High Accuracy Liquid Propellant Slosh Predictions Using an Integrated CFD and Controls Analysis Interface

Brandon Marsell[*]
a.i. solutions, Launch Services Program, Kennedy Space Center, FL 32899

David Griffin[†]
a.i. solutions, Launch Services Program, Kennedy Space Center, FL 32899

Dr. Paul Schallhorn[‡], Jacob Roth[§]
NASA, Launch Services Program, Kennedy Space Center, FL 32899

Coupling computational fluid dynamics (CFD) with a controls analysis tool elegantly allows for high accuracy predictions of the interaction between sloshing liquid propellants and the control system of a launch vehicle. Instead of relying on mechanical analogs which are not valid during all stages of flight, this method allows for a direct link between the vehicle dynamic environments calculated by the solver in the controls analysis tool to the fluid flow equations solved by the CFD code. This paper describes such a coupling methodology, presents the results of a series of test cases, and compares said results against equivalent results from extensively validated tools. The coupling methodology, described herein, has proven to be highly accurate in a variety of different cases.

I. Introduction

The effects of sloshing propellant on the control system of a launch vehicle are very important and cannot typically be ignored. Many different tools have been developed to study this interaction and attempt to predict the magnitude of the slosh effects. One such tool is called the Universal Controls Analysis Tool (UCAT). This tool is a Mathworks MATLAB based model developed and used extensively by the Launch Services Program (LSP) at the Kennedy Space Center (KSC) to analyze the performance of many launch vehicles on a mission by mission basis. It has the capability of including a multitude of dynamics such as structural bending, actuators, winds, gravity, aerodynamics, sensors, and propellant slosh. The Launch Services Program has developed a version of this tool for most vehicles and configurations (number of boosters, stages, and engines) used by the program. Since most of the details of these models are proprietary in nature, this paper will focus on a generic version of the tool with no reference to any proprietary information.

The computational fluid dynamics (CFD) solver used in this study is Flow3D distributed by Flowscience inc. It is a robust solver that is widely used in the aerospace industry.[3] It is a leader in free surface flow simulations such as water in ducts, liquid metal castings, and liquid propellant tank slosh. This code is used extensively by the Launch Services Program to analyze fluid flow in propellant tanks and address any concerns with propellant orientation within the tanks during any point in the mission. It is also used to develop slosh parameters for the mechanical analogs used in some control system simulations and is capable of simulating many different fluids, from cryogens like liquid oxygen, to storable propellants like hydrazine.

The motivation for this study arose from the need for a more robust and accurate method of determining the effects of propellant slosh on control systems with the goal of accurately predicting the location of the propellants within the tanks. In the past, this information was needed to evaluate the effects of propellant slosh during abrupt vehicle maneuvers. Issues such as ullage collapse, wall wetting, thermal conditioning, liquid venting, control stability, and cryogenic boil-off have all been of concern to a multitude of missions. The problem is that during these abrupt vehicle maneuvers, the simple pendulum analog used to model propellant effects is no longer valid. By

---

[*] Fluid Systems Engineer, Launch Services Program, Mail Code: AIS-20, Kennedy Space Center, FL 32899
[†] Control Systems Engineer, Launch Services Program, Mail Code: AIS-20, Kennedy Space Center, FL 32899
[‡] Branch Chief, Launch Services Program, Mail Code: VA-H3, Kennedy Space Center, FL 32899
[§] Fluid Systems Engineer, Launch Services Program, Mail Code: VA-H3, Kennedy Space Center, FL 32899

replacing the pendulum analog with a full CFD solution, the model remains valid and will not only produce vehicle control response effects, but also produce accurate propellant location information. From this, all of the afore mentioned issues may be resolved.

This analysis was previously accomplished using a two step process. First, the controls systems engineer determines the appropriate acceleration environment based on a rigid body assumption. These accelerations are then sent to the CFD engineer who uses them as an input to a CFD simulation that determines the fluid location. The problem with this method is that it is a loosely coupled system. The fluid forces are not affecting the control system. This can lead to large errors in the prediction of fluid location. The new coupled method described in this paper allows both solvers to run simultaneously. This allows the fluid to affect the control system and the control system to affect the fluid, which leads to much higher accuracy in the final result.

