

M. Carr
J. L. Volakis

Semi-annual Report

**Doppler Shift Analysis of
Rotating Helicopter Blades**

Sikorsky Aircraft Corporation
1201 South Avenue
Bridgeport, CT 06604

July, 1997

PROJECT INFORMATION

PROJECT TITLE: Fast Integral Algorithms for Rotorcraft Blade Scattering in the VLF and UHF Bands

REPORT TITLE: Doppler Shift Analysis of Rotating Helicopter Blades

U-M REPORT No.: 374968-1-T

CONTRACT

START DATE: 1 January 1997

END DATE: 31 December 1997

DATE: July 1997
Semiannual Report

SPONSOR: Daniel C. Ross
Sikorsky Aircraft Corp.
Mail Stop B101A
1201 South Ave.
Bridgeport, CT. 06604
Phone: (203) 384-7010
Fax: (203) 384-6701
Email: dross@sikorsky.com

SPONSOR

CONTRACT No.: P.O. N5600918

U-M PRINCIPAL

INVESTIGATOR: John L. Volakis
EECS Dept.
University of Michigan
1301 Beal Ave
Ann Arbor, MI 48109-2122
Phone: (313) 764-0500 FAX: (313) 647-2106
volakis@umich.edu
<http://www-personal.engin.umich.edu/~volakis/>

PROJECT PEOPLE: Michael Carr, Mikhael Smelianskiy, John Volakis

Doppler Shift Analysis of Rotating Helicopter Blades: Interim Report 3

Michael Carr (mcarr@umich.edu)

John Volakis (volakis@umich.edu)

Radiation Laboratory
Department of Electrical Engineering and Computer Science
The University of Michigan at Ann Arbor
Ann Arbor, Michigan 48109

July 23, 1997

1. INTRODUCTION

In our previous report, we examined enhanced TF analysis techniques, namely the adaptive Gaussian and adaptive optimal kernel (AOK) Gaussian. An improved spectrogram employing a Blackman window was also analyzed. Because the AOK method gave the best results, it will be used for all future TF analysis.

In this second interim report, the mathematics of the AOK are examined closely. We also discuss the implementation of a TF visualization tool. A parallel, optimized PTRI code is developed for the SGI PowerChallenge, and this code is used to simulate helicopter blade pitching.

2. ADAPTIVE OPTIMAL KERNEL (AOK)

For a given signal $s(t)$, any time-frequency distribution, $P(t,\omega)$, falling within Cohen's bilinear class can be represented by

$$P(t,\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} A(\theta, \tau) \Phi(\theta, \tau) e^{-j\theta\alpha - j\tau\omega} d\theta d\tau \quad (1)$$

where

$$A(\theta, \tau) = \int_{-\infty}^{\infty} s^* \left(t - \frac{\tau}{2} \right) s \left(t + \frac{\tau}{2} \right) e^{j\theta t} dt \quad (2)$$

The bilinear function $A(\theta,\tau)$ is called the ambiguity function (AF) for the signal, and is multiplied by the weighting function $\Phi(\theta,\tau)$, known as the kernel, to ultimately form the TFD. The AF represents the one-dimensional signal in two-dimensional time/frequency space, and because it is a bilinear transformation, it contains both auto-components and cross-components of the signal. These cross-components can reduce auto-component resolution and obscure the TFD. The purpose of the kernel in the

analysis is to attenuate these cross-components while emphasizing auto-components. Only after this is accomplished can an attractive and useful TFD be computed.

The best way to describe the process is through illustrations. Two standard benchmark signals used as a measure of TFD quality are Gaussian pulses

$$s_1(t) = e^{-a(t+T_0)^2} + e^{-a(t-T_0)^2} \quad (3)$$

and the sum of two linear-FM chirp signals

$$s_2(t) = e^{-a(t+T_0)^2 + jc(t+T_0)^2 - j\omega_1(t+T_0)} + e^{-a(t-T_0)^2 + jc(t-T_0)^2 - j\omega_2(t-T_0)} \quad (4)$$

Figures 1(a) and 1(b) show plots of s_1 and s_2 vs. time, while Figures 2(a) and 2(b) show plots of the ambiguity functions for s_1 and s_2 .

