Simscape Modeling of a Custom Closed-Volume Tank

Nathaniel P. Fischer
Kennedy Space Center

Major: Mechanical Engineering
KSC Spring Session
Date: 1 April 2015

Simscape Modeling of a Custom Closed-Volume Tank

Nathaniel P. Fischer[*]
University of Wisconsin-Platteville, Platteville, WI, 53818

The library for Mathworks® Simscape™ does not currently contain a model for a closed-volume fluid tank where the ullage pressure is variable. In order to model a closed-volume variable ullage pressure tank, it was necessary to consider at least two separate cases: a vertical cylinder, and a sphere. Using library components, it was possible to construct a rough model for the cylindrical tank. It was not possible to construct a model for a spherical tank, using library components, due to the variable area. It was decided that, for these cases, it would be preferable to create a custom library component to represent each case, using the Simscape language. Once completed, the components were added to models, where filling and draining the tanks could be simulated. When the models were performing as expected, it was necessary to generate code from the models and run them in Trick (a real-time simulation program). The data output from Trick was then compared to the output from Simscape and found to be within acceptable limits.

Nomenclature

P	=	pressure
k	=	specific heat ratio of gas
r	=	tank radius
d	=	diameter
A	=	area
P_0	=	initial pressure
m	=	mass
c_p	=	constant pressure specific heat
V_0	=	initial volume of fluid
V_f	=	volume of fluid at time t
V_t	=	volume of tank
A_s	=	surface area
$\mathrm{d}t$	=	time step
ρ	=	fluid density
h_0	=	initial fluid height
h_f	=	height of fluid at time t
h_t	=	height of tank
$\mathrm{d}V_f/\mathrm{d}t$	=	fluid flow rate
T_0	=	initial temperature
T	=	temperature
Q	=	heat flow
KSC	=	Kennedy Space Center
NASA	=	National Aeronautic and Space Administration
Simscape	=	MathWorks® Simscape™
CSV	=	Comma-Separated Value

[*] Simulation Intern, NE-C1, Kennedy Space Center, University of Wisconsin-Platteville.

I. Introduction

MathWorks® Simscape™ (Simscape), is modeling and simulation software that can be used to represent a variety of physical systems. The National Aeronautic and Space Administration (NASA) is investigating further use of Simscape by the Application & Simulation Software Engineering Branch, NE-C1, at Kennedy Space Center (KSC). Simscape models must be proven to be capable of running in real-time, using Trick (a real-time simulation program), and with an acceptable level of accuracy. I have been tasked with creating a functional model of a fluid tank, where the gas pressure inside of the tank will increase or decrease with a corresponding change in liquid volume. There are two major tank geometries that will be considered for this model: a vertical cylinder, and a sphere.

II. Simscape Components and Models

A. Requirements

1. *The component should have one liquid inlet/outlet, located at the bottom.*
2. *The tank pressure should vary with changes in the liquid volume.*
3. *The component should account for heat transfer between the tank and the liquid in the tank.*
4. *The component will be considered to be well insulated, so the heat transfer effects with the environment will be assumed to be negligible.*

B. Rough Model

In order to have a basis for comparison, I constructed a model that is representative of a cylindrical tank, using Simscape library components. It uses a hydraulic cylinder that is physically connected to a pneumatic cylinder through the piston rods of the cylinders. When using this model, if the dimensions of the tank need to be changed, the changes have to be incorporated into both cylinders, this is more time consuming and could introduce the possibility for errors. Additionally, the thermal port on the pneumatic cylinder is meant to interact with the gas, and I am more interested in the heat transfer between the liquid and the tank.