II. Pendulum Analog

Typically, control system analysis tools that include propellant slosh effects do so by means of a simplified mechanical analog. One such mechanical analog is illustrated in the Figure 1. The pendulum analog is a simple system that models propellant slosh with the use of two masses. Previous studies have shown that sloshing fluid inside a container can be split into two parts. The first is referred to as the fixed mass. This is a non-moving mass which models the portion of the propellant that is relatively stationary near the bottom of the tank. The "slosh mass" is attached to the tank as a pendulum representing the moving portion of the fluid near the free surface.[1,2,5,6]

By including a system like the pendulum analog in the control system model it is possible to simulate the effects of sloshing fluid inside the tanks of a launch vehicle.[1] Since the physics of a pendulum can be fully described by using a few simple equations, it adds very little computational overhead, allowing the analyst to make many runs and adequately test the control system. This model works well but has several drawbacks. The first is that it is only valid for low amplitude slosh. If the vibration environment gets too violent and the slosh waves begin to break, the pendulum analog assumption breaks down. Another drawback is that it provides no indication of the propellant location within the tank. While the pendulum mass may be applying the correct slosh forces, there is no way to tell where the propellant is inside the tank.[1,6] This information is crucial to thermal and venting analysis among others. Another disadvantage is that the model is not appropriate for low-g environments. In this low acceleration environment, the cohesive forces due to surface tension dominate, and the fluid acts much differently. The final drawback to using the pendulum analog deals with resolving higher order slosh modes. Most pendulum slosh models use a single pendulum. This only captures the primary slosh mode and ignores the higher order modes. Though higher order modes usually have small amplitudes, a more accurate model should take these into account.

The coupling tool aims at bypassing all of this by replacing the pendulum analog with a full CFD computation. The fully integrated CFD run will take many times longer to run which may restrict its application to those cases that require high accuracy results along with propellant orientation information.[6] Though it may be slower, this new coupling tool should provide higher accuracy simulations which will yield a deeper understanding of the dynamics behind the interaction between sloshing liquids and vehicle control systems.
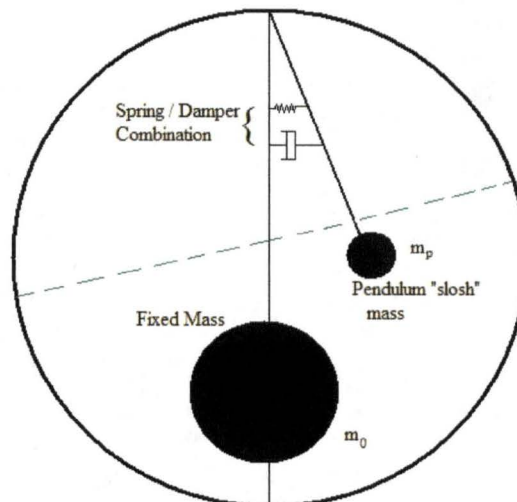

Figure 1. Mechanical Pendulum Analog

III. Methodology

The objective of this study was to develop a software analysis tool that couples a fluid flow CFD solver (Flow-3D) with a controls model (UCAT) in order to produce more accurate predictions of vehicle dynamics. By setting up regular (or dynamic) communication intervals, both programs can interchange information at specified times and accurately predict the effects of fluid motion on a vehicle. Below is a flow chart describing the communication exchange between the two programs.
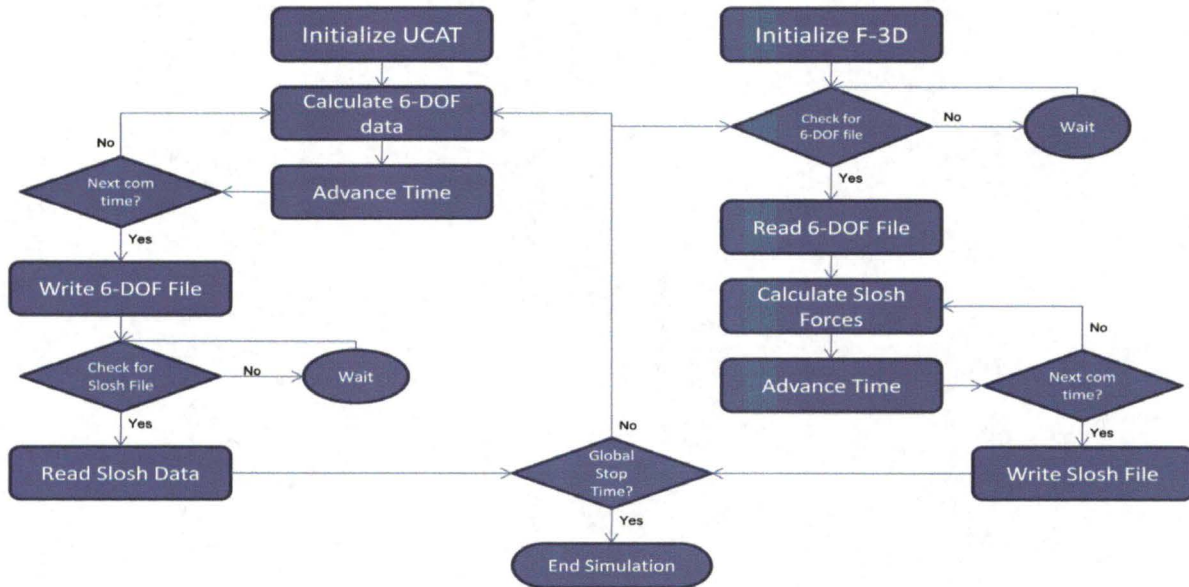

