

Simulation and Animation of Mechanical Systems

Jan Danek¹

HUMUSOFT, Prague, Czech Republic

Arkadiy Turevskiy² and Terry Denery³
The MathWorks, Inc., Natick, MA, 01760

This paper introduces a process for simulating and animating mechanical components in the context of a larger system. Using the NASA HL-20 flight vehicle's main landing gear as an example, the paper shows how the CAD assembly can be used as a starting point for creating a dynamic model as well as a realistic animation of the mechanical components. The paper also describes how a dynamic model created from the CAD assembly can be integrated into the larger model of the overall flight vehicle and how the animation created from the CAD assembly can be incorporated into a larger flight vehicle animation. The resulting model and animation are used for detailed analysis and visualization of a main landing gear collapse due to a sudden wind gust shortly before landing.

Nomenclature

$I(n)$ = Unity matrix of n rows and n columns, with diagonal elements equal to one and off-diagonal elements equal to zero.

I. Introduction

MODELING of mechanical systems and components often starts with CAD tools. These tools are useful for specifying the detailed three-dimensional (3-D) mechanical design of a component. Once the CAD model of a mechanical component is complete, engineers will often need to design a control system for it. With the control design task accomplished, engineers will want to analyze how the mechanical component affects the performance of the overall vehicle and how the mechanical components work together with other vehicle systems.

For example, the landing gear group for an aircraft manufacturer may develop a mechanical design of a landing gear; the controls group may develop a controller for that mechanical design; and the systems engineering group may then model how the landing gear operation affects the dynamics of the overall vehicle and how the hydraulic actuator works with the landing gear mechanical design and control system.

Often, different software packages are used for each step of this process, making it difficult to move from mechanical 3-D modeling to control design and then from control design to a system level simulation. This paper introduces a streamlined workflow that enables quick and easy transition from mechanical 3-D modeling to control design and then to system level modeling and simulation. This workflow provides a compelling solution to the control design and system modeling tasks. This solution will be based on using the commercial off-the-shelf software tool, Simulink[®], from The MathWorks, Inc. Using the proposed approach, we will translate a CAD file into an environment where control design can be easily achieved. We will also explain how to easily model and simulate multi-domain systems, including mechanical, hydraulic, and electrical components. We will discuss how to include models of mechanical components into models of larger systems. In addition, we will show how models for environmental conditions, such as wind turbulence and changes in atmospheric pressure and temperature with altitude can be easily integrated into the model of the overall vehicle.

As we discuss the overall process, we will specifically focus on two areas of this workflow. First, we will show how engineers can create Simulink models of mechanical components using CAD models as a starting point. Second, we will discuss how to create high quality 3-D animations from these models. We will present a process for exporting CAD models into Virtual Reality Modeling Language (VRML) files. VRML enables high quality online

¹ Lead Developer, Pobrezni 20, Prague, 18600, Czech Republic.

² Technical Marketing Manager, 3 Apple Hill Drive, Natick, MA, 01760, AIAA member

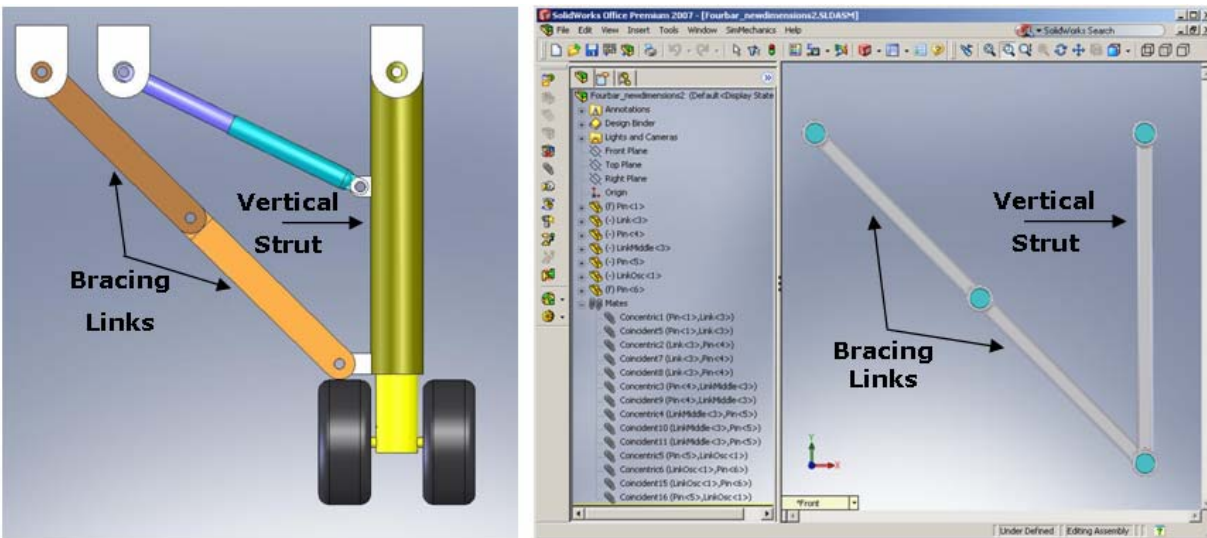
³ Principal Application Engineer, 3 Apple Hill Drive, Natick, MA, 01760, AIAA member

3-D animation. Once the VRML file of the CAD model is available, we will show how to easily connect it to a model of a respective mechanical component and create a 3-D online animation. To illustrate our proposed approach we will use a specific example of a CAD model representing the HL-20 flight vehicle's main landing gear.

The end result of this workflow is a combined model of a mechanical component and a compelling 3-D online animation. With this setup we were able to easily model the dynamics of the landing gear component by itself and then include this component into a system level simulation. The 3-D animation capability provided invaluable insight into the component dynamics and improved the understanding of control system performance and the component's effect on the whole vehicle.

II. Modeling Landing Gear Dynamics as Part of the HL-20 Vehicle Model

Denery, Ghidella, Mosterman, and Shenoy¹ showed how a model of a mechanical component can be developed using a tool for modeling multi-body systems, such as SimMechanics². Specifically, they showed how to develop a dynamic model of HL-20 flight vehicle's landing gear using this tool. The benefit of this approach is that dynamic models of components are much easier to construct using the domain-specific building constructs that SimMechanics provides than by explicitly writing out differential equations governing component dynamics. For instance, in the considered example, a model of a landing gear was constructed using constructs called "body" blocks and "joint" blocks in SimMechanics. Through body blocks users describe the center of gravity (CG) location, mass, and inertia tensor of the links that form a mechanical assembly. In the example, landing gear is constructed of four links: a vertical strut, two bracing links, and a sprung extension (shock absorber). Joints are analogous to physical connections like hinges, slots, and ball and socket connections. For the landing gear, the links connect to the airframe and to each other through revolute joints, which are directly analogous to hinges. These revolute joints are all aligned to rotate about the axes parallel to the longitudinal axes of the airframe. Figure 1 shows a



a) Main landing gear in deployed condition. b) Four-bar linkage SolidWorks assembly representing main landing gear.

