SANDIA REPORT

SAND2004-4378
Unlimited Release
Printed September 2004

An Automated Procedure for Analyzing the Effects of Vortex-Induced Fin Pressure on Roll Torque for a Finned Body of Revolution

Shazib Z. Vijlee

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2004-4378
Unlimited Release
Printed September 2004

An Automated Procedure for Analyzing the Effects of Vortex-Induced Fin Pressure on Roll Torque for a Finned Body of Revolution

Shazib Z. Vijlee
Department of Mechanical Engineering
The University of Texas at Austin
Austin, Texas 78712

Department of Aerosciences and Compressible Fluid Mechanics (9115)
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-0825

Abstract

In flight tests, certain finned bodies of revolution firing lateral jets experience slower spin rates than expected. The primary cause for the reduced spin rate is the interaction between the lateral jets and the freestream air flowing past the body. This interaction produces vortices that interact with the fins (Vortex-Fin Interaction (VFI)) altering the pressure distribution over the fins and creating torque that counteracts the desired spin (counter torque). The current task is to develop an automated procedure for analyzing the pressures measured at an array of points on the fin surfaces of a body tested in a production-scale wind tunnel to determine the VFI-induced roll torque and compare it to the roll torque experimentally measured with an aerodynamic balance. Basic pressure, force, and torque relationships were applied to finite elements defined by the pressure measurement locations and integrated across the fin surface. The integrated fin pressures will help assess the distinct contributions of the individual fins to the counter torque and aid in correlating the counter torque with the positions and strengths of the vortices. The methodology produced comparisons of the effects of VFI for varying flow conditions such as freestream Mach number and dynamic pressure. The results show that for some cases the calculated counter torque agreed with the measured counter torque; however, the results were less consistent with increased freestream Mach numbers and dynamic pressures.

Acknowledgements

I would like to especially thank Walt Wolfe, Steve Beresh, and Carl Peterson for allowing me to help them with this project and to simply learn so much from them. Also, my manager, Basil Hassan for streamlining and simplifying the student intern process. Lastly, Art Ratzel for helping me get this great opportunity at Sandia.

Contents

Figures

Tables

Nomenclature

$\mathbf{A}$	area vector
A_{fin}	fin area
$\mathbf{A}_i$	triangle element area
A_{tr}	tapped region area
$\mathbf{F}$	force vector
$\mathbf{F}_i$	triangle element force vector
J	nozzle dynamic pressure ratio
$\mathbf{L}$	first triangle leg vector
$\mathbf{M}$	second triangle leg vector
N	number of triangle elements
p	pressure
p_i	triangle element pressure
$Q_{freestream}$	freestream dynamic pressure
Q_{nozzle}	nozzle dynamic pressure
$\mathbf{r}$	radius vector
$\mathbf{r}_i$	triangle element radius vector
r_x	radius vector x-component
r_y	radius vector y-component
r_z	radius vector z-component
$\mathbf{T}_{fin}$	fin torque vector
$\mathbf{T}_{tr}$	tapped region torque vector

Note: Vector parameters have been distinguished with bold face type.

An Automated Procedure for Analyzing the Effects of Vortex-Induced Fin Pressure on Roll Torque for a Finned Body of Revolution

Introduction

In flight tests, certain finned bodies of revolution (FBR) firing lateral jets experience slower spin rates than expected. The primary cause for the reduced spin rate is the interaction between the lateral jets and the freestream air flowing past the body, producing vortices that interact with the fins (Vortex-Fin Interaction (VFI)) altering the pressure distribution over the fins and creating torque that counteracts the desired spin (counter torque). External, production-scale wind tunnel experiments measured roll torque, fin pressures, and vortex strengths and positions while varying airspeed, angle of attack, fin orientation, and jet strength for a finned body of revolution [1,2]. The current task is to develop an automated procedure for analyzing the fin pressures to determine the VFI-induced roll torque and comparing it to the roll torque experimentally measured with an aerodynamic balance. Accomplishing the task involves integrating the pressures over the fin surface and automating the integration algorithm to expedite the data analysis. The fin pressures provide additional information about the spin rates that complement the balance-measured torque. The integrated fin pressures will help assess the distinct contributions of the individual fins to the counter torque and aid in correlating the counter torque with the positions and strengths of the vortices.

Vortex-Fin Interaction

The reduced spin rate can be attributed to a phenomenon called vortex-fin interaction. VFI begins with the spin motor nozzles firing lateral jets on the finned body of revolution (FBR); see Figure 1. The jets fire in opposite directions and create a counterclockwise torque that causes the FBR to spin about its longitudinal axis. The nozzle jets are in crossflow with respect to the freestream air, and the jet-freestream interaction creates vortices that alter the flow field. The vortex-fin interaction induces pressure differentials on the fin surface. The induced pressure differential creates an opposing torque in the clockwise direction. This opposing torque is the primary cause for the reduction in spin rate.

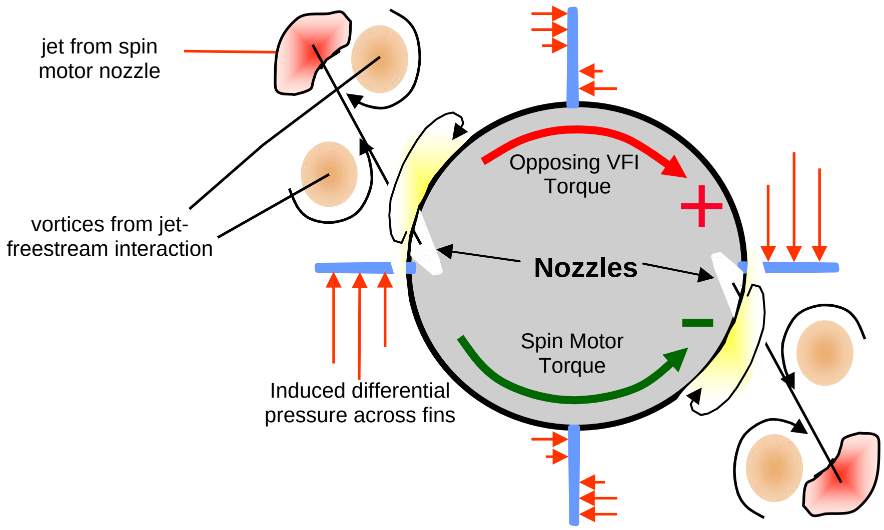

Figure 1. FBR with Components of VFI (View from Nose Looking Aft).

Experiments

The FBR used in our wind tunnel tests has four fins oriented 90° apart resembling an X, as seen in Figure 2. Figure 3 shows the FBR side view with the fins rotated into the vertical plane. Two of the four fins were instrumented and studied: Fin A and Fin B. The two sides of both fins are equipped with 24 pressure taps each, similar to Figure 4. The taps are numbered from 1 to 48, with the even numbered and odd numbered taps on opposite sides of the fin. The fin is too thin for the pressure taps to be exactly opposite one another, so they are slightly offset. This arrangement creates eight distinct torque-creating surfaces, but symmetry allows us to measure only the four in Table 1.

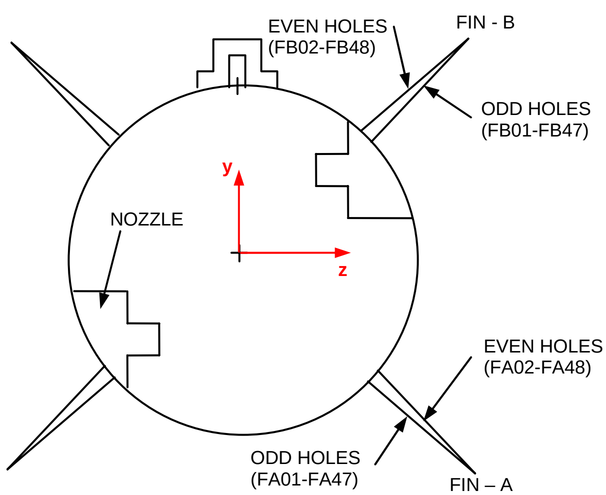

Figure 2. Aft View of Fins (View from Nose Looking Aft).

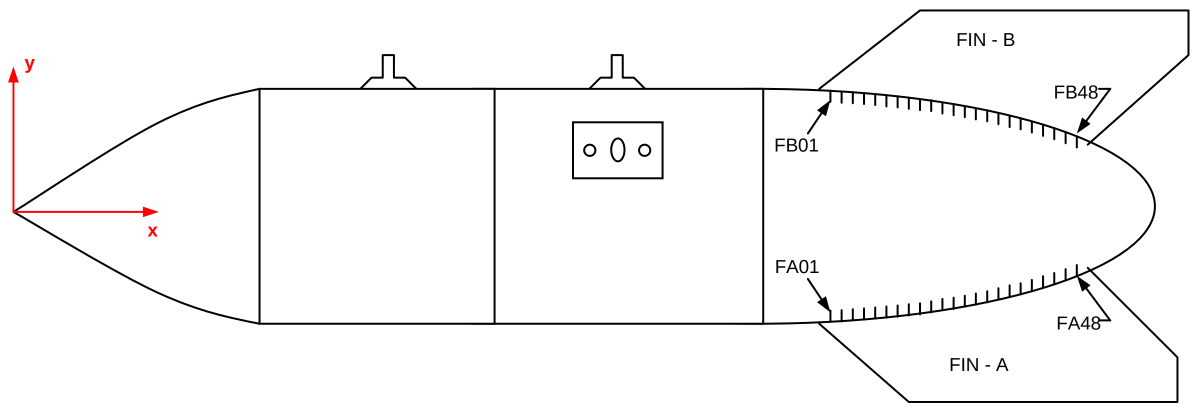

Figure 3. Side View of FBR (not to scale).