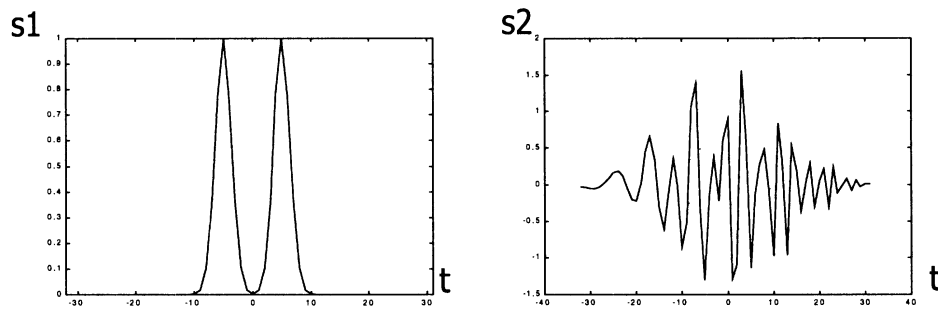


Figure 1: Plot of (a) s_1 and (b) s_2 vs. time

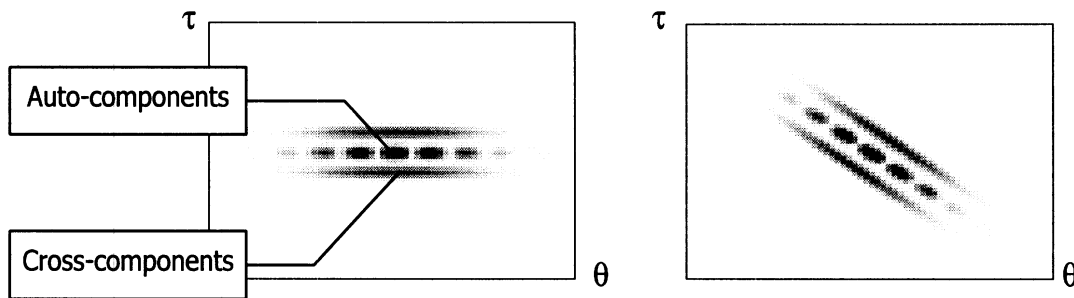


Figure 2: Ambiguity functions of (a) s_1 and (b) s_2

The Gaussian basis function employed by [4] creates Gaussians oriented along the θ (time) or τ (frequency) axis only, as shown in Figures 3(a) and 3(b). For s_1 , the θ -oriented Gaussian works very well, while for s_2 neither orientation fits. To analyze a large group of signals, therefore, it is necessary to develop an algorithm that can adapt its kernel to the orientation of the ambiguity function.

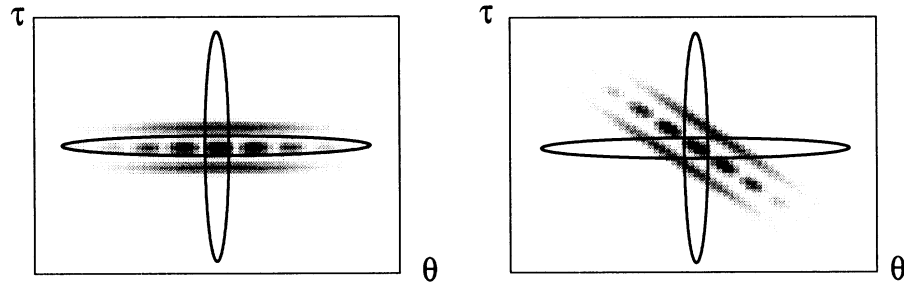


Figure 3: Gaussian basis functions of [4] overlaid onto (a) s_1 and (b) s_2

The optimal kernel Gaussian employed in our blade analysis is basically a *radial* Gaussian, defined by

$$\Phi(\theta, \tau) = e^{-(\theta^2 + \tau^2) / [2\sigma^2(\psi)]} \quad (5)$$

where

$$\psi = \arctan \frac{\tau}{\theta} \quad (6)$$

and the function $\sigma(\psi)$ controls the spread of the Gaussian at radian angle ψ . Figure 4(a) shows a typical spread function, while Figure 4(b) shows its associated Gaussian kernel. Once the kernel is defined in this manner, the problem becomes one of optimization to find a spread function that best attenuates the cross-components.

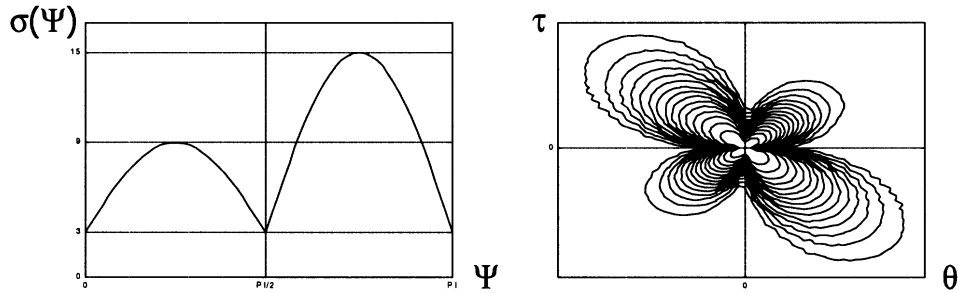


Figure 4: An example spread function (a) and its associated Gaussian kernel (b)

The equation to optimize is

$$\max_{\psi} \int_0^{2\pi} \int_0^{\infty} |A(r, \psi) \Phi(r, \psi)|^2 r dr d\psi \quad (7)$$

subject to the following constraints:

$$\Phi(r, \psi) = e^{-r^2 / 2\sigma^2(\psi)} \quad (8)$$

$$\frac{1}{2\pi} \int_0^{2\pi} \int_0^{\infty} |\Phi(r, \psi)|^2 r dr d\psi \leq \alpha \quad (9)$$

$$\alpha \geq 0 \quad (10)$$

where $A(r, \psi)$ is the ambiguity function in polar coordinates.

Constraint (8) limits the kernel to one of a Gaussian basis function, where the shape of the Gaussian is determined by the shaping function $\sigma(\psi)$. Since the AF is symmetric about the origin--

$$|A(r, \psi)|^2 = |A(r, \psi + \pi)|^2 \text{ -- } \sigma(\psi) \text{ only needs to be determined for the top-half of the kernel.}$$

Constraints (9) and (10) limit the volume of the optimal kernel to a user-specified amount α , determined, to some extent, through trial and error. It is constraint (9) that ultimately forces the kernel to behave as a low-pass filter, enclosing the auto-components rather than the cross-components. A low-pass kernel is desirable, because the auto-components tend to be clustered around the origin of the AF while cross-components fall away. Note that the constraints do not specify an angular orientation; unlike the Gaussian defined by [4], this Gaussian is free to take any radial shape.

Optimization criteria (7) dictates that the optimal kernel will be of a form maximizing the integral over the ambiguity plane. Therefore, the kernel must *maximize* this integral (optimization criteria) while *minimizing* its own area (Constraint (9)). It is for this reason that the kernel encircles the auto-components so well—surrounding both the auto- and cross-components would occupy valuable area while making little contribution to the integral due to the “valleys” between the two. Instead, the kernel surrounds only the auto-components, which tend to be clumped together. The whole procedure assumes that the auto- and cross-components are somewhat separated in the AF, as they are in most applications.

