Store Separation Trajectory Simulation for the High Speed Anti-radiation Demonstrator (HSAD) Program

E. Hallberg[1]
United States Naval Academy, Annapolis, MD 21402

E. Ray[2]
ATK Ordnance Systems, Plymouth, MN 55442

R. Fitzwater[3]
Virginia Polytechnic Institute & State University, Blacksburg, VA 24061

The High Speed Anti-radiation Demonstrator (HSAD) is a follow-on replacement for the AGM-88E HARM and is planned for carriage on the F/A-18C aircraft. Several store separation trajectory simulation tools (including NAVSEP and AEDC TGP) were used with Wind Tunnel test data to clear HSAD for a its first free flight test from the F/A-18C parent aircraft at a Mach number of 0.8 and a pressure altitude of 30,000 feet. Comparative analysis between the tools revealed a deficiency in the TGP based codes when predicting the trajectory of stores with off centerline axis thrust. A method was devised to incorporate autopilot functionality, and to Mach interpolate freestream wind tunnel data in NAVSEP simulations using MATLAB/SIMULINK.

Nomenclature

Store trajectories are defined in the Aircraft Axis System, which has its origin at the store center of gravity at release. The origin is fixed with respect to the aircraft and thus translates along the current flight path at the freestream velocity. The axes rotate to maintain constant angular orientation with respect to the current flight path direction.

ψ = PSI = Store yaw angle, positive nose right as seen by the pilot, deg.
θ = THETA = Store pitch angle, positive nose up, deg.
ϕ = PHI = Store roll angle, positive right wing down, deg.
P = Store roll rate, positive right wing down, deg/sec.
X_A = Store CG location relative to carriage, positive forward, parallel to aircraft centerline, ft.
Y_A = Store CG location relative to carriage, positive right, as seen by the pilot, ft.
Z_A = Store CG location relative to carriage, positive down, perpendicular to aircraft centerline, ft.
XFS = Fuselage Station. Distance of the store CG from the absolute Aircraft-axis system origin in the negative X_A direction, inches, full scale.
YBL = Butt Line. Distance of the store CG from the absolute aircraft-axis system origin in the Y_A direction, inches, full scale.
ZWL = Water Line. Distance of the store CG from the absolute aircraft-axis system origin in the negative Z_A direction, inches, full scale.
C_N = CN = Normal force coefficient, positive up, perpendicular to the store axis.
C_Y = CY = Side force coefficient, positive right, looking forward along store centerline.
C_A = CA = Axial force coefficient, positive rearward, along store centerline.

[1] Assistant Professor of Aerospace Engineering. AIAA Member.
[2] Aerospace Engineer. Mail Stop MN07-MW44, 4700 Nathan Lane N. AIAA Member.
[3] Student AIAA Member.

C_l = CLL = Rolling moment coefficient, positive right wing down.
C_m = CLM = Pitching moment coefficient, positive nose up.
C_n = CLN = Yawing moment coefficient, positive nose right.
$\alpha_{AIRCRAFT}$ = Aircraft Angle of Attack, deg.
α_S =ALPHAS = Store Angle of Attack (without induced upwash), positive nose up, as seen by the pilot, deg.
β_S= BETAS = Store Sideslip angle (without induced sidewash), positive nose left as seen by the pilot, deg.
6-DOF = Six-Degree of Freedom.
AEDC = Arnold Engineering Development Center
CG = Center of Gravity.
CTS = Captive Trajectory System.
HARM = High-speed Anti Radiation Missile.
HSAD = High Speed Anti-radiation Demonstrator.
ITALD = Improved Tactical Air Launched Decoy.
KCAS = Knots, Calibrated Air Speed.
NAVSEP = Navy generalized Six-Degree of Freedom separation simulation software.
TGP = Trajectory Generation Program.

I. Introduction

THE High Speed Anti-radiation Demonstration Program (HSAD) addresses the Navy's requirement to strike swiftly and decisively control time critical mobile forces. It intends to accomplish this by providing the capability to defeat short dwell mobile targets at range. The approach taken in developing the HSAD was to combine the proven capability of the AGM-88E High-speed Anti Radiation Missile (HARM) with advancements in propulsion technology, specifically advanced integral nozzleless rockets and variable flow ducted rocket ramjets. The program expects to demonstrate a substantial increase over current tactical capabilities including an increase of over the baseline HARM through a series of three flight tests from an F/A-18 C/D aircraft.

All three HSAD flights will be conducted from a modified LAU-118, on station 3 of the F/A-18C test aircraft. Figure 1 shows a mockup HSAD on a baseline LAU-118. The first free flight test of the HSAD missile is planned for December 2005. The release condition will be a Mach number of 0.8 and an altitude of 30,000 feet.

Prior to the store receiving an airworthiness certification for release, a detailed simulation was conducted to estimate the HSAD trajectory immediately after launch. The simulation was based, in part, on wind tunnel testing at the Arnold Engineering Development Center (AEDC).[1]

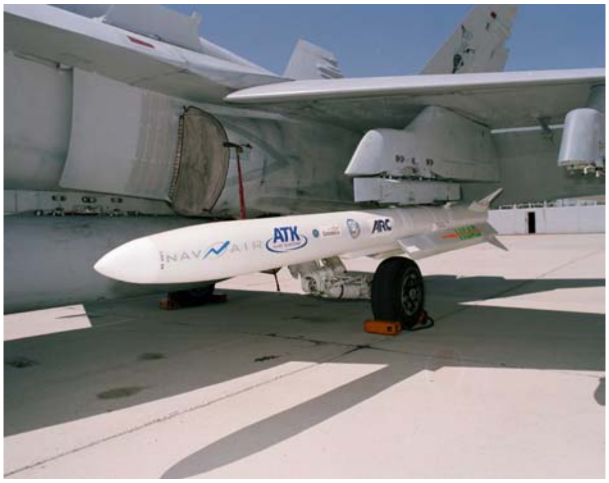

Figure 1. HSAD Missile on F/A-18C (baseline launcher).

