Simscape Modeling Verification in the Simulink Development Environment

Christopher E. E. Volle
Kennedy Space Center
Major: Mechanical Engineering

KSC Fall Session
Date: 22 10 2014

Simscape Modeling Verification in the Simulink Development Environment

C. Volle[1]
University of Nebraska – Lincoln, Lincoln, Nebraska, 68508

The purpose of the Simulation Product Group of the Control and Data Systems division of the NASA Engineering branch at Kennedy Space Center is to provide a real-time model and simulation of the Ground Subsystems participating in vehicle launching activities. The simulation software is part of the Spaceport Command and Control System (SCCS) and is designed to support integrated launch operation software verification, and console operator training. Using Mathworks Simulink tools, modeling engineers currently build models from the custom-built blocks to accurately represent ground hardware. This is time consuming and costly due to required rigorous testing and peer reviews to be conducted for each custom-built block. Using Mathworks Simscape tools, modeling time can be reduced since there would be no custom-code developed. After careful research, the group came to the conclusion it is feasible to use Simscape's blocks in MatLab's Simulink. My project this fall was to verify the accuracy of the Crew Access Arm model developed using Simscape tools running in the Simulink development environment.

I. Introduction

The main goal of the project was to test the accuracy of fixed-step versus variable-step solvers on the Space Launch System (SLS) Crew Access Arm (CAA) Actuation subsystem in a MatLab Simscape development environment; The secondary goal is to adjust the model component configuration and parameters in order to more accurately simulate the expected real-life motion of the CAA. Currently, the Simulation Product Group uses MatLab's Simulink development environment to build their simulation models. Those models were then compiled using variable-step solvers in a desktop environment, followed by fixed-step solvers to prepare the model for conversion to C or C++ to be run in Trick. Trick is a Government off the Shelf (GOTS) software which executes simulations in a real-time environment. To make models in Simulink, the Group must code custom library blocks to simulate the real-life hardware used in building various systems and subsystems. Programing custom blocks costs valuable time and money due to required rigorous testing and peer reviews to be conducted for each custom-built block to ensure there are no coding bugs before use in the simulation models. Simscape library blocks do not require the same level of careful scrutiny, since they are part of the MatLab program. The Simulation Product Group's plan is to reduce the cost and shorten the lengthy component development process by utilizing Simscape library blocks to build the required simulation models.

II. MatLab and the Crew Access Arm

A. Learning about Simscape

MatLab has its own language and is a computer algebra system used for complex computations and data analysis. MatLab is made by Mathworks and commonly used by NASA for data processing and analysis as well as modeling and simulation. Modeling is completed in MatLab Simulink, which is an add-on with a graphical user interface and library of blocks representing mathematical operations which can be linked together to visually represent equations and systems. Simscape is located in Simulink and has blocks to represent physical parts in order to represent entire physical systems. Using Simscape, it is possible to build models of entire systems. The blocks used in Simscape have two different sections, which are referred to as First Generation and Second Generation SimMechanics.

[1] Software Engineer, Application and Simulation Software Engineering Branch, NE-C1.

B. The Crew Access Arm

The SLS Crew Access Arm is the platform which, when fully extended, is the walkway that allows the astronauts to move between the Mobile Launch Tower and the SLS Orion Capsule. At its fully retracted position, the side of the CAA is against the side of the Mobile Launch Tower. To be fully extended, the CAA must rotate approximately 195° in 120 seconds. Figure 1 shows the Mobile Launch Tower with the CAA in this fully extended position in contact with the SLS. In order to keep contaminates from entering the crew module, the Environmental Chamber located at the end of the CAA is capable of creating a seal and decontaminating its interior, as well as anyone within the chamber.

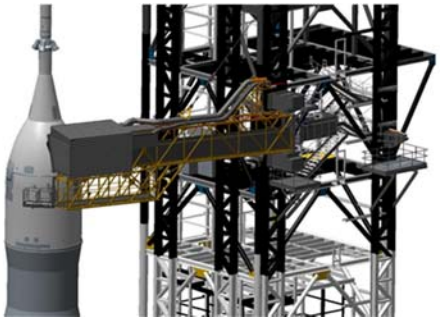

Figure 1. The fully extended Crew Access Arm (Ref. 2).