Figure 2. Methodology Flow Chart

The idea is to initialize both programs with information from a common, known starting point (settled propellant) and allow them to progress simultaneously using a time domain that is discretized into small time-steps, while communicating only at pre-determined time intervals. So as to not require communication at every single time-step (though this option is available), each individual program is allowed to maintain a separate overall time-step, as long as they eventually reach the next communication point. For example, if the communication interval is set to two seconds, each program will use its own time-step and ensure that they each hit the two second mark exactly. At that point they exchange information and carry on to the next communication point (in this case at 4 seconds). Allowing each program to control its own internal time-step allows each solver to use the most appropriate time-step for its specific job and therefore avoid extra computation. Since the time-step for the CFD side is usually one to two orders of magnitude smaller than the time-step for UCAT, this dual time-stepping technique allows the CFD code to run at the smaller time-step while UCAT can use a much larger one.

At the very beginning of the run both programs have been initialized. Following code initialization, UCAT calculates the appropriate acceleration environment and advances in time. At the same time, Flow-3D begins searching for the appropriate file containing the acceleration environment. When UCAT reaches the communication time, it writes out this file and starts looking for the appropriate communication file from Flow-3D. Once UCAT finishes writing the file, Flow-3D reads in the accelerations and applies them to the fluids. Flow-3D then calculates all of the forces on the walls of the tank and writes them out to a file. This allows UCAT to continue and the cycle to repeat.

As indicated in the paragraphs above, the communication between the two programs is achieved by means of a communication file. These files are time-stamped text files that contain all of the necessary information for each part to continue. The format of these files is very specific and is outlined in the table below. The Flow-3D file must contain the forces and moments caused by the fluid on the tank walls and are referenced to a predetermined point on the vehicle. This point can be anywhere on the vehicle but care must be taken to ensure that both programs reference the same point. The very first number in the communication file is the current time. This is used to verify that the file was written at the appropriate communication time point. The next value in the file is called the "max wait time." This designates the maximum amount of time (in seconds) to spend looking for the next communication file.

This is merely a "catch" to prevent an infinite loop in the case of a failure by any part of either code. The next six values correspond to the six degrees of freedom force and moment computations. The final value in the file sets the communication interval for use when a dynamic communication interval is desired. The UCAT communication file is very similar except instead of writing forces and moments, it will write out linear and angular accelerations for use by Flow-3D.

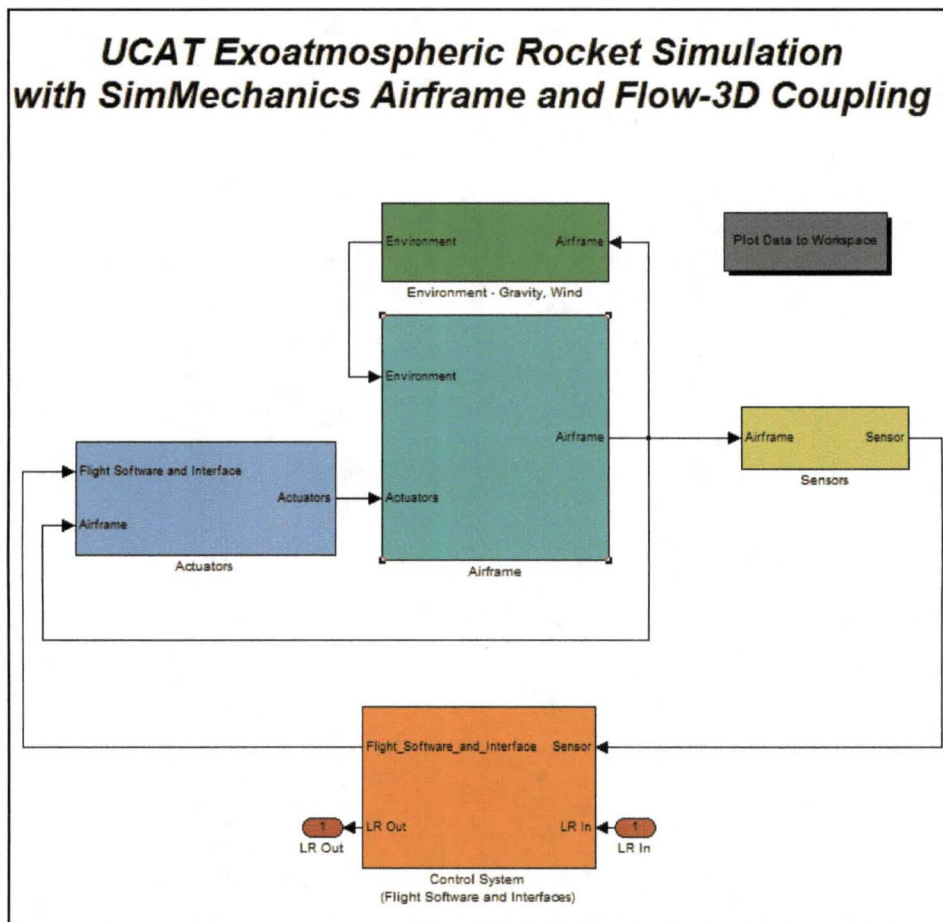| FLOW-3D File	UCAT File
Time	Time
Max wait time	Max wait time
Force in X direction	Acceleration in the X direction
Force in Y direction	Acceleration in the Y direction
Force in Z direction	Acceleration in the Z direction
Moment about X	Angular acceleration about X
Moment about Y	Angular acceleration about Y
Moment about Z	Angular acceleration about Z
Communication interval	Communication interval

