Enhancing Simulation Studies with 3D Animation

Saurabh Mahapatra, MathWorks

The use of simulation studies to better understand the dynamic behavior of a system under investigation is at the core of verifying designs early in the development process. Despite the amount of data that such studies produce, a 3D representation of the system creates a more complete understanding of system behavior. This article describes the use of 3D animation in simulation-centric workflows to augment early verification activities, such as those used in Model-Based Design. The evolution of technology and domain specialization in the simulation and 3D graphics fields presents several challenges for using 3D animation in simulation-centric studies. A set of examples illustrates how to meet these challenges.

Harvesting Data from Simulations
The last decade has seen the increasing use of computer technology to prototype engineering designs before hardware manufacture or deployment. One area that has generated significant interest is the study of dynamic systems in which nonlinear equations determine the governing dynamics. Providing the means of representing these systems as software programs enables multiple simulations to be run based on various inputs to the system. This methodology presents opportunities for the simulation engineer, such as the ability to vary system parameters or environmental inputs to reflect possible-use scenarios, and to do so exhaustively. Early verification through simulations minimizes the risk inherent in the design process by reducing the probability of discovering errors late in the development process.

Effective early verification requires the careful analysis of simulation data to enable understanding of system behavior. For example, an engineer could analyze data sets to discover patterns of unintended behaviors resulted during multiple runs. The analysis of such data, however, can pose a challenge as it creates the need for domain-specific experts who decipher it.

Towards a Better Understanding of Multidimensional Simulation Data
What makes the understanding of simulation data from a dynamic system difficult?

Typically, it is the interdependencies in the multidimensional data caused by the equations describing the system. In addition to the mathematical analyses that can be done on raw data, visual representations of the data such as 2D and 3D plots for cluster or trend analysis would enrich our understanding of the data relationships due to their placement relative to each other or with respect to a parameter, such as time.

Despite the obvious advantages of using the analytic and visual methods together, this approach would provide some challenges to our understanding in specific situations, especially when the dimensionality of the data is large. Consider the simple physics of an aircraft whose physics modeled a rigid body having six degrees of freedom (6DoF). To analyze the motion of this aircraft, the dimension of the data in time would be six, representing three positional and three rotational coordinates. Besides the obvious limitations in deciphering trends in the raw data, graphical representations such as 2D and 3D plots will be insufficient to improve our understanding of the rigid body motion in this six-dimensional data space. The task becomes harder if we add dimensional complexity to this example by measuring temperature data in time as a function of surface geometry.

Much has been written in 3D graphics literature about a picture being worth a thousand words. Some authors have ambitiously extrapolated that moving images are worth even more. Over the decades, the success of the 3D video game industry and advances in 3D graphics hardware and software lend credibility to this observation. But from a simulation-centric standpoint, data representation using 3D scene representations is promising, if the system under study has interesting interdependencies that can be perceived by the analyst. For example, a 3D animation of multiple agents collaborating to achieve a common or different goal would provide insights in the area of swarm studies. In general, dynamic simulations with their associated physics, such as mechanics, thermodynamics, acoustics or multibody interaction present opportunities for visualization through 3D animation.

A combined strategy that leverages the above approaches can enhance early verification. Figure 1 shows an example of vehicle dynamics simulation in Simulink as a hybrid representation that combines the three approaches. The dimensionality of the data associated with this simulation is 78—mechanical parameters of the 4 wheels (64) each consisting of positions (3), rotations of wheel axis (9), spin (1), and forces (3). The remaining (14) are associated with the braking signal (1), steering input signal (1), car position (3), and car rotation (9).