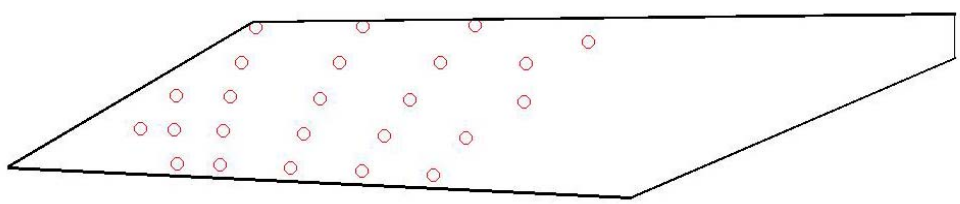

Figure 4. Fin Shape (in black) and Pressure Taps (in red).

Table 1. Surfaces of Interest.

Surface	Fin	Taps
1	A	odd
2		even
3	B	odd
4		even

Wind tunnel experiments were performed at the Arnold Engineering Development Center (AEDC) and NASA Ames with several configurations for a single FBR. Each run varied the flow conditions and model geometry of the FBR. Hundreds of cases were run and the data stored in specific directories. For example, Run #0376 is stored as file *C2_Q220_M8/RUN00376.dat*. *C2* is for model Configuration 2, *Q220* is for a freestream dynamic pressure of 220 psf, and *M8* is for Mach 0.8.

Fin pressures induced by VFI were recorded and stored while an aerodynamic balance directly measured torque. The balance-measured torque will be used for comparison after the fin pressures have been reduced to torque. Equation 1 shows the Nozzle Dynamic Pressure Ratio, J, which is used to determine trends and relationships between spin rate, roll torque, and fin pressure. J represents the strength of the nozzle jets. The Nozzle Dynamic Pressure Ratio is an experimental quantity and is the primary correlation factor for jets in cross flow.

$$J = Q_{nozzle} / Q_{freestream} \quad (1)$$

Integration Methodology

The first step in assessing the effect of VFI on counter torque is integrating the fin pressures across the fin surface. The integration methodology is identical for each of the surfaces in Table 1, so discussion can be limited to a generic case. The real surfaces have 24 taps; our simplified case will have nine. In Figure 5 the nine pressure taps have been meshed with a series of triangles, lettered from A through G. In this simple example, the triangles are arbitrarily created, but in the proper methodology the triangles are determined through Delaunay Triangulation [3]. The torque due to surface pressure can be determined with Equation 2. The torque vector for the tapped region ($\mathbf{T}_{tr}$) is the cross product of the radius vector (**r**) and the force vector (**F**), but for the entire region it is the vector sum of the torque produced by each triangle. The surface will experience a total torque value that is the sum of the torques on the N triangles. As shown in Equation 3, the force vector is the product of the discrete pressure (p) and the discrete area vector (**A**).

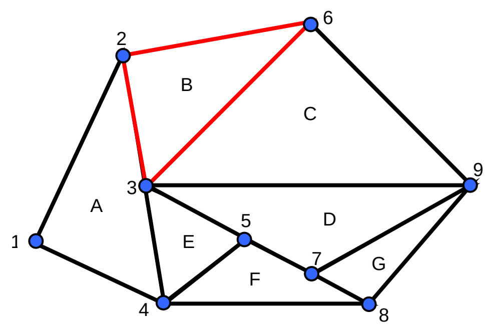

Figure 5. Hypothetical Tap Configuration and Triangulation.

$$\mathbf{T}_{tr} = \mathbf{r} \times \mathbf{F} \approx \sum_{i=1}^{N} \mathbf{r}_i \times \mathbf{F}_i \qquad (2)$$

$$\mathbf{F}_i = p_i \mathbf{A}_i \qquad (3)$$

The first step in the integration process is to determine the discrete pressure associated with each triangular element. Assuming a linear pressure variation between taps, the nominal pressure for a single triangle is the mean of the pressures at the three taps that constitute the three vertices of the triangle. For example, in Triangle B above, the pressure is shown as Equation 4.

$$p_B = \frac{1}{3}\left(p_2 + p_3 + p_6\right) \qquad (4)$$

The magnitude of the area for Triangle B is calculated with Equation 5 [4]. The cross product of two of the triangle's legs (**L** and **M**) will result in a vector that is perpendicular to the triangle. The area vector is the product of the area magnitude and the area unit vector; see Equation 6. Arbitrarily selecting which leg is **L** and which leg is **M** will result in variations in the direction of the area vector; it will sometimes point into the plane of the triangle and other times it will point out of the plane of the triangle. However, based on the coordinate system in Figures 2 and 3, we know that a clockwise torque is positive. The pressures will always be acting into the surface because they are absolute pressures, so the torque produced by the even tapped surfaces should always be positive and those produced by the odd tapped surfaces should always be negative.

$$A_B = \frac{1}{2}\sqrt{\begin{vmatrix} y_6 & z_6 & 1 \\ y_2 & z_2 & 1 \\ y_3 & z_3 & 1 \end{vmatrix}^2 + \begin{vmatrix} z_6 & x_6 & 1 \\ z_2 & x_2 & 1 \\ z_3 & x_3 & 1 \end{vmatrix}^2 + \begin{vmatrix} x_6 & y_6 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}^2} \quad (5)$$

$$\mathbf{L} = \begin{bmatrix} x_6 - x_2 \\ y_6 - y_2 \\ z_6 - z_2 \end{bmatrix} \quad \mathbf{M} = \begin{bmatrix} x_3 - x_2 \\ y_3 - y_2 \\ z_3 - z_2 \end{bmatrix} \qquad \mathbf{A}_B = A_B \cdot \frac{\mathbf{L} \times \mathbf{M}}{\left\| \mathbf{L} \times \mathbf{M} \right\|} \quad (6)$$

The mean triangle pressure is assumed to act at the centroid of the triangle in three-dimensional space; the centroid for Triangle B is calculated with Equations 7a-7d. The centroid of a triangle is the mean coordinates of the three vertices. The coordinates of the centroid also act as the radius vector from the coordinate system origin to the centroid (**r**), which is the same radius we will use as the moment arm for our torque calculation.

$$\mathbf{r}_B = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \quad (7a)$$

$$r_x = \frac{1}{3}\left(x_2 + x_3 + x_6\right) \quad (7b)$$

$$r_y = \frac{1}{3}\left(y_2 + y_3 + y_6\right) \quad (7c)$$

$$r_z = \frac{1}{3}\left(z_2 + z_3 + z_6\right) \quad (7d)$$

The accuracy of the method for triangulating the pressures depends on the taps being physically and computationally coplanar. Three points will always make a plane, but the physical fin surface is curved in some regions, so some of the triangles do not accurately represent the surface. The triangulation may connect three taps with a single computational plane, even if the same three taps are not on a single physical plane. Revising the tap coordinates to include coordinates for the slope discontinuities could solve this problem. Though the present analysis neglects this error source, it is believed to be small as the surface curvature is generally mild and present only over a portion of each fin.

Physically, we could not measure pressures over the entire fin surface. The pressure taps are holes drilled normal to the surface and into spanwise tubes that are drilled into the fin. Limitations on the diameter and aspect ratios of the tubes (which must be linear) prevented us from tapping the entire surface. A subsurface, the tapped region, was used to approximate the effects over the entire fin; however, we must scale the torque to get a better estimate. Equation 8 scales the torque from the tapped region to the entire fin area. For the final values, we must also include a factor of two to account for the two un-instrumented fins.

$$\mathbf{T}_{fin} = \mathbf{T}_{tr} \frac{A_{fin}}{A_{tr}} \quad (8)$$

Determining the triangle pressures, area vectors, and radius vectors are all we need to determine the torque produced by each triangle. The sum of the torque over the surface will be compared with the torque values measured by the aerodynamic balance. Our simple example was for nine taps and discussion was limited to a single triangle, but the real situation involves 24 taps and approximately 35 triangles for several hundred values of J. The easiest way to expedite the data analysis is to automate it in MATLAB [3].

MATLAB Implementation

The main goal of the project was to develop an automated procedure for analyzing the data, so only a small subset of data was used to develop the software. The overarching main program is *VFI_torque.m*. In the following discussion each indentation represents a descending level of hierarchy. For example, *VFI_torque.m* calls the surface specific function *FAodd.m*; however, it could have called the functions specific to the other three surfaces in Table 1. The reader should recall that each surface in Table 1 can be interchanged, so the discussion is limited to the first surface: Fin A, odd taps. The surface subfunctions call *vert_and_area.m*, *triangle_pressures.m*, and *moment.m*. The third level function *vert_and_area.m* calls the lowest function *tri3Darea.m*. The following discussion "walks" through the analysis using the code's actual variable names. The complete code is included in Appendix A.