III. UCAT

Figure 3. UCAT Top Level Block Diagram

The Universal Controls Analysis Tool was developed by the controls group in the Launch Services Program at the Kennedy Space Center to evaluate the performance of many different launch vehicles. The generic version used for this study is applied to a hypothetical two engine upper stage vehicle. The tool is written using the Mathworks MATLAB Simulink environment. Figure 3 depicts the top level of the UCAT Simulink model.[4] This top level block diagram is the same in any version of UCAT. The launch vehicle dynamics are all included inside the "Airframe" block. The inputs to the block are environment (wind, gravity, etc...) and actuators (actuator position information). The Airframe block outputs the position, orientation, and acceleration of the vehicle (in various coordinate systems). This output signal is processed by the "Sensors" block which models the onboard INU. The output from the "Sensors" block is then routed to the Control System block (orange). This block models the autopilot logic that calculates the actuator commands to maintain the vehicle on its specified trajectory. Finally, these actuator commands are sent to the "Actuators" block which calculates the engine gimbal deflections that get passed back to the airframe block which is now ready for the next time-step.

Figure 4. Airframe Block

Figure 4 illustrates the inner workings of the "Airframe" block. This block consists of two main sub-blocks. One of these blocks is the "SimMechanics Airframe" block which is the one responsible for keeping track of all of the vehicle dynamics. It tracks all of the mass properties and reference coordinate frames, and calculates rigid body motion of the launch vehicle. The other block is named the "Flow-3D Coupling" block. As its name implies, this block contains all of the mechanisms that allow UCAT to communicate with Flow-3D. Though the communication process involves reading and writing formatted text files and applying transfer functions, the basic function of this block is to read in vehicle accelerations and output the corresponding slosh forces and moments.

The communication between Flow-3D and UCAT is accomplished using text files. Sequentially named text files contain the necessary information for each part to run properly. This file read/write process is done using a MATLAB function that is called from within the Flow-3D Coupling block. The MATLAB function takes in a 9 element vector corresponding to the format described above and writes the appropriate information to a text file. The function then falls into a while loop where it waits for the next file to be written by Flow-3D. Once the new file is read in by the MATLAB function, a series of transfer functions interface the Flow-3D data into the rest of UCAT. Without these transfer functions, the feedback loop becomes unstable and the solver diverges. The Flow-3D coupling block showing the MATLAB function call and transfer functions is illustrated in Figure 5.
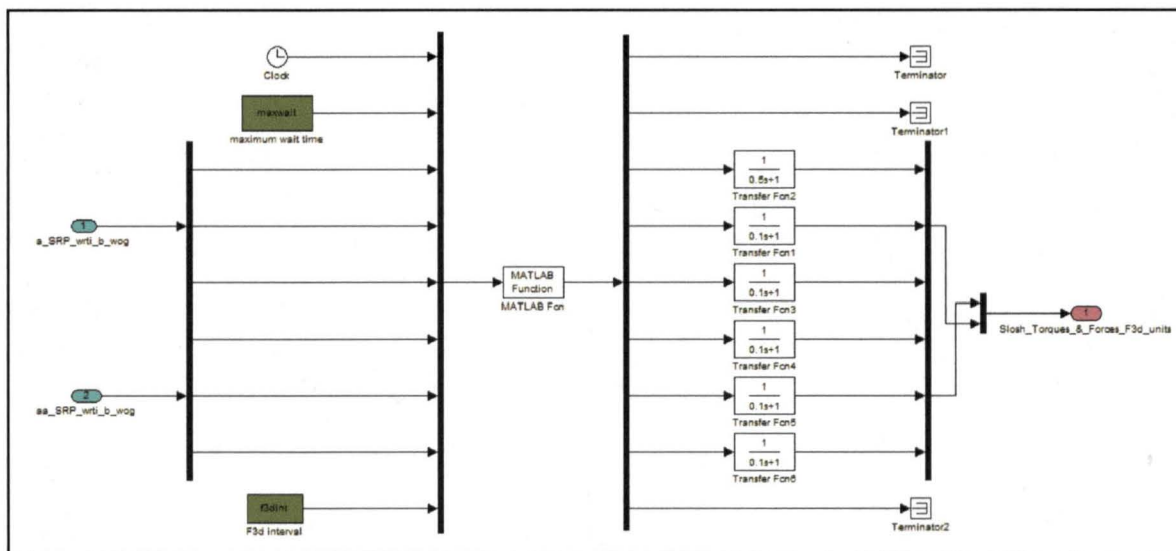
Figure 5. Flow-3D Coupling Block