Figure 1. Main landing gear and its four-bar linkage representation in SolidWorks®.

SimMechanics model of HL-20 landing gear.

The same paper showed how the model of a mechanical component could be integrated into a simulation model of a larger system. Specifically, it was shown how the SimMechanics model of a landing gear could be integrated into the overall HL-20 flight simulation model. This was done by interfacing a SimMechanics model of the landing gear with the rest of the simulation model implemented in Simulink³. With this approach system engineers have a complete vehicle simulation model that they can use for trade-off studies, for example, to evaluate how the design of a landing gear component affects the performance of the overall flight vehicle. It was also shown how environment could also be modeled in the overall simulation. Using Aerospace Blockset⁴ authors included models of the atmosphere, Earth's gravity field, and the effects of wind disturbances in the simulation. The authors then evaluated how well different designs of the landing gear's hydraulic actuator were able to prevent a collapse of the landing gear after a strong wind gust immediately before the touch-down.

While it provided system engineers with useful tools for evaluating effects of mechanical component design on the performance of the overall vehicle, the approach proposed by Denery, Ghidella, Mosterman, and Shenoy did not include one important workflow improvement that is proposed and described in this paper. Mechanical design of mechanical components is typically done using a CAD tool such as SolidWorks⁵ or Pro/ENGINEER^{®6}. One significant improvement to the process described by Denery, Ghidella, Mosterman, and Shenoy could result from automatically creating a dynamic model of a mechanical component from its CAD file. System engineers or control engineers would then not have to build a dynamic model themselves. Instead of manually constructing a dynamic model from SimMechanics blocks, the engineer would simply initiate the conversion process of the CAD file to generate the dynamic model automatically. In addition to significantly simplifying the workflow, this enhancement would eliminate the potential errors that system engineers or control engineers could make when manually creating a dynamic model from the CAD assembly.

This process improvement is possible using two translators developed by The MathWorks. One of these translators⁷ automatically exports SolidWorks assemblies into SimMechanics models, while the other does the same for Pro/ENGINEER assemblies. The built-in SimMechanics visualization is useful for understanding the behavior of the modeled mechanisms. However, SimMechanics animation does not provide realistic rendering of the modeled component. An alternative animation method, based on the use of virtual reality graphics, makes it possible to create realistic animations for a SimMechanics model created from CAD assembly. In the rest of the paper we will use the HL-20 main landing gear component example to illustrate how a CAD assembly of a mechanical component can be used to automatically create a SimMechanics model and a realistic animation to connect to it. We will then show how this SimMechanics model and the animation can fit into the overall HL-20 flight simulation.

III. Creating SimMechanics Model of HL-20 Landing Gear from SolidWorks Assembly

As described by Denery, Ghidella, Mosterman, and Shenoy, the main landing gear represents a well-known mechanical construct, the four-bar linkage. As Fig. 1a shows, when the landing gear is deployed, the vertical strut is rotated by 90 degrees relative to the airframe. The lower bracing link (link 1) is rotated by 45 degrees relative to the vertical strut, and the upper bracing link (link 2) is collinear with the lower bracing link. Since the lengths of vertical strut, bracing link 1, and bracing link 2 were defined by Denery, Ghidella, Mosterman, and Shenoy to be 1 meter, 0.707 meters, and 0.707 meters respectively, the four-bar linkage composed of the airframe, vertical strut, and two bracing link forms an isosceles right triangle. In this triangle the airframe and the vertical strut serve as catheti, and the two bracing links form the hypotenuse.

Figure 1b shows a SolidWorks assembly of a four-bar linkage corresponding to the described configuration of the HL-20 landing gear. In addition to three links corresponding to vertical strut and two bracing links, the SolidWorks assembly shown also contains four pins that are used for illustrating the rotational motion of links relative to each other and to the airframe. The top two pins represent the connection of the three links to the airframe and are fixed relative to the airframe. The lower right pin enables the rotation of bracing link 1 relative to vertical strut. Finally, the lower left pin enables the rotation of bracing link 2 relative to bracing link 1.

To create a SimMechanics model from this assembly it is necessary to download and install SolidWorks-to-SimMechanics Translator. Once this free software is installed, it is possible to generate a textual description of the assembly that lists the mass properties for each body and the characteristics of each joint defined in the SolidWorks assembly. This description is saved to a SimMechanics XML file as shown in Fig. 2. Once the SolidWorks assembly has been saved as a SimMechanics XML file, it is possible to automatically create a SimMechanics model out of it. As Fig. 3a shows, this can be done with a single command in MATLAB[®]. The resulting SimMechanics model with blocks rearranged and background colors added for easier understanding is shown in Fig. 3b. It should be noted that not all CAD assemblies can be exported into SimMechanics models. In SolidWorks the

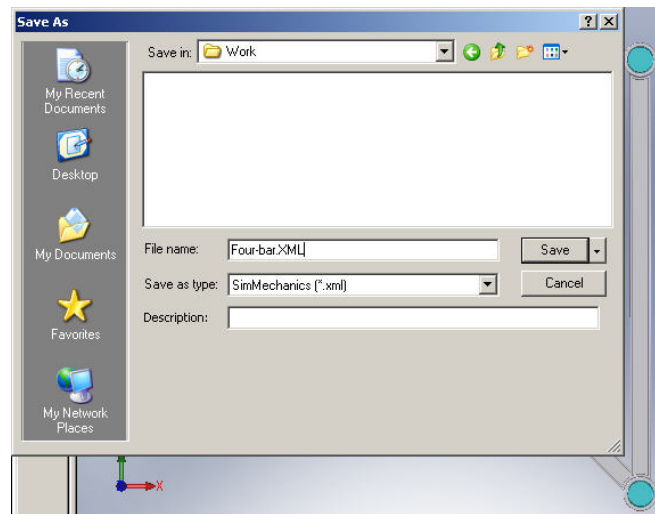
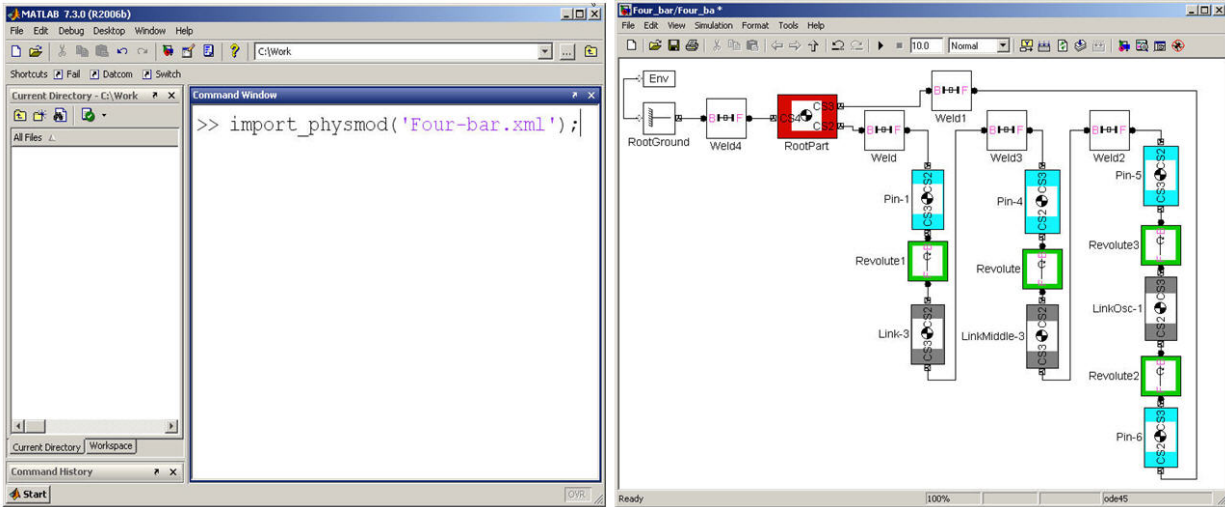