VFI_torque.m

The main program *VFI_torque.m* requests the user to input the directory (**dir**) with the data files and the run number of interest (**run**). The information in the DAT files is loaded into **data** and **tap_coords** and then filtered for the VFI correlation parameter (**J**), the pressure values for Fin A (**FA_p**) and Fin B (**FB_p**), and the roll torque measured by the aerodynamic balance (**T_roll**). The total area of the whole fin is calculated as **fin_area**. The program then calls the individual surface functions: *FAodd.m*, *FAeven.m, FBodd.m, FBeven.m*.

FAodd.m

The surface subfunction first filters **FA_p** and **tap_coords** to maintain only the pressures for this surface (**p**) and the **x**, **y**, and **z** tap coordinates for this surface. It then calls the sub-subfunction *vert_and_area.m* to calculate the triangle vertices (**vertices**), triangle area vectors (**area_vectors**) and triangle centroid radius vectors (**r**).

vert_and_area.m

The generic sub-subfunction accepts the **x**, **y**, and **z** tap coordinates and uses MATLAB's Delaunay function to create a mesh of triangular elements. The triangulation is based on combined y and z values. For this case, the triangulation results in **vertices,** a 35x3 matrix (35 triangles with 3 vertices each). Then the 35x3 variables **tri_xcoord**, **tri_ycoord**, and **tri_zcoord** are created and sent to *tri3Darea.m*.

tri3Darea.m

This sub-sub-subfunction accepts **tri_xcoord**, **tri_ycoord**, and **tri_zcoord** and calculates the area of each three dimensional triangle with Equation 5, then returns the area magnitudes (**area**) to *vert_and_area.m*.

The program checks **area** for extraordinarily small values (less than 1% of the mean) and deletes them. It also calculates the total area (**a_surf**) included in the tapped region, this will help scale the results over the entire fin. The radius vector **r** for each triangle is the location of the centroid of the triangle. The area normal vector (**area_vectors**) for each triangle is determined as the cross product of two of the triangle's legs. The sub-subfunction returns **vertices**, **area_vectors,** and **r** to *FAodd.m*.

The subfunction calls *triangle_pressures.m* to calculate the pressures (**p_triangle**) acting on the discrete triangles.

triangle_pressures.m
The sub-subfunction accepts the pressure values (**p**) and the triangle vertices (**vertices**). It determines the pressure for each triangle as the mean of the three pressures at the three vertices of the triangle. The result, **p_triangle**, is sent to *FAodd.m* as a 320x35 matrix (320 J values, 35 triangle pressures).

The subfunction calls *moment.m* to calculate the torque (**T_tot**) acting over the entire surface.

moment.m
The sub-subfunction accepts **p_triangle**, **area_vectors**, and **r** and calculates the force, **F**, on each triangle resulting in a 320x3x35 array (320 J values, 3 components, 35 triangles). It then uses **r** and **F** and calculates the torque, **M**, experienced at the centroid of each triangle; the torque values are also in a 320x3x35 array. The torque values are all made positive to maintain a sign convention since the sign of the area vectors is ambiguous. The torque is then summed along the 3rd dimension to determine a singular torque vector for the entire surface (Σ35 triangles = 1 surface), the result is **T_tot**, a 320x3 matrix (320 J values, 3 torque components) that is returned to *FAodd.m*.

The x-values of **T_tot**, **T_tot_x**, are assigned the appropriate sign. The subfunction then plots **T_tot_x** vs. **J** and saves the JPG in **dir** with the same name as the DAT file. The filename also contains an indicator of the surface being plotted. The pressure torque is scaled with the ratio of fin area (**fin_area**) and tapped area (**a_surf**) to better approximate the total roll torque from fin pressures over the entire fin surface. The subfunction sends the value for the surface specific, pressure induced torque (**T_FAodd**) to the main function *VFI_torque.m*.

The main function sums all the torque calculated from the pressure data from all surfaces (**T_pressure**), and calculates the sum of each fin (**T_FA** and **T_FB**). On a single graph it plots **T_pressure** and **T_roll** vs. **J**. On two other plots it displays **T_FA** and **T_FB** vs. **J**. All plots are saved to **dir** with filenames similar to the DAT file, but with a unique identifier.

Results

Since a small subset of data was used to create the software, the only cases run were of AEDC Configuration 2 with an angle of attack of 0^o. Lugs were not installed on Configuration 2. The software produces seven plots of various torque parameters versus the Nozzle Dynamic Pressure Ratio, J. Qualitatively, the plots show the effect of VFI versus the strength of the nozzle jets. Each of the plots is saved in the same directory as the original DAT file. For reference, the plots for Run #0376 are shown here. Run #0376 is at Configuration 2, 0.8 Mach, and $Q_{freestream}$ = 220 psf. Each run involves different configuration settings and flow conditions so each run will produce different plotted results, but these are examples of the information that the software reports.

The first plot, shown in Figure 6, is the plot of the Fin A, Odd side, counter torque versus J.

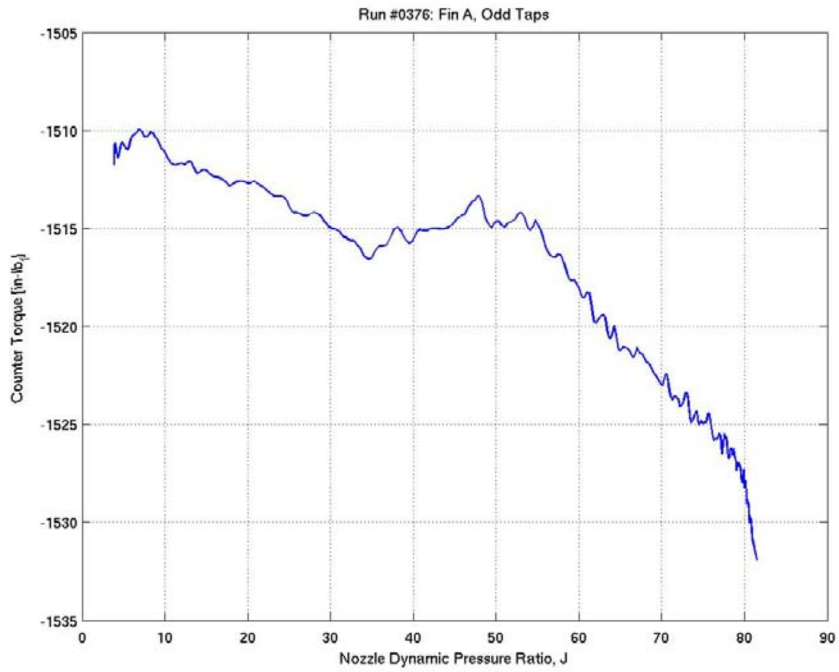

Figure 6. Fin A, Odd Counter Torque Versus Nozzle Dynamic Pressure Ratio for Run #0376.

The second plot, Figure 7, is the counter torque from Fin A, Even side, versus J.

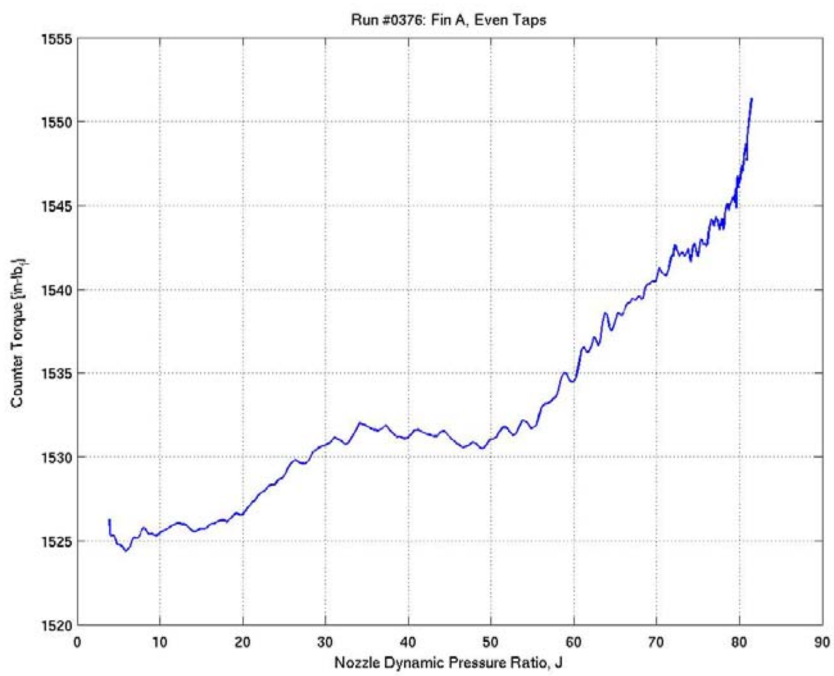

Figure 7. Fin A, Even Counter Torque Versus Nozzle Dynamic Pressure Ratio for Run #0376.