The Flow-3D block format not only makes the workflow easy to follow, but also allows for the coupling block to be used in any version of the program. The "Flow-3D Coupling" block is the heart of the communication interface between both programs and it can be used in any Simulink based model. One of the more challenging aspects of this study was the need to modify the current version of UCAT to remove variable mass properties capabilities. Many of the advanced functions in UCAT are aimed at trying to account for the time varying mass properties as the propellant is depleted. The Flow-3D coupled version no longer needs to calculate these values in UCAT. They are all accounted for within Flow-3D and transferred to UCAT in the form of forces and moments applied at a point. This allows the coupling interface tool to accurately predict vehicle dynamics for any vehicle and for any mission.

IV. FLOW-3D

Flow-3D is an industry leading commercial CFD tool that specializes in free surface flows. It features both explicit and implicit solvers, a multitude of turbulence models, non-inertial reference frame capabilities, and a unique volume of fluid (VOF) boundary treatment option. This VOF treatment is unique to Flow-3D and uses a pressure boundary at the fluid to gas interface. This allows the solver to ignore any cells that contain gas and only solve the fluid flow equation in the regions containing fluid. This greatly reduces computation time for any given problem.

Flow-3D is primarily written in the FORTRAN programming language. Flow Science inc. provides access to various subroutines in the code that allow for user customization. These subroutines are mostly meant to allow for the use of nonstandard boundary conditions, new material properties, external forces, etc. In our case, the subroutines are used to construct the communication interface. A combination of two subroutines allow the code to read in the time-stamped text file written by UCAT containing 6 DOF acceleration data and write out the next communication file. For example: at time = 1.5s there is supposed to be an information exchange between the programs. When Flow-3D reaches time=1.5 s. it will fall into a loop and start looking for a file named 'UCAT1000150.dat' This file will contain all of the 6 DOF acceleration information from UCAT. This subroutine runs at the end of every time-step.

V. Verification and Validation

The coupled fluids/controls system problem is inherently difficult to validate. Since most of the slosh effects of interest only happen during long periods of microgravity, there is very little data available to compare the model results to. The ideal experiment would have a video camera, or propellant mapping device that allows for a one to one comparison with the CFD propellant location output as a function of time. Since this type of data is not yet available, it was decided to use a well validated pendulum analog to compare against. As long as the slosh amplitudes remain fairly low, and there are no breaking waves, the pendulum analog is appropriate.

Several test cases were devised to test the coupling tool. The first test case applies a steady 100% thrust to a generic upper stage vehicle with two liquid propellant tanks at a 50% fill level. After 5 seconds, the simulation imparts a sudden impulse in an off axis (z) direction (Figure 6). This off axis push will impart energy to the fluid and cause it to start sloshing inside the tank. This perturbation should be sensed by the vehicle which will adjust the thrust vector (by moving the engine gimbal) in response. The linear and angular acceleration of the vehicle's center of mass was tracked throughout the entire scenario. The exact same scenario was applied to both a pendulum analog and fully coupled CFD/UCAT run.

The figure below shows the results from a fully coupled run with an off axis push. There is excellent agreement between the pendulum model (green) and the fully coupled model (blue). This is to be expected because these are the conditions under which the pendulum model has been shown to be accurate. A push along the positive z axis, above the vehicle center of mass, will produce a positive acceleration in the z direction and a negative angular acceleration about the y axis (into the page in Figure 6). Also shown, is the gimbal angle for the engine nozzle. As evident in the graph, the flight computer on-board the vehicle detects the push and immediately commands the nozzle to respond to the disturbance. After the initial push, the sinusoidal signal is the

Figure 6. Test Case Free Body Diagram

force response from the sloshing fluid within the tank. Finally, the slosh is also evident in the axial acceleration signal. The initial push not only impacts the lateral acceleration of the vehicle, but the fore and aft shift in fluid center of gravity also causes the axial acceleration to slightly fluctuate.