Figure 2. The SolidWorks assembly can be saved as SimMechanics XML file.



a) A SimMechanics model can be created with one line of code. b) Resulting SimMechanics model.

Figure 3. A SimMechanics model can be automatically created from a SolidWorks assembly.

restrictions limiting the location of one body relative to another are called mates. SolidWorks-to-SimMechanics Translator attempts to find a SimMechanics joint corresponding to each mate in SolidWorks assembly. As there is no one-to-one correspondence between SolidWorks mates and SimMechanics joints, the conversion is not always possible, or it does not translate as intended. SolidWorks-to-SimMechanics Translator manual lists supported SolidWorks mates. As Fig. 1b shows, the SolidWorks assembly for the landing gear uses concentric and coincident mates (the list of mates is shown in the lower left corner of the SolidWorks screen). These types of mates are supported by SolidWorks-to-SimMechanics Translator, and therefore the SolidWorks assembly can be successfully translated into a SimMechanics model. When a SolidWorks assembly contains unsupported mates, the translation process will still proceed, but the unsupported mates will be replaced by weld joints in the resulting SimMechanics model. In this situation, even though the resulting SimMechanics model does not capture the component dynamics correctly, it is still valuable because it contains the mass and inertia properties of the bodies, automatically created from the SolidWorks assembly.

A closer examination of the SimMechanics model in Fig. 3b shows that the model contains eight bodies: seven bodies representing four pins (highlighted in blue) and three links (highlighted in grey), as expected, as well as a body called *RootPart* that represents the airframe. The top two pins of the assembly are connected to *RootPart* by weld joints. This type of connection ensures that these two pins are fixed relative to *RootPart*. Each of the three links is connected to two pins. For example, as Fig. 3b shows, the vertical strut (represented by block *LinkOsc-1*) is connected by a revolute joint to upper right pin (block *Pin-6*) and by a revolute joint to lower right pin (block *Pin-5*). SolidWorks-to-SimMechanics Translator created similar connections for the other two links.

As mentioned above, in addition to placing the bodies into the SimMechanics model and connecting them with appropriate joints, SolidWorks-to-SimMechanics Translator also automatically determines mass, inertia, and coordinate system properties for all bodies.

Once the SimMechanics model is automatically created, it can be verified by running a simulation. Figure 4 shows a SimMechanics visualization for the simulation of the created model. The four-bar linkage moves under the influence of the gravity force in this scenario.

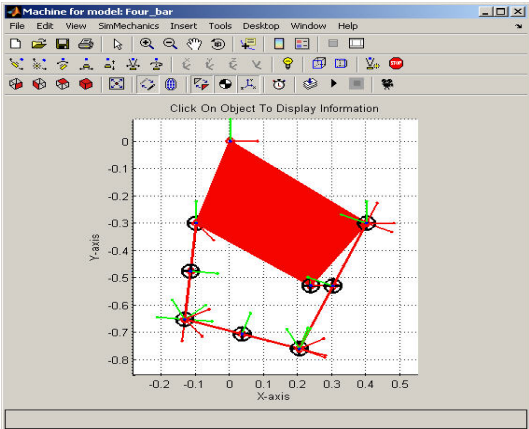


Figure 4. A visualization of four-bar linkage motion in SimMechanics.

IV. Creating a Virtual Reality Animation from the SolidWorks Assembly

As Fig. 4 shows, SimMechanics visualization, while providing useful insight into the behavior of mechanical component, does not produce realistic animation. High quality realistic animation can be achieved using Virtual Reality Toolbox⁸. Virtual Reality Toolbox represents an interface between MATLAB and Simulink data on one hand and virtual reality graphics on the other hand. Virtual reality graphics are based on VRML, an open standard for describing 3-D scenes. A detailed description of VRML can be found in multiple sources, including a comprehensive textbook by Marrin and Campbell⁹, as well as several other publications^{10,11}. Virtual Reality Toolbox has been successfully used in multiple applications for visualizing results of Simulink simulations^{12,13}. Often, however, system and control engineers who are unfamiliar with VRML find it difficult to create a VRML file describing a 3-D scene they would like to visualize.

For a design that starts with CAD assemblies, the VRML file creation process can be significantly simplified. In the HL-20 landing gear example, the SolidWorks assembly can be saved as a VRML file, as shown in Fig. 5. Several options have to be set properly for the process to be successful. Figure 6a shows one of these options, image quality. This setting can be accessed through the Tools->Options menu. The top slider (**Shaded and draft quality HLR and HLV resolution**) should be set as shown in Fig. 6a. Setting its value too low will result in low quality animation. Setting its value too high will result in unnecessarily high number of polygons in the resulting VRML file, leading to slow animation. Several other options are shown in Fig. 6b. The window in Fig. 6b is opened by choosing the **Options...** button in the “Save as VRML” dialog shown in Fig. 5. In the window shown in Fig. 6b it is necessary to choose “VRML97” as the version of