The remaining portion of the TF analysis is rather straightforward. Once the optimal kernel has been found, it is multiplied by the AF. Two Fourier transforms are applied, as shown in (1), to arrive at the final OK TFD. The procedure works best when the components of the signal are of linearly changing frequency, as shown in Figure 2. Rotating blades, however, produce a signal with frequencies that are changing *sinusoidally*, which in turn creates the complicated AF shown in Figure 5. One kernel alone cannot analyze this signal and produce a high-quality TFD.

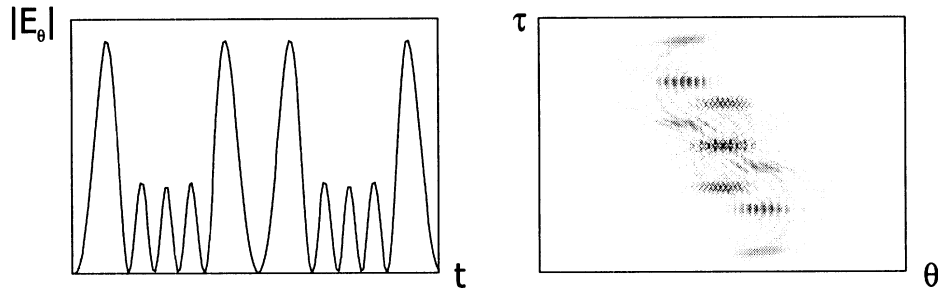


Figure 5: RCS of rotating 2.5λ wire (a) and its ambiguity function (b). Note that there is no way to separate the auto-correlation from the cross-correlation.

The adaptive optimal kernel fixes this problem by varying its kernel across the signal. It does this by employing a *short-time* ambiguity function, where, just as in the short-time Fourier transform, only a user-specified time-window of the entire signal is analyzed. The window is moved across the signal, and at each position the entire process is carried out to create a single column of the TFD. During these small regions of time, the sinusoidal signal can be linearly approximated, and the Gaussian kernel applied. Figure 6 shows a windowed ambiguity function, as it appears for the spinning wire case. The total signal was made up of 150 samples.

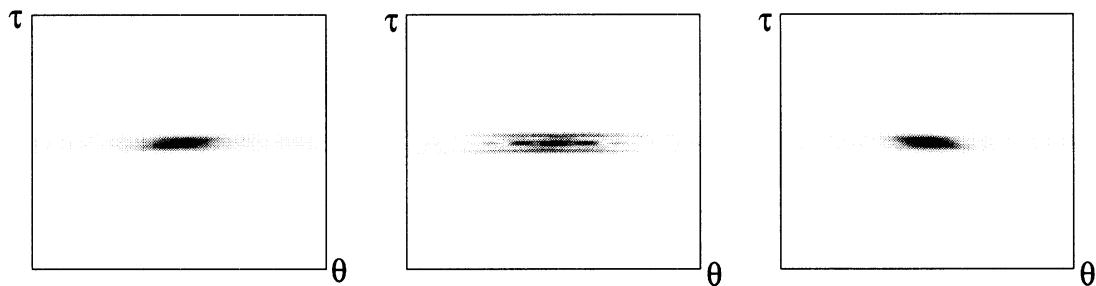


Figure 6: Some windowed ambiguity functions of the 2.5λ wire. (a) samples 1-17, (b) samples 22-38, (c) samples 59-75

Through knowledge of this procedure, we can make several conclusions. In order to approximate a portion of a sinusoid with a line, we must have a sufficient number of samples. The three ambiguity functions shown in Figure 6 contain *constant* frequencies—we see no tilting of the AF as we do in Figure 2(b). And, the frequencies contained are near-DC. This suggests that a window of 16 samples is much too small, and that a larger window should be used. For efficient Fourier transforms, however, the window size must be a power of 2, so we must insure that a window of size 32 isn't so large that the auto- and cross-components become mixed. A hybrid solution would be to increase the window size to 32 while increasing the size of the entire signal from 150 to 200 or 250 samples. More experiments need to be performed before a final conclusion can be reached.

3. HELICOPTER BLADE PITCHING

IMPLEMENTATION

Helicopters use blade “pitching” to direct their thrust forward, backward, left and right. This pitching occurs along the longitudinal axis of the blade, and changes as the blade rotates through its 360° arc. For instance, when the pilot wants to travel forward, the blades pitch to direct thrust toward the back of the helicopter.

In our simulation, a counter-clockwise spinning of the blade, without pitching and when the blade is located in the XY plane, is accomplished by holding theta constant at 90° and moving phi clockwise through a 360° arc. Because the geometry itself is not altered during the simulation, the impedance matrix needs to be filled only once. The computation of each look angle takes little time comparatively.

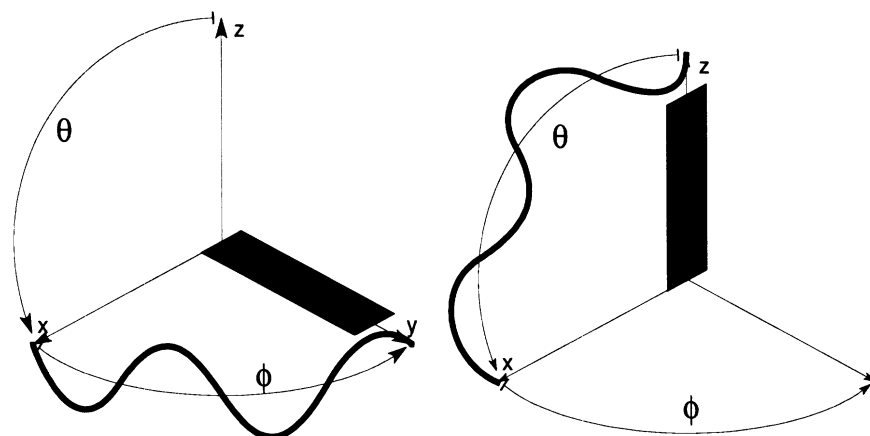


Figure 7: Blade in (a) XY and (b) XZ planes

It was desired to take this pitching action into account when analyzing the blades for their TF response. One might first suppose that the pitching can be accomplished by moving the incident theta angle sinusoidally, instead of holding it constant. However, with the blade located in the XY plane (see Figure 7(a)), the pitching action is impossible to replicate by simply altering the look angle. We see that the blade does indeed pitch if phi is held at 0° and theta varies up and down from 90°, but at a phi of 90°, the entire blade moves up and down as theta varies. This is obviously not the desired behavior.

