

The Reduction of Acoustic Noise Emissions from a Hard Disk Drive

by

William H. Ray V
B.S., Engineering and Applied Science
California Institute of Technology
(1980)

Submitted to the Department of
Electrical Engineering and Computer Science
In partial fulfillment of the Requirements
for the Degree of

Master of Science
in Electrical Engineering

at the

Massachusetts Institute of Technology

January 26, 1994

© 1994 William H. Ray V. All Rights Reserved

The author hereby grants to MIT permission to reproduce and to distribute publicly
paper and electronic copies of this thesis document in whole or in part.

Signature of Author _____
Department of Electrical Engineering and Computer Science
January 26, 1994

Certified by _____
Steven B. Leeb
Assistant Professor of Electrical Engineering
Thesis Supervisor

Certified by _____
Jeffrey H. Lang
Professor of Electrical Engineering
Thesis Supervisor

Certified by _____
Dennis Brown
Company Supervisor

Accepted by _____
Fredric R. Morgenthau
Chairman, Departmental Committee on Graduate Students

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

APR 06 1994

LIBRARIES

ARCHIVES

The Reduction of Acoustic Noise Emissions from a Hard Disk Drive

by

William H. Ray V

Submitted to
the Department of Electrical Engineering and Computer Science
on January 26, 1994 in partial fulfillment of the requirements
for the Degree of Master of Science in Electrical Engineering

Abstract

This thesis develops a method for reducing the acoustic noise emissions from a hard disk drive by shaping the control current to the actuator. Audio frequency structural vibrations are excited by the movement of a disk drive actuator following conventional move-command trajectories. The vibrations result in objectionable acoustic noise. This thesis presents a method of implementing shaped-command trajectories to reduce the acoustic noise problem.

Command trajectories, including position, velocity, and acceleration commands, are shaped to reduce their audio frequency content. The shaping is implemented via convolution with smoothing pulses. Two move algorithms are developed, both trajectory following.

The first algorithm presented uses temporally anti-symmetric accelerations to implement short-length moves. This method permits scaling of both the amplitude and duration of unit-move trajectories to realize moves of varying length and duration. Although the implementation of a move algorithm utilizing anti-symmetric accelerations has aesthetic appeal because of its simplicity it ignores asymmetries inherent in the available acceleration during long-length moves.

A second algorithm is developed for long-length moves. The second algorithm constructs command trajectories by joining several previously-computed trajectory segments. The command segments correspond to acceleration, deceleration, and a switch between acceleration and deceleration.

The author presents comparative measurements of sound-pressure levels, actuator-coil-current power spectrums, seek performance, and post-seek tracking accuracy. The comparisons are made between disk drives following conventional move-command trajectories and the same disk drives following shaped-command trajectories.

Thesis Supervisors: Assistant Professor Steven B. Leeb and Professor Jeffrey H. Lang.

Acknowledgments

I sincerely appreciate the considerable help that both friends and family provided over the course of this project. This thesis would not have been possible without their contributions.

Tom Ray, my brother, spent many days reviewing and correcting the text of the first draft. Tom improved the clarity of the presentation by at least an order of magnitude. He is the one responsible for making the text readable for the benefit of the other reviewers.

John Scaramuzzo, of Digital, contributed greatly by reviewing the thesis for technical accuracy and readability.

Steven Leeb and Jeff Lang, my advisers at MIT, must be thanked for guiding my efforts and suggesting areas for exploration.

I would like to thank my friends at Digital's Colorado Springs facility, Garry Gray and Irvan Hildreth for their generous help. Irv spent several days collecting and arranging for the shipment, from Colorado Springs to Massachusetts, of nearly all of the equipment required during the project. Garry rewrote firmware; enabling me to implement the methods developed in this thesis in a working disk drive.

Mike Strzepa's assistance during the acoustic testing was greatly appreciated.

My positive attitude was sustained by my friendships at MIT. These friends were a constant source of relief: Kim, Deron, John, JP & Laura, Steve, Fari, Zoher, Lara, and many more.