The third plot, Figure 8, is the plot of the total counter torque on Fin A versus J.

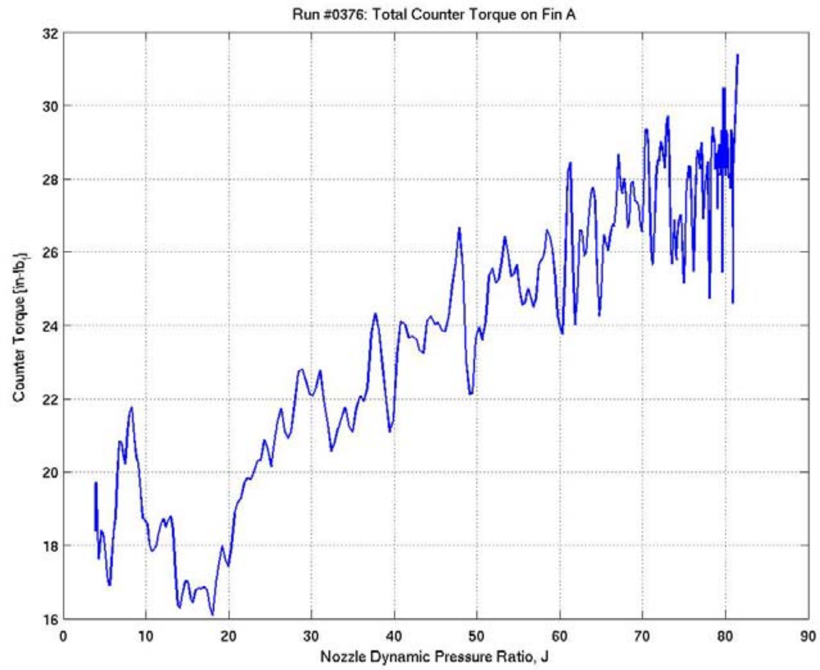

Figure 8. Fin A, Total Counter Torque Versus Nozzle Dynamic Pressure Ratio for Run #0376.

The fourth plot, Figure 9, is the plot of Fin B, Odd side, counter torque versus J.

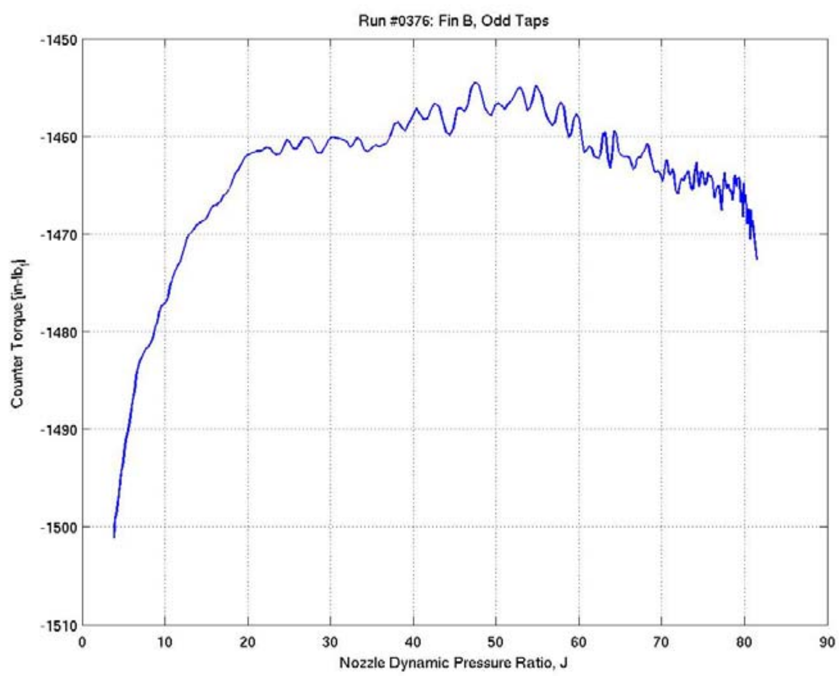

Figure 9. Fin B, Odd Counter Torque Versus Nozzle Dynamic Pressure Ratio for Run #0376.

The fifth plot, Figure 10, is the counter torque from Fin B, Even side, versus J.

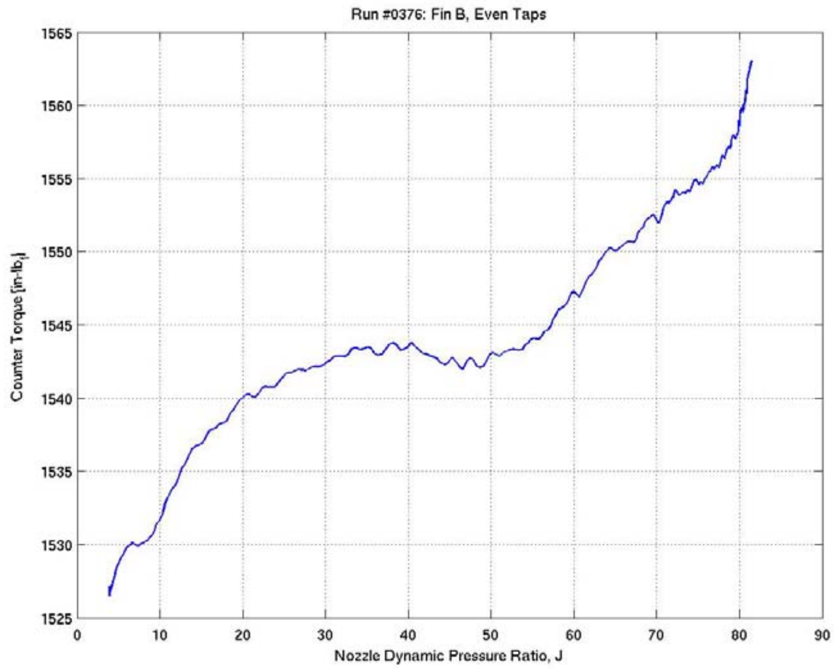

Figure 10. Fin B, Even Counter Torque Versus Nozzle Dynamic Pressure Ratio for Run #0376.

The sixth plot, Figure 11, is the total counter torque on Fin B versus J.

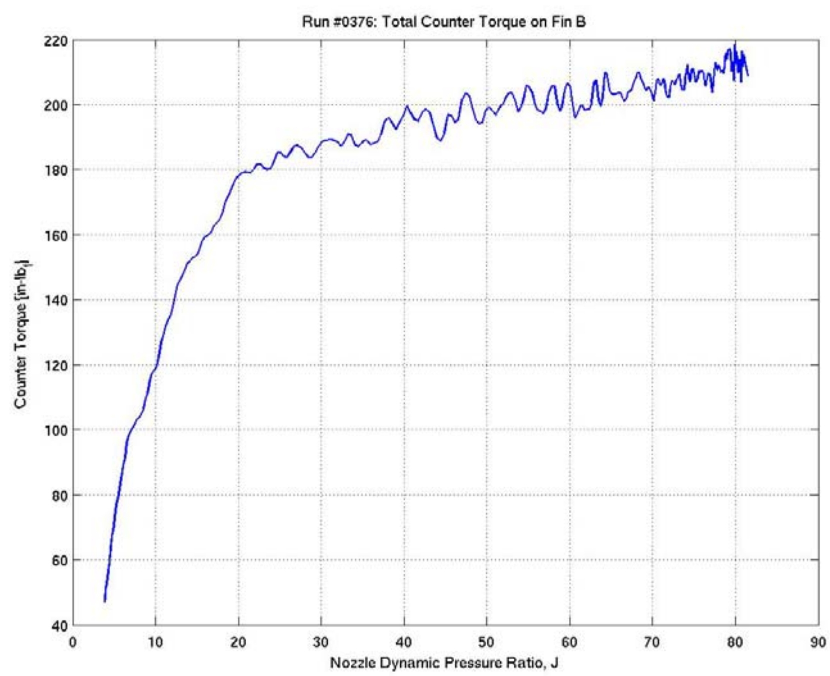

Figure 11. Fin B, Total Counter Torque Versus Nozzle Dynamic Pressure Ratio for Run #0376.

The seventh plot is the total counter torque on the FBR versus J. This includes the pressure torque and the torque measured by the aerodynamic balance for comparison; see Figure 12.