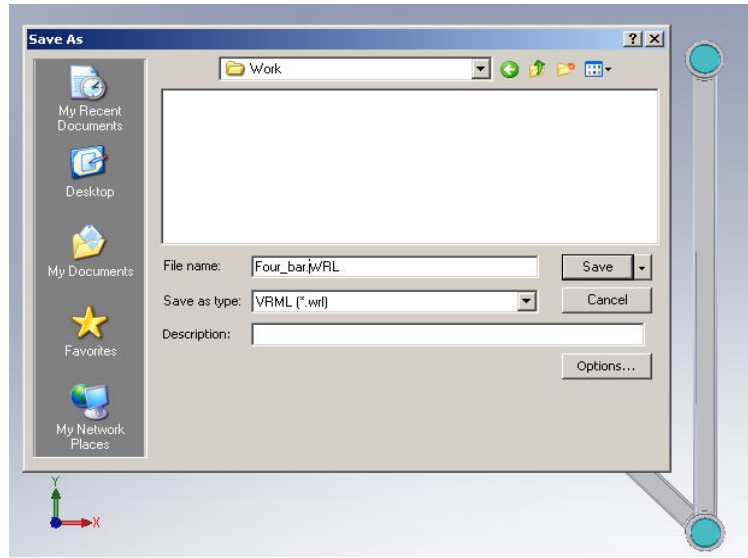
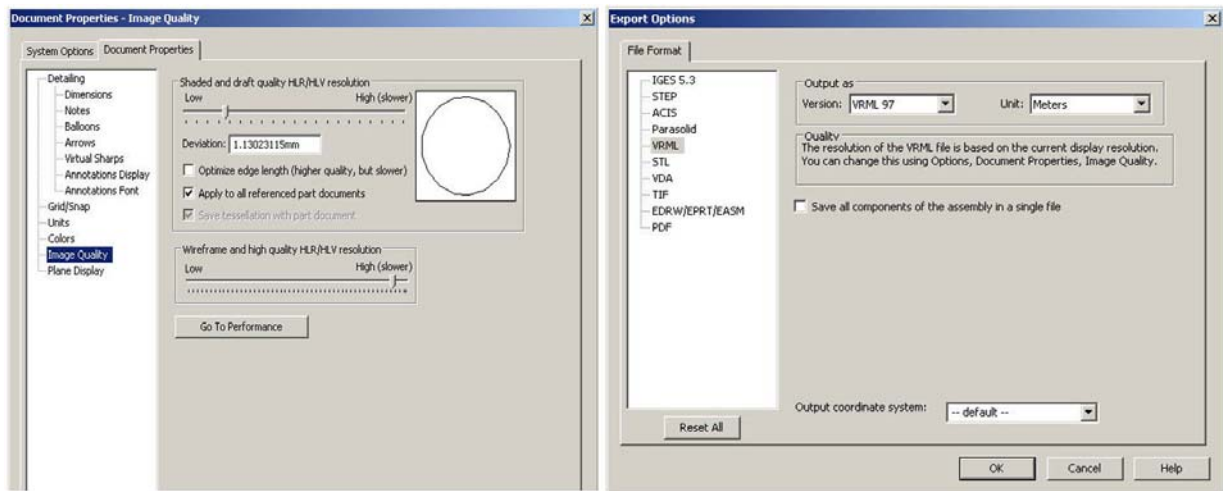


Figure 5. The SolidWorks assembly can be saved as a VRML file.

VRML97” as the version of



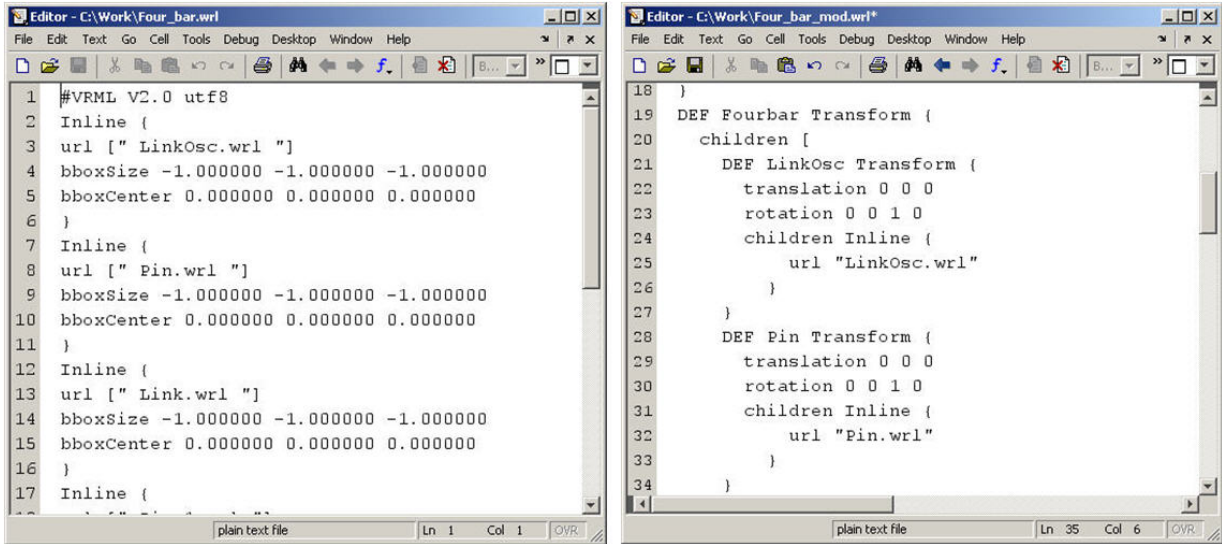
a) The image quality option.

b) Options for saving as VRML.

Figure 6. VRML exporting options have to be set properly to create realistic animation.

VRML and “meters” as the unit. It is also necessary to keep the box “Save all the components of the assembly into a single file” unchecked. When all the options are selected as described, and the SolidWorks assembly of a landing gear is saved as VRML, eight VRML files are generated. These eight files contain 3-D geometries of three links and

four pins. There is also a file describing the whole mechanical component. This VRML file was called Four_bar.wrl. Since VRML files are text documents, the content of a VRML file can be viewed in any text editor. Figure 7a shows



a) A VRML file automatically created from the SolidWorks assembly.

b) A VRML file with extra spaces removed and Transform nodes added.

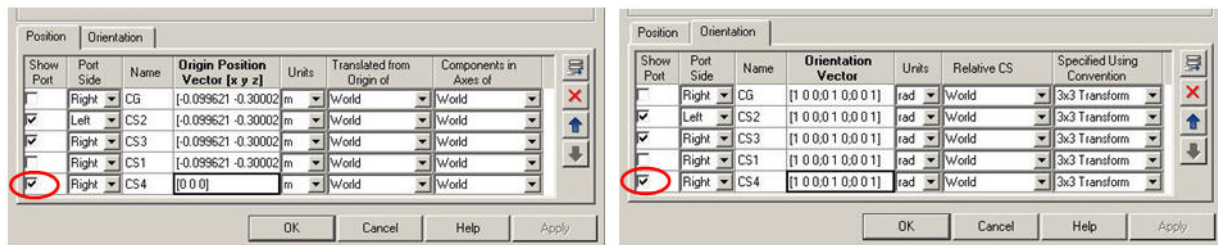
Figure 7. A VRML file created from SolidWorks assembly and the required modifications.

the content of Four_bar.wrl in the MATLAB editor. Four_bar.wrl file contains seven Inline objects that reference four pins and three links forming the landing gear component. Some modifications to the Four_bar.wrl file are needed before it can be used with Virtual Reality Toolbox to animate the SimMechanics model automatically generated from SolidWorks assembly.