An interesting feature of the coupling method is the initialization. As seen in the first second of the axial acceleration plot, there is a small amount of time needed for the solution to converge to the proper initialization point. This is caused by the transfer functions used to couple the CFD and UCAT dynamics. The transfer functions delay dynamic information and it takes a small amount of time for them to reach steady state. This can be fixed with proper initialization of the transfer function states.
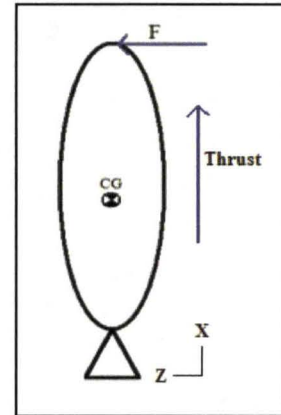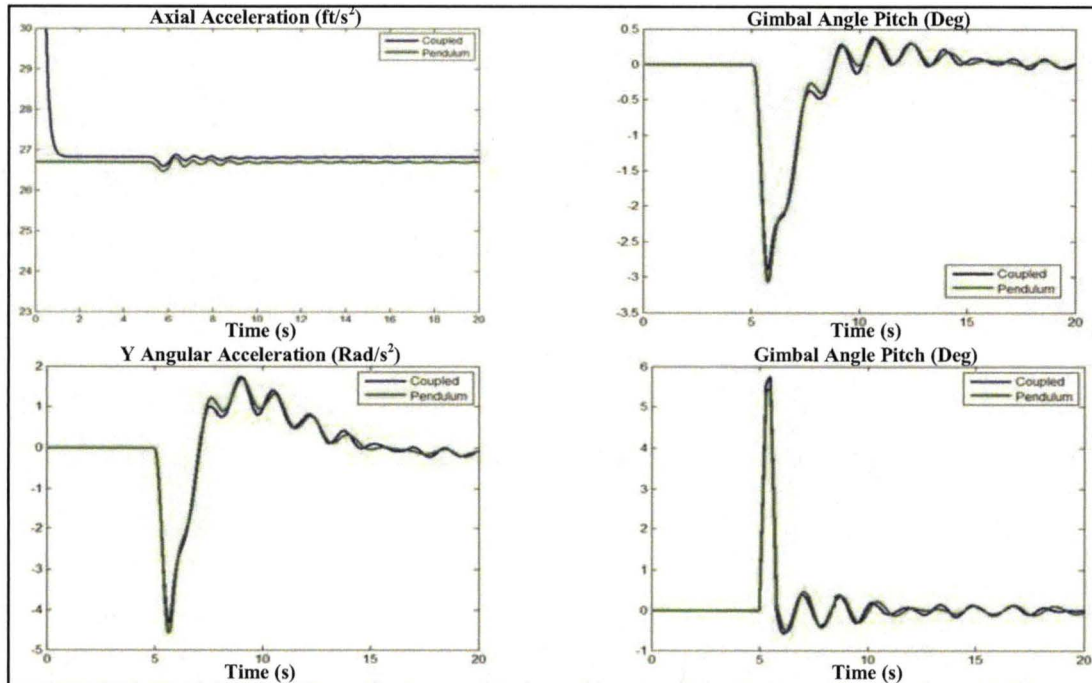

Figure 7. Simulation Results Comparison

The final test scenario is very similar to the one illustrated in Figure 7. The only difference is that for this simulation the propellants from both tanks in the vehicle are being depleted. This is made obvious by the constant increase in axial acceleration. Since the engines are assumed to produce a constant amount of thrust, the reduced

vehicle mass as the propellants are being depleted will result in a constant rise in acceleration. This effect is properly captured by both the coupled and pendulum slosh models as seen below.
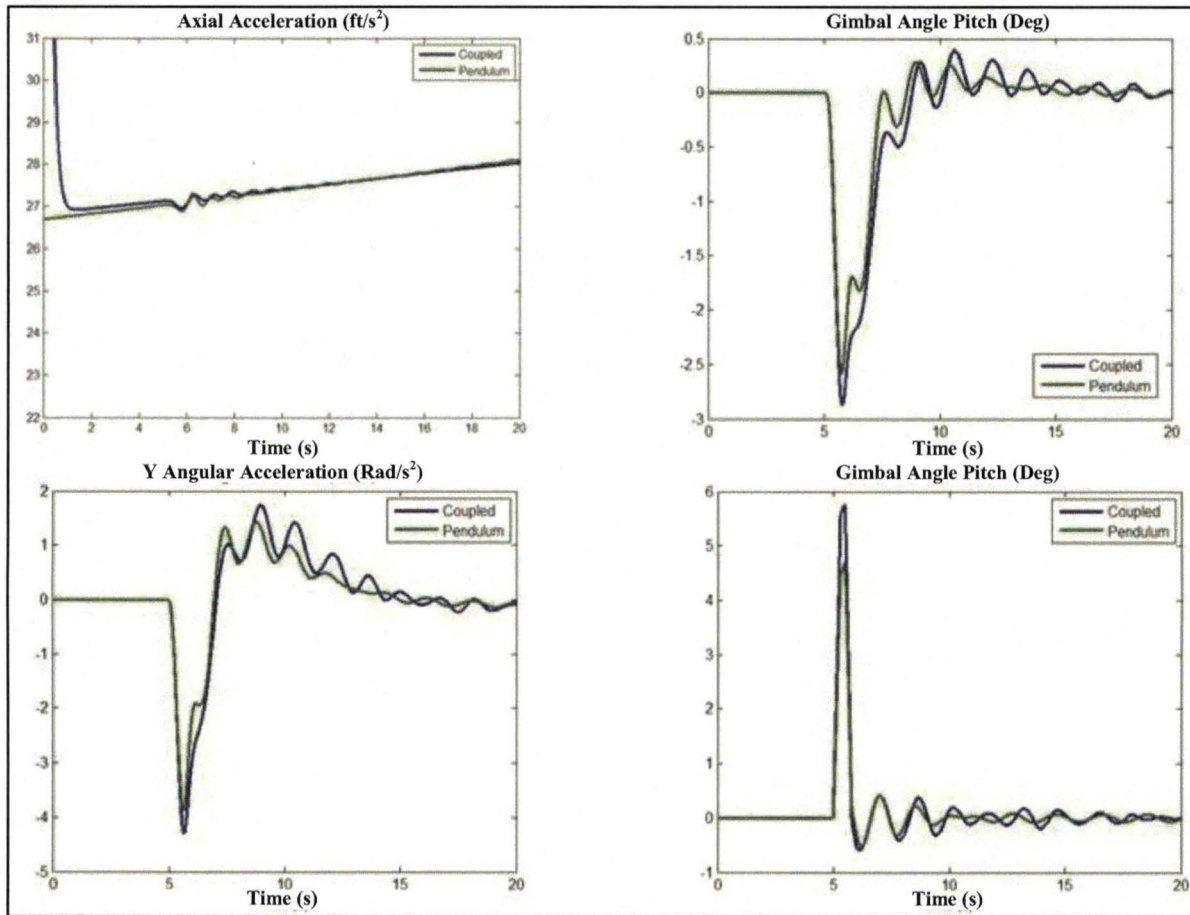