C. The Model

The model consists of various Second Generation SimMechanics Simscape blocks from both SimHydraulics and SimMechanics. The SimHydraulics blocks are the pressure source, valves, and double-acting hydraulic cylinders; they are arranged in such a way that the model is capable of both extension and retraction. SimMechanics are the two pinions and four racks.

There are two valves, one Discrete and one Analog, for both extension and retraction, bringing the total to 4 valves. The discrete valves are designed to have only two states, open and closed, while the analog valves can be throttled between 0% and 100% open. In theory, having only one out of any of the four valves open should result in no movement in the system. Opening both the Discrete and Analog extension valves should extend the arm.[2] Conversely, if the retraction valves are open, the extended Crew Access Arm should retract.[3] Opening all four should put the system into hydraulic lock. If only one valve each from extend and retract is opened, the system will either be in hydraulic lock, or the hydraulic fluid will flow right back to the source, causing no motion in the system.

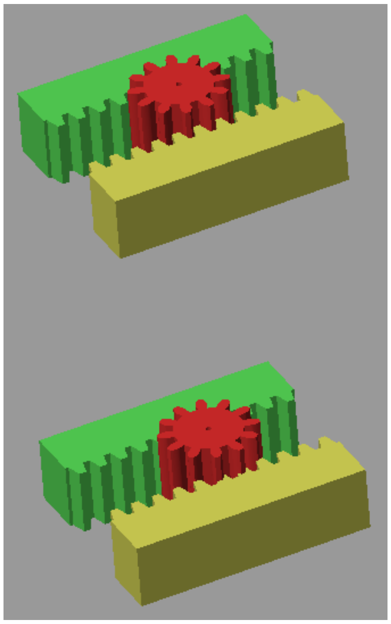

Figure 2. Crew Access Arm hinge in its locked position before movement.

At both ends of each rack is a chamber of hydraulic fluid connected to the subsystem in the previous paragraph. Between two of the racks is a pinion. The rack and pinion assembly[4] moves by the pressure difference created between the two chambers. When a rack moves, it sends the rack on the other side of the pinion to which it is attached the other direction. Figure 2 and Figure 3 show the assembly created as a visual representation in MatLab in its locked and fully extended positions, respectively.

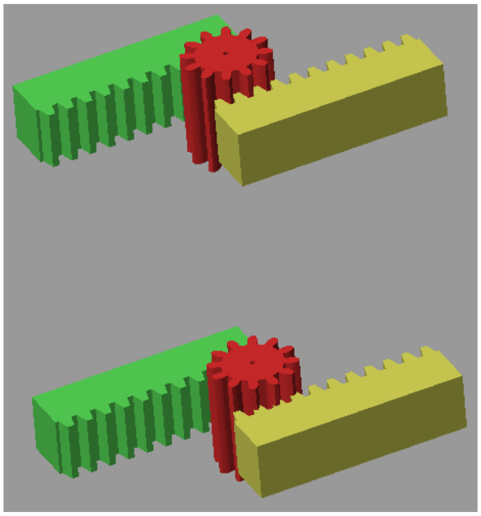

Figure 3. Crew Access Arm hinge in its fully extended position after movement.

The arm for the current model is to be assumed already either fully extended or fully retracted when starting the simulation, even though visually the center of the racks are aligned with the center of the pinions at start. Though it is possible to adjust the model so it visually starts at one of the extremes, a script would

[2] Assuming the retraction valves are closed and the Arm is fully retracted.

[3] Assuming the extension valves are closed and the Arm is fully extended.

[4] Or hinge.

need to be executed in order to prepare the model for actuation in the opposite direction. The script would have to change the variables within the rigid transform blocks, which dictate the starting position and movement of the racks according to the force applied by the double-acting hydraulic cylinders. Other assumptions include extension occurs for movements that result in graphs of motion[5] with plots above the x-axis, i.e., in the first quadrant, and retraction occurs for movements that result in graphs of motion below the x-axis, i.e., in the fourth quadrant.