I owe thanks to Frank Bernett for suggesting, several years ago, that a smoother coil current might make a disk drive quieter.

John Ray, my other brother, is to be thanked for encouraging me to take leave of a fine job to pursue a degree at MIT. He is fortunate that I have a short memory for difficult times.

For over thirty years my parents have supported my every endeavor. I love them deeply and know that I will continue to enjoy their support throughout all our years.

Finally, I would like to thank my grandfather, William Hamilton Ray III. It's been nearly a quarter of a century since Grandpop placed a Steelcase desk inside an early sixties Volkswagen bus. The desk didn't fit at first. He used a bigger hammer. It fit... after a window shattered and fell into the street. His perseverance put that desk in the van. He taught me a lot about practical engineering. Never give up. Never quit.

Contents

Chapter 1:	Introduction	15
1.1	Conventional Methods of Noise Reduction in Disk Drives	15
1.2	The Reduction of Noise Emission by Command Shaping	17
1.3	Description of the RA73 Disk Drive	17
1.3.1	Mechanical Description	17
1.3.2	The RA73's Original Seek Algorithm	21
1.4	Overview of disk drive noise emission	24
Chapter 2:	Command Shapes	31
2.1	The Ideal Acceleration Waveform	31
2.2	Filtering, Convolution, and Smoothing Pulses	31
2.3	Anti-Symmetric Acceleration Waveforms	34
2.3.1	Gaussian Shaped Acceleration Trajectory	36
2.3.2	'SMART' Acceleration Trajectory	37
2.3.3	Piecewise Trigonometric Forcing Functions	38
2.3.4	Ramped Sinusoid Function	40
2.4	The Smooth Anti-Symmetric (SAS) Acceleration Waveform	41
Chapter 3:	Implementation	44
3.1	The Trajectory-Following Loop	44
3.2	The Short-Seek Algorithm	49
3.2.1	The Short-Seek Continuous-Time Reference Trajectories	49
3.2.2	The Short-Seek Discrete-Time Reference Trajectories	52
3.2.3	Time and Amplitude Scaling	54
3.2.4	Short-Seek Algorithm Summary	59
3.3	The Long-Seek Algorithm	60
3.3.1	Motivation for the Development of the Long Seek Algorithm	60
3.3.2	Long-Seek Algorithm Overview	61
3.3.2.1	Constructing Profiles from Segments	63
3.3.2.2	Review of Long-Seek Algorithm Concepts	66
3.3.3	Voltage-Limited Acceleration and Constant-Level Deceleration Models	67

3.3.3.1	The Coil Model	68
3.3.3.2	The Actuator Model with a Step of Applied Voltage	68
3.3.3.3	The Voltage-Limited Acceleration Model	69
3.3.3.4	The Constant-Level Deceleration Model	71
3.3.4	The Long-Seek Algorithm Smoothing Pulse	72
3.3.5	Description of the Cosine Acceleration-Reversal Segment	73
3.3.5.1	Scaling and Offsetting the Cosine Acceleration-Reversal Segment	74
3.3.6	Determining the Duration of the Deceleration Phase	78
3.3.7	Determining Total Move Time and Distance	78
3.3.8	The Discrete Time Implementation	79
3.3.9	Long-Seek Algorithm Summary	85
Chapter 4:	Testing and Results	86
4.1	Acoustic Noise Emission	88
4.1.1	Baseline Noise	90
4.1.2	Total Noise	92
4.1.3	Seek Noise	94
4.1.4	Seek Noise Difference	96
4.1.5	'Worst Case' Measurements	98
4.2	Average Coil Current Power Spectrum	102
4.3	Coil Current Shapes and Power Spectrums for Selected Seek Lengths	105
4.3.1	Single-Track Seek	108
4.3.2	8-Track Seek	110
4.3.3	64-Track Seek	112
4.3.4	160-Track Seek	114
4.3.5	1,000-Track Seek	116
4.3.6	Full-Stroke Seek	118
4.4	Average Seek Times	120
4.5	Post-Seek Settling Characteristics	124
4.5.1	Comparing the Post-Seek Vibration of the Quiet and the Original Seek Algorithm	124
4.5.2	Comparing the Post-Seek Vibration of the Quiet Algorithm and a Surrogate Bang-Bang Controller	129
4.6	Testing and Results -- Conclusions	134