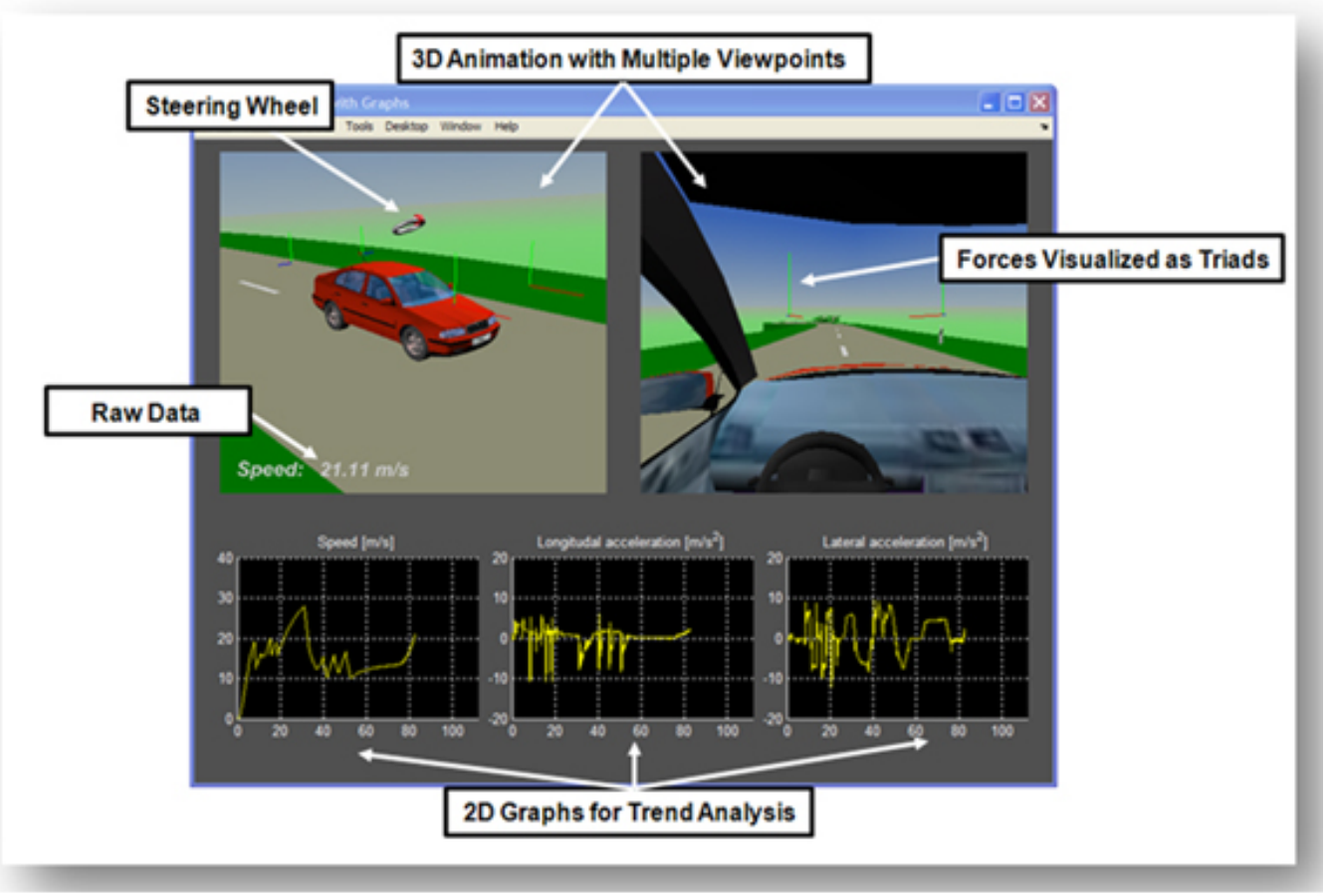

Figure 1: Hybrid representation that combines 1D raw data, 2D graphs, and 3D animation approaches.

Augmenting Early Verification Using 3D Animation Techniques

Figure 2 shows examples of simulation performed in the Simulink modeling environment with Simulink 3D Animation for 3D modeling in Virtual Reality Modeling Language (VRML). Engineers can leverage these 3D animation techniques in the early verification process.