First, extra spaces have to be removed between the quotation marks and the names of inlined files. Second, Transform objects have to be wrapped around inlined files. Figure 7b shows the described changes applied to file Four_bar.wrl and saved as Four_bar_mod.wrl. Virtual Reality Toolbox works with Transform objects by changing properties, such as translation and rotation, with values from Simulink models or MATLAB data. Finally, lighting, viewpoints, background, world information, and navigation information can all be added to the VRML file as needed.

V. Connecting a Virtual Reality Scene to the SimMechanics Model

With both the SimMechanics model and the virtual scene completed, the next step is to connect the model to the virtual scene in order to create a realistic high-quality animation. The first step of this process is to add an additional coordinate system to each of the bodies representing the four pins and three links in the SimMechanics model. Figure 8 shows this for left upper pin where a new coordinate system, CS4, is being added. The new coordinate



a) The new coordinate system center is at (0, 0 0).

b) The orientation matrix is $I(3)$.

Figure 8. A new coordinate system should be added to each body in SimMechanics model.

system should have its origin at (0, 0, 0) and its orientation matrix should be $I(3)$, as shown in Fig. 8a and Fig. 8b respectively. The new coordinate system is defined relative to the world coordinate system. By placing the check in the box next to this coordinate system, as shown in Fig. 8, an extra output is generated for each of the four pins and

three links. The signal connected to this output carries the information about position and rotation of the corresponding body in the world coordinate system. These positions and rotations should be measured and fed to the virtual scene. Therefore, the second step is to connect a body sensor block for each of the four pins and three links

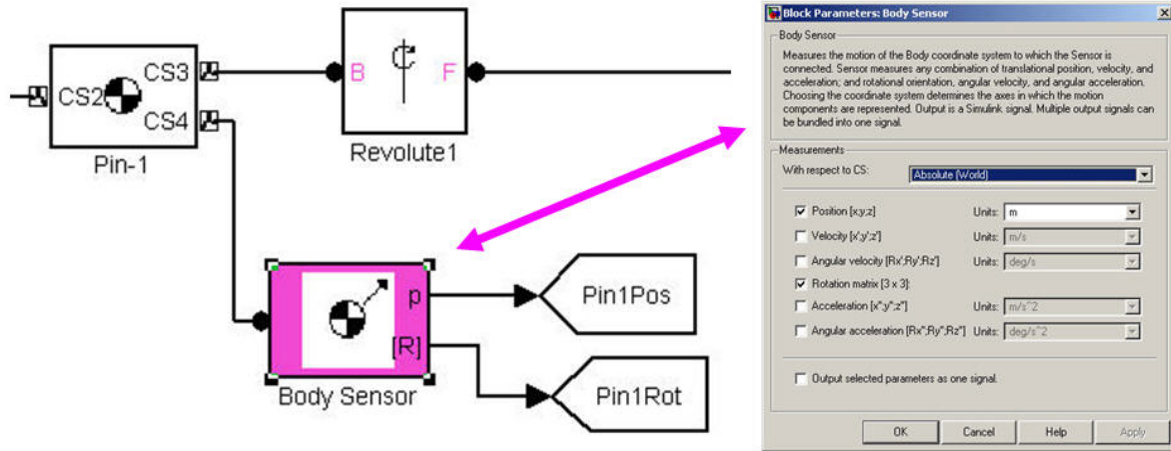
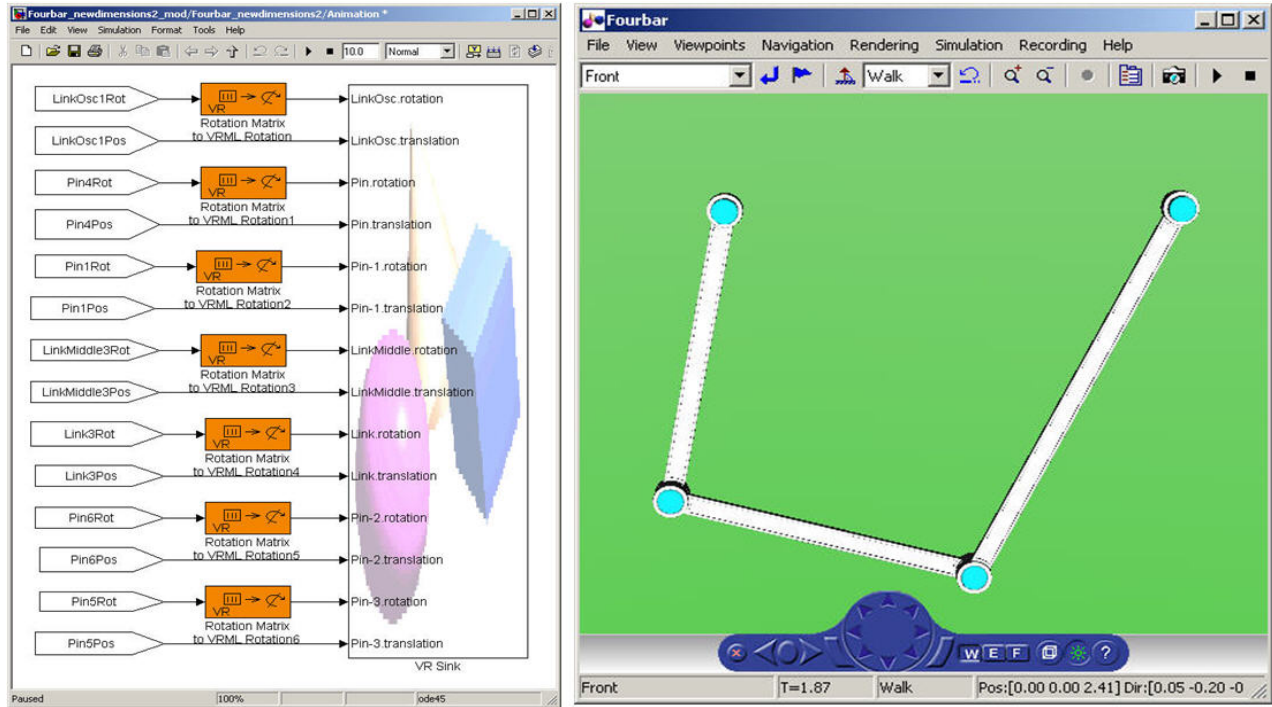


Figure 9. A Body Sensor block is used to measure the position and rotation of each SimMechanics body.

through the output created in the first step.

Figure 9 shows this connection as well as the options for the Body Sensor block (highlighted in magenta). These options determine that the outputs from the Body Sensor block contain the rotation matrix and position. Step three is to convert the rotation matrix into axis-angle representation, as VRML uses that representation to define the body's orientation. The conversion from rotation matrix to axis-angle representation is shown in Fig. 10a and is accomplished by using the appropriate blocks (highlighted in orange) from either the SimMechanics library or the



a) Connection of the SimMechanics model to the virtual reality scene.

b) Simulation results can now be animated with the virtual reality scene.