There is at least one good reason to have the blade in the XY plane: simulation of a ground- or air-based radar where the incident wave is not exactly edge-on to the blade. In this type of simulation, it is necessary to tilt the entire blade instead of just rotating it around its own axis. Unfortunately, there is no way to do both types of analysis--tilting and pitching--at the same time. It may be possible, if necessary, to implement a rotation algorithm which can “rotate” the impedance matrix without the time-consuming matrix fill, but this needs to be looked at more carefully. If it is possible to rotate the impedance matrix quickly, the pitching can be done during the matrix rotation while the tilting is done using incident angles.

To give the blade its pitching action, we move it from the XY plane into the XZ plane, as shown in Figure 7(b). With this orientation, the phi angle is inversely proportional to the pitch angle, and the theta

angle is the instantaneous angle through which the blade is spinning. Rotation with pitching can be accomplished by moving theta through 360° while varying phi sinusoidally.

RESULTS

Appendices C and D contain results from each of the pitching simulations. Appendix C compares the TFDs where the spinning blade reaches a constant maximum pitch (45°) at varying rotation angles. Appendix D compares the TFDs where the spinning blade reaches varying maximum pitch angles at a constant rotation angle (90°). For each diagram, the pitching period of the blade was 1 Hz. Refer to Figure 7(b) for view angle reference when observing Appendices C and D. For all figures, the magnitude of the TFD is displayed in a logarithmic scale.

For each of these simulations, the basic shape of the TFD remains identical to that of the non-pitching blade throughout the series, because no part of the blade is spinning any faster or slower than it was before. The maximum amplitude of the TFD does change, however, because different amounts of radiation are now reflected from the target; blades with a larger surface area exposed to the radar source generally reflect more energy to it.

While these results are not as interesting as we would hope for, they do highlight an interesting observation—there is no black magic to the art of time-frequency analysis. The TFD of a signal is fairly upon observation of the geometry and its environment. In this case, the blade's speed was not altered. Therefore, we should expect no change in the basic shape of the TFD. The amplitude of the field returns from the blade, on the other hand, changed because of the pitch angles. So we should expect that the magnitudes of the components of the TFD would change somewhat.

4. LEAD-LAG ANALYSIS

IMPLEMENTATION

Lead-lag is the tendency for a helicopter blade to speed up and slow down during its rotation due to Coriolis effects. Mechanical resonations cause the blade to “flap,” or move upwards or downwards, thus changing the center of gravity. If the CG moves toward the center of the helicopter, the blade will experience a force pushing it faster. The blade experiences a slowing effect as the CG moves away from the center.

Because the lead-lag is directly proportional to the flapping of the blade, it is this flapping that is of main interest. For a specific geometry, data must be collected as to the frequency, magnitude, and phase of the flapping effect—from this can be derived the frequency, magnitude, and phase of the associated lead-lag. We have implemented lead-lag as a sinusoidal component added to the rotation of the blade. As the blade rotates through 360 degrees, this sinusoidal component either pushes it ahead or pulls it back, effectively speeding it up and slowing it down.

RESULTS

Because we are changing the speed of the blade directly, we should expect a shift in the basic shape of the TFD. This normally sinusoidal-shaped TFD should now contain “ripples” similar to the sinusoidal lead-lag component. If the lead-lag sinusoid is not exactly in phase with the blade’s rotation, major components in the TFD, such as the edge-on angle response, will move off of their normal locations.

Appendix E contains results from our lead-lag analysis. As the amplitude of the lead-lag sinusoid is varied, a proportionally sized ripple appears in the TFD.

5. PTRI: PARALLEL TRICODE (MOMENT METHOD) FOR SGI POWERCHALLENGE

Two simulation approaches are of interest in this project. The first, method of moments, is a very standardized integral equation code. The second, adaptive integral method (AIM), is a relatively new code which boasts large memory savings and speedups for nearly linear geometries. We were interested in making a time and memory comparison between the two methods. During our initial observations of the rotating blades, it was realized that our existing moment-method software, first written in 1991 and named **TRICODE**, needed large-scale enhancements before it could be fully utilized for TF analysis. We named the new code **PTRI**--short for parallel **TRICODE**.

The first improvement was to design the moment method code to read incident angles from a file rather than using the limited “cut” approach. When defining a cut, either phi or theta is held constant while the variable angle is rotated through some arc. In order to implement blade pitching, it was necessary to alter both phi and theta simultaneously. A separate code was developed to generate the proper file containing phi and theta angles, and then **PTRI** was given the ability to read and interpret this file. Saving these incident angles in a file also saves a great amount of time during code development and when using the same incident angles for several geometries, as the data defining the path need not be re-entered for each run. The format of this file is given in Appendix A.

Secondly, the code was designed around the client hardware—an SGI PowerChallenge with 16 parallel processors. Each section of the code was hand-parallelized for maximum speed-up using parallel computing. Other optimization techniques such as loop unrolling and scalar optimization were used as well.