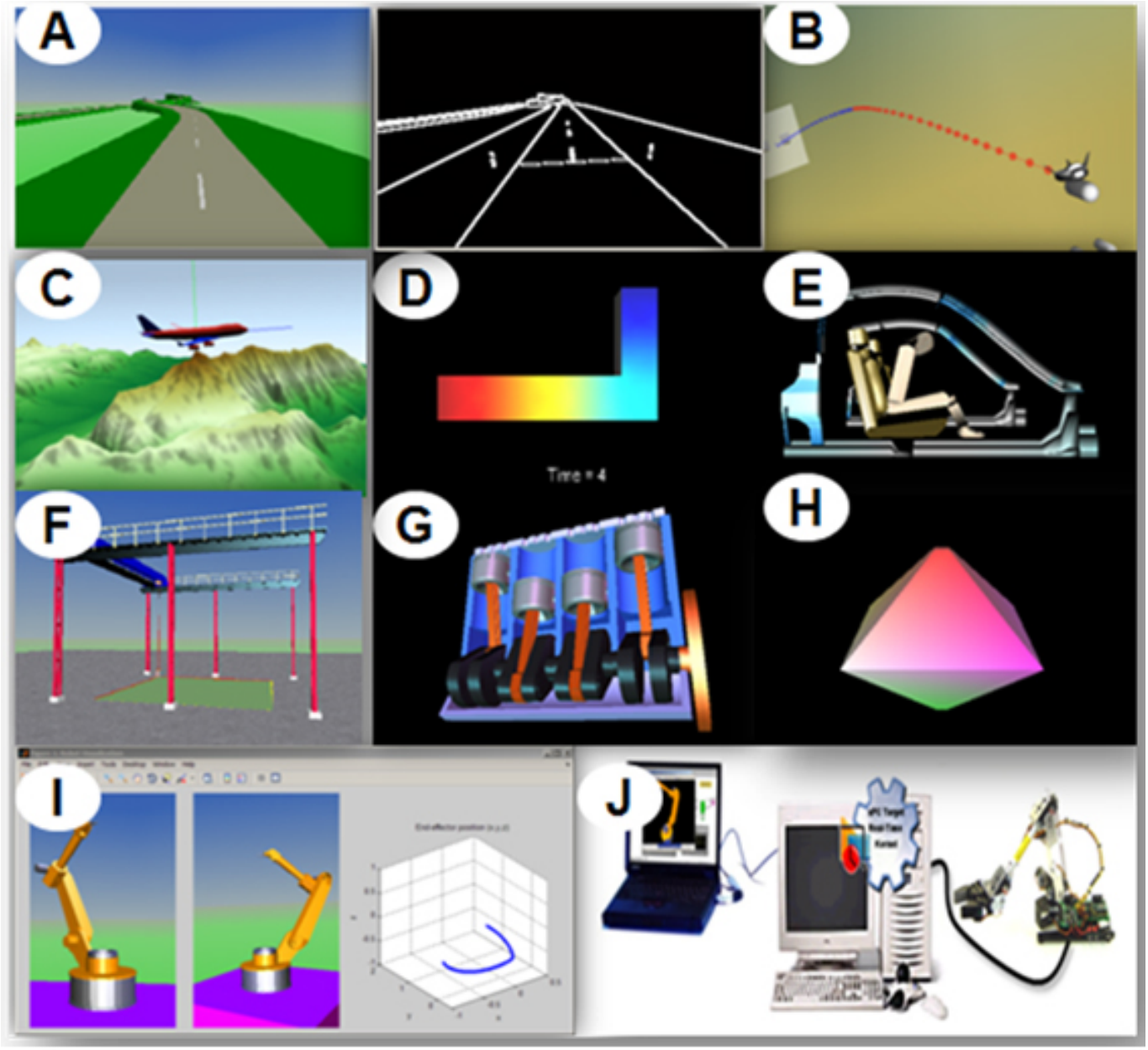

Figure 2: Collage of 3D animation examples that augment early verification in simulation-centric studies.