Figure 10. The Body Sensor block is used to measure the position and rotation of each SimMechanics body.

Virtual Reality Toolbox library (blocks from the Virtual Reality Toolbox library were used in the case shown in Fig. 10). The final step is to connect SimMechanics model to Virtual Reality scene using a VR Sink block from Virtual Reality Toolbox. This is shown in Fig. 10a as well.

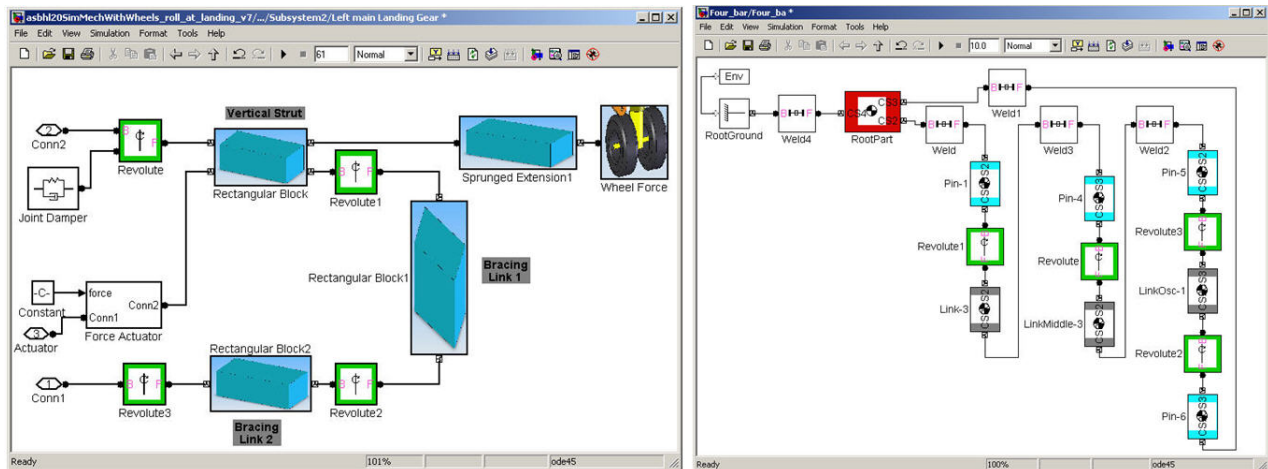
Once the connection has been established, the simulation can be started and the animation can be observed in the Virtual Reality window, as shown in Fig. 10b.

VI. Multi-Domain Modeling and Control Design using a SimMechanics Model of a Mechanical Component

The SimMechanics model and corresponding Virtual Reality Toolbox animation are useful in providing insight into the dynamic behavior of the mechanical component. SimMechanics model can be further used for designing a controller for the mechanical component in question and for performing multi-domain modeling. For example, a controller could be designed that would apply the appropriate amount of torque to the vertical shaft to keep it in the desired position. To accomplish that, the SimMechanics model would be linearized appropriately, and the controller would be designed for the linearized plant. Both the linearization task and the control design can be conveniently performed using Simulink[®] Control Design. Additionally, the mechanical rigid body SimMechanics model could be interfaced with models of components of other physical domains. For example, the controller that was just discussed could be implemented with a hydraulic actuator or electric motor. The appropriate modeling tools, SimHydraulics[®], SimPowerSystems, and SimDriveline allow multi-domain modeling and simulation of this type in the Simulink environment. This topic was covered in detail by Denery, Ghidella, Mosterman, and Shenoy and therefore will not be discussed again here.

VII. Incorporating a Model of Landing Gear Mechanical Components into a Simulation of the HL-20 Flight Vehicle

The obtained SimMechanics model of the landing gear can be incorporated into a flight simulation model of the HL-20 flight vehicle. This was shown and explained in detail by Denery, Ghidella, Mosterman, and Shenoy. Figure 11a shows a SimMechanics model for the main landing gear developed by the aforementioned authors. Figure 11b



a) A SimMechanics model of the main landing gear in a HL-20 simulation.

b) A SimMechanics model of main landing gear automatically created from SolidWorks.

Figure 11. A SimMechanics model obtained from the SolidWorks assembly can be incorporated into the HL-20 simulation model with minor changes.

shows the same SimMechanics model as shown in Fig. 3b – the SimMechanics model obtained by automatically translating the SolidWorks assembly of a four-bar linkage. Comparing SimMechanics models in Fig. 11a and Fig. 11b it can be observed that the SimMechanics model in Fig. 11b has three links (highlighted in gray) connected by four revolute joints (highlighted in green) just like the model in Fig. 11a does (the three links are shown by textboxes next to them highlighted in gray). The SimMechanics model in Fig. 11a does not model pins as the SimMechanics model in Fig. 11b does (pins are highlighted in blue in Fig. 11b), and it also includes a model of sprung extension to model wheel forces. Therefore, it can be seen that the SimMechanics model obtained from the SolidWorks assembly can be used to model a landing gear as a part of the overall flight vehicle model with only minor modifications required.

In addition to incorporating a SimMechanics model of the landing gear into the overall vehicle simulation, the animation of a four-bar linkage representing the landing gear had to be incorporated into a virtual scene showing a landing of the HL-20. The preparation of this virtual scene will be described first. One of the standard shipping Aerospace Blockset demos, aerblk_HL20_vr, was used as a starting point. This demonstration file has a virtual scene linked to it with Virtual Reality Toolbox. The virtual scene shows the landing of the HL-20 flight vehicle and is shown in Fig. 12. As can be seen from that figure, the 3-D model of the HL-20 flight vehicle used in the virtual scene is low resolution and does not include 3-D models of landing gear components. Therefore, this virtual scene was used as a starting point but was modified to include a higher resolution model of the HL-20 flight vehicle. This higher resolution model of the HL-20 flight vehicle was obtained by using a 3-D Model of HL20 created in

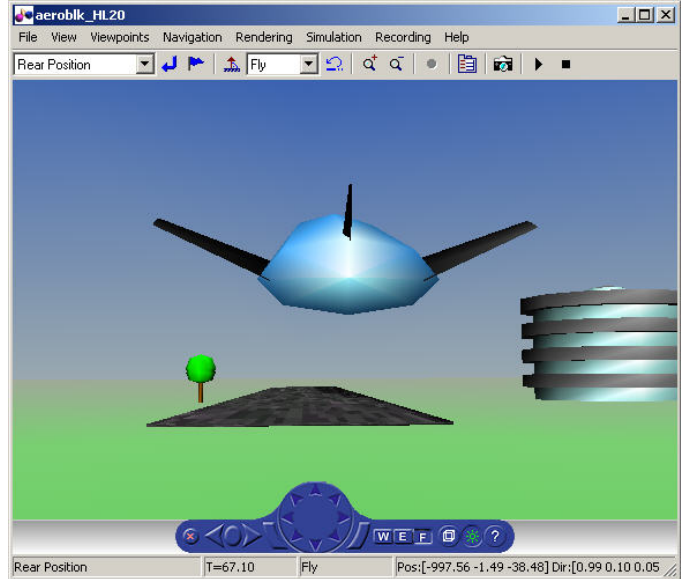


Figure 12. The HL-20 landing virtual scene with a low resolution VRML model of the HL-20 was used as a starting point.

