which case the motion of all the objects on the path is coordinated in order to keep a certain clearance between the objects.

Among other things, the logical module ensures that the statuses of the assembly objects remain consistent throughout the simulation. For example, the assembly tasks are normally assigned to the assemblers. At that point the program determines which part the assembler is currently working on and updates the part's status accordingly in order to make the assembly task data available in case the part is interrogated by the users of the application.

Furthermore, the logical module coordinates the graphical states of the assembly objects according to the current status of the assembly process. For instance, it changes the color of the assembler models to indicate whether the assembler is busy, waiting for the next task, or away from the assembly station. Similar actions are performed in order to correctly represent the assembly parts and the assembly vehicles in the virtual environment.

A set of virtual menus is one of the components of the interaction module. They provide full control over the application's functionality. The location of a menu is determined from the position and orientation of the interaction device, a wireless wand in this case, so that the menu appears attached to the device. This allows the users to keep the menu system from being obstructed by the objects in the environment. The menu system is toggled and navigated with the wand's buttons.

Control of the simulation time flow is one of the tasks accessible from the menu system. Users have the ability to access any time segment of the simulation and play back the animation at any speed. Time readout in the VR environment is provided with either an analog clock, a standard digital clock, or the elapsed simulation time counter.

Menus can be also used to control the amount of information simultaneously displayed in the virtual environment. For instance, users can toggle on and off all the data labels associated with the assemblers or the assembly parts.

Data interrogation of the individual objects in the environment is also performed with the wireless wand. By depressing the wand's trigger button in the immediate vicinity of an assembly part or an assembler the user toggles the data label containing all the information associated with that object. In case of the assembly part this information includes part type, model type, its unique simulation ID, its status (busy, delayed, or idle), current task performed on the part, and an alphanumeric message that identifies the cause of a delay for the part if such situation occurs.

In order to access areas of the simulated environment located beyond the physical extents of the VR facility, users are provided with the ability to translate in any direction with variable speed. Since some VR facilities lack the rear screen, navigational controls also include the ability to rotate the simulated environment around the current position of the user. This allows the users to investigate areas of the virtual environment otherwise located outside of their visual range.

5 TECHNOLOGY TESTING

A tractor assembly line was chosen to test the linkage between operations simulation results and the immersive virtual reality environment. An ALiSS application of the entire line was developed to provide system information and event timing to the virtual reality environment.

The assembly line consists of 15 work stations plus over 50 subassembly stations. Served by an overhead electrified monorail system of 10 carriers, and an AGV system of 7 vehicles, the line is manned by 28 assemblers working in a single 8-hour shift and flexing between work stations.

As a first step towards virtual reality visualization of the entire assembly line, testing was limited to a restricted area of the line. Such area includes five stations along the main line, plus six subassembly stations, where work is

performed on shift plates, fuel lines and frames prior to their installation.

The focus of the analysis is on a single station, which consists of three floor assembly fixtures. In the first fixture, the tractor main frame is assembled. In the second fixture, the left and right rear frame components are added to the main frame. The third fixture is located directly under the overhead electrified monorail system.

The overall line sequence of events starts by loading an empty monorail carrier with the next transaxle to be built as determined from the current production line-up. Once the main frame assembly is positioned in fixture #3, the corresponding transaxle is lowered onto the frame where the two components are 'mated' to form the chassis. An overhead bridge crane moves the frame assembly between the three fixtures. Once mated, the entire chassis assembly is raised up to the monorail carrier, that travels to the AGV line and queues up and awaits unload by the next available AGV. After unload, the empty monorail carrier returns to the transaxle load area.

The application has been installed at both six-screen and four-screen virtual reality systems, and extensive testing and validation has been performed.



Figure 3: Application Testing in the Six-Screen VR System

6 CONCLUSIONS AND FUTURE WORK

The project long-term vision is to develop a Virtual Reality training environment and laboratory for production assemblers. Potential benefits of achieving this vision would include understanding and applying the relationship between product quality, assembler training, and product optionality.

The first phase of the project successfully linked results from an ALiSS simulation application to an immersive Virtual Reality visualization of a portion of the simulated assembly line. Phase I entailed establishing a communication link from the simulation that translated into executable commands in the VR environment.

The next project phase proposes to take the next logical step towards a fully immersive assembler training laboratory, by effectively inserting dynamic user input as a part of the simulation model.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support provided by D. Brad Keleher, Scott C. Miller, Dan R. Teel and Thomas C. Johnson. Special thanks go to Jerry R. Duncan for his valuable comments on the paper.