III. Testing

A. Running the Simulations

When running a simulation in Simscape, there must be a designated solver that Simscape will use to solve the model. The task for this project was to compare the accuracy of these solvers. The recommended variable-step solvers, suggested in MatLab's webinar Real-Time Simulation of Physical Systems Using Simscape, are ode15s and ode23t; for fixed-step, the recommended solver is ode14x.

The model tested is very stiff[6], which is exactly what these solvers are designed to handle. Since the specific motion is not important and measured angular velocity and acceleration for each run of the simulation are simply derivatives of the angle traveled by the arm, focus will be on comparing[7] graphs of angular distance traveled by the arm.

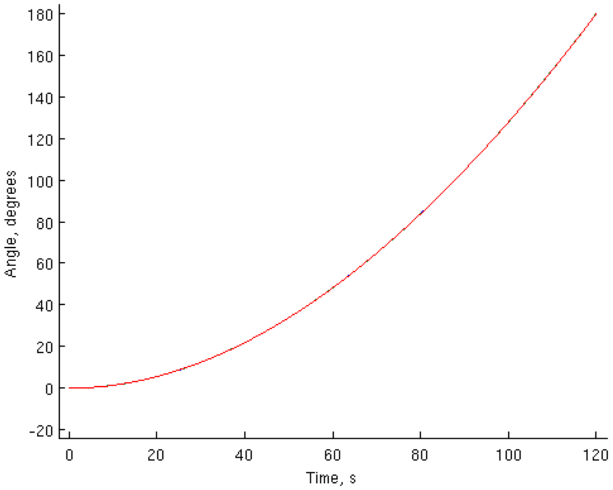

Figure 4. Position of the Crew Access Arm. *The model has no friction, so the only parameters the motion relies on are the forces created by a change in pressure between the the chambers of a rack, inertial mass, and the damping coefficent. Damping of the system causes the curve to straighten, even though the valves*

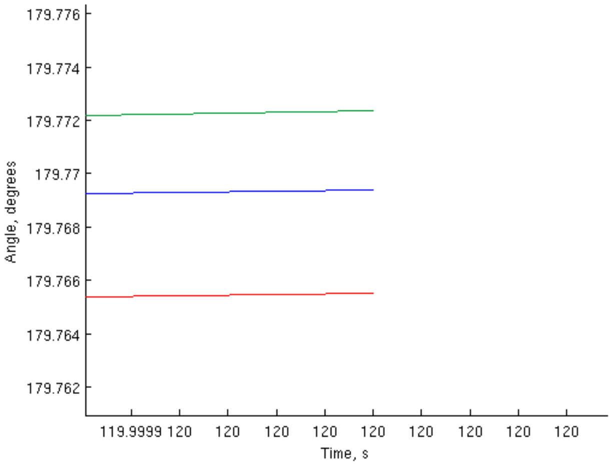

Figure 5. Magnified view of the Fig. 4 at time = 120s. *Close-up is at 120s because that is where the differences between the data of the solvers is the largest.*

Not only does a solver have to be selected, but there are also numerous Model Configuration Parameters to tune. After testing the effects of these parameters, I have come to a conclusion that, for the Crew Access Arm model, the following parameters do not affect accuracy by a significant amount[8]:

1) ode15s: Solver Jacobian method, Shape preservation, Maximum Order, Solver reset method, Number of consecutive minimum steps

2) ode23t: Solver Jacobian method, Shape preservation, Solver reset method, Number of consecutive minimum steps

3) ode14x: Solver Jacobian method, extrapolation order, Number Newton's iterations

After testing the parameters, the next step was to compare the results from the different

[5] Position, velocity, and, acceleration.

[6] Stiff models have very few degrees of freedom. In MatLab, a stiff model is very large, or difficult for Simscape to solve.

[7] Taking the data from one plot and subtracting the data from the other plot to make a graph of the difference

[8] If the chosen solver's parameters are changed, the results before and after the change differ by less than 0.001°. Depending on their value, list of parameters may have an effect on the solving speed of the simulation.

solvers. Because the model does not actually have the Arm attached to the rack and pinion, the system has been adjusted to have a very large mass[9] for internally calculating inertia of the rack and pinion and very large coefficient of damping. These work together to temporarily simulate natural resistance to move of the Arm in its absence. Even with these parameters, the motion of the graphs does not depict the true motion of the CAA.