Next, the outdated LINPACK matrix solver was replaced with a LAPACK solver supplied by SGI and optimized specifically for the PowerChallenge. Because the LAPACK solver has the capability to solve for multiple right-hand-sides, the excitation vector inside the **PTRI** code was increased to a matrix containing each look angle in a separate column.

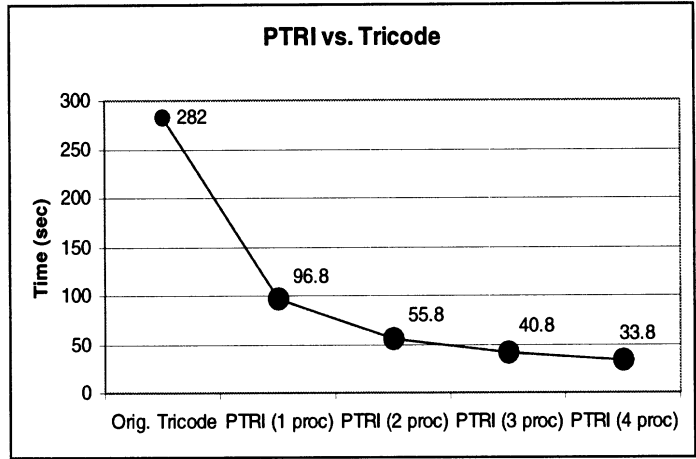


Figure 8: Comparison of TRICODE and PTRI

Figure 8 shows a comparison between **TRICODE** and the new **PTRI**. Both are running on an SGI PowerChallenge with 16 200 MHz R10000 processors. The machine has 2 GB of RAM and 6 GB of hard disk. The test geometry was the 400 MHz flat plate with 1767 unknowns. While the system shown was solved for 175 right-hand-sides, the number of RHS contributes little to the total computation time—the majority of time is spent in the LU factorization of the impedance matrix.

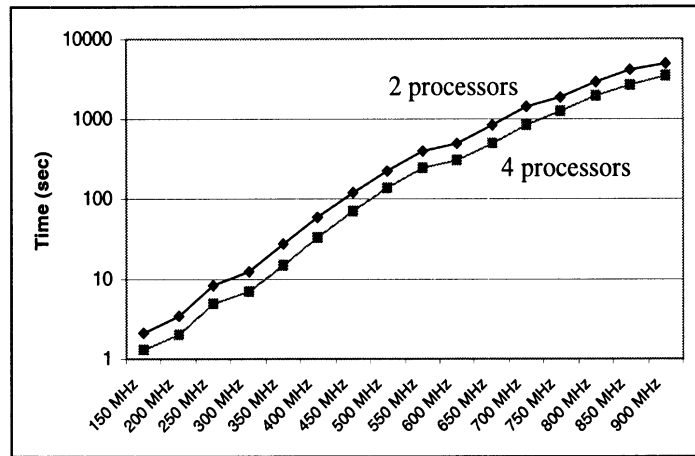


Figure 9: Solution times for flat plates

Figure 9 illustrates the complete solution times for the flat plate geometries ranging from 150 MHz to 900 MHz using **PTRI**. Again, 175 RHS were calculated. This graph will become especially important when comparing **PTRI** to an AIM approach, because AIM must refill its impedance matrices for each look angle.

A commercial solver known as FMS (Fast Matrix Solver) was also examined. FMS has the advantage that it can solve matrices out-of-core very quickly. For our systems, FMS decreased the matrix (in-core) inversion time to about 2/3 of LAPACK; these time savings are not significant enough to warrant the \$2000+ price tag of FMS. However, if extremely large geometries are to be examined, such as those

above 1 GHz, we recommend a move to the FMS solver because the impedance matrix will quickly become larger than available memory.

6. FUTURE STEPS

The main push of the next phase of the project involves examining improvements AIM will offer to the analysis. AIM capabilities will be added to the **PTRI** code, and the resulting benefits weighed for their actual usefulness. Because AIM is known to require refilling the impedance matrix for each look-angle, the speed-up of AIM will actually be a function of the size of the geometry, the number of look-angles, and the “flatness” of geometry itself.

Another issue which will be examined in the following weeks concerns the effects of lead-lag on the TFD. While lead-lag capability has been implemented in software, simulations have not yet been run to generate conclusive data. Discussion of the implementation as well as the results of any simulations will be included in the next report.

Table 1 shows future goals of the project and their target dates of completion.

Table 1: Future Project Goals

Target Date	Goal
	Weigh benefits of AIM and establish criteria for choosing a method
	Simulate realistic helicopter blade with AIM
	Implement materials in AIM

REFERENCES

1. Chatterjee, Volakis, and Kent. "Scattering by a Perfectly Conducting and a Coated Thin Wire Using a Physical Basis Model." IEEE Transactions on Antennas and Propagation. Vol. 40, No. 7, pp 761-769. July '92.
2. McCormack, Chris. "Time-Frequency Analysis in Radar Backscatter Problems." University of Michigan Graduate Thesis. '96.
3. Ling, Hao and Trintinalia. "Extraction of Waveguide Scattering Features Using Joint Time-Frequency ISAR." IEEE Microwave and Guided Wave Letters, Vol. 6, No. 1. pp. 10-12. January '96.
4. Chen, Dapang and Qian. "Signal Representation Using Adaptive Normalized Gaussian Functions." Signal Processing 36. pp. 1-11. '94.
5. Chen, Dapang and Qian. Joint Time-Frequency Analysis. Upper Saddle River, New Jersey. '96.
6. Baraniuk, Richard. "Optimal-Kernel Time-Frequency Representations." <http://jazz.rice.edu/software/TFA/optkernel.html>
7. Baraniuk, R. and Jones, "A Signal-Dependent Time-Frequency Representation: Optimal Kernel Design," IEEE Transactions on Signal Processing, vol. 41, no. 4, pp. 1589-1602. April '93.
8. Baraniuk, R. and Jones, "Signal-Dependent Time-Frequency Analysis Using a Radially Gaussian Kernel", Signal Processing, vol. 32, no. 3, pp. 263-284. June '93.
9. Jones, D. and Baraniuk, "An Adaptive Optimal-Kernel Time-Frequency Representation," IEEE Transactions on Signal Processing, vol. 43, no. 11, pp. 2361-2371. October '95.
10. Glisson, D., Rao, and Glisson. "Potential Integrals for Uniform and Linear Source Distributions on Polygonal and Polyhedral Domains." IEEE Transactions on Antennas and Propagation, Vol. AP-32, No. 3. March '84.
11. Haddad, Pamela. "Scattering by Non-planar Surfaces." University of Michigan Masters Thesis.
12. Rao, Sadasiva, Wilton, and Glisson. "Electromagnetic Scattering by Surfaces of Arbitrary Shape." IEEE Transactions on Antennas and Propagation, Vol. AP-30, No. 3. May '82.
13. Prouty, R. W. "Coriolis Effects and Main Rotors." Rotor & Wing. p. 49. January '97.