HSAD utilizes an advanced guidance section to control four moving control surfaces mounted on the rear fins of the vehicle. At release, the fins are initially locked. Fin unlock occurs approximately 0.7 seconds after release as the autopilot takes active control of the HSAD vehicle. Initial simulations were performed as if the fins remained locked until the store moved a considerable distance ahead of the aircraft. Next, it was necessary to incorporate the effect of the active HSAD autopilot during the initial release phase. In all cases, it had to be proved that the store would remain stable and not interfere with the operation of the aircraft. Autopilot failure modes were also evaluated. Following the principle of independent verification, the Navy desired to validate the HSAD autopilot with at least one complete 6-DOF simulation code other than the proprietary one used by the contractor.

A variety of candidate Six-Degree Of Freedom (6-DOF) codes are available to simulate store trajectories. AEDC uses a code known as the Multi-Dimensional Interpolation Trajectory Generation Program (TGP). The U.S. Navy generalized store separation code is known as NAVSEP, and is based on an early version of the AEDC TGP code. Both codes have undergone numerous improvements and modifications during the past 15 years.

Both TGP and NAVSEP required the addition of a separate subroutine for autopilots. Incorporating source code into NAVSEP is labor intensive with the results invariably tailored only to a specific store. Verification of the autopilot functionality is difficult principally because of the embedded structure. A better way to incorporate autopilots is found in the Royal Australian Air Force STEME (Store Trajectory Estimation in a MATLAB Environment) 6-DOF code.[2] In a similar fashion the autopilot modeling was accomplished in block diagrams in a MATLAB/SIMULINK[3,4] environment and exchange protocols were developed for interaction with the NAVSEP code. A benefit of the resulting structure was the relative ease with which Mach interpolated wind tunnel data could be incorporated.

II. HSAD Wind Tunnel Tests

The U.S. Navy relies on wind tunnel testing using the Captive Trajectory System (CTS) to obtain store freestream and aircraft flowfield aerodynamics.[5] Ejector force characteristics, mass properties, and rocket motor thrust profiles are obtained through separate testing prior to flight test. The carriage loads and aerodynamic effects on the store after release are computed using data from three types of wind tunnel tests. Freestream data captures the aerodynamic loads on the store away from the influence of the aircraft. The second type of test captures the flowfield effects of the interaction of the store with the aircraft. Aerodynamic data for various store orientations and positions are taken in a grid in proximity to the aircraft. The results of these tests are naturally referred to as the grid data. Finally, CTS trajectory data are obtained by using the store mass, thrust, and aerodynamic loads in a quasi-steady trajectory simulation while the Wind Tunnel is blowing.

The freestream data to be implemented into any 6-DOF code is often a much larger scale model than that of separation testing. To this end, a 40% scale HSAD vehicle manufactured by Trimodels (shown in Figure 2) underwent freestream testing in the Allied Tunnels 7-FT Trisonic Tunnel prior to the separation loads testing.

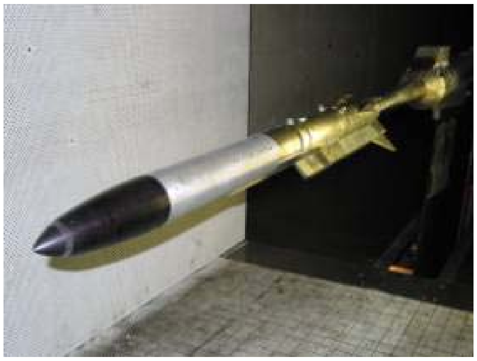

Figure 2. HSAD 40% Freestream Model.

Before separation Wind Tunnel testing, the Navy performed some conceptual studies of HSAD separations, using the HSAD 40% freestream data and mass properties, together with a HARM grid data from a previous Wind Tunnel test. During a parametric simulation study, it was discovered that changing the store Center of Gravity (CG) location relative to the point of action relative to the thrust line of action did not affect the trajectory. Upon investigation, this non-intuitive response was attributed to an error, where the thrust offset from the store CG was never accounted for in the legacy NAVSEP code.[6] This defect was immediately corrected in NAVSEP. Because the HSAD CG is below the store centerline (due to the presence of dual ventral ramjet inlets), the corrected version of NAVSEP predicted that a coupled moment should cause the HSAD to pitch downward.

Next, the 6% scale HSAD and F/A-18C aircraft models manufactured by Boeing underwent separation/loads testing in the AEDC 4T facility.[7] During CTS testing, performed with the AEDC Multi-Dimensional Interpolation Trajectory Generation Program code, it was noticed that the HSAD did not exhibit the expected pitch-down behavior, but a satisfactory explanation was not offered until after the test had concluded. Since the NAVSEP and TGP shared a common development thread some 15 years back, it was postulated that the thrust offset defect discovered in the NAVSEP code might also be present in the AEDC TGP code. A thorough investigation of the AEDC TGP code showed this to be the case.

Appropriate corrections were implemented in the code and the CTS trajectory run was post-processed to reflect the changes. The resulting processed CTS trajectory showed excellent agreement with the modified NAVSEP. Because all the CTS launch trajectories were incorrect, this experience demonstrates the utility of acquiring sufficient Wind Tunnel grid data to perform off-line simulation.

Despite their inaccuracy, the original CTS trajectories could still perform an important function. A standard practice to validate grid data is to run 6-DOF simulations at the same condition as a CTS trajectory. Ideally the trajectories should be identical, since they are both based on the same wind tunnel models. CTS trajectory Run number 1210 serves as a good test case because of the similarity to the planned first free flight HSAD launch. NAVSEP was run in an unguided mode (with the thrust offset purposely not included) at the test conditions summarized in Table 1. The simulated trajectory from NAVSEP matched the CTS run extremely well. As shown in Figure 3, the trajectories are virtually indistinguishable at 0.39 seconds after release. The axis refers to the Absolute Aircraft Axis System (XFS, YBL, ZWL System).