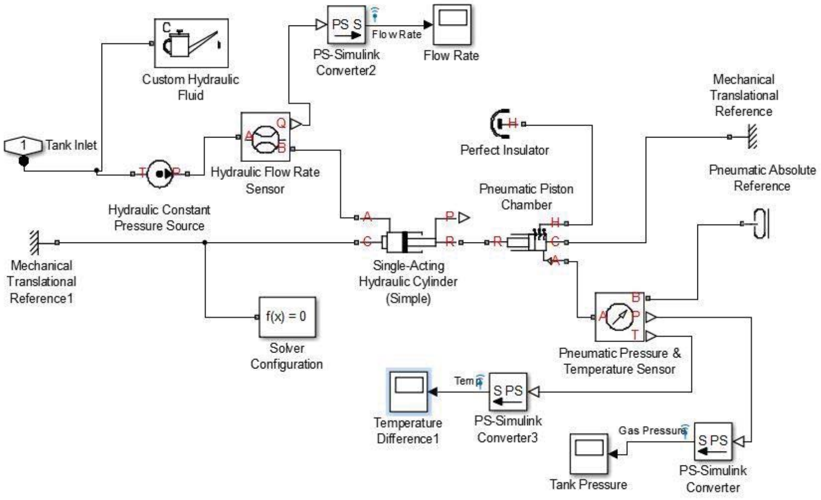


Figure 1. Image of rough Simscape model.

C. Custom Cylindrical Tank Component

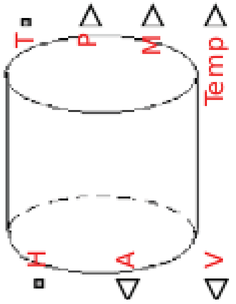


Figure 2. Image of custom Simscape cylindrical tank component.

Using the Simscape language, it is possible to modify some library components, or, to create complete custom components. In order to satisfy the requirements for the fluid tank, I decided to start from the beginning. To simplify the equations, I made certain reasonable assumptions: The gas in the tank is assumed to be an ideal gas, temperature change of the gas due to volume change (compression or expansion), is considered to be negligible, heat transfer between the gas and liquid is considered to be negligible, the tank is considered to be well insulated, change of tank volume due to pressure change is not considered. The port labels for the component require some clarification. H is the hydraulic port, where the fluid flows into or out of the tank. A is a physical signal port that outputs the contact surface area between the tank and the fluid, it is used with the custom heat transfer components during simulation. V is a physical signal port that outputs the fluid volume. T is a thermal port that is used to model the heat transfer, between the tank and the fluid in the tank. P is a physical signal port that outputs the pressure in the tank. M is a physical signal port that outputs the fluid mass, and it is used with the custom thermal mass component to model heat transfer. Temp is a physical signal port that outputs the fluid temperature.

D. Cylindrical Tank Equations

The following equations were used to create the cylindrical tank component.

$$A = \pi r^2 \tag{1}$$

$$V_{t,f,0} = A h_{t,f,0} \tag{2}$$

$$\frac{\mathrm{d}V_f}{\mathrm{d}t} = A\frac{\mathrm{d}h_f}{\mathrm{d}t} \tag{3}$$