APPENDIX A – FORMAT OF INCIDENT ANGLE FILE

<n, number of incident angles>

<phi₁> <theta₁>

<phi₂> <theta₂>

.

.

.

<phi_n> <theta_n>

[EOF]

APPENDIX B - VISUALIZATION

An OpenGL-based visualization tool called **MESHVIEW** was developed to aid interpretation of TF results. Because it uses low-level OpenGL calls and the GLUT user interface, the code is machine independent and will run just as well on WindowsNT, Windows95, or any OpenGL-capable Unix machine. Currently, the machine must support OpenGL version 1.1, but the code can be altered to work with OpenGL version 1.0 if necessary.

MESHVIEW was originally developed as a cross-platform tool to view TRICODE (Moment Method) input files, but it has been expanded to include viewing surface currents and complete TF results. When used to view mesh files only, the program presents a screen such as that in Figure 10. This mode allows the engineer to check geometry or create images of the geometry under test.

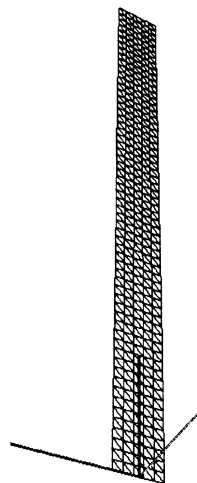


Figure 10: 300 MHz flat blade mesh

When the appropriate version of PTRI is used to create a “current file” (a file containing information about the currents at each node for each view angle), these currents can be superimposed upon the mesh using a color spectrum to reflect current magnitude. The display of the current is very useful for “seeing” where hot and cold spots lie along the blade. Viewing the current can also aid in mesh creation-- if it is known that a hot spot should lie along a certain discontinuity, yet it isn’t showing up on the model, smaller discretization is needed in that area. The true power of the program is seen in its animation mode, however, where the currents “move” along the object as the view angle changes. When used to superimpose currents on the mesh, the program presents a screen such as that in Figure 11.

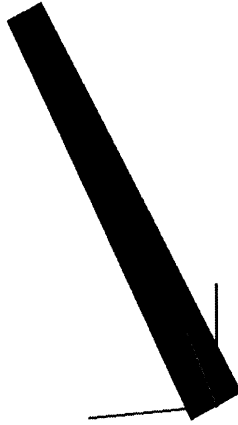


Figure 11: 300 MHz mesh with superimposed currents

Finally, the TF analysis generated by the AOK program can be loaded into the program and displayed as a colored, semi-transparent band around the object. The color and transparency of the band are altered to show the TF values at each look angle. When used to animate currents, a moving vertical black line highlights the portion of the band corresponding to the present view angle of the model. When used in this mode, **MESHVIEW** presents a screen such as that in Figure 12.

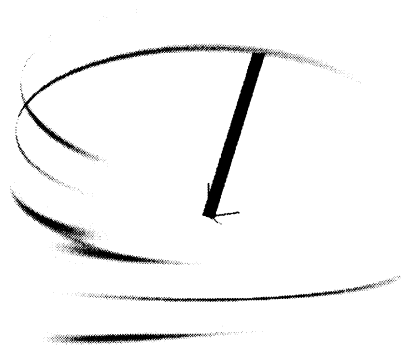
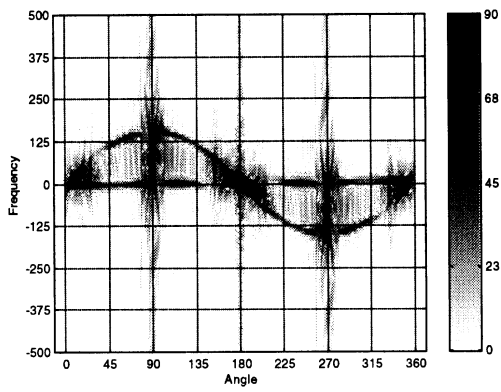
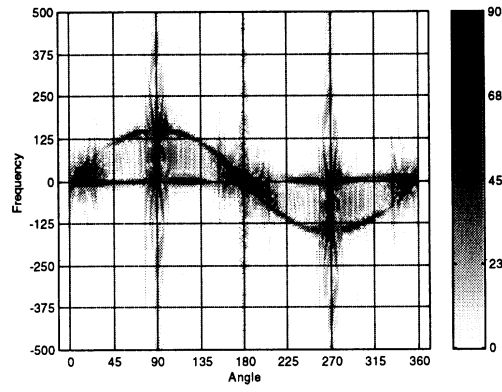