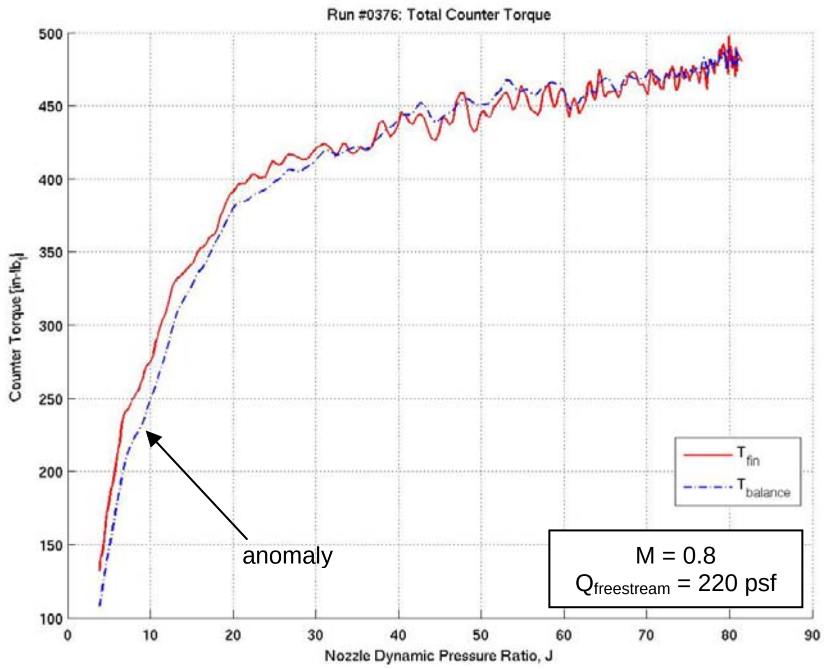

Figure 12. Fin and Balance Counter Torque Vs. Nozzle Dynamic Pressure Ratio for Run #0376.

For each analyzed run, we witnessed the same occurrence at approximately J = 7. The arrow on Figure 12 shows this anomaly in both the balance-measured torque and the reduced-pressure torque. Valuable information could be obtained by understanding what physically occurs at Nozzle Dynamic Pressure Ratios of about 7.

The seven plots shown above are the results reported for a single run. We performed the integration methodology on 8 additional runs; see Table 2. At 0° angle of attack, roll angle has no effect, so essentially each set of test conditions has a repeat run at a body roll angle of 45°. We only integrated one of the pair to minimize redundancy.

Table 2. Runs with Reported Results.

Integration Run	Configuration	$Q_{freestream}$	Mach	Experimental Run
1	2	220	0.8	0376
2			0.95	0386
3			1.1	0396
4		440	0.8	0426
5			0.95	0416
6			1.1	0406
7		880	0.8	0451
8			0.95	0461
9			1.1	0471

The most helpful results are from plots like Figure 12, so Figures 13 to 20 are similar to Figure 12 for each respective run; they show the two torque curves for the runs in Table 2.

The solid red lines are the torque determined from our integration methodology, and the dashed blue lines are the torque directly measured with the aerodynamic balance.

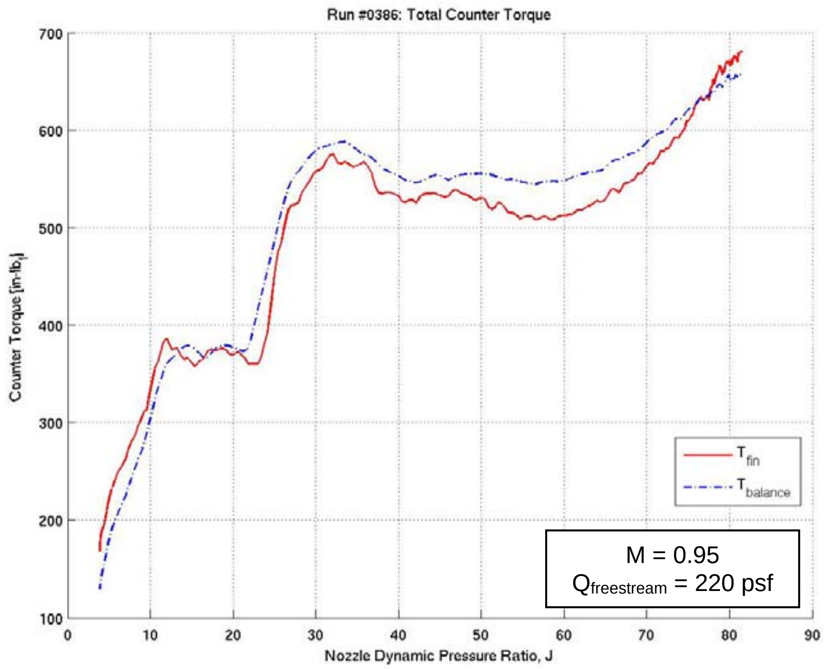

Figure 13. Fin and Balance Counter Torque Vs. Nozzle Dynamic Pressure Ratio for Run #0386.

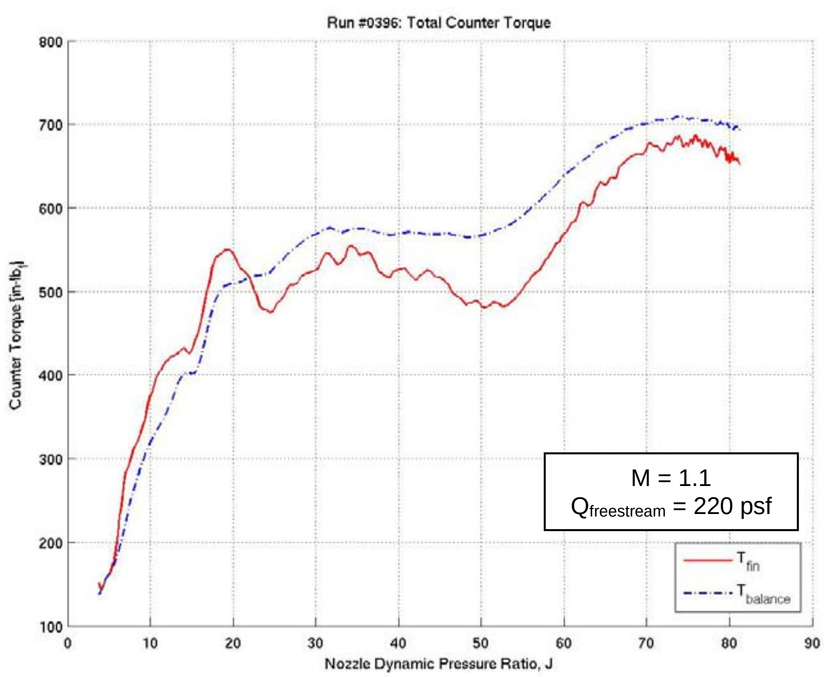

Figure 14. Fin and Balance Counter Torque Vs. Nozzle Dynamic Pressure Ratio for Run #0396.

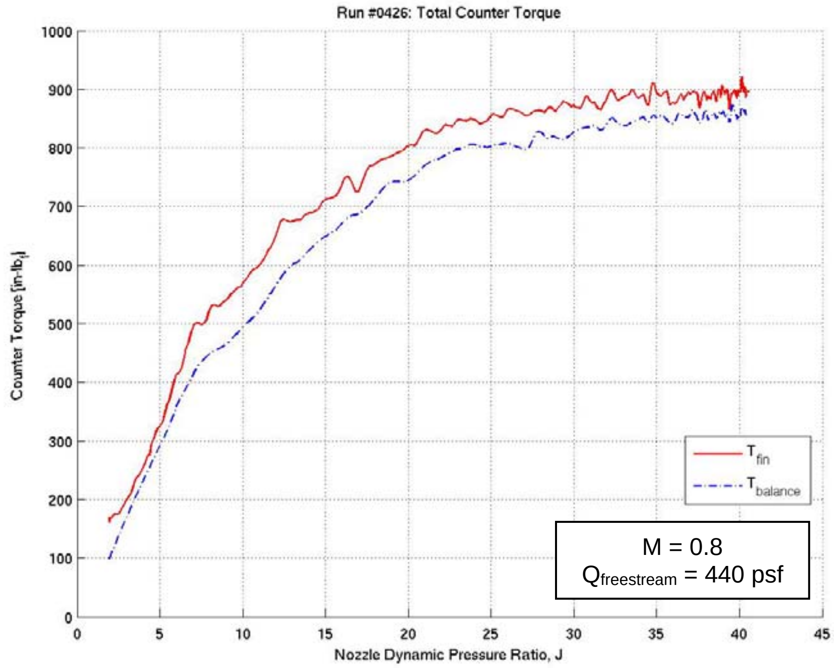

Figure 15. Fin and Balance Counter Torque Vs. Nozzle Dynamic Pressure Ratio for Run #0426.