Figure 8. Draining Tanks Results Comparison

Although the signal match is fairly good, there are some differences. It is important to remember that this is a fairly complex system that is modeling the coupled interaction of two draining, sloshing fluids of different densities in different tanks with a control system. Though it is difficult to say precisely what causes the differences, there are several possible explanations. The first is the tuning of the transfer functions. As stated earlier, the transfer functions serve to delay dynamic information between the two models. This allows the system to be stable. The problem with this is that the transfer function must be tuned appropriately to add enough delay to stabilize the system while not damping out the force response from the fluid. Too much delay can cause the fluid to have an inaccurate effect on the vehicle and reduce the time accuracy of the system. The second possible explanation lies in the accuracy of the CFD solver. The model used in this case includes turbulence, mass loss (draining), non-inertial reference frame motion, fluid compressibility (LH2), and a Volume of Fluid (VOF) free surface formulation. Each of these complications may add error to the CFD output. This brings up the third possibility. The pendulum model must also take into account the changes in mass as the tanks drain. There has been very little validation of the pendulum model for draining tanks, but this is normally accounted for by changing the pendulum parameters (pendulum mass, length, hinge point location, fixed mass, fixed mass location, and damping) as the fill level changes. The pendulum model uses a table of these parameters as a function of fill level and interpolates between the values. This interpolation method may be a source of error and could explain some of the differences between the coupled and pendulum models. Lack of long duration, low gravity, liquid slosh data makes it very difficult to fully explain the differences between two models.

VI. Future Work

The pendulum method is a very effective means of modeling liquid propellant slosh assuming the amplitudes remain relatively low and there are no breaking waves. The validation presented in this document shows excellent agreement between the fully coupled CFD tool and the well validated pendulum analog. The advantage of the coupled CFD tool is that its validity goes beyond low amplitude and gentle slosh. This method should theoretically be valid under any condition. Lack of data forced this project to validate against the pendulum analog but, optimally, this tool would be best validated against experimental data where a large range of motions are tested. For this reason, the Launch Services Program has begun work on a new project aimed at collecting this data.

The SPHERES Slosh Experiment will use the currently operational SPHERES (Synchronized Hold Engage and Reorient Experimental Satellites) micro-satellites aboard the International Space Station (ISS), along with a partially filled liquid container to simulate maneuvers encountered by launch vehicle upper stages.[7] The data gathered by this experiment will be used to further validate the fully coupled CFD to control system analysis tool. An array of sensors will allow the experiment to gather dynamics data such as position, rates, and accelerations as well as video of the fluid inside. This will serve to both quantitatively and qualitatively validate the coupling tool.

Figure 9. SPHERES Satellite[7]

The SPHERES aboard the ISS are three small, self-contained vehicles with on-board propulsion, power, sensors, communication, and computer subsystems that can maneuver autonomously within the ISS. An example of one of these SPHERES is seen in Figure 9. All three spheres have an array of twelve fixed cold-gas thrusters that are fed by a liquid CO_2 propellant tank. They have a fixed thrust which can be pulse modulated to produce variable forces on the vehicle. The control system uses both inertial and external sensors for position and attitude determination. Gyroscopes and accelerometers provide rapid inertial state estimates, while ultrasonic measurements from wall mounted beacons are used to update the state estimate with respect to the space station internal volume.[7] This extensive amount of controllability makes it the perfect platform for testing the effects of liquid propellants on the control system of a vehicle in a microgravity environment.