Chapter 5: Summary, Conclusions, and Suggestions for Future Work	137
5.1 Conclusions	139
5.2 Suggestions for Future Work	141
5.2.1 Reducing Controller Complexity	141
5.2.2 Power Amplifier Dissipation	146
Appendix A	150
Appendix B	154
Appendix C	155
References	157

Figures

Figure 1-1 RA73 Mechanics	18
Figure 1-2 Actuator Assembly	19
Figure 1-3 Coil Current to Servo-Head Position Transfer Function	20
Figure 1-4 RA73 Original Seek Control Loop	21
Figure 1-5 Actuator-Coil Current, 1,000-Track Seek	23
Figure 1-6 Velocity & Velocity Command, 1,000-Track Seek	23
Figure 1-7 Baseline Noise, Drive #1	25
Figure 1-8 Baseline Noise, Drive #2	25
Figure 1-9 Total Noise, Drive #1	26
Figure 1-10 Seek Noise, Drive #1	26
Figure 1-11 Coil-Current to Sound-Pressure Measurement, Drive #1	28
Figure 1-12 Coil-Current to Sound-Pressure Measurement, Drive #2	28
Figure 1-13 Current Spectrum, Drive #1	28
Figure 1-14 Current Spectrum, Drive #2	28
Figure 2-1 Comparison of Smoothing Pulses and Their Spectrums	33
Figure 2-2 Anti-Symmetric Acceleration	34
Figure 2-3 Amplitude Scaling	35
Figure 2-4 Duration Scaling	35
Figure 2-5 Gaussian Shaped Acceleration Trajectory	36
Figure 2-6 SMART Acceleration Trajectory	38
Figure 2-7 Piecewise Trigonometric Anti-Symmetric Acceleration Trajectory	39
Figure 2-8 Piecewise Trigonometric Asymmetric Acceleration Trajectory	39
Figure 2-9 Ramped-Sinusoid Basis Function	40
Figure 2-10 Forming the SAS Waveform by Convolution	41
Figure 2-11 SAS Acceleration Profile, 1 Meter Move	43

Figure 2-12	SAS Acceleration Spectrum, 1 Meter Move	43
Figure 3-1	Trajectory-Following Seek Loop	45
Figure 3-2	Command Shapes	51
Figure 3-3	Short Seek Duration vs. Distance	56
Figure 3-4	Short Seek Peak Current vs. Distance	56
Figure 3-5	Full-Stroke Seek Current	61
Figure 3-6	'Ideal' Acceleration	62
Figure 3-7	Smoothing the 'Ideal' Acceleration	62
Figure 3-8	'Ideal' Accelerations	64
Figure 3-9	Smoothed Accelerations	64
Figure 3-10	Three Acceleration Segments	64
Figure 3-11	Cosine Acceleration-Reversal Segment	65
Figure 3-12	Actuator Coil Model	68
Figure 3-13	Actuator and Saturated Power Amplifier Model	69
Figure 3-14	Acceleration Current, Modeled versus Measured	71
Figure 3-15	Smoothing Pulse for the Long-Seek Algorithm	73
Figure 4-1	Baseline Noise, Drive #1	91
Figure 4-2	Baseline Noise, Drive #2	91
Figure 4-3	Total Noise, 'Normal' Random Seek Rate, Drive #1	93
Figure 4-4	Total Noise, 'Normal' Random Seek Rate, Drive #2	93
Figure 4-5	Seek Noise, 'Normal' Random Seek Rate, Drive #1	95
Figure 4-6	Seek Noise, 'Normal' Random Seek Rate, Drive #2	95
Figure 4-7	Seek Noise Difference, 'Normal' Random Seek Rate, Drive #1	97
Figure 4-8	Seek Noise Difference, 'Normal' Random Seek Rate, Drive #2	97
Figure 4-9	Total Noise, 'Worst Case' Random Seek Rate, Drive #1	99
Figure 4-10	Total Noise, 'Worst Case' Random Seek Rate, Drive #2	99
Figure 4-11	Seek Noise, 'Worst Case' Random Seek Rate, Drive #1	100
Figure 4-12	Seek Noise, 'Worst Case' Random Seek Rate, Drive #2	100
Figure 4-13	Seek Noise Difference, 'Worst Case' Random Seek Rate, Drive #1 ...	101
Figure 4-14	Seek Noise Difference, 'Worst Case' Random Seek Rate, Drive #2 ...	101
Figure 4-15	Coil Current Power Spectrum, 'Normal' Random Seek Rate, Drive #2	103
Figure 4-16	Coil Current & Sound Pressure Differences, 'Normal' Seek Rate, Drive #2	103
Figure 4-17	Coil Current Comparison, Single-Track Seek, Drive #2	109
Figure 4-18	Coil Current Comparison, 8-Track Seek, Drive #2	111
Figure 4-19	Coil Current Comparison, 64-Track Seek, Drive #2	113
Figure 4-20	Coil Current Comparison, 160-Track Seek, Drive #2	115
Figure 4-21	Coil Current Comparison, 1000-Track Seek, Drive #2	117
Figure 4-22	Coil Current Comparison, Full-Stroke (2,680 Track) Seek, Drive #2 .	119