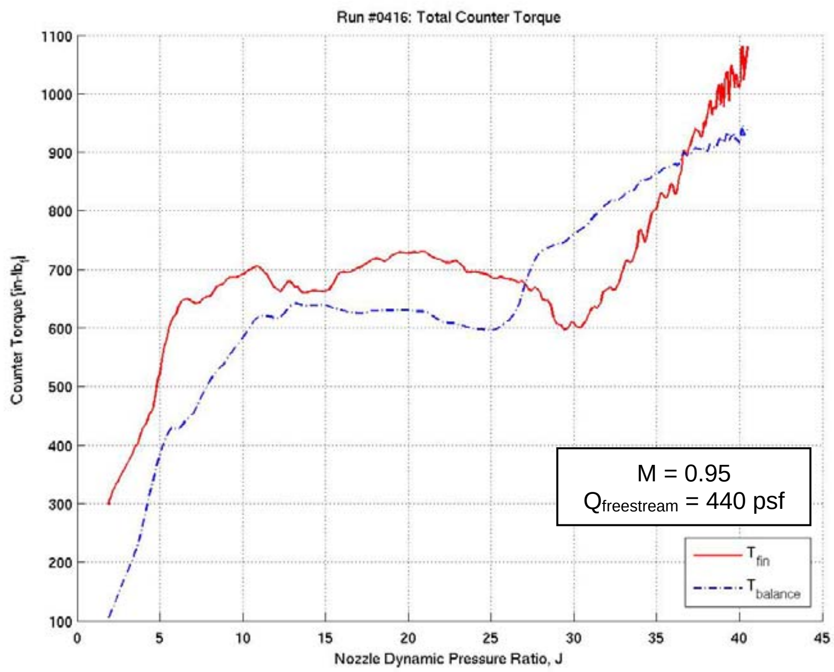

Figure 16. Fin and Balance Counter Torque Vs. Nozzle Dynamic Pressure Ratio for Run #0416.

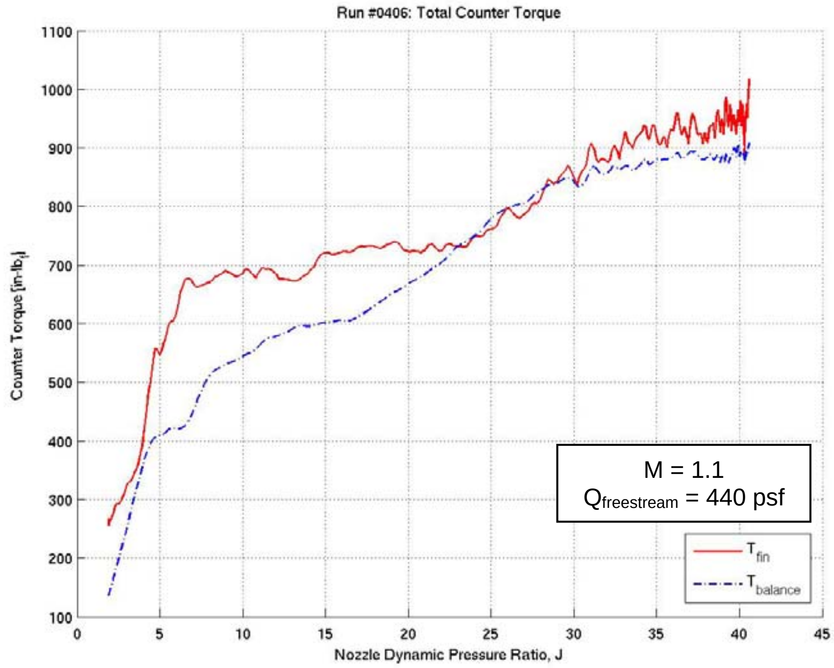

Figure 17. Fin and Balance Counter Torque Vs. Nozzle Dynamic Pressure Ratio for Run #0406.

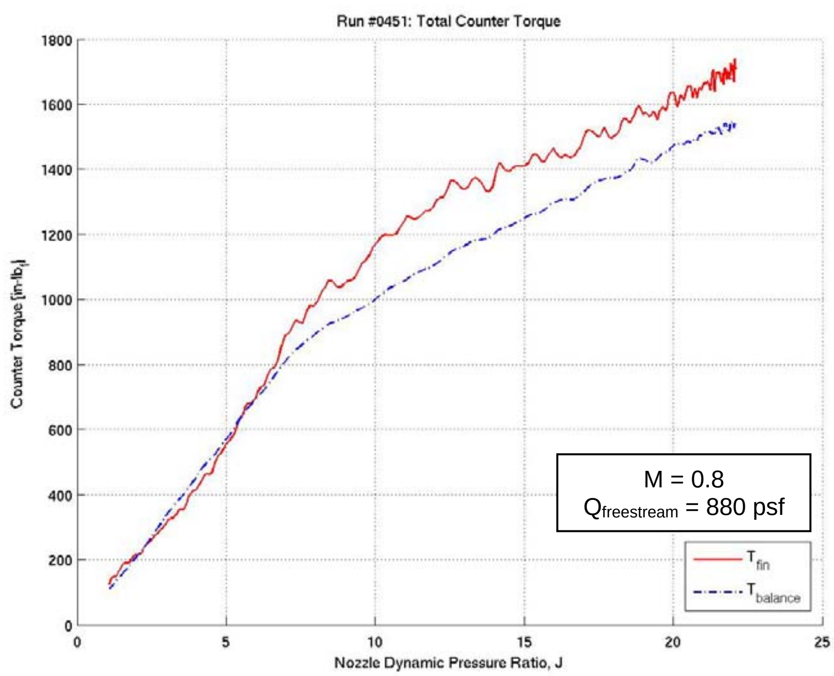

Figure 18. Fin and Balance Counter Torque Vs. Nozzle Dynamic Pressure Ratio for Run #0451.

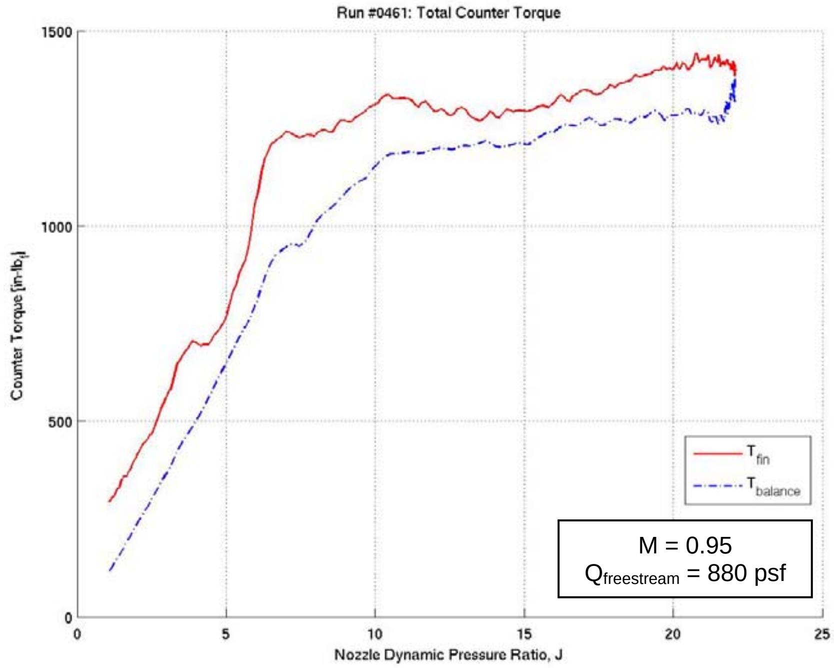

Figure 19. Fin and Balance Counter Torque Vs. Nozzle Dynamic Pressure Ratio for Run #0461.

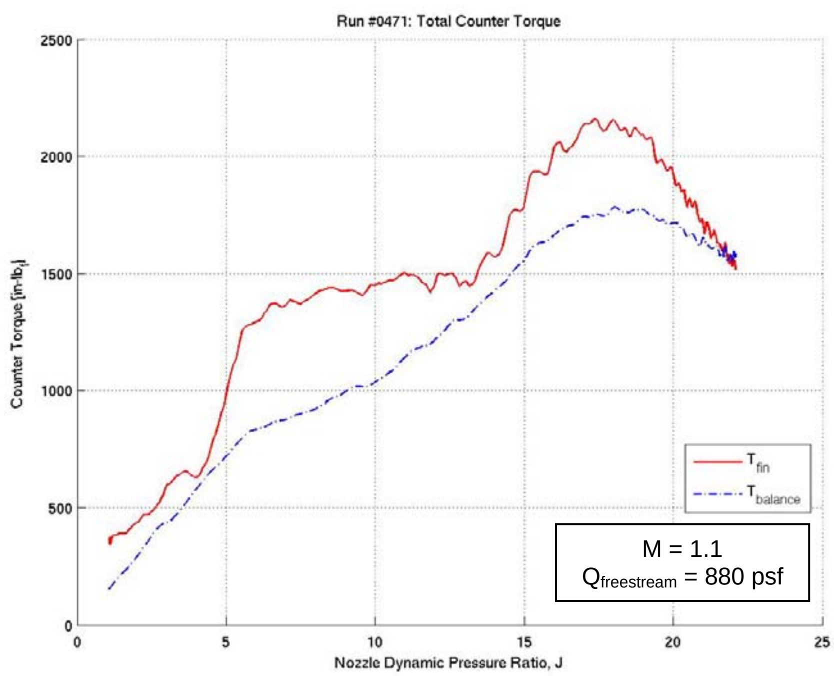

Figure 20. Fin and Balance Counter Torque Vs. Nozzle Dynamic Pressure Ratio for Run #0471.