Figure 4[10] shows the angle of the Crew Access Arm in terms of degrees using all of the solvers being tested. On this graph, only the data for ode23t can be easily distinguished. This is because the data for the ode15s and ode14x simulations occur so close to the data for ode23t, that it overlaps the both ode15s and ode14x. This shows that, if the motion of the Arm was observed while running the course depicted by these results, there will be no visual difference in the angle traveled, nor its speed to get to its end location. To see how accurate the fixed-step solver is, the data plotting setting was changed from the stair-step-like configuration to the line configuration so it would be easier to read the values of each simulation. Figure 5 shows a closer look at the final second of Fig. 4. Ode23t and ode14x have the smallest difference at less than 0.0001°; this is a percent error of 0.000115%. The largest difference occurs between the two variable-step solvers, ode23t and ode15s, at about 0.0002°.

B. Obtaining Realistic Motion

A computer model has numerous parameters to tweak in order to make the model a realistic representation of a system that is in design with plans to be manufactured. As long as the model correctly represents the connections of the hardware, then the more accurate the values of the hardware attributes, such as pipe length, chamber pressure, or weight of a gear, the more accurate the results will be in simulating the true motion of the system. Simulating the true motion of the system is important for the end product of the CAA model because it will be converted to C or C++ and combined with other systems in order to simulate full launches.

Originally, when the CAA model simulation was executed, the Arm rotated the full ~195° in merely 5 seconds. Simulation also showed constantly increasing acceleration, meaning it never slowed down to stop. Increasing the simulation runtime would allow the Arm to swing thousands of degrees in just 60 seconds. The model is now adjusted to move 179° in 120 seconds, shown in Figure 4. This would reflect a more desired motion of the CAA.

Some parameters were correctly set under the creator of the model, but others had to be adjusted. Most parameters are now more accurate in their values. Even with more correct parameter values, the motion will still not reflect the correct motion. Ways to correct this are discussed in a later section.

Accurate motion was obtained by greatly increasing the inertial masses, and by adding friction to the system. As described previously, an increase in the inertial masses was necessary in order to simulate the existence of the effect the CAA would have upon the actuation system. Originally, the model was configured in such a way that the forces applied by the double-acting hydraulic cylinders were constant and without friction. Thus, no matter the distance traveled or the angular velocity of the CAA, the Arm would build angular speed exponentially in accordance with Newton's First Law of Motion. To correct this issue, resistance had to be added to the system.

Adding resistance, i.e., friction, proved to be a difficult task because there are only a few logical locations in the model that it can be inserted. The double-acting hydraulic cylinders only exerts a force upon the rack and is not latched to it, so making the hydraulic cylinder slow to a stop to exert no force is essential ensuring the rack does not continually increase velocity, but it will not stop the rack. This means adding resistive forces anywhere in the hydraulics subsystem will not accomplish the goal, but it is still necessary to close the valve. This means resistive forces must solely reside in the mechanics.

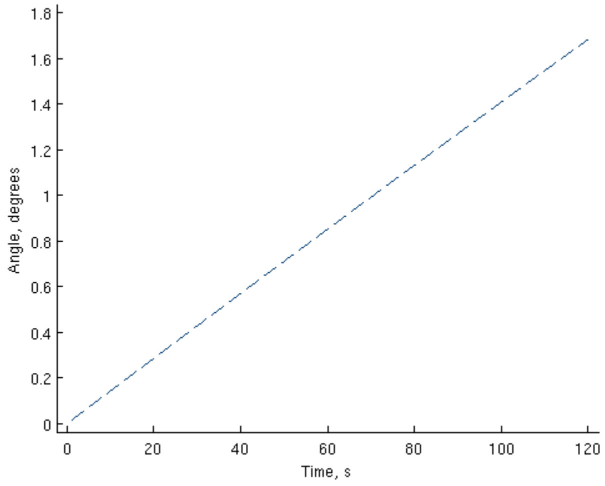

Figure 6. Simulation run without damping. *Note the y-axis has a maximum value of 1.8°.*

[9] Somewhere around 1 billion for each part slows the system down enough to have more preferred motion readings; Instead of traveling thousands of degrees in a few seconds, the model moves about 179° in 120s.