$$P = P_0\left(\frac{V_t - V_0}{V_t - V_f}\right)^k \tag{4}$$

$$m = \rho V_f \tag{5}$$

$$A_s = A + \pi d h_f \tag{6}$$

$$Q = m c_p (T - T_0) \tag{7}$$

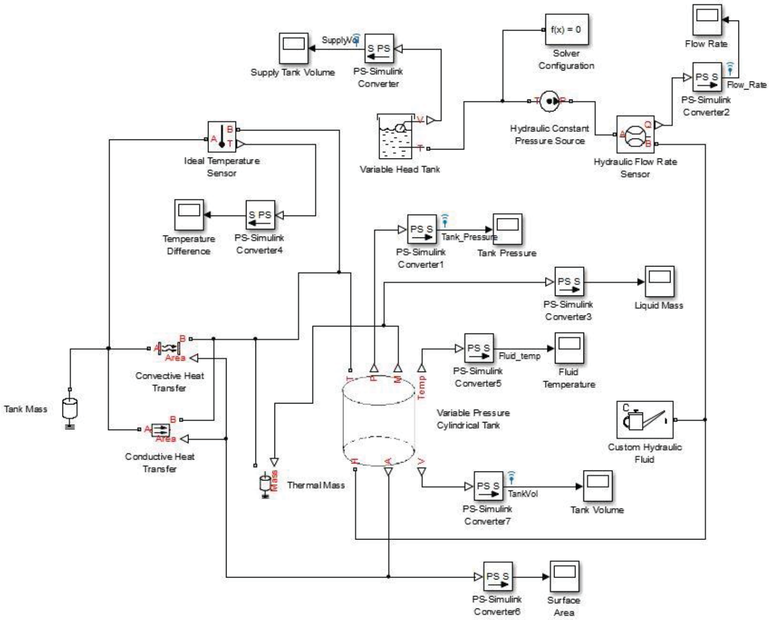


Figure 3. Image of custom cylindrical tank model (spherical model similar).

E. Testing Custom Tank Models

In order to verify the functionality of the custom tank components, they were added to a model, where the interaction with other components could be tested and verified. In the model, the "Variable Head Tank" supplies the fluid, and the "Hydraulic Constant Pressure Source" provides the pressure to initiate flow. Heat transfer in the tank component is modeled through the thermal port "T", the thermal mass associated with the fluid in the tank is represented by "Thermal Mass", and the thermal mass of the tank is represented by "Tank Mass". The heat transfer between the tank and the fluid is represented by the "Convective Heat Transfer" and "Conductive Heat Transfer" components.

F. Custom Spherical Tank Component

The assumptions for the spherical tank are the same as the assumptions listed for the cylindrical tank. In order to represent the relationship between the area change and the height change of the fluid, it was necessary to develop this component from the beginning, using the Simscape language. The inputs and outputs for this tank are similar to those found on the cylindrical tank, described in section (II.C).

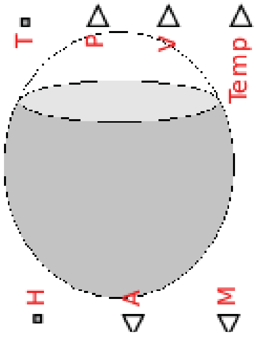


Figure 4. Image of custom spherical tank Simscape component.

G. Spherical Tank Equations

The following equations were used to create the spherical tank component. The derivations for the volume and area equations are in the Appendix.

$$A = \pi\left(2rh_f - {h_f}^2\right) \tag{8}$$

$$V_t = \frac{4}{3}\pi r^3 \tag{9}$$

$$V_0 = \pi {h_0}^2 \frac{3r - h_0}{3}$$ (10)[1]

$$P = P_0\left(\frac{V_t - V_0}{V_t - V_f}\right)^k \tag{11}$$

$$m = \rho V_f \tag{12}$$

$$V_f = V_0 + \pi(r\left({h_f}^2 - {h_0}^2\right) - \frac{{h_f}^3}{3} + \frac{{h_0}^3}{3}$$ (13)[2]

$$A_s = 4\pi r^2 - 2\pi r\left(2r - h_f\right) \tag{14}$$

$$\frac{\mathrm{d}V_f}{\mathrm{d}t} = \frac{\mathrm{d}m}{\mathrm{d}t}/\rho \tag{15}$$

$$Q = mc_p(T - T_0) \tag{16}$$

H. Custom Library

After writing the code for the components in Simscape language, it is necessary to save them in a separate folder. In order to be considered as a Simscape library, this folder must begin with a “+”, so, an example would be: +Folder_name, I would suggest adding this directory to the Matlab path. From the Matlab command line, the command to compile and build the components in the library folder is: ssc_build Folder_name. If there are no errors in the Simscape code files, the library should resemble Fig. 3. The default image for a custom Simscape component is a labeled box with the corresponding inputs and outputs. If a different image is desirable, it is relatively simple to add. Save a copy of the image in the “+Folder_name” directory. The file name for the image has to match the file name of the component, for example: for the Simscape file, “custom_file.ssc” the image would need to be “custom_file.img”, where img is a supported file format for the image. The supported image file formats are: jpg, bmp, and png[3].