REFERENCES

- Barfield, W., and T.A. Furness III. 1995. *Virtual Environments and Advanced Interface Design*, New York, NY, Oxford University Press.
- Bierbaum, A., C. Just, P. Hartling, K. Meinert, A. Baker, and Carolina Cruz-Neira. 2001. "VR Juggler: A Virtual Platform for Virtual Reality Application Development", *Proceedings of the IEEE Virtual Reality 2001 Conference (VR'01)*, Yokohama, Japan, March 2001.
- Henriksen, J. O. 2000a. "SLX: The X is for Extensibility", In *Proceedings of the 2000 Winter Simulation Conference*, ed. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 183–190.
- Henriksen, J. O. 2000b. "Adding Animation to a Simulation Using Proof™", In *Proceedings of the 2000 Winter Simulation Conference*, ed. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 191–196.
- Kelsick, J., J.M. Vance, L. Buhr, C. Moller. 2003. "Discrete Event Simulation Implemented in a Virtual Environment", *ASME Journal of Mechanical Design*, Vol. 125, No. 3, September 2003, 428–433.
- Kesavadas, T., and M. Ernzer. 1999. "Design of Virtual Factory Using Cell Formation Methodologies", *Industrial Virtual Reality: Manufacturing and Design Tool for the Next Millennium, Proceedings of the ASME Symposium on Virtual Reality Environment for Manufacturing*, November 14–19, 1999, Nashville, Tennessee. MH-Vol.5/MED-Vol.9, 201–208.
- Kibria, D., and C. McLean. 2002. "Virtual Reality Simulation of a Mechanical Assembly Production Line", In *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yücesan, C.-H. Chen, J.L. Snowdon, and J.M. Charnes, 1130–1137.
- Stuart, R. 2001. *The Design of Virtual Environments*, Ft. Lee, NJ, Barricade Books.
- Whitman, L., V. Madhavan, D. Malzahn, and J. Twomey. 2002. "Virtual Reality Model to Aid Case Learning", In *Proceedings of the Industrial Engineering Research Conference*, May 19–21, 2002.
- Woo, M., J. Neider, T. Davis, D. Shreiner. 1999. *OpenGL® Programming Guide*, Reading, MA, Addison-Wesley Co.

AUTHOR BIOGRAPHIES

DENIS DOROZHKIN is a Ph.D. student at the Mechanical Engineering Department at Iowa State University and a Graduate Research assistant working in the Virtual Reality Applications Center. His research interests are in virtual reality applications in mechanical design and spatial kinematics. He received his M.S. (2002) in Mechanical Engineering from Iowa State University, and B.S (2000) in Engineering Physics from Loras College, Iowa. His email address is <dorodv@iastate.edu>.

MARCO LEMESSI, Ph.D. is a member of the Simulation Group in the Industrial Engineering Department of Deere & Company, the worldwide corporate headquarters of John Deere. He received his Ph.D. (2002) in Traffic Engineering and M.S. (1998) in Civil Engineering from the University of Rome "La Sapienza", Italy. Before joining Deere & Company in 2002, he has been expert consultant in several EU transportation research projects, and has lectured as expert in the Department of Environmental Engineering of the University of Perugia and in the Transportation Department of the University of Rome "La Sapienza". His email address is <LemessiMarco@JohnDeere.com>.

GORDON D. REHN supervises the Simulation Group in the Industrial Engineering Dept. of Deere & Company. He received his B.S.M.E. from Iowa State University, and is a registered Professional Engineer in the State of Illinois. He has performed discrete event simulation analysis of manufacturing operations since 1976 for internal Deere projects, as well as consulted on simulation projects outside the Deere organization. He has presented at various technical conferences for IIE, ASME, and INFORMS, as well as speaking at academic symposiums sponsored by the University of Iowa, and the Monterrey Institute of Technology in Monterrey, Mexico. He is a member of IIE. His email address is <RehnGordonD@JohnDeere.com>.

JUDY M. VANCE, Ph.D. is Professor and Chair of the Mechanical Engineering Department at Iowa State University and a faculty fellow of the Virtual Reality Applications Center. Her research expertise is in virtual reality applications in mechanical design, numerical methods that facilitate interactive design, and spatial mechanism design. She received her B.S. (1980), M.S. (1987), and Ph.D. (1992) degrees from Iowa State, all in Mechanical Engineering. Dr. Vance spent several years in industry working for both Maytag and John Deere before returning to ISU to pursue her graduate education. She is a member of the ASME Design Division Executive Board and President of the Women in Engineering Leadership Institute. She teaches courses in mechanisms, kinematics, machine design, optimization and computer graphics. Her email address is <jmvance@iastate.edu>.