The scales in Figures 8 and 11 lead us to believe that the majority of the pressure induced torque occurs on Fin B and, consequently, on its symmetric counterpart. Figure 12 is the most accurate of our results. In Figure 12, the torque determined from the fin pressure data and the torque measured directly from the aerodynamic balance seem to overlap. If this is accurate, it means virtually all of the spin-reducing, counter-torque is caused by VFI; however, their actual agreement is difficult to predict until we quantify the uncertainty. The result shown is for the case with the lowest dynamic pressure and lowest Mach number. In Figures 12, 15, and 18, you can see the changes as $Q_{freestream}$ is increased and M is held constant. In Figures 12, 13, and 14, you can see the changes as $Q_{freestream}$ is held constant and M is increased. When these two parameters change, the results are noticeably less consistent. The reason for this discrepancy likely is due to the assumptions surrounding Equation 8. We have assumed that the physics within the tapped region is indicative of the physics over the entire fin. We believe that the fin's leading edge is the most effective, and consequently, the fin's trailing edge is the least effective. The most valuable information from these comparisons is the validation of our methodology. It produces results qualitatively consistent with those from the aerodynamic balance, so we can now supplement the balance-derived counter torque values with the results from the integration of the pressure data. Pressure data are more detailed than the net torque; we can investigate phenomenon on a much finer scale.

Conclusions and Future Work

In conclusion, we have only begun to analyze the effects of vortex-induced fin pressure on roll torque. VFI is the primary phenomenon that reduces spin rate of a FBR in flight. The wind tunnel experimental data was reduced with a simple, fundamental integration methodology, but the crux of the project was to completely automate the procedure. The entire process is now coded in MATLAB so further reduction can be accomplished with a few simple commands. The MATLAB program outputs results in the form of seven plots for each run; the plots can be used for comparing different flow conditions and model geometries.

The next steps in understanding the effects of VFI on spin-reducing counter-torque are to perform the same analysis at the many other test conditions. Currently, only the basic conditions have been studied; virtually no information was studied to understand the effects of changing the FBR's geometry and angle of attack. Also, valuable information could be obtained by analyzing the pressure gradients along the surface and determining the spin-reducing effects of different regions of a fin; this requires combining the fin pressure results with the Particle Image Velocimetry (PIV) results from NASA Ames. The analysis of different parts of a fin will lead to a better approximation than Equation 8 and, consequently, a better model.

References

1. Peterson, C. W., Wolfe, W. P., and Payne, J. L., "Experiments and Computations of Roll Torque Induced by Vortex-fin Interaction," Paper No. AIAA-2004-1069, 42nd AIAA Aerospace Sciences Meeting, Reno, NV, January 2004.
2. Peterson, C. W., Henfling, J. F., Erven, R. J., and McWherter-Payne, M. A., "Experimental Characterization of Supersonic Spin Motor Nozzle Flow," Paper No. AIAA-2004-1295, 42nd AIAA Aerospace Sciences Meeting, Reno, NV, January 2004.
3. "MATLAB – The Language of Technical Computing," The Math Works, Inc., Natick, MA, November 2000.
4. Beyer, William; *Standard Mathematical Tables*, 25th Edition; CRC Press; West Palm Beach, FL; 1978; pp. 294.

Appendix A: MATLAB M-Files

VFI_torques.m

```
clear all
clc

%global variable declaration
global dir run J FA_p FB_p tap_coords fin_area

%conversions
psf2psi = 0.0069444;%convert from psf to psi

%user input
dir = input('Enter directory name (ex. C2_Q440_M8):  ','s');%the directory with DAT
files
run = input('Enter the run number (ex. 0426):  ','s');%the first DAT file

%file input
data = load([dir '/RUN0' run '.dat']);%loads entire DAT file
tap_coords = load(['fin_taps.dat']);%loads entire taps file that contains
coordinates for the taps

%calculating area of whole fin
fin_dim = load('fin_shape.dat');%loads coordinates of fin edge points
fin_x = fin_dim(:,1);%maintains edge x-coordinates
fin_y = fin_dim(:,2);%maintains edge y-coordinates
fin_area = polyarea(fin_x,fin_y);%calculates area of whole fin

%data filter
J = data(:,427);%maintains Nozzle Dynamic Pressure Ratio from DAT file, in column
427
FA_p = data(:,[87:134]) * psf2psi;%maintains 48 columns of pressure data for Fin A
FB_p = data(:,[135:182]) * psf2psi;%maintains 48 columns of pressure data for Fin B
T_roll = abs(data(:,29));%maintains Roll Torques from Aerodynamic Balance

%function calls
T_FAodd = FAodd;%calls function FAodd stores fin pressure, roll torques on Fin A,
Odd
T_FBodd = FBodd;%calls function FBodd stores fin pressure, roll torques on Fin B,
Odd
T_FAeven = FAeven;%calls function FAeven stores fin pressure, roll torques on Fin
A, Even
T_FBeven = FBeven;%calls function FBeven stores fin pressure, roll torques on Fin
B, Even

%calculations
T_pressure = 2*(T_FAodd + T_FAeven + T_FBodd + T_FBeven);
%determines total torque from fin pressures, factor of 2 is for the other 2
symmetric fins
T_FA = T_FAodd + T_FAeven;%determines total torque on Fin A
T_FB = T_FBodd + T_FBeven;%determines total torque on Fin B

%plots both balance-measured torque and fin pressure torque on same axis
%and saves it in the same directory with DAT file
hold on
plot(J,T_pressure, 'r', 'LineWidth', 1);
plot(J,T_roll,'b-.', 'LineWidth', 1);
grid on
legend('T_f_i_n', 'T_b_a_l_a_n_c_e',0);
title(['Run #' run ': Total Counter Torque']);
```

```
xlabel('Nozzle Dynamic Pressure Ratio, J');
ylabel('Counter Torque [in-lb_f]');
saveas(gcf,[dir '/RUN0' run 'total'],'jpg');
hold off
close

%plots total roll torque on Fin A versus J and saves it in the directory with
%the DAT files
plot(J, T_FA, 'LineWidth', 1);
grid on
title(['Run #' run ': Total Counter Torque on Fin A']);
xlabel('Nozzle Dynamic Pressure Ratio, J');
ylabel('Counter Torque [in-lb_f]');
saveas(gcf,[dir '/RUN0' run 'finAtotal'],'jpg');
close

%plots total roll torque on Fin B versus J and saves it in the directory with
%the DAT files
plot(J, T_FB, 'LineWidth', 1);
grid on
title(['Run #' run ': Total Counter Torque on Fin B']);
xlabel('Nozzle Dynamic Pressure Ratio, J');
ylabel('Counter Torque [in-lb_f]');
saveas(gcf,[dir '/RUN0' run 'finBtotal'],'jpg');
close
```

FAodd.m

```
function T_FAodd = FAodd

%global variable declaration
global dir run J FA_p tap_coords fin_area

%data filter
p = FA_p(:,[1:2:47]);%maintains pressure data for Fin B, odd pressure taps
x = tap_coords(:,1);%maintains x coordinates for Fin A, odd pressure taps
y = tap_coords(:,2);%maintains y coordinates for Fin A, odd pressure taps
z = tap_coords(:,3);%maintains z coordinates for Fin A, odd pressure taps

%function calls
[vertices, area_vectors, r, a_surf] = vert_and_area(x, y, z);%calls function to
determine triangle vertices,
...area vectors, triangle centroids, and area on Fin A, Odd within tapped region
p_triangle = triangle_pressures(p, vertices);%calls function to determine mean
pressures for each triangle
T_tot = moment(p_triangle, area_vectors, r);%calls function to determine torque for
the entire surface, Fin A:Odd

%calculations
T_tot_x = -1.*T_tot(:,1);

%plots roll torque for Fin A, Odd versus J and saves it in directory with
%DAT file
plot(J, T_tot_x, 'LineWidth', 1)
grid on
ylabel('Counter Torque [in-lb_f]')
xlabel('Nozzle Dynamic Pressure Ratio, J');
title(['Run #' run ': Fin A, Odd Taps']);
saveas(gcf,[dir '/RUN0' run 'finAodd'],'jpg');
close
```

```
%scales torque to approximate the torque over the entire Fin A, Odd
T_FAodd = T_tot_x*(fin_area/a_surf);
```

FAeven.m

```
function T_FAeven = FAeven

%global variable declaration
global dir run J FA_p tap_coords fin_area

%data filter
p = FA_p(:,[2:2:48]);%maintains pressure data for Fin A, even pressure taps
x = tap_coords(:,4);%maintains x coordinates for Fin A, even pressure taps
y = tap_coords(:,5);%maintains y coordinates for Fin A, even pressure taps
z = tap_coords(:,6);%maintains z coordinates for Fin A, even pressure taps

%function calls
[vertices, area_vectors, r, a_surf] = vert_and_area(x, y, z);%calls function to
determine triangle vertices,
...area vectors, triangle centroids, and area on Fin A, Even within tapped region
p_triangle = triangle_pressures(p, vertices);%calls function to determine mean
pressures for each triangle
T_tot = moment(p_triangle, area_vectors, r);%calls function to determine torque for
the entire surface, Fin A:Even

%calculations
T_tot_x = T_tot(:,1);

%plots roll torque for Fin A, Even versus J and saves it in directory with
%DAT file
plot(J, T_tot_x, 'LineWidth', 1)
grid on
ylabel('Counter Torque [in-lb_f]')
xlabel('Nozzle Dynamic Pressure Ratio, J');
title(['Run #' run ': Fin A, Even Taps']);
saveas(gcf,[dir '/RUN0' run 'finAeven'],'jpg');
close

%scales torque to approximate the torque over the entire Fin A, Even
T_FAeven = T_tot_x*(fin_area/a_surf);
```