Figure 4-23	Post-Seek Settles, Position-Error Signal, Drive #2	126
Figure 4-24	Post-Seek Settles, Position-Error Average Power Spectrums, Drive #2	128
Figure 4-25	Open-Loop vs. Quiet-Seek Algorithm, Single-Track Seek, Drive #2..	130
Figure 4-26	Open-Loop vs. Quiet-Seek Algorithm, 8-Track Seek, Drive #2	131
Figure 4-27	Open-Loop vs. Quiet-Seek Algorithm, 64-Track Seek, Drive #2	132
Figure 5-1	Smoothing the Rectangular-Pulse Pair with another Rectangular Pulse	142
Figure 5-2	One Millisecond Rectangular Pulse and Spectrum Magnitude	145
Figure 5-3	Smoothing-Pulse Spectrum Comparison	146
Figure 5-4	Original AlgorithmCoil Voltage, 1,000-Track Seek	147
Figure 5-5	Quiet-Seek AlgorithmCoil Voltage, 1,000-Track Seek	147
Figure 5-6	Original AlgorithmTransistor Voltage, 1,000-Track Seek	147
Figure 5-7	Quiet-Seek Algorithm Transistor Voltage, 1,000-Track Seek	147
Figure 5-8	Original AlgorithmTransistor Power Dissipation,1,000-Track Seek	148
Figure 5-9	Quiet-Seek AlgorithmTransistor Power Dissipation,1,000-Track Seek.	148
Figure A-1	RA73 Velocity Reference as a Function of Distance to Target	150
Figure A-2	Feedforward Profile	152
Figure A-3	Original Algorithm, Actuator-Coil Current, 1,000 Track Seek	153
Figure A-4	Original Algorithm, Tail Current, 1,000 Track Seek	153

Tables

Table 3-1	Counter Increment Interpolation Tables	57
Table 3-2	RA73 Parameter Values	70
Table 4-1	Hewlett-Packard 3563A Measurement States	87
Table 4-2	Coil Current Measurement Tracks	106
Table 4-3	Seek-Time Performance Comparison, Drive #1	121
Table 4-4	Seek-Time Performance Comparison, Drive #2	122
Table 4-5	Post-Seek Settling Measurement Tracks	125
Table 4-6	Open-Loop vs. Quiet-Seek Move-Time Comparison, Drive #2	129

Equations

(1.1)	Actuator Torque Balance	19
(1.2)	Actuator Position Related to Current	20
(2.1)	Gaussian Pulse Fourier Transform	32
(2.2)	Gaussian Shaped Acceleration Trajectory	36
(2.3)	SMART Cost Function	37
(2.4)	SMART Acceleration Trajectory	38
(2.5)	Ramped-Sinusoid Basis Function	40
(2.6)	SAS Acceleration	41