Create virtual test beds for designing tests: As shown in Figure 2 (A), testing the algorithm for a lane departure system in this computer vision application would require the capture of several input videos, which can be costly to acquire and difficult to determine in the initial stages of the project. After building the test environment virtually, these input videos can be acquired by recording the motion of a camera through this world and then preprocessing them.

Leverage trajectory tracing: As shown in Figure 2 (B), by viewing the trajectory tracing of a spacecraft from different viewpoints, a visual notion of the shape characteristics is obtained. The trajectory is initially colored blue and then changes to red due to the transitioning between different system operating modes. While a state chart animation enables the understanding of the transition between the modes as a function of time, this methodology provides an additional lens by viewing this information in 3D mathematical space. In the later stages of the design,

trajectories can be compared by overlaying the simulation and experimental data.

Understand environment models: As shown in Figure 2 (C), the environment model consists of 3D terrain data of a city visualized for understanding aircraft travel.

Use color dynamics to understand heat and temperature phenomena: As shown in Figure 2 (D), the heat distribution data obtained from a finite element analysis on a solid metallic block is visualized through color dynamics. Understand test failures: As shown in Figure 2 (E), in situations that require multiple tests on a model where a subset of those encounter failure, then 3D animation can help visualize the nature of the fault, as shown in the crash-test visualization.

Conduct interactive tests: As shown in Figure 2 (F), in this dynamic simulation of a portal crane that has been programmed to follow the positional input command, the test engineer can specify the 3D position of the object to be lifted as a test vector for the simulation. The test vector is then mapped to the 3D visual coordinates, providing creative opportunities during interactive testing.

Gain insight into multiple components or object interaction: As shown in Figure 2 (G), this figure shows the motion of the coupled four-cylinders inside an internal combustion engine. The visualization of the motion of the crank, connecting rods, and pistons helps to understand the timing of the entire system. 3D graphic technologies also enable objects to be rendered transparently, thus enabling the appearance of other objects within them.

Understand variable-sized input signals: As shown in Figure 2 (H), this figure shows the morphing of a 3D polyhedral object whose vertices and their associated motion signals appear and disappear in time, which helps understanding of systems where the dimensionality in the input test vector to a system varies in time.

Use GUIs to abstract out implementation details: As shown in Figure 2 (I), this figure shows simulation that has had details abstracted out such that only a couple of parameters are required for tuning. This abstraction can be used to visually verify and quickly reach the answers.

Visualize real-time simulations: As shown in Figure 2 (J), this figure shows the example of rapid prototyping with 3D animation. A 3D animation containing the plant in tandem with the actual view of the hardware provides insight into the differences between the behavior in simulation and the actual implementation.

By using these 3D animation techniques, engineers can overcome the challenges of understanding complex data or sharing information with stakeholders who may not be interested in the technical details of the solutions. The mapping of data in the software model to visual coordinates provides an effective means for understanding and sharing information across teams within an organization. From a marketing standpoint, a recorded animation of a proposed solution is convincing to customers or project sponsors.

Challenges in Using 3D Animation for Simulation-Centric Studies
Despite the advantages of using 3D animation in early verification of designs, several technological challenges exist for the simulation engineer. The fields of computer 3D graphics and engineering simulation have evolved differently, with important consequences. Increasing specialization has led to widely divergent skill sets and has affected the growth and development of technology and tools in the two fields. Figure 3 shows the range of possible applications by plotting 3D graphics fidelity against simulation fidelity.

The economic cost of any application that brings these domains together depends on the fidelities of the simulation and the 3D models. For example, a large-scale 3D flight simulator project requires domain experts such as graphic designers and engineers, as well as computing hardware. The result is the creation of a highly specialized application that meets user needs but is not easy to customize (Figure 3, top right). A range of applications such as solid modeling and cartoon animation (Figure 3, top left) are not simulation-intensive, but depend on high-quality graphics.