Figure 12: 300 MHz mesh with currents and TF

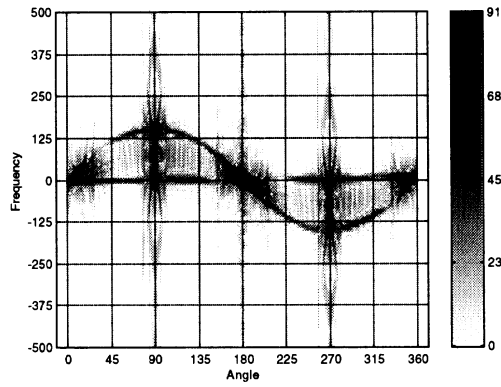
APPENDIX C – BLADE PITCHING RESULTS, EFFECTS OF MAXIMUM ANGLE



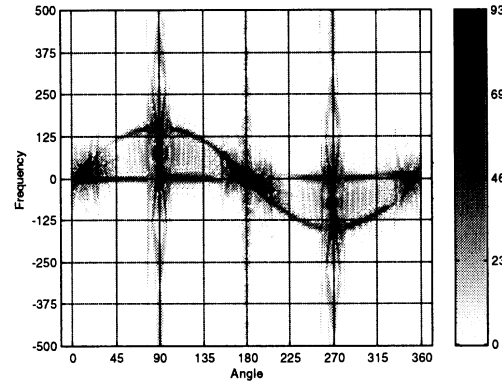
$\theta_{\max}=0$ $|\phi_{\max}|=45$ $|TF|_{\max}=83840$



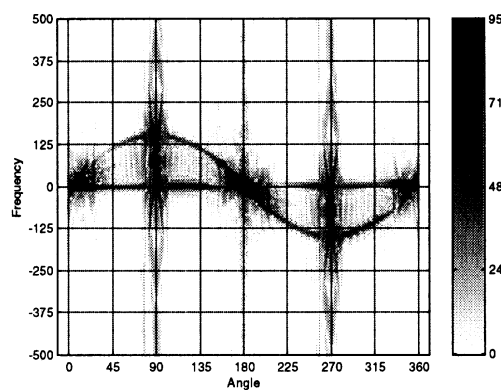
$\theta_{\max}=15$ $|\phi_{\max}|=45$ $|TF|_{\max}=83159$



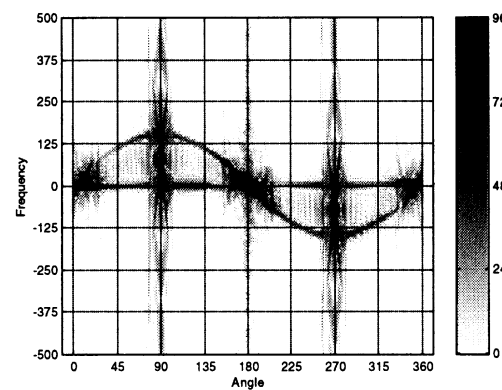
$\theta_{\max}=30$ $|\phi_{\max}|=45$ $|TF|_{\max}=85720$



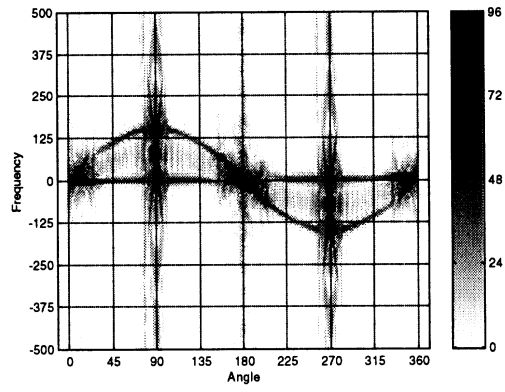
$\theta_{\max}=45$ $|\phi_{\max}|=45$ $|TF|_{\max}=105258$