Previous studies have attempted to use the liquid CO_2 in the SPHERES propellant tanks to collect low gravity liquid slosh data. Unfortunately, the small amount of liquid inside the CO_2 tanks was not sufficient to produce a measurable response to the vehicle's attitude. For this reason, the new SPHERES Slosh Experiment will use a purpose built partially filled acrylic tank that attaches in between two SPHERES as shown in figure 10. In this configuration, the amount of liquid in the tank should be enough to produce a measurable force response that can be used to compare against the results of the fully coupled CFD to controls system tool.
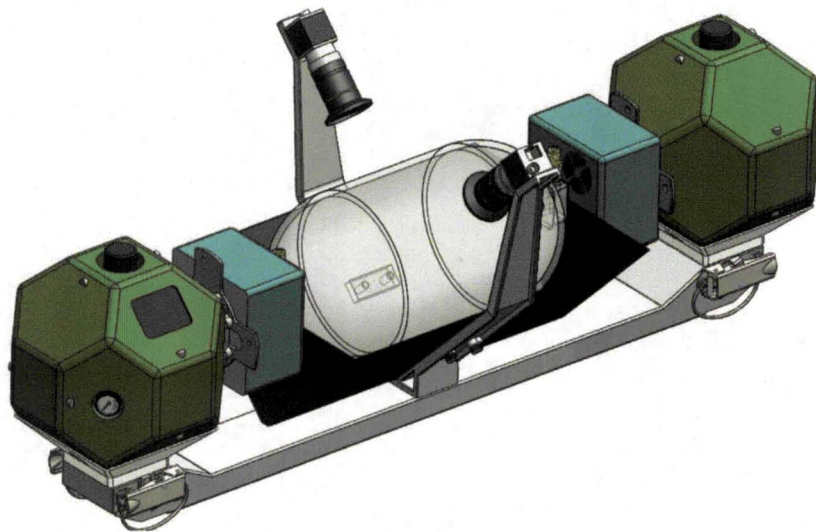

Figure 10. Proposed Configuration for SPHERES Slosh Experiment

The SPHERES slosh project is currently in the preliminary design phase. Though the basic design has been determined (Figure 10), the exact tank size is still being developed. Several trade studies have been completed with this goal in mind. These trade studies include determining the optimal tank size by minimizing the bond number while maximizing fluid mass and kinetic energy. The main goal is to have enough fluid to affect the system while ensuring the thrusters aren't overwhelmed.

The intent of this project is to create a database of long duration, microgravity liquid slosh data. Currently, the only sources of microgravity liquid slosh data consist of short duration (less than 30 seconds) slosh data sets collected aboard an aircraft. These are usually of limited value due to their short duration and fluctuating microgravity levels. These limitations are removed aboard the ISS. The constant microgravity environment coupled with the possibility of long duration tests, makes this project especially useful. This data set will not only be used to validate the CFD to controls coupling tool, but will also be made available to the slosh research community providing a unique perspective into the behavior of fluids in a microgravity environment.

VII. Conclusion

A fully coupled CFD-to-control system analysis tool has been created. This tool allows for high accuracy simulations of the fluid dynamic behavior in propellant tank slosh events and their influence on vehicle control dynamics. The validation runs against a well understood and trusted pendulum model show good agreement which instills confidence in the results obtained by the coupling tool. Future work will produce a rich database of microgravity slosh data that can be used to further validate this coupling tool. This novel multidisciplinary approach to solving the slosh analysis problem is a huge step forward that will advance our understanding of liquids in propellant tanks enabling the design of better vehicles that will be part of the future of human space exploration.

Acknowledgments

References

[1]Dodge, F.T., *The New Dynamic Behavior of Liquids in Moving Containers,* Southwest Research Institute, Technical Report, San Antonio, Texas.

[2]Hubert, C., *Behavior of Spinning Space Vehicles with Onboard Liquids*, Hubert Astronautics, 2nd Edition August 2008

[3]FLOW-3D, Version 9.4.2.2, Flow Science, Inc, Santa Fe, NM, 2010

[4]MATLAB, Version 7.8.0.347 (R2009a), The Mathworks, Inc. Natick, MA, 2009

[5]Weihs, D., and Dodge, F.T., *Liquid Motions in Nonaxisymmetric, Partially Filled Containers Rotating at Zero Gravity*, AIAA Journal of Spacecraft and Rockets, Vol. 28 1991.

[6]Marsell, B. Gangadaharan, S. Chatman, Y. and Sudermann, J. *Using CFD Techniques to Predict Slosh Force Frequency and Damping Rate*, AIAA Structures, Structural Dynamics, and Materials Conference, May 4-7, Palm Springs, CA, 2009

[7]Hilstad, Mark O., Enright, John P., Richards, Arthur G., and Mohan, Swati *The SPHERES Guest Scientist Program*, Massachusetts Institute of Technology Space Systems Laboratory, Cambridge, MA, 2010