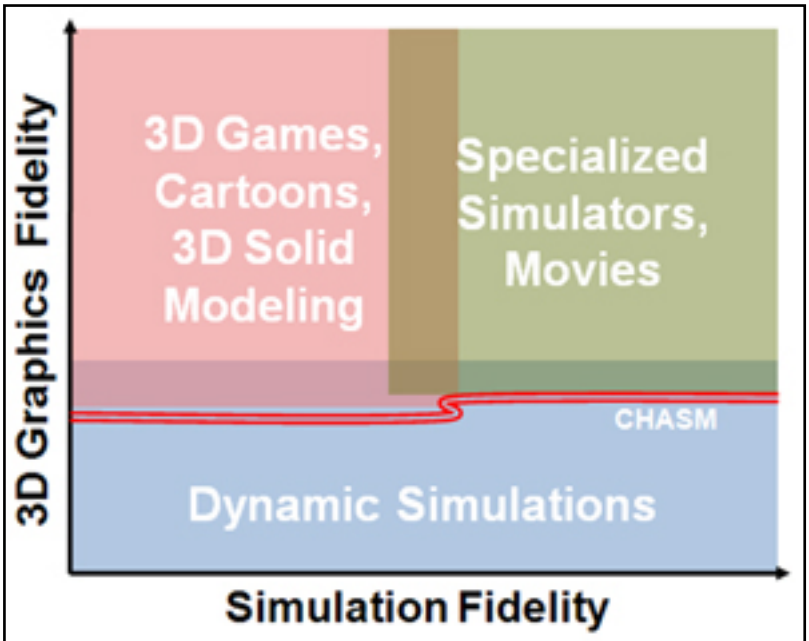

Figure 3: A map showing applications with simulation fidelity plotted against 3D graphics fidelity. The double red line indicates the challenge that simulation engineers must overcome to leverage 3D animation with dynamic simulations.

But what about dynamic simulation projects on an engineer's desktop that may not require high-fidelity 3D graphics (Figure 3, bottom)? Engineers use 2D graphics to visualize the data in their simulations and face the challenge of trying to build compelling 3D animations to run on top of their simulations. This challenge is represented by the chasm shown in Figure 3. Engineers must cross the chasm for simulation studies to be more effective.

Crossing the Chasm

From a simulation engineer's standpoint, to cross this chasm will require that the tools and technologies must enable the following-
• Complexity management of 3D world authoring including portability across different computing platforms
• Seamless connectivity from the simulation environment including 3D rendering
• Lower cost to the individual user to contain project costs

These ideas are explained in more detail below.

Open Standards for 3D Modeling
The plethora of proprietary 3D authoring tools presents a challenge to simulation engineers who want to import the tools into a simulation environment. Open standards for 3D scene authoring enable engineers to build, share, and connect to the tools across simulation environments. VRML, a standard accepted by the International Organization for Standards (ISO), has evolved to the emerging XML-based standard called X3D. The language contains sufficient intuitive 3D constructs that can be used to author 3D scenes. As an example, Simulink 3D Animation provides connectivity to drive 3D worlds authored in VRML from Simulink, thus enabling easy collaboration across different computing platforms. Choosing an open 3D standard also encourages vendors of proprietary tools to support them.

Leveraging 3D Authoring Tools and Web-Based Content
Over last several years, the 3D authoring tools segment has been fiercely competitive as several free and inexpensive tools have emerged and gained popularity. Some examples of these tools are grouped below according to their use in the simulation environment.