Table 1. CTS Run 1210 & First HSAD Free Flight

Configuration	Flight Condition
HSAD on IB Station 3	Mach number of 0.80
330 gallon external fuel tank on CL Station 5	Pressure altitude of 30,000 ft
ATFLIR on MW Station 4	Aircraft dive angle of 0 deg
OB Station empty	Aircraft angle of attack of 3.5 deg
	Store initial carriage pitch of -3.0 deg
	Store initial geometric angle of attack of 0.5 deg

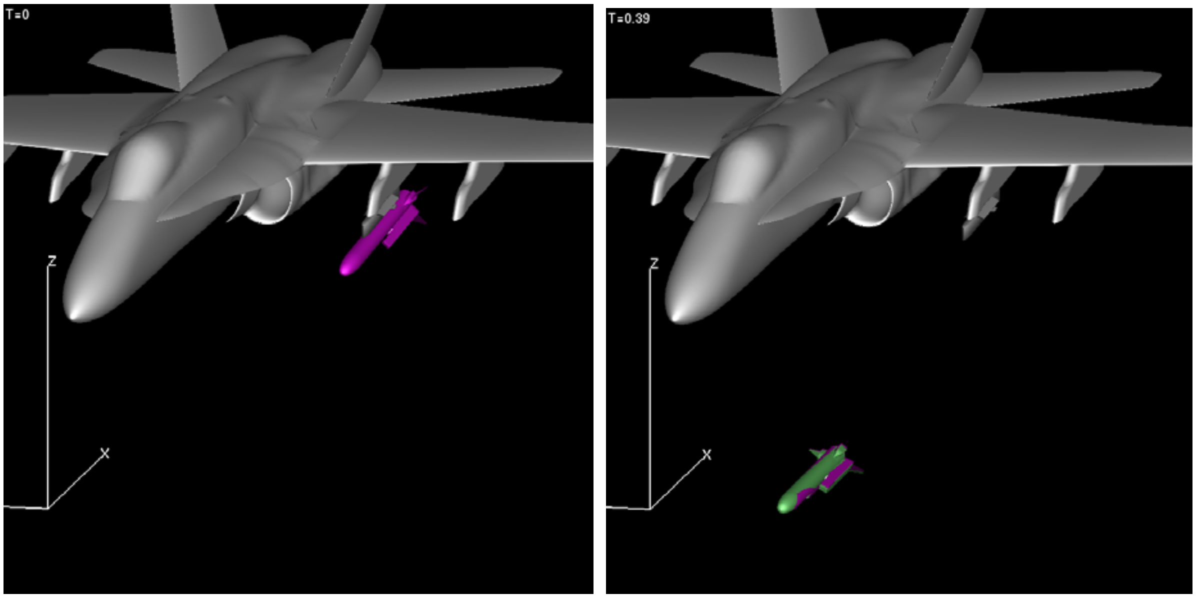

Figure 3. Visualization of NAVSEP Unguided trajectory and CTS Run 1210 at T=0 seconds and T=0.39 seconds. No CG/Thrust offset is applied.

Since the grid data were now validated, incremental changes were made to the simulation to better reflect reality. Grid data are generally implemented into a 6-DOF code as “delta” coefficients; that is, they are added to baseline store freestream values to account for the influence of the aircraft. To reduce systematic errors from the Wind Tunnel (sting aft end model distortion, scale effects, etc.), the freestream values subtracted from the “total” grid coefficients are from the same model as used for grid measurement (6% scale, in this case). However, NAVSEP has the ability to then use a better resolution freestream database to improve the quality of the simulation. The Navy took advantage of the 40% scale HSAD data for all further trajectory simulations.

Next, the rocket booster profile was updated to include a transient at the beginning of the launch, as well as varying the mass properties of the store as the propellant is expended. Figure 4 shows the HSAD at 0.7 seconds after release for three cases: without the CG/Thrust Offset (**purple**); with the corrected offset (**green**); and with the offset as well as the updated booster profile (**orange**).

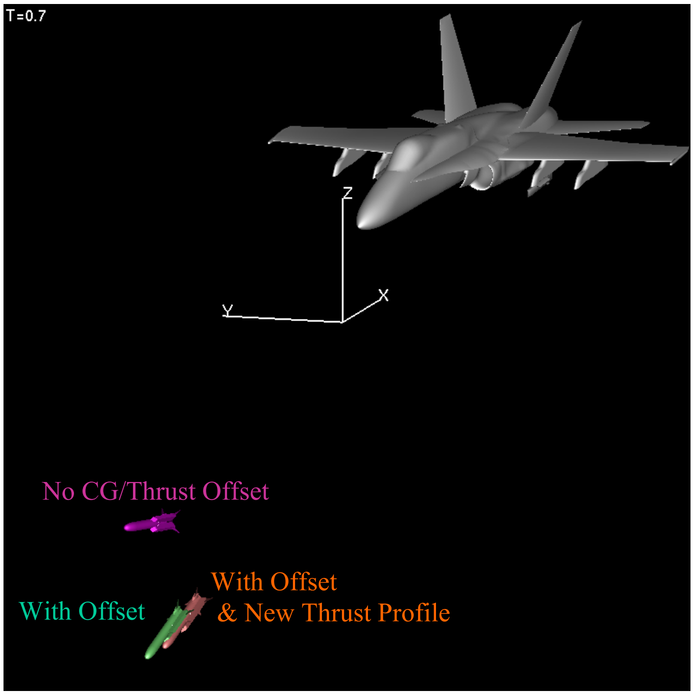

Figure 4. HSAD Trajectory Modeling Improvement.