(2.7)	SAS Acceleration Fourier Transform	42
(2.8)	SAS Acceleration Fourier Envelope	42
(3.1)	Compensator Gain	46
(3.2)	Open-Loop Function	46
(3.3)	Closed-Loop Function	46
(3.4)	Controller Output	47
(3.5)	Controller Output with Interdependent Reference Profiles	47
(3.6)	Closed-Loop Function with Interdependent Reference Profiles	48
(3.7)	Unit-Move Position Trajectory	50
(3.8)	Unit-Move Velocity Command	50
(3.9)	Unit-Move Acceleration Command	50
(3.10)	Position Command, Unit Time	50
(3.11)	Velocity Command, Unit Time	50
(3.12)	Acceleration Command, Unit Time	50
(3.13)	Reference Commands, Move Time 'T'	51
(3.14)	Position Command	51
(3.15)	Interpolated Position Reference	53
(3.16)	Continuous to Discrete -Time Conversion	53
(3.17)	Difference Equation for $v[n]$	53
(3.18)	Difference Equation for $a[n]$	53
(3.19)	Sampling $x(t)$	53
(3.20)	Short Seek Distance, Duration, and Peak Current Relationships	56
(3.21)	15-Track Seek Counter Increment Calculation	58
(3.22)	Scaled Interpolated Commands	58
(3.23)	Voltage-Limited Acceleration Model	70
(3.24)	RA73 Voltage-Limited Acceleration Model	70
(3.25)	Smoothing Pulse Definition	73
(3.26)	Cosine Segment, Acceleration	74
(3.27)	Cosine Segment, Velocity	74
(3.28)	Cosine Segment, Position	74
(3.29)	Cosine Segment, ΔA	74
(3.30)	Cosine Segment, Average A	74
(3.31)	Cosine Segment, ΔV	74
(3.32)	Cosine Segment, ΔX	74
(3.33)	$A(t)$ and $V(t)$ Approximations	75
(3.34)	$A(t)$ related to $V(t)$	75
(3.35)	Cosine Segment, Scale-Factor, α	76
(3.36)	Cosine Segment, Offset-Factor, β	76
(3.37)	Cosine Segment, Final Velocity	77
(3.38)	Cosine Segment, Final Position	77

(3.39)	Constant-Deceleration Time	78
(3.40)	Difference Equation Relating $v[n]$ to $a[n]$	81
(3.41)	Difference Equation Relating $v[n]$ to $x[n]$	82
(3.42)	Velocity Ramp Correction	84
(5.1)	Continuous to Discrete-Time Conversion	142
(5.2)	Continuous-Time Rectangular Smoothing Filter	144
(5.3)	Discrete-Time Rectangular Smoothing Filter	144
(A.1)	Original RA73 Seek Algorithm Velocity Reference	151
(A.2)	Velocity Reference, Position > 946 Tracks	151
(A.3)	Velocity Reference, Position < 19.3 Tracks	151
(A.4)	Feedforward Definition	152
(B.1)	Fourier Transform of Unit Area Sine Pulse	154
(B.2)	Fourier Transform of Unit Area Raised Cosine Pulse	154
(C.1)	Voltage-Limited Acceleration Derivation	155
(C.2)	Voltage-Limited Acceleration Model	155
(C.3)	Simplified RA73 Acceleration Model	156

Chapter 1: Introduction

Today's distributed computing environment has replaced yesteryear's noisy computer rooms with networks of workstations and PC's spread throughout quiet office areas. These small computers are generally quiet machines. Even so, in a quiet office, workstations and desktop computers may be a major source of acoustic noise. A significant component of the noise generated by these systems is the sound produced by their disk drives.

Disk drive noise can be distracting; consequently, customers avoid buying noisy drives. Often the objectionable noise is associated with the movement of the disk drive's head positioning actuator.