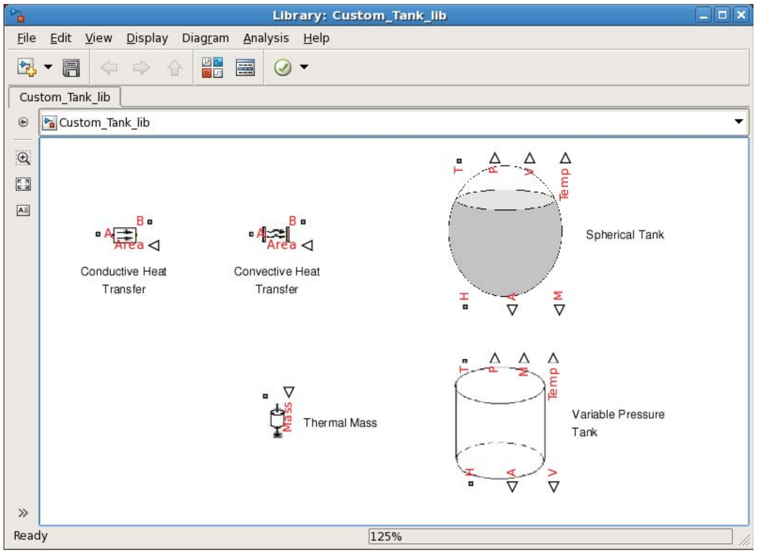


Figure 5. Image of custom component library.

I. Additional Custom Components

When attempting to model the heat transfer between the fluid and the tank walls, I found that the standard library components were missing some necessary functionality. In the “Thermal Mass” standard library component, the mass is a fixed parameter that cannot easily be adjusted during simulation. For the tank components, the fluid mass is changing continuously with fluid volume. In order to consider the change in mass, I modified the standard “Thermal Mass” component to accept the value for mass as a physical signal input. Also, both tank components were modified to output the fluid mass as a physical signal. A similar problem was encountered with the heat transfer library components (Conductive Heat Transfer and Convective Heat Transfer), where the area is a fixed parameter. The surface area of the liquid that contacts the tank changes continuously with fluid volume as well. A similar solution was employed to account for the changing area, the heat transfer components were modified to accept the area as a physical signal input, and the tank components were modified to output the area as a physical signal.

III. Verification of Models

A. Cylindrical Tank Comparison

The cylindrical tank component was incorporated into a model, similar to the model shown in Fig. 3. The simulation was run, and the results were recorded. This procedure was repeated for the rough model, shown in Fig. 1. Next, some of the results from the two simulations were plotted and compared, shown in Fig. 6 and 7.

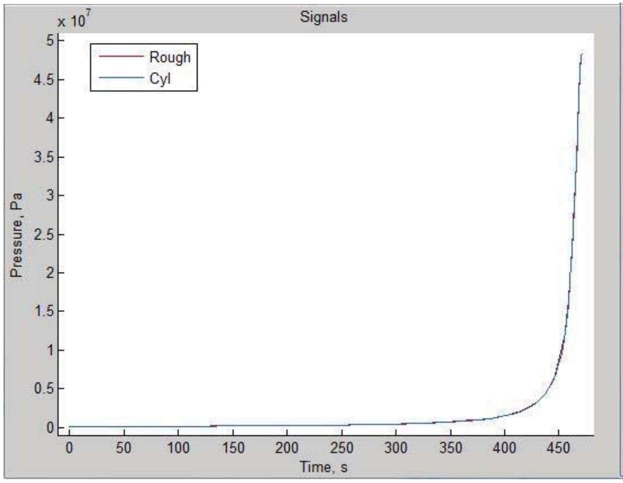


Figure 6. Pressure as a function of time.