AC3D™ software and used in a FlightGear flight simulator visualization with the asbhl20 demo from Aerospace Blockset. The AC3D model of the HL-20 was converted into VRML and incorporated into the virtual scene. Virtual scene was also updated with a more realistic terrain model.

Next, a virtual scene had to be connected to the model. To do this, latitude, longitude, and altitude were converted into X, Y, and Z VRML coordinates to define the HL-20 position in the virtual scene. Pitch, yaw, and roll were converted to axis-angle format and fed into VRML scene to define the HL-20 vehicle attitude. Figure 13 shows the connection of the HL-20 flight simulator model to the virtual scene.

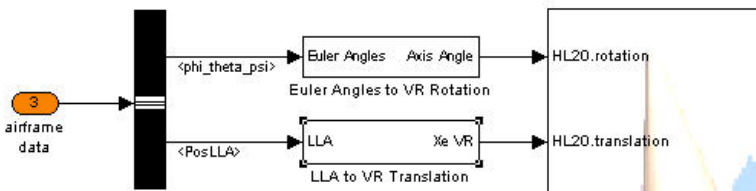
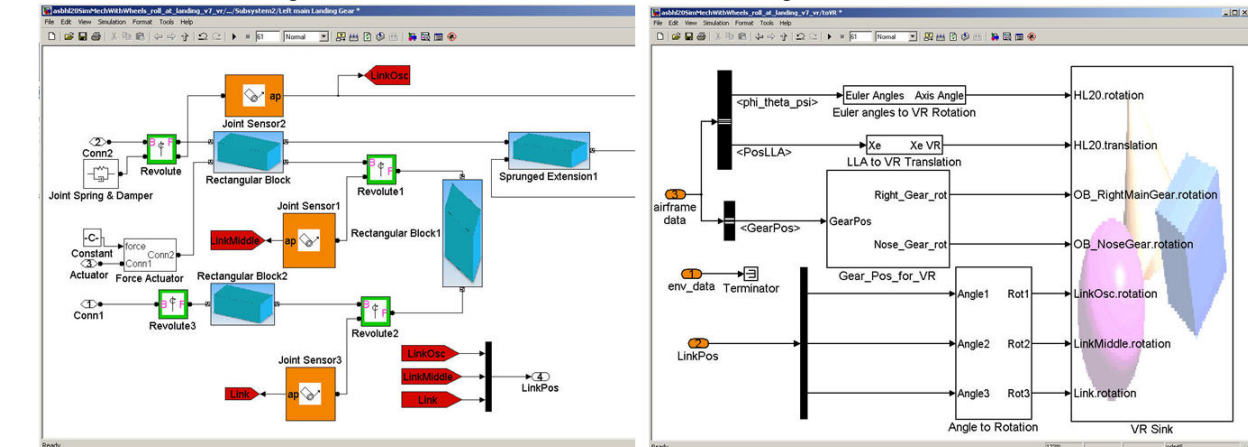


Figure 13. The HL-20 pitch, yaw, and roll angles as well as latitude, longitude and altitude had to be converted into VRML rotation and translation to connect to the virtual scene.



a) The model of the main landing gear was modified to send link rotation angles to the virtual scene. b) Connection of Simulink model to the virtual scene.

Figure 14. SimMechanics model obtained from SolidWorks assembly can be incorporated into HL-20 simulation model with minor changes.

With the high resolution model of the HL-20 incorporated into the virtual scene, the next step was to include a four-bar linkage 3-D model previously generated from the SolidWorks assembly into the virtual scene as well. To do that, an extra transform node was added to the virtual scene to allow proper positioning and orientation of the four-bar linkage relative to the HL-20 airframe. The HL-20 flight simulation model had to be modified as well. Figure 14a shows how the SimMechanics subsystem for modeling the main landing gear was modified. Three *Joint Sensor* blocks (highlighted in orange) were connected to three revolute joints to sense the three rotation angles (highlighted in red) of the landing gear links and send these signals to the virtual scene. The measured rotation angles were the rotation angle of the vertical strut relative to the airframe, the rotation angle of bracing link 1 rotation angle relative to the vertical strut, and the rotation angle of bracing link 2 relative to bracing link 1. Figure 14b shows the resulting model connection to the virtual scene. With this setup the model could be run and the deployment of landing gear could be visualized, including how individual links rotate relative to each other and relative to the airframe.

To test the resulting model and the virtual reality scene, the simulation scenario described by Denery, Ghidella, Mosterman, and Shenoy was used. In this scenario, a wind gust is introduced shortly before the touch down. This wind gust caused the HL-20 to roll and touch down on only one wheel. This causes the landing gear to collapse and the flight vehicle crashes. The described scenario was run with the adjusted model, and as expected, the model provided link rotation angles describing the collapse of the landing gear. Figures 15-17 show several snapshots of

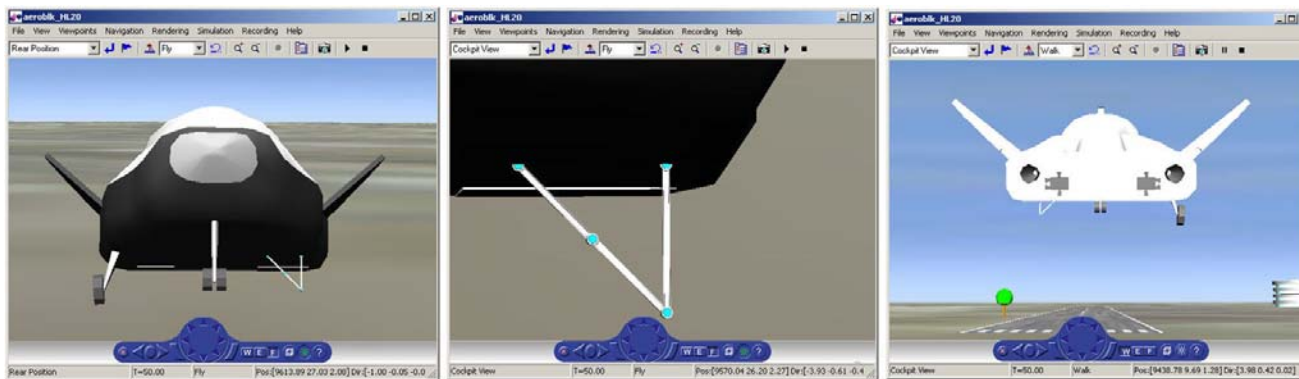


Figure 15. The HL-20 shortly before landing with landing gear deployed.