This thesis reduces acoustic noise emissions from a hard disk drive actuator using a sampled data control system by limiting the frequency content of the disk drive actuator control signal. Command shaping techniques are employed to reduce vibrations excited by actuator-coil currents during actuator movements. The methods developed reduce the noise power caused by a disk drive's actuator movements by an order of magnitude.

The developed methods achieve low-vibration movements by employing trajectory following algorithms. In a trajectory following algorithm, the control loop reference follows either stored or synthesized profiles from an initial state to a commanded end state. Position, velocity, and acceleration profiles are shaped to produce smooth actuator-coil current commands, reducing acoustic noise emission.

A comparison of noise emission and other important disk drive measurements between disk drives operating with and without the new techniques is presented.

1.1 Conventional Methods of Noise Reduction in Disk Drives

Much of the work aimed at reducing acoustic emissions from disk drives focuses on mechanical solutions. Frequently the solutions involve the addition of a dampening mechanism to a resonant mode and/or the redesign of a structure to alter a mode's character. Several examples from industry can be given.

C. M. Woldemar and A. Kumano of Seagate Technology¹ achieved a 4 dB to 5 dB reduction in noise emission between 2 KHz and 4 KHz as a result of adding a sheet of felted foam to the base of a disk drive housing. The felted foam reduced the airborne transmission of acoustic energy from the base to an adjacent printed circuit board. The felted foam may also have introduced some dampening to an acoustically resonant cavity between the base and the circuit board. Woldemar and Kumano also potted spindle motor coils in varnish to damp a mode at 8.6 KHz. They were able to reduce the amplitude of the mode by a factor of two while moving the mode out to 9.0 KHz at the same time.

The redesign of an impeller in a cooling blower lowered the noise power emission level of a Control Data removable disk drive by 0.67 Bels according to L. F. Luttrell and E. K. Dunens².

Disk drive mechanical engineers have employed many techniques to reduce vibration in their designs. Actuator-coil modes have been dampened by impregnating the coil with epoxy. Tuned-beam dampers have been used on other actuator coils. Constrained-layer dampening has been used to dampen both flexure modes and housing modes. Arms and housings have been redesigned to change mode shapes and frequencies. Often this work is motivated by servo loop stability requirements or a need to reduce vibration while track following; however, the reduced vibration can also yield quieter operation. At times the solutions are effective for several modes, such as when constrained-layer dampening is added; other times only an individual mode is improved.

Mechanical solutions typically increase the cost of the product. Adding dampening material requires an additional manufacturing process with additional material, scrap, and (potentially) testing. Redesigning piece parts also requires time, effort, and money. Finally, mechanical changes risk the delay or cancellation of a product introduction.

An alternative and general method of reducing vibration is desirable. Shaping the power spectrum of a control signal to avoid exciting mechanical modes of vibration is the alternative pursued here.

¹"Noise Control of Disk Drives for Desk-Top Computers," Proceedings of the 1987 National Conference on Noise Control Engineering, Ilene J. Busch-Vishniac, Ed. (Institute of Noise Control Engineering, Poughkeepsie, NY, 1990) pp. 89-94.

²"Noise Sources in Disk Drives," Proceedings of the 1987 National Conference on Noise Control Engineering, Jiri Tichy, Ed. (Institute of Noise Control Engineering, Poughkeepsie, NY, 1987) pp. 141-146.

1.2 The Reduction of Noise Emission by Command Shaping

The methods to reduce acoustic noise emission developed in this thesis take the mechanical plant as a 'given' and reduce the noise emissions by not exciting the plant's modes. The approach is to avoid exciting the system with undesirable energy, rather than absorbing energy from the system's unwanted dynamics. If the plant's modes are approximately linear and the coil is driven with currents that lack energy at the modal frequencies, then the modes will not be excited and the drive will operate quietly.

Less vibration excitation can result in benefits other than noise reduction. Specifically, less excitation shortens the time needed to allow vibrations to settle out after an actuator movement. It therefore allows better tracking performance after a movement. Better tracking performance yields better read/write error rates and potentially tighter track spacing and higher data-storage capacities. Less vibration may also reduce fatigue and stress in the mechanical components, potentially improving product reliability.