FBodd.m

```
function T_FBodd = FBodd

%global variable declaration
global dir run J FB_p tap_coords fin_area

%data filter
p = FB_p(:,[1:2:47]);%maintains pressure data for Fin B, odd pressure taps
x = tap_coords(:,7);%maintains x coordinates for Fin B, odd pressure taps
y = tap_coords(:,8);%maintains y coordinates for Fin B, odd pressure taps
z = tap_coords(:,9);%maintains z coordinates for Fin B, odd pressure taps

%function calls
[vertices, area_vectors, r, a_surf] = vert_and_area(x, y, z);%calls function to
determine triangle vertices,
...area vectors, triangle centroids, and area on Fin B, Odd within tapped region
```

```
p_triangle = triangle_pressures(p, vertices);%calls function to determine mean
pressures for each triangle
T_tot = moment(p_triangle, area_vectors, r);%calls function to determine torque for
the entire surface, Fin B:Odd

%calculations
T_tot_x = -1.*T_tot(:,1);

%plots roll torque for Fin B, Odd versus J and saves it in directory with
%DAT file
plot(J, T_tot_x, 'LineWidth', 1);
grid on
ylabel('Counter Torque [in-lb_f]')
xlabel('Nozzle Dynamic Pressure Ratio, J');
title(['Run #' run ': Fin B, Odd Taps']);
saveas(gcf,[dir '/RUN0' run 'finBodd'],'jpg');
close

%scales torque to approximate the torque over the entire Fin B, Odd
T_FBodd = T_tot_x*(fin_area/a_surf);
```

FBeven.m

```
function T_FBeven = FBeven

%global variable declaration
global dir run J FB_p tap_coords fin_area

%data filter
p = FB_p(:,[2:2:48]);%maintains pressure data for Fin B, even pressure taps
x = tap_coords(:,10);%maintains x coordinates for Fin B, even pressure taps
y = tap_coords(:,11);%maintains y coordinates for Fin B, even pressure taps
z = tap_coords(:,12);%maintains z coordinates for Fin B, even pressure taps

%function calls
[vertices, area_vectors, r, a_surf] = vert_and_area(x, y, z);%calls function to
determine triangle vertices,
...area vectors, triangle centroids, and area on Fin B, Even within tapped region
p_triangle = triangle_pressures(p, vertices);%calls function to determine mean
pressures for each triangle
T_tot = moment(p_triangle, area_vectors, r);%calls function to determine torque for
the entire surface, Fin B:Even

%calculations
T_tot_x = T_tot(:,1);

%plots roll torque for Fin B, Even versus J and saves it in directory with
%DAT file
plot(J, T_tot_x, 'LineWidth', 1)
grid on
ylabel('Counter Torque [in-lb_f]')
xlabel('Nozzle Dynamic Pressure Ratio, J');
title(['Run #' run ': Fin B, Even Taps']);
saveas(gcf,[dir '/RUN0' run 'finBeven'],'jpg');
close

%scales torque to approximate the torque over the entire Fin B, Even
T_FBeven = T_tot_x*(fin_area/a_surf);
```

vert_and_area.m

```
function [vertices, area_vectors, r, a_surf] = vert_and_area(x,y,z)

%calculations
c = sqrt(y.^2 + z.^2);%computes value to use in 2D triangulation from 3D
coordinates
t = delaunay(x,c);%triangulates points
tri_xcoord = x(t);%creates matrix of x coordinates for triangles
tri_ycoord = y(t);%creates matrix of y corrdinates for triangles
tri_zcoord = z(t);%creates matrix of z coordinates for triangles

%function call
area = tri3Darea(tri_xcoord, tri_ycoord, tri_zcoord);%computes area magnitudes of
triangles

%calculations
avg_a = mean(area);%calculates the mean area
norm_a = area./avg_a;%normalizes the areas
ix = find(norm_a < 0.01);%finds the indices of areas less than 1% of the mean

%deletions and adjustments
t(ix,:) = [];%deletes triangles with areas less than 1% of the mean
area(ix) = [];%deletres areas less than 1% of the mean
tri_xcoord = x(t);%creates NEW matrix of x coordinates for triangles
tri_ycoord = y(t);%creates NEW matrix of y coordinates for triangles
tri_zcoord = z(t);%creates NEW matrix of z coordinates for triangles

%calculations
a_surf = sum(area);%determines the area of the tapped region
rx = mean(tri_xcoord,2);%determines mean x coordinates for each triangle (i.e. the
x coordinate of the triangle's centroid)
ry = mean(tri_ycoord,2);%determines mean y coordinates for each triangle (i.e. the
y coordinate of the triangle's centroid)
rz = mean(tri_zcoord,2);%determines mean z coordinates for each triangle (i.e. the
z coordinate of the triangle's centroid)
r = [rx ry rz];%creates a matrix of centroid locations for each triangle

for i = 1:length(t)%steps through triangles
    n = [tri_xcoord(i,2)-tri_xcoord(i,1) tri_ycoord(i,2)-tri_ycoord(i,1)
tri_zcoord(i,2)-tri_zcoord(i,1)];%creates vector from one triangle leg
    m = [tri_xcoord(i,3)-tri_xcoord(i,1) tri_ycoord(i,3)-tri_ycoord(i,1)
tri_zcoord(i,3)-tri_zcoord(i,1)];%creates vector from another triangle leg
    s(i,:) = cross(n,m);%cross product of two adjacent triangle legs
    b(i) = norm(s(i,:));%norm of newly found vector that is perpendicular to
triangle plane
    p(i,:) = s(i,:)./b(i);%determines unit vector for vector that is perpendicular
to triangle plane
    q(i,:) = area(i).*p(i,:);%determines area vector for each triangle plane
end

%variable return
vertices = t;%triangle vertices (N x 3), N triangles, 3 vertices
area_vectors = q;%area vectors (N x 3), N triangle, 3 vector components
```

tri3Darea.m

```
function area = tri3Darea(x,y,z)

%from CRC Standard Mathematical Tables 25th Edition p. 294
```

```
%calculation region
for i = 1:length(x)%stepping through triangles
    m = x(i,:);%temporary variable with x coordinates for the 3 vertices of 1
triangle
    n = y(i,:);%temporary variable with y coordinates for the 3 vertices of 1
triangle
    o = z(i,:);%temporary variable with z coordinates for the 3 vertices of 1
triangle

    a = [n(1) o(1) 1
         n(2) o(2) 1
         n(3) o(3) 1];%the first matrix in area equation

    b = [o(1) m(1) 1
         o(2) m(2) 1
         o(3) m(3) 1];%the second matrix in area equation

    c = [m(1) n(1) 1
         m(2) n(2) 1
         m(3) n(3) 1];%the third matrix in area equation

    d(i) = 0.5*(sqrt((det(a))^2 + (det(b))^2 + (det(c))^2));%calculating the area
with the equation
end

%returning variable to parent function
area = d;
```

triangle_pressures.m

```
function p_triangle = triangle_pressures(p, v)

%calculations
for i = 1:size(p,1)%varying the values of J
    a = p(i,:);%temporary variable holding the 24 pressure values for the 24
pressure taps at a single J value
    b = a(v);%temporary variable creating a matrix of Nx3, where N is the number of
triangles and the 3 is the vertices of each triangle
    for j = 1:size(b,1)% varying triangle
        c(i,j) = mean(b(j,:));%calculating the mean pressure at the three vertices
for 1 triangle
    end
end

%variable return to parent function
p_triangle = c;
```

moment.m

```
function T_tot = moment(p, a, r)

%calculation region
for i = 1:size(p,1)%varying values of J
    for j = 1:size(a,1)%varying triangle
        for k = 1:3%varying vector component
            F(i,k,j) = p(i,j)*a(j,k);%calculating force vectors for a single J
value, for a single triangle
```

```
        end
        M(i,:,j) = cross((r(j,:)), F(i,:,j));%calculating moment vectors for a
single J value, for a single triangle
    end
end

M_temp = abs(M);%ensuring that all values have the same sign
T_tot = sum(M_temp,3);%summing the torques for each triangle, returns 1 torque
vector
```

Distribution

1	MS 0825	B. Hassan, 9115
1	MS 0825	C. W. Peterson, 9115
1	MS 0825	W. P. Wolfe, 9115
1	MS 0834	J. E. Johannes, 9112
1	MS 0834	S. J. Beresh, 9112
2	MS 0899	Technical Library, 9616
1	MS 9018	Central Technical Files, 8945-1

1 Shazib Vijlee
2207 Wickersham Lane #401
Austin, Texas 78741