$\theta_{\max}=60$ $|\phi_{\max}|=45$ $|TF|_{\max}=136226$

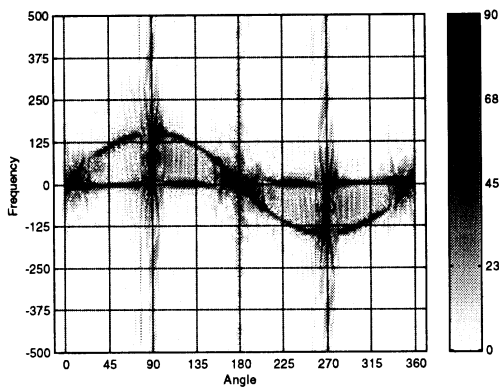


$\theta_{\max}=75$ $|\phi_{\max}|=45$ $|TF|_{\max}=152577$

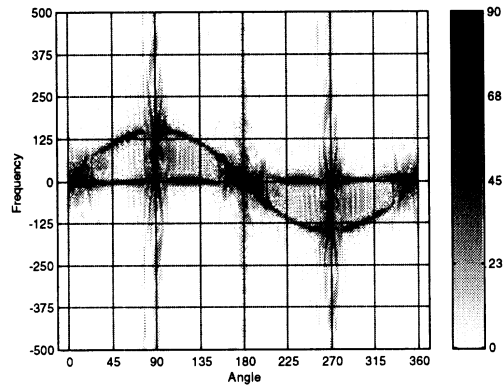


$\theta_{\max}=90$ $|\phi_{\max}|=45$ $|TF|_{\max}=154862$

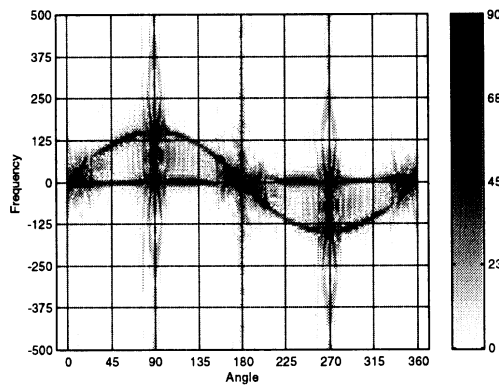
APPENDIX D – BLADE PITCHING RESULTS, EFFECTS OF MAXIMUM PITCH



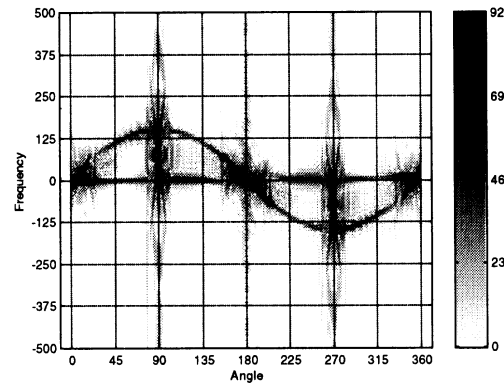
$\theta_{\max}=90$ $|\phi_{\max}|=0$ $|TF|_{\max}=83805$



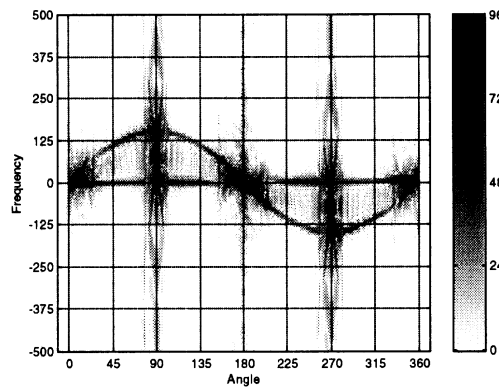
$\theta_{\max}=90$ $|\phi_{\max}|=10$ $|TF|_{\max}=83168$



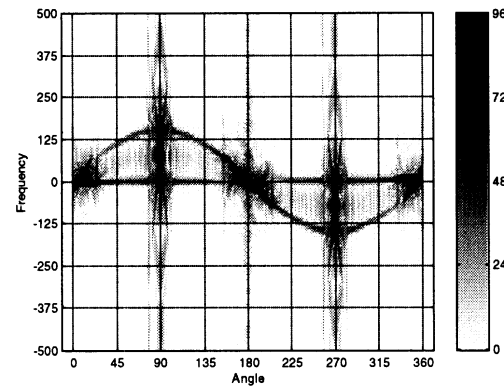
$\theta_{\max}=20$ $|\phi_{\max}|=90$ $|TF|_{\max}=83884$



$\theta_{\max}=30$ $|\phi_{\max}|=90$ $|TF|_{\max}=98968$

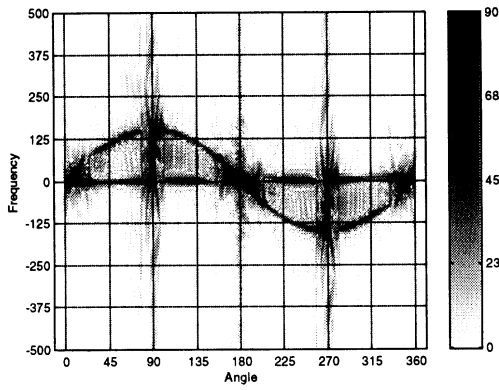


$\theta_{\max}=40$ $|\phi_{\max}|=90$ $|TF|_{\max}=140453$

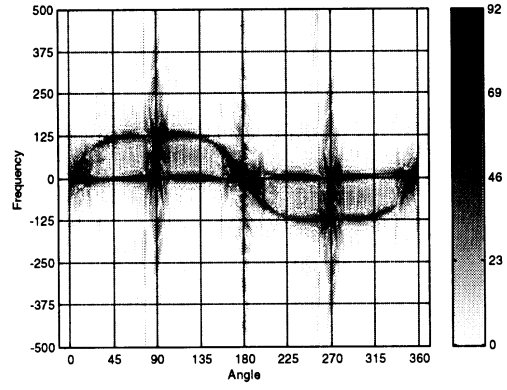


$\theta_{\max}=50$ $|\phi_{\max}|=90$ $|TF|_{\max}=143630$

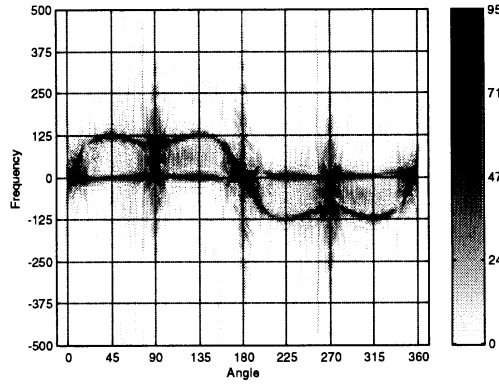
APPENDIX E – LEAD-LAG ANALYSIS RESULTS



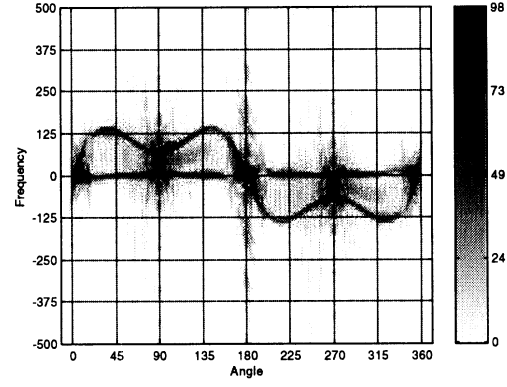
$|LL|=0^\circ, LL_s=0$



$|LL|=5^\circ, LL_s=0$



$|LL|=10^\circ, LL_s=0$



$|LL|=15^\circ, LL_s=0$