Limiting the frequency content of the actuator control signal also has negative consequences. Shortening a signal in the frequency domain lengthens it in the time domain. Shaping commands to reduce their high-frequency content extends their duration and makes theoretical move times, excluding settling, longer. Use of smoothly shaped commands, as developed in this project, requires that the control effort never saturates, further extending move times. The total of move time and settle time is a key measure of disk drive performance and should not be compromised.

1.3 Description of the RA73 Disk Drive

Digital's RA73 disk drive was used to demonstrate the noise reduction techniques described in this thesis. This section describes the RA73 to help the reader appreciate the complexity of the disk drive system and the opportunities that exist for noise production within the disk drive. The section first describes the RA73's mechanical configuration, then it describes the RA73's seek algorithm, finally it discusses the RA73's acoustic noise emissions.

1.3.1 Mechanical Description

The RA73 is a full height 5 ¼" rigid-media disk drive. It has a formatted-data capacity of 2 gigabytes and an average positioning time of 12 milliseconds. It contains 11 platters, 21 read/write data heads, and 1 dedicated servo head. The surface of each

platter is divided circumferentially into 2,682 tracks spaced at 2,468 tracks per inch. The spindle rotates at 3,600 RPM. The drive is powered by +5.0V and +12.0V.

The RA73's mechanics set, shown in Figure 1-1, consists of a drive housing (aluminum shell) enclosing a spindle assembly and an actuator assembly. The spindle assembly is comprised of a spindle motor and 11 platters.

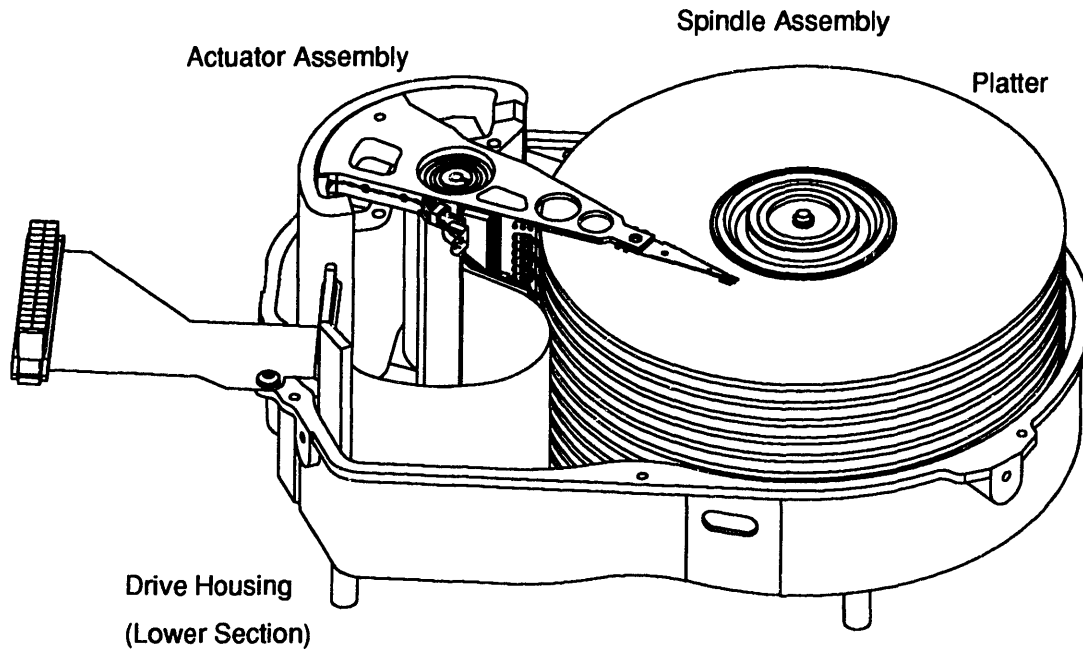


Figure 1-1 RA73 Mechanics

The actuator assembly, shown in Figure 1-2, includes the actuator motor, arms, flexures, and read/write heads. The actuator motor is a moving coil DC motor. The function of the actuator is to rapidly and accurately reposition the read/write heads over the desired customer data. It rotates on a shaft anchored to the disk housing. It is located near the edge of the drive's 11 spinning platters. Attached to the actuator's shaft, and extending out over the surfaces of the platters, are twelve thin, rigid, arms. Each of the ten interior arms has two flexures (a thin piece of stamped and formed steel) attached to its tip. Each flexure has a read/write head attached to its tip. A rectangular coil, extending the length of the actuator, attaches to the other side of the shaft. It is held in the field of the actuator motor's permanent magnets.