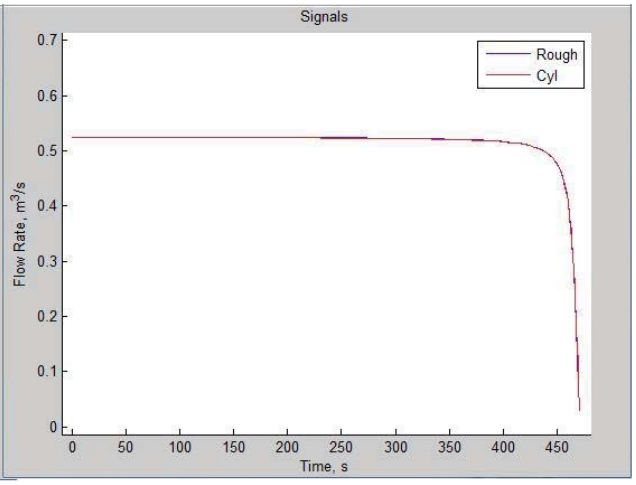


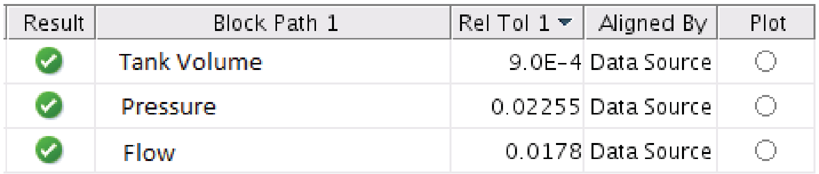
Figure 7. Fluid flow rate as a function of time.

I then compared the two simulations, using the “Simulation Data Inspector” in Simscape, the results are shown in Fig. 8. The two runs match each other closely, the Tank Volume curves are within 0.09%, the Pressure curves are within 2.26%, and the Flow curves are within 1.78%.

B. Code Generation

After comparing the two models and finding the results to be satisfactory, the next step is to generate code from the Simscape model, so it can be run in the Trick real-time simulation environment. The results from the Trick simulation will then be compared to the results from the Simscape simulation, to determine if any variation exists between the two simulations. Also, it is important that the model is “real-time capable”, meaning that the computations performed during one second of simulation time must take less than one second of real-time to complete.

Certain configuration parameters in Simscape must be set before code generation can be done. I saved a copy of the model onto the “KSC Template”, where the parameters were already set to the required values. I then ran the simulation to verify that it would compile and run with the new settings. In the Simscape model, I replaced all of the Scopes with output ports, and used Pulse Generators to generate an input signal. Using Matlab Coder, as a test, I was able to successfully generate C++ code from the model. In order to



Result	Block Path 1	Rel Tol 1 ▾	Aligned By	Plot
✓	Tank Volume	9.0E-4	Data Source	○
✓	Pressure	0.02255	Data Source	○
✓	Flow	0.0178	Data Source	○

Figure 8. Comparison of simulation runs.

generate all of the necessary files to run the simulation in Trick, a series of steps must be taken. First, I ran sim_mc_buildScript with the appropriate command line arguments and file paths, from a terminal. If there were any errors in the build, I addressed them, and repeated the command until the build was successful. Next, I created a data_logging.txt file with input and output names that matched the model inputs and outputs, using the Subsystem_Test_Bench Excel template. I then saved a copy of the data_logging.txt file into file_path/SIM/data. Next, I included the data_logging.txt file in the header of file_path/SIM/RUN/input. I ran the sim_tcid_buildScript with the appropriate command line arguments from a terminal, if any errors appeared, I addressed the errors and repeated until the script ran successfully. I verified that an S_main_operatingsystem.exe file existed in the file_path/SIM directory.

To run the simulation in Trick, I changed directory to file_path/SIM. The next step was to run the S_main_operatingsystem.exe file with argument RUN/input. Using Ctrl Z to stop the command, and then typed “bg” (to move the program to the background). I opened the Trick simulation control by typing: sim_control.tcl –m machine host name –port number. The Trick control panel opened with the simulation paused. I ran the simulation for a sufficient time to gather the necessary data. Once the simulation was complete, there was a log_filename.trk file in the file_path/SIM/RUN directory. I converted this file to a comma-separated value (CSV) file, using the command trk2csv with the appropriate arguments[4]. Next, I imported the CSV data and compared it to data that was saved from a Simscape simulation, using Matlab to plot the results.