[10] For Fig. 4 and Fig. 5, ode23t is red; ode15s is green; ode14x is blue.

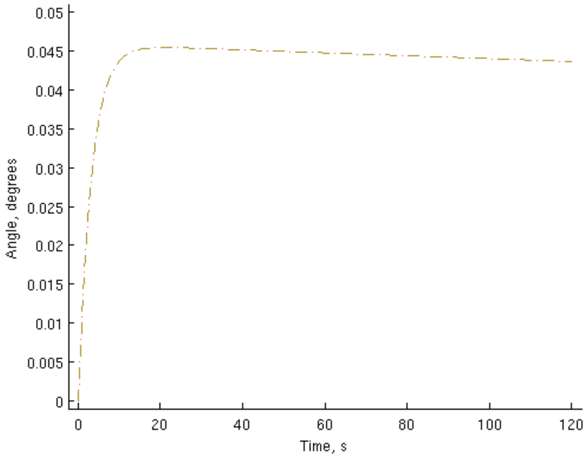

Figure 7. Simulation run with damping coefficent at 500000 lbf*ft/(rad/s). *Note the y-axis has a mimimum value of 0.05°.*

In all of the mechanics, there are four logical locations to insert friction. After doing a bit of trial-and-error, only adding a value to the damping coefficient for the revolute joint had any effect on slowing the system. Figure 6 and Figure 7 respectively show the difference between a damping coefficient existing or not existing. For these simulation runs, inertial mass has been decreased in order to more easily notice the differences between the two graphs. The biggest difference is the shape of the graph. Figure 6 continues to increase velocity even after the valve is shut off because of the pressure difference explained in a following section; Figure 7 quickly increases velocity, but after the valve is shut off, damping steadily decreases the velocity.

C. Back Pressure

Even with damping added to the system, the motions from model simulations do not reflect the expectations from various control settings. Figure 8 is a basic representation of the Actuation subsystem. The following are theoretical scenarios based on the position of valves in order to better understand what to expect from the current hydraulics configuration and explain the results of the simulations.

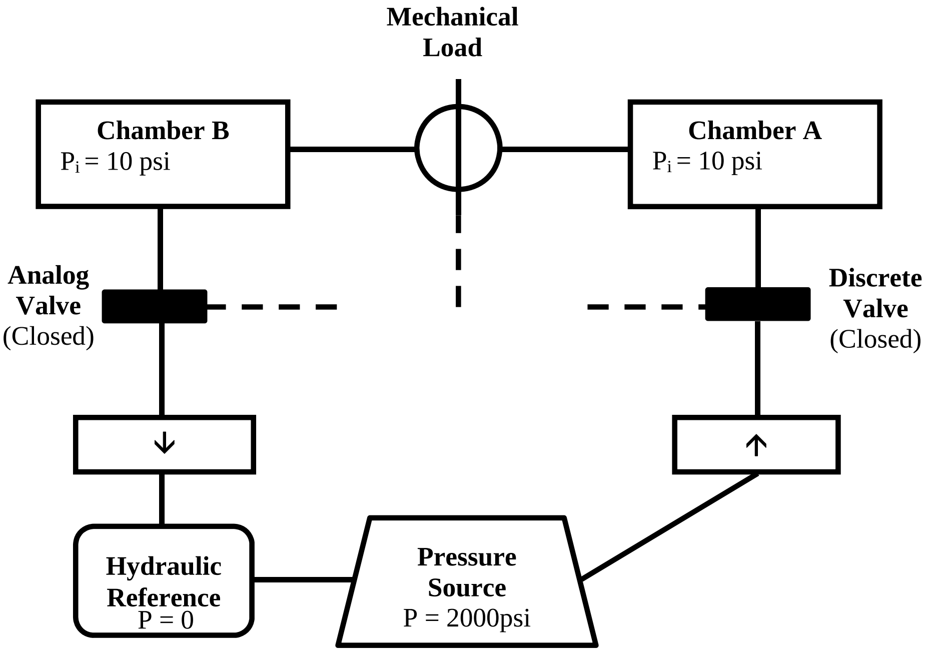

Figure 8. Simplified drawing of the CAA's hydraulics subsystem.