3D VRML/X3D editors: An advantage of standardizing on an open standard such as VRML/X3D is the availability of free or inexpensive tools for authoring. Simulation engineers who want to author or modify existing 3D models in VRML can use these editors:
• V-Realm Builder
• Flux Studio
• Internet Scene Assembler
• SwirlX3D
• VrmlPad

3D non-VRML editors: Non-VRML editors are usually powerful and designed for specific 3D modeling tasks. Most have a rich set of examples that engineers can use for simulations. Legacy 3D work in these environments, such as CAD models, can be reused. These tools are generally expensive and complex, but some are relatively inexpensive or free. Examples include:
• 3ds Max
• AC3D
• Autodesk Maya
• Blender
• CATIA
• Google SketchUp
• LightWave

• modo
• Pro/ENGINEER
• SolidWorks

The following editors support a VRML/X3D-export feature. Engineers can define precision and complexity of the exported tessellated files to meet performance requirements. Specialized model conversion and optimization tools offer polygon reduction and model optimization capabilities. Examples include:
• PolyTrans
• Deep Exploration
• VizUp

3D authoring tools that facilitate model creation in highly specialized application areas such as terrain generation, urban modeling, laser scanning, and reverse 3D engineering usually combine connectivity to external data sources with the ability to automatically create 3D models. Examples include:
• 3DEM
• World Construction Set/ Scene Express
• FARO Laser Scanner software tools

3D content on the Web: Market competition in the online 3D content segment has made the acquisition of 3D elements from a vendor for assembly inside a VRML editor very affordable and free in many cases. Popular sites that have emerged in recent years for acquiring such content are:
• www.turbosquid.com [1]
• www.amazing3d.com/ [2]
• www.3dexport.com [3]
• www.the3dstudio.com [4]
• www.3d02.com [5]
• www.top3d.net/ [6]
• www.sketchup.google.com/3dwarehouse [7]

Seamless Connectivity from the Simulation Environment
The simulation tool must provide connectivity and features that support 3D animation. The following examples show different kinds of 3D animation support that are available within the Simulink environment, as shown in Figure 4.