III. NAVSEP Autopilot Development

Up to this point, the simulated trajectories were only valid to the time of fin unlock, after which point the control surfaces would guide HSAD. The next question was whether the autopilot design would have sufficient control authority to capture the natural tendency of the HSAD to pitch downward, roll, etc., and level it out for pre-programmed maneuvers. The time of fin unlock, approximately 0.7 seconds, was specifically chosen such that the HSAD would be a safe distance in front of the aircraft. However, it was necessary to demonstrate that the autopilot would never allow the missile to come back towards the aircraft.

The Navy decided to make moderate modifications to NAVSEP to allow for its execution in a MATLAB/SIMULINK environment. The developmental goals were to 1) provide robust capability to incorporate store autopilots in SIMULINK block diagram form, and 2) retain the capability to run the same NAVSEP source code (without an autopilot) in its native batch mode. This strategy would leverage on decades of corporate knowledge and compatibility with legacy NAVSEP data sets while providing an accepted standard environment for analysis and inclusion of control algorithms.

Because MATLAB has the capability to incorporate legacy FORTRAN code in its computational environment, it was first suggested to modify the NAVSEP code to run as a MATLAB executable (mex-file) in the MATLAB environment. Data would be exchanged between NAVSEP and MATLAB quickly via memory pointers. However, because such a version of NAVSEP would not be able to run in the native batch mode, this would lead to source code bifurcation. Any upgrades in one code would require a parallel effort in the other code. This was not in keeping with the design goal to modify a single NAVSEP version to run in either a stand-alone or MATLAB mode. Nevertheless, this option has the potential to execute faster than the method the Navy ultimately chose and the idea may be revisited in the future.

Over the years, the Store Separation Branch of NAVAIR had developed at least 18 unique version of NAVSEP as different engineers tailored the code to meet a specific requirement. This led to considerable confusion as engineers shared data. A recent effort was made to consolidate these fragmented codes into a single, well-maintained code, named NAVSEP 3.0. As part of the program, the source code was upgraded from FORTRAN IV to FORTRAN 90. Concurrently, several fundamental bugs were removed, documentation was written, and example test cases were developed. Input files were changed from a cryptic and rigid format to the more self-descriptive FORTRAN Namelist format, whose syntax was more forgiving. Output files were better labeled and contained more useful information.

The new system uses a SIMULINK block that can run arbitrary MATLAB scripts (M-files). During each iteration SIMULINK calls a MATLAB script, which executes NAVSEP in a shell using the "unix" command. The new version of NAVSEP, designated version 3.1, is therefore used as a 6-DOF "kernel" at every time step. Output files from NAVSEP 3.1 now include the store state in a format easily read into MATLAB, for propagation to the autopilot and the next time step. A diagram of the process is shown in Figure 5. File I/O creates a slower system than using pointers, but the final wall clock times were still acceptable.

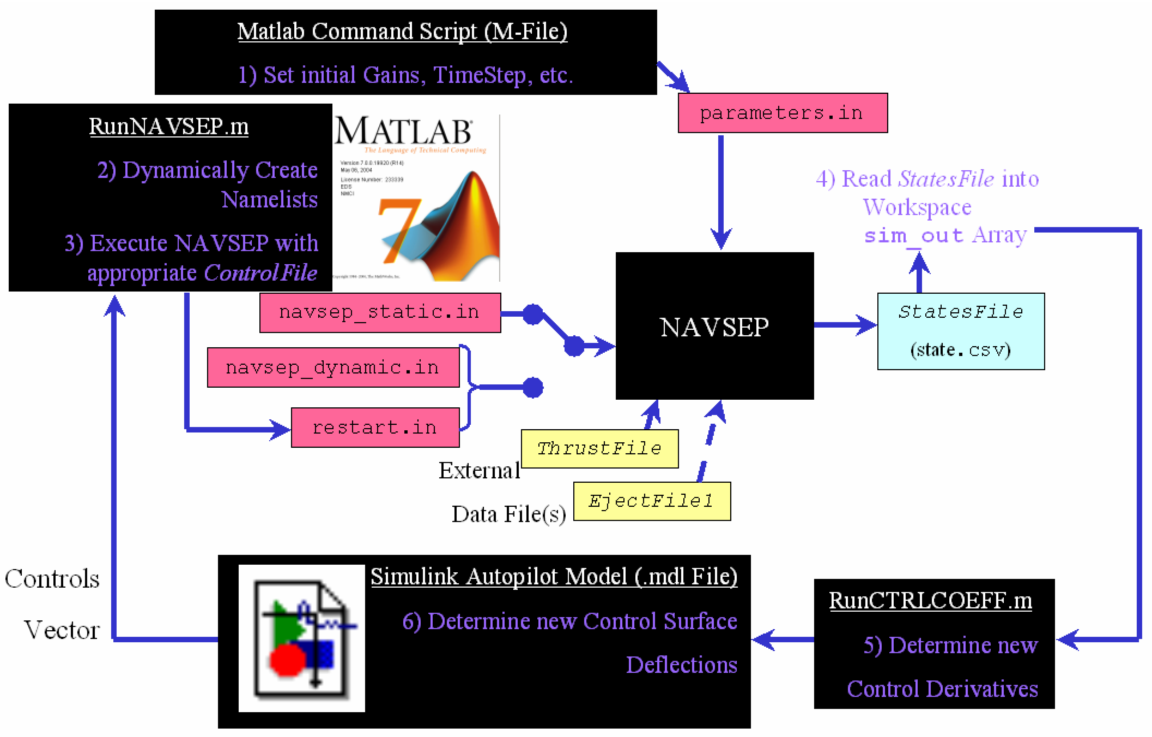
Matlab Command Script (M-File)
1) Set initial Gains, TimeStep, etc.
parameters.in
RunNAVSEP.m
2) Dynamically Create Namelists
3) Execute NAVSEP with appropriate ControlFile
MATLAB
4) Read StatesFile into Workspace sim_out Array
navsep_static.in
navsep_dynamic.in
restart.in
NAVSEP
StatesFile (state.csv)
ThrustFile
EjectFile1
External Data File(s)
Controls Vector
Simulink Autopilot Model (.mdl File)
6) Determine new Control Surface Deflections
RunCTRLCOEFF.m
5) Determine new Control Derivatives