First, with both of the valves closed, the reservoir attempts to pump fluid to Chamber A, but cannot since the discrete valve is closed. The pressure in Chamber A is unaffected. The pressure in Chamber B is unaffected. The pressure in the Return Line is zero because of the Hydraulic Reference (0 psi). The CAA does not move, as is the expected outcome when both valves are closed. Simulations of this configuration in the model reflect the predictions stated; no movement occurs.

The second scenario is where the discrete valve is open and the analog valve is closed. The pressure in Chamber A starts at 10 psi. Because the Discrete valve is open, the reservoir fills Chamber A with fluid, resulting in the pressure on the right side of the Mechanical Load being 2000 psi. The Analog valve is closed, so the fluid in Chamber B has nowhere to go, making the pressure on the left side of the Mechanical Load 10 psi; thus, the pressure difference between the right and left sides is 1990 psi. This causes movement of the CAA. There should not be motion if only one valve is open.

Thirdly is the case where the Discrete valve is closed, but the analog valve is open. The pressure in Chamber A starts at 10 psi. Because the Discrete valve is closed, the reservoir cannot raise the pressure in Chamber A. The Analog valve is open, so the fluid in Chamber B drains to the Hydraulic reference, creating a pressure difference of 10 psi across the Mechanical Load. The pressure difference causes very small movement of the CAA. Simulations of this configuration in the model reflect the predictions stated; very small motion of the CAA occurs.

The fourth and final is when both the discrete and analog valve open. The pressure in Chamber A starts at 10 psi. With the Discrete valve open, fluid from the pressure source quickly raises the pressure to 2000 psi. With the Analog valve open, Chamber B empties to the Hydraulic Reference. The pressure difference created at the Mechanical Load is 2000 psi, causing only slightly faster movement than in the second scenario, where the pressure difference is 1990 psi.

The resulting motions in two out of the four scenarios do not reflect the anticipated motion of the real-life CAA. Tests were conducted on changing the pressures in chambers A and B to 2000 psi would result in the correct motion with no success. Therefore, the model must be adjusted to account for these errors. Analyzing the errors suggests the solution lies in adding back-pressure to the system. Back-pressure would allow the system to accelerate much more slowly and make the CAA not move when only one of the valves is opened.

IV. Conclusion

The accuracy of the ode14x solver versus the ode15s and ode23t solvers has been confirmed by running the simulations described. There is more testing to do on other types of models, such as ones containing electronics, but the Simulation Product Group will be excited to start using Simscapes library blocks hopefully very soon. The model has been adjusted to show a more realistic motion of the CAA, because it now slows down when the valves are shut off. This is important because the model will eventually be merged with other SLS models for future testing. Back pressure must be added in order to obtain full realistic motion of the CAA. This may be a task for a future intern project.

Acknowledgments

I'd like to first and foremost thank Samantha Harris and Samantha Testa, the summer interns who built the initial model used for this project. Next, I'd like to thank Adam Ayer, who was extremely helpful, dropping what he was doing to help my co-worker and me whenever we had questions. Speaking of whom, I'd like to thank Billy J. Albritton, a helpful person who is great to work with, and is an all-around awesome friend. Last but not least, I'd like to thank Lien Moore for offering me the opportunity of a lifetime with this internship. Thank you all.

References

[1] Miller, Steve. "Real-Time Simulation of Physical Systems Using Simscape." *Simulink Webinar.* Mathworks, 18 Feb. 2010. Web. 22 Oct. 2014. http://www.mathworks.com/videos/real-time-simulation-of-physical-systems-using-simscape-81697.html

[2] Moore, R. D., "Concept of Opperations SLS ML Crew Access Arm (CAA)," K0000026962-PLN_CAA_ConOps, 2014.