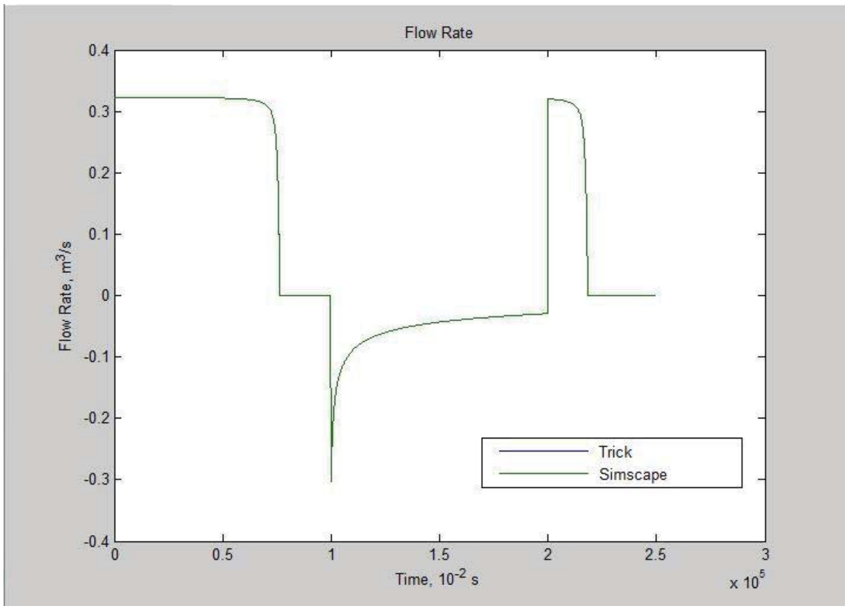


Figure 9. Comparison of Simscape data and Trick data.

As shown in Fig. 9, the data from the Trick simulation overlays the plot from the Simscape simulation exactly. The results shown are from the cylindrical tank model, the results from the spherical tank model were similar when the simulations were compared.

IV. Conclusion

When comparing the custom cylindrical tank to the one provided Simscape library components, the results are promising. Unfortunately, I have no rough Simscape model to compare the custom spherical tank model against, to verify the accuracy of the model, other than the equations that define the tank geometry, though the model performs as expected. When the need arises, The use of the Simscape language to create custom components, or to extend existing Simscape library components is straight forward, feasible and recommended.

Appendix

Derivation of Spherical Tank equations:

$$r(h)^2 = R^2 - (R - h)^2,$$

Or

$$r(h)^2 = 2Rh - h^2$$

Where r(h) is the radius of the fluid level at height h and R is the tank radius.

So, the volume would be:

$$V = \int A\, dh,$$

Where $A = \pi * r(h)^2$

$$V = \int_{h_0}^{h} (2\pi Rh - \pi h^2)\, dh + \int_{0}^{h_0} (2\pi Rh - \pi h^2)\, dh$$

$$V = \pi \left\{ R(h^2 - h_0^2) - \frac{h^3 - h_0^3}{3} + Rh_0^2 - \frac{h_0^3}{3} \right\}$$

Acknowledgments

I would like to thank my mentor, Lien Moore, for offering me this amazing, once in a lifetime opportunity. Also, I would like to thank Adam Ayer, who was very helpful in getting me set up with accounts and initial training. Additionally, Mike Waldorf was very helpful when I had issues with code generation and Trick questions. Last but not least, I would like to thank my family for their support and encouragement. This entire experience has been challenging in all of the right ways, and I feel as though I have learned so much in such a short time.

References

[1] “Fluid Level in a Spherical Tank,” URL: http://spacemath.gsfc.nasa.gov/Calculus/5Page45.pdf [cited 2 April 2015].

[2] “Fluid Level in a Spherical Tank,” URL: http://spacemath.gsfc.nasa.gov/Calculus/5Page45.pdf [cited 2 April 2015].

[3] “Build custom library from collection of Simscape files,” URL: http://www.mathworks.com/help/physmod/simscape/ref/ssc_build.html [cited 2 April 2015].

[4] “Trick User’s Guide”, November 1, 2009.