Figure 5. Interface between NAVSEP and MATLAB/SIMULINK Autopilot

Several tangential improvements were made concurrently with the NAVSEP upgrade. These include providing output formatting options to facilitate easy plotting for a variety of plotting programs (GNUPLOT, Microsoft EXDEL, AEDC's DATAMINE, MATLAB, etc.). Also, additional Ejector Force parameters were added to provide modeling of arbitrary phenomena such as "roll spikes" and "imparted yaw."

IV. Validation with ITALD Autopilot

In order to validate the results of NAVSEP integrated with an autopilot, the results from a test flight that involved an Improved Tactical Air Launched Decoy (ITALD) release from an F/A-18C were used. This was an ideal test case since the ITALD uses an autopilot like the HSAD, and data from actual flight tests were accessible. The initial ITALD flight tests conducted at Patuxent River Naval Air Station in 1996 led to some separation failures. After improvements in the aerodynamics and control system of the ITALD, wind tunnel tests and simulation analyses were conducted to clear more flight tests. These flight tests were conducted in 1998 and were successful.[8]

The telemetry data from ITALD Flight Test Point 3 (October 8, 1998) was used for the comparison with NAVSEP. The flight test release was from the outboard Improved Triple Ejector Rack (ITER) Shoulder station from the port outboard pylon on F/A-18C Station 2. An ajacent ITALD was located on the inboard ITER shoulder, while an adjacent TALD was mounted on the center station of a second ITER on the inboard aircraft pylon. Because of the original Wind Tunnel grid data, all simulation was conducted as if the release was from the mirror-image starboard side of the aircraft. The release condition was at 397 KCAS at a pressure altitude of 12,000 ft, which is a Mach number of 0.90. The aircraft was straight and level at release with an Angle of Attack reading of 3.5 deg. Because the store is mounted at 3.0 deg downward relative to the aircraft, it had an initial geometric Angle of Attack of 0.5 deg.

The results from NAVSEP match the flight test data very well. Figure 6 displays a multi-exposure visualization of the store location at 0.7 sec after release for Telemetry (**purple**), as well as NAVSEP simulation with the autopilot on (**orange**) and off (**green**). The simulation with the active autopilot maintains a similar attitude as Telemetry, while the simulation with the autopilot off experiences uncontrolled roll.

The main difference between simulation and Telemetry lies with the store displacement. However, because no Photogrammetric data were available, it is difficult to confirm if this was the case. As other studies have shown,[9,10] because Telemetry post-processing means integrating accelerometer data twice in inertial space, it often falsely shows the store closer to the aircraft than Photogrammetrics, which takes into account relative motion of wing flexure, rolling, etc. However, Telemetry is the preferred method for obtaining store attitudes, because Photogrammetrics usually cannot place cameras in an ideal position. Therefore we can say with confidence that the simulated autopilot is controlling store attitude similar to the real store.

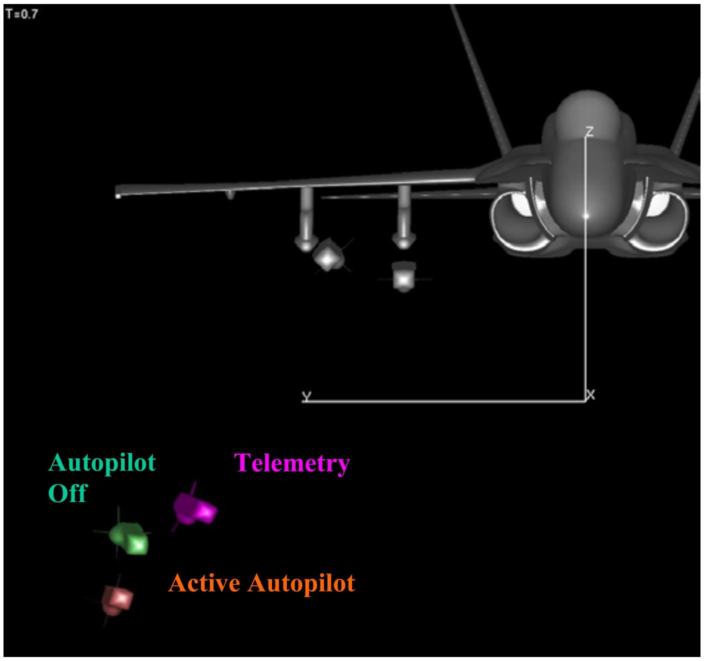

Figure 6. Comparison of Simulated ITALD Trajectory with Flight Test Telemetry.

Further indication of correct implementation of the ITALD autopilot is shown in Figure 7. Deflection angles of the three control surfaces (right elevon, left elevon, and rudder) from NAVSEP match well with both the commanded and measured angles from Telemetry. The confidence gained in the model allowed the Navy to apply NAVSEP 3.1 to the HSAD missile.