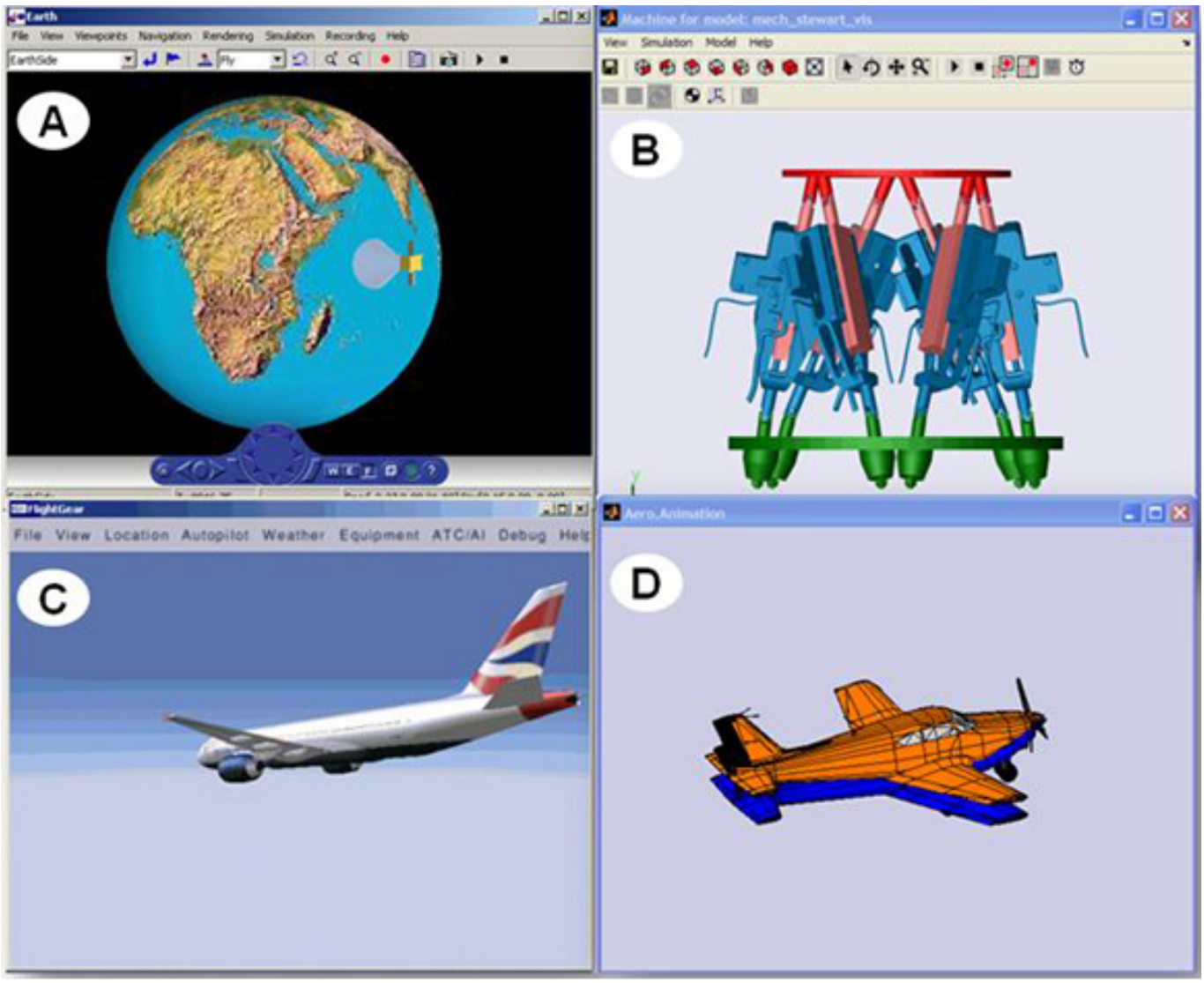

Figure 4: Examples of connectivity to 3D animations driven from the Simulink environment.

Figure 4 (A) Direct connectivity to 3D VRML worlds with Simulink 3D Animation: This product contains blocks that enable the connectivity of Simulink signals to their 3D counterpart. A high quality animation viewer is brings animation into the simulation environment. Figure 4 (B) Automatic CAD model import into SimMechanicsTM models: This CAD model for a Stewart platform was imported automatically with built-in animation into the Simulink environment as SimMechanics blocks. The model is useful for the visualization of mechanical parts modeled in CAD. Multiple parts can be combined to form a scene using the VRML-export approach. Figure 4 (C) Direct connectivity to FlightGear simulator: The FlightGear flight simulator project is an open-source, multiplatform, cooperative flight simulator development project. It is a sophisticated flight simulator framework for visualizing flight simulation. Aerospace Blockset provides blocks that can be used to plug data from simulations within the Simulink environment into this simulator. Figure 4 (D) Direct connectivity to MATLAB Handle Graphics: Most simulation tools will have simple 3D geometrical primitives that can be used to author simple 3D objects. This figure shows the overlay of flight simulation and actual trajectory data for a simple aircraft created with Handle Graphics in MATLAB.

Author's bio:

Saurabh Mahapatra is a product marketing manager for the Simulink platform products at MathWorks. His current interests are in understanding how new and upcoming technologies can leverage Model-Based Design for effective product development. He can be reached at saurabh.mahapatra@mathworks.com [8]

Source URL (retrieved on *12/17/2014 - 4:15pm*):
http://www.ecnmag.com/articles/2010/09/enhancing-simulation-studies-3d-animation

Links:
[1] http://www.turbosquid.com
[2] http://www.amazing3d.com/
[3] http://www.3dexport.com
[4] http://www.the3dstudio.com
[5] http://www.3d02.com
[6] http://www.top3d.net/
[7] http://www.sketchup.google.com/3dwarehouse
[8] mailto:saurabh.mahapatra@mathworks.com