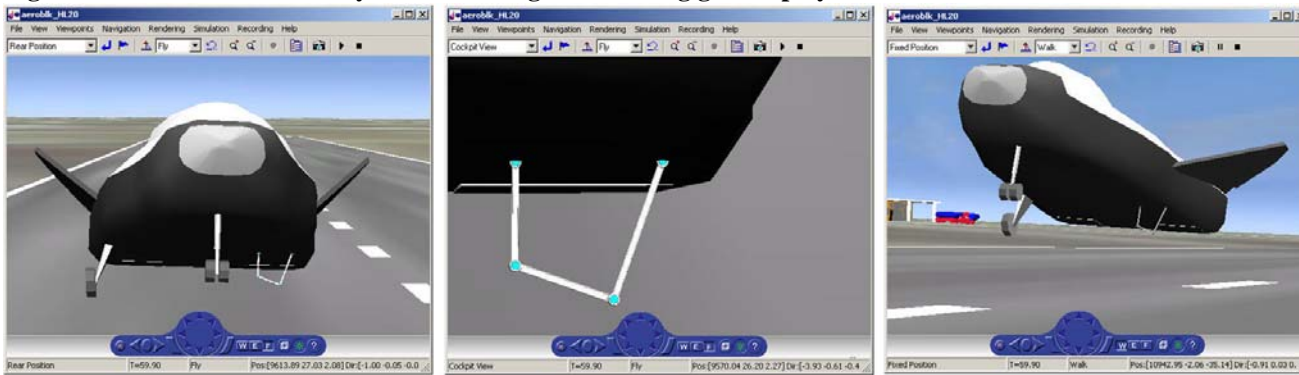


Figure 16. The HL-20 immediately after touch down. Landing gear is starting to collapse.

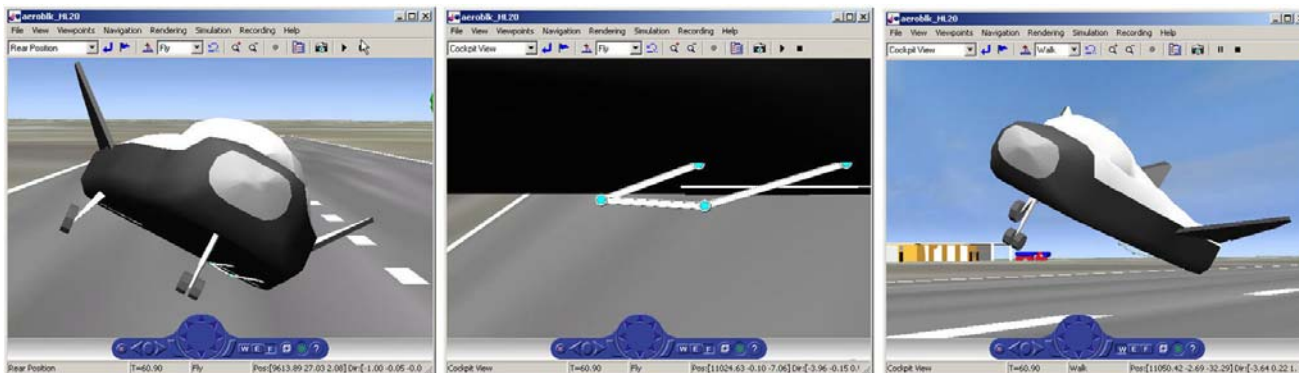


Figure 17. The HL-20 after main landing gear collapses completely.

the virtual scene with a clear representation of how the landing gear collapses and the flight vehicle crashes. This simulation proved the validity of the model and the related virtual scene.

VIII. Conclusion

The paper presented a workflow for utilizing CAD assemblies of mechanical components for creation of dynamic models, control design, multi-domain modeling, and system simulation. It was shown how a realistic animation can be created and connected to the model to help understand system dynamics. Using an example of HL20 landing gear it was shown how the collapse of a main landing gear at landing can be modeled and animated using the proposed approach. The value of the proposed approach is in the simplified model development, reuse of CAD assemblies developed by mechanical engineers, and better understanding of component and system dynamics through realistic animation.

References

-
- ¹ Denery, T., Ghidella, J. R., Mosterman, P. J., and Shenoy R., "Creating Flight Simulator Landing Gear Models Using Multidomain Modeling Tools," *AIAA Modeling and Simulation Technologies Conference and Exhibit*, Keystone, Colorado, 2006.
 - ² SimMechanics, "SimMechanics User's Guide", The MathWorks, Inc., Natick, MA, March 2007.
 - ³ Simulink, "Simulink User's Guide", The MathWorks, Inc., Natick, MA, March 2007.
 - ⁴ Aerospace Blockset, "Aerospace Blockset User's Guide", The MathWorks, Inc., Natick, MA, March 2007.
 - ⁵ SolidWorks Office Premium 2007, Software Package, SolidWorks Corporation, Concord, MA, 2007.
 - ⁶ Pro/ENGINEER Wildfire 3.0, Software Package, PTC, Needham, MA, 2007.
 - ⁷ SolidWorks to SimMechanics Translator, Software Package, The MathWorks, Inc., Natick, MA, 2007.
 - ⁸ Virtual Reality Toolbox, "Virtual Reality Toolbox User's Guide", The MathWorks, Inc., Natick, MA, March 2007.
 - ⁹ Marrin., C., and Campbell, B., *Teach Yourself VRML 2 in 21 Days*, 1st ed., Sams.net, Indianapolis, Indiana, 1997.
 - ¹⁰ The Virtual Reality Modeling Language, International Standard ISO/IEC 14772-1:1997.
 - ¹¹ Carey, R., and Bell, G., *The Annotated VRML 2.0 Reference Manual*, Addison-Wesley Publishing, Boston, MA 1997.
 - ¹² Castillo-Effen, M., Alvis, W., Castillo, C., Valavanis, K. P., and Moreno, W. A., "Modeling and Visualization of Multiple Autonomous Heterogeneous Vehicles," *IEE Systems, Man and Cybernetics International Conference*, Vol. 3, October 2005, pp. 2001-2007.
 - ¹³ Mayhew, E. J., "Creating Standalone Virtual Environments," *MATLAB Digest*, The MathWorks Inc., Natick, MA, March 2007

©1994-2007 by The MathWorks, Inc., MATLAB, Simulink, Stateflow, Handle Graphics, Real-Time Workshop, SimBiology, SimHydraulics, SimEvents, and xPC TargetBox are registered trademarks and The MathWorks, the L-shaped membrane logo, Embedded MATLAB, and PolySpace are trademarks of The MathWorks, Inc. Other product or brand names are trademarks or registered trademarks of their respective holders.