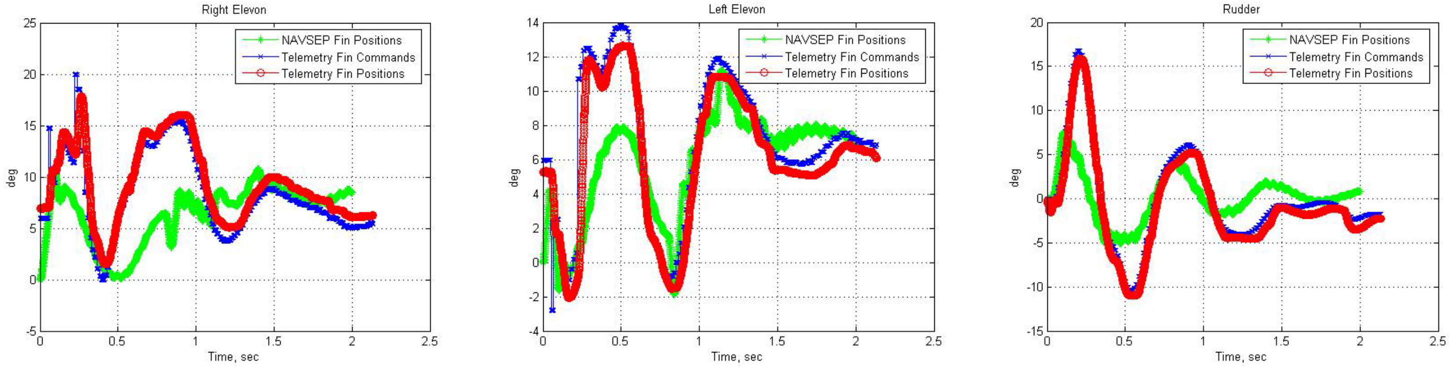

Figure 7. Comparison of Simulated ITALD Control Surface Deflection with Flight Test Telemetry.

V. HSAD Autopilot Simulation with NAVSEP

Using the same methods as the ITALD, NAVSEP 3.1 could be used to simulate the HSAD with an active autopilot. In addition to the autopilot, MATLAB allows for pre-processing of freestream data, which is critical to the accuracy of any trajectory.

NAVSEP is programmed to sort freestream data according to Store Angle of Attack (ALPHAS) sweeps at discreet Store Sideslip (BETAS) values. However, a more efficient mode of collecting freestream data in the Wind Tunnel is by using the aeroballistic axis method, whereby a store is placed at a particular angle of incidence and then rolled 180° or 360°. Figure 8 shows the preferred NAVSEP data format (left) as well as the actual data locations for the HSAD 40% Wind Tunnel test (right). ALPHAS sweeps appear as vertical lines, a BETAS sweep is a horizontal line, a coupled sweep has a slope of –1, and an aeroballistic sweep appears as a half-circle (with a radius of 4 deg.).

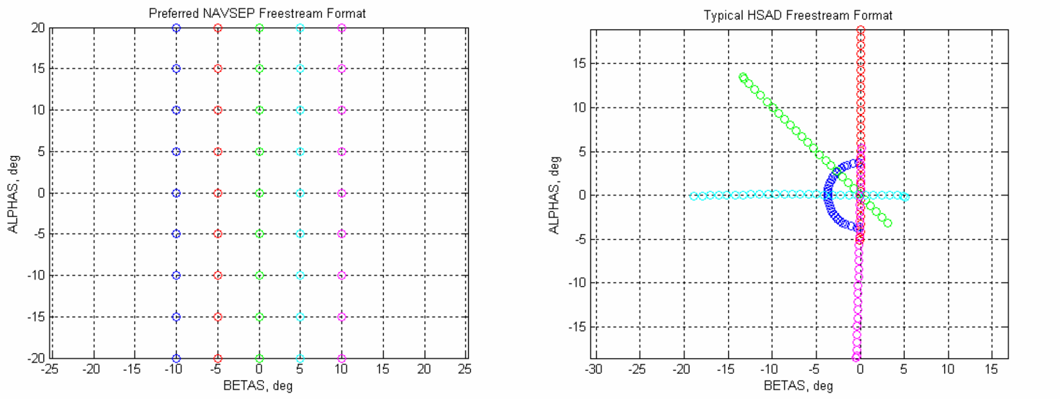

Figure 8. Typical Freestream Matrices.

Using the MATLAB "meshgrid" command, it is simple to create a regularly-spaced matrix of ALPHAS and BETAS locations which are easily read by NAVSEP. In this case, it is advantageous to cluster points around the origin because the HSAD is unlikely to depart from small angles. MATLAB was then used to interpolate the raw data at all of these points with the "griddata" command. Viewing such meshes as surfaces, such as the normal force coefficient (CN) and yawing moment coefficient (CLN) surfaces shown in Figure 9, can generate insight into data quality and store aerodynamic properties. Because the HSAD is symmetric in BETAS, data taken at positive values could be used to generate the surfaces (as shown with CN), but does not need to be included in the final NAVSEP database (as shown with CLN).

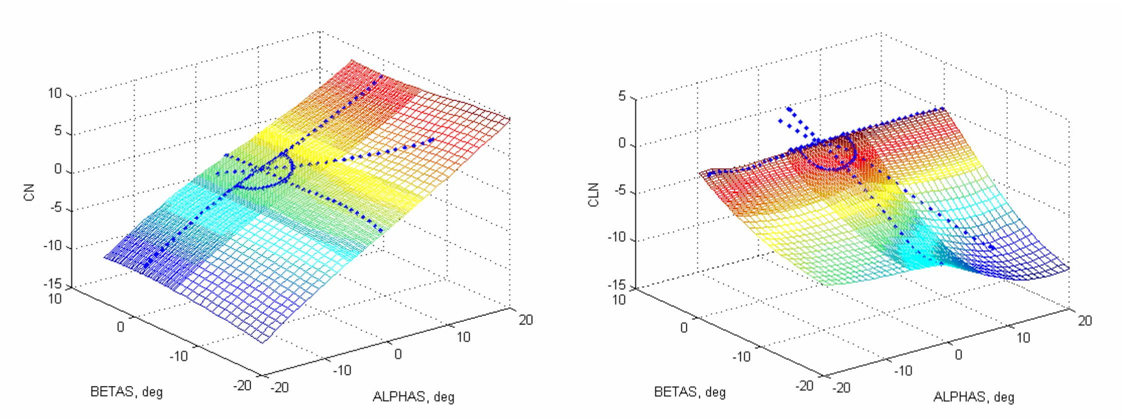

Figure 9. HSAD Normal Force (CN) and Yawing Moment Coefficient (CLN) Interpolated Surfaces. Original data are depicted as blue dots.

The ability to interpolate freestream according to Mach number would be required to adequately simulate HSAD. HSAD freestream data exists for Mach numbers of 0.80, 0.90, 1.20, and 1.80. Up to this point, NAVSEP simulations of HSAD either assumed a constant Mach number of 0.80, or would periodically re-start the simulation with data from a higher constant Mach number. An option to modify the NAVSEP FORTRAN source code to interpolate by Mach number was immediately discarded as impractical for the time allotted. Rather, it was decided to use MATLAB to perform the interpolation and feed the appropriate instantaneous data to NAVSEP as needed. This task was simplified because the freestream mesh surfaces could be created uniformly at various Mach numbers. Figure 10 depicts the CA surfaces for 4 Mach numbers (right) as well as a typical interpolated surface (left). Note that values for Mach numbers 1.2 and 1.8 appear concurrent. The freestream aerodynamic coefficients were pre-processed to create a matrix of layers at 0.01 Mach number increments.

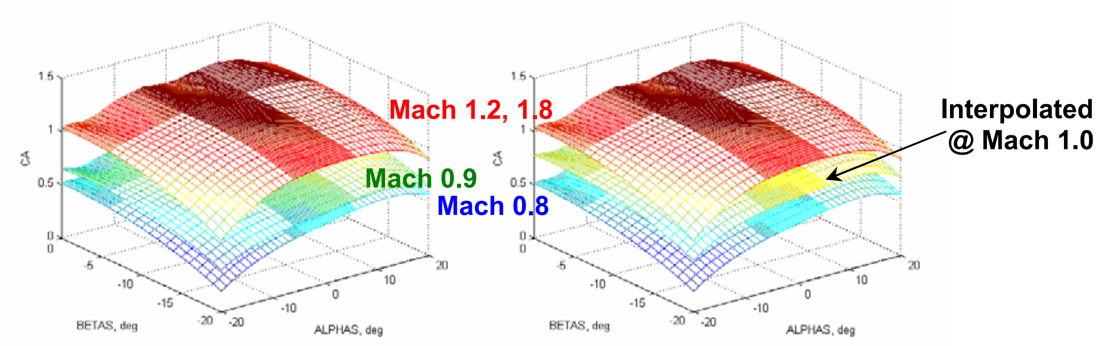

Figure 10. HSAD Axial Force Coefficient (CA) Freestream Interpolation in Mach number

Trajectory simulation of the HSAD was performed by successively adding more features to make the scenarios more realistic. A baseline trajectory was first obtained by NAVSEP in the offline mode without any controls. Next, a simulation was run with the autopilot in the MATLAB/SIMULINK environment. Finally, the autopilot trajectory was amended to perform freestream interpolation according to the instant Mach number experienced by the HSAD. The utility of the autopilot is perhaps best illustrated in its ability to capture roll rate, as shown in Figure 11, which shows all three trajectories. The uncontrolled HSAD would roll in excess of 500 deg/sec at 1.4 sec after release. Note that the time-step resolution for all trajectory plots has been decreased for clarity.

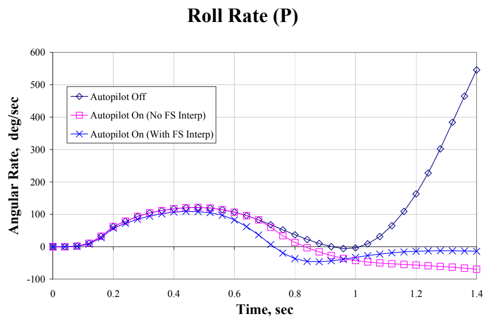

Figure 11. HSAD Simulated Roll Rate, demonstrating the effect of active control.

The full trajectory displacement and attitude history is shown in Fig 12. The HSAD autopilot does not affect the store displacement significantly, with the exception of some control in the lateral (YA) direction after approximately 1.0 sec. Large differences in store attitude can be seen with the addition of the autopilot.

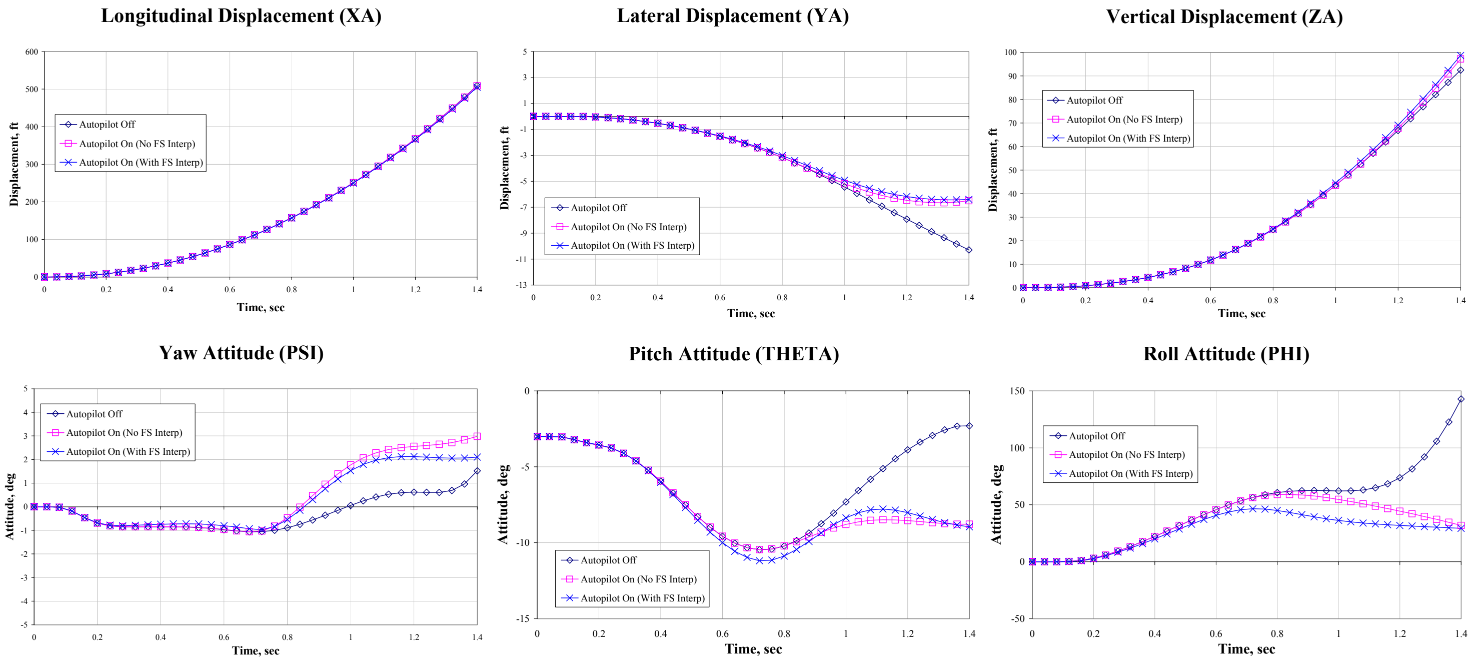

Figure 12. Comparison of Simulated HSAD Trajectories.

At first glance, both trajectories using the autopilot appear similar. However, the difference in using freestream interpolation becomes apparent when examining a time history of the axial force coefficient (CA) in Figure 13. The axial force coefficient, and therefore store drag, has increased by a factor of two after about 0.8 sec of acceleration in Mach number. Obviously simulating the correct missile loadings is critical to accurate simulation.

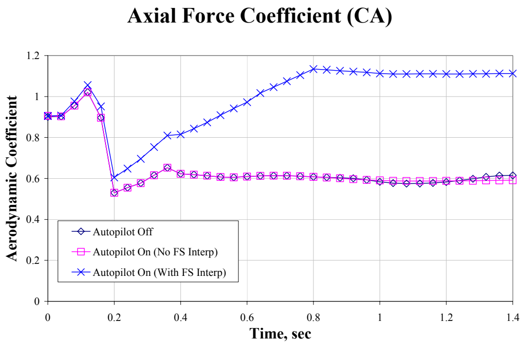

Figure 13. HSAD Simulated Axial Force Coefficient time history.

VI. Future Direction

Following the first free flight of the HSAD vehicle, flight telemetry will be used to validate and improve trajectory simulations from NAVSEP. Current plans for a second free flight of the HSAD vehicle at similar release condition as the first free flight. There is a growing consensus, however, that the second free flight could be used to better demonstrate the increased capability of the HSAD vehicle. This would most likely require a modification to the nominal launch condition, notably a release at a high transonic Mach number. The improvements and validation of the NAVSEP code made during the road to the HSAD first free flight test will be important in assuring a safe release for future releases from U.S. Navy aircraft.

It is always desirable to improve the speed of the simulation. A unique MATLAB executable version of NAVSEP was able to reduce wall clock time by about a factor of two by exchanging data via pointers rather than file I/O. However, the overall time was still several orders of magnitude larger than real-time. The multiple runs of Monte Carlo parametric studies require fast simulation times to be practical. Further efforts are warranted to produce an executable to be run outside of the MATLAB environment. Such a method could take advantage of high-level MATLAB features while still providing a fast stand-alone tool.

References

[1] Huhes, D., *et al*., "F/A-18C HSAD Test," Proj. No. 10425, Test No. TC-1098. Arnold Engineering Development Center, Arnold AFB, TN, April 2004.

[2] Akroyd, G., Tutty, M., "Theory Basis for STEME – Store Trajectory Estimation in a Matlab Environment," Aircraft Research and Development Unit, Royal Australian Air Force, Formal Report – Project I1018, 1999.

[3] SIMULINK Release 14 User's Manual:
http://www.mathworks.com/access/helpdesk/help/toolbox/simulink/index.html

[4] MATLAB Release 14 User's Manual: http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.html

[5] Taverna, F. P., Cenko, A., "Navy Integrated T&E Approach to Store Separation," Paper 13, RTO Symposium on Aircraft Weapon System Compatibility and Integration, Chester, UK, October 1998.

[6] Moyer, S. A., "NAVSEP – Navy Generalized Separation Package," AVCSTD Report 93011-6053, Sep. 1993.

[7] Veazey, D. T., Hopf, J. C., "Comparison of Aerodynamic Data Obtained in the Arnold Engineering Development Center Wind Tunnels 4T and 16T," AIAA, 1998.

[8] Richardson, Philip A. and Talbot, Matthew, "Analysis of the Separation Failures of a Decoy from the F/A- 18 Aircraft," AIAA-96-2411, June 1996.

[9] Ray, E., "CFD Applied to Separation of SLAM-ER From the S-3B," AIAA Paper 2003-4226. AIAA Applied Aerodynamics Conference, June 2003.

[10] Ray, E., “Authorizing SLAM-ER use from the P-3C with CFD,” AIAA Paper 2005-0845. AIAA Aerospace Sciences Meeting, January 2005.

[11] Tutty, M., “Aircraft Stores Compatibility Systems Engineering – The Fundamentals & Future,” Aircraft Stores Compatibility XIII Symposium, ITEA 2003.