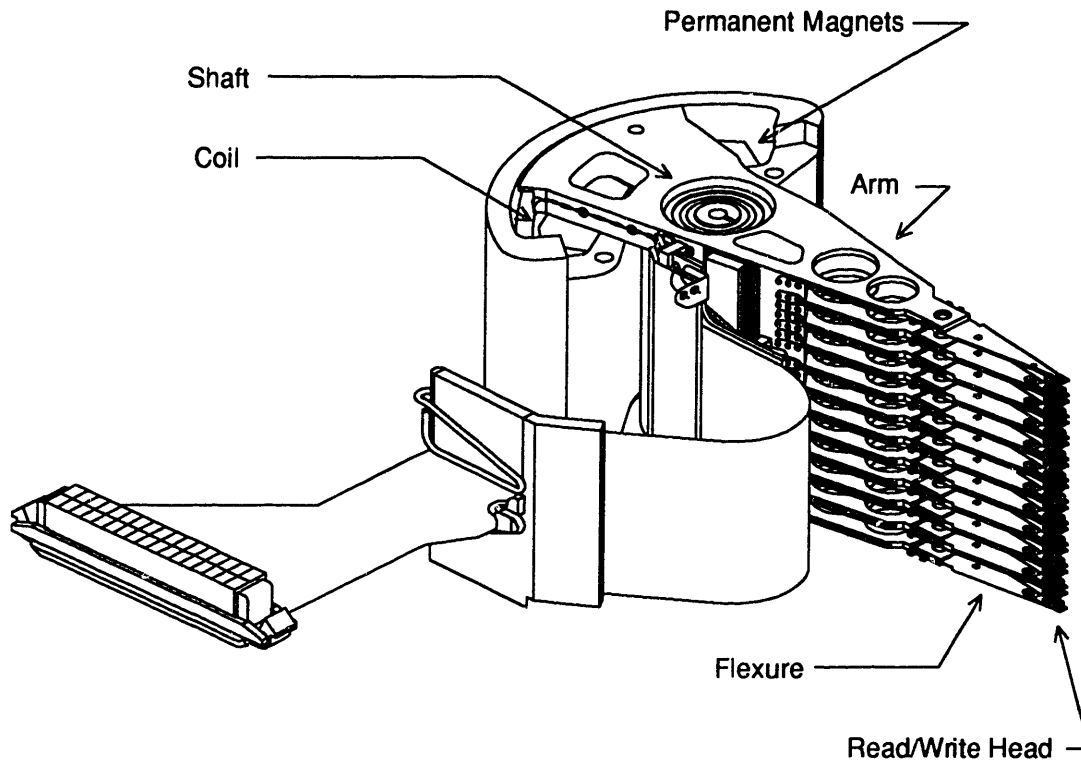


Figure 1-2 Actuator Assembly

The interaction of the actuator's magnetic field and the current flowing in the actuator's coil creates a force on the active sides of the coil. This in turn, creates a torque on the actuator and causes the actuator to accelerate. The actuator torque balance equation is shown below.

$$\text{Torque} = K_{\text{torque}} I_{\text{coil}} = J \frac{d^2\theta}{dt^2} \quad (1.1) \text{ Actuator Torque Balance}$$

where:

$$K_{\text{torque}} = \text{Torque Constant} = 0.262 \frac{\text{Nm}}{\text{A}}$$

$$I_{\text{coil}} = \text{Actuator Coil Current (Amps)}$$

$$J = \text{Actuator Inertia} = 94.34 \times 10^{-6} \frac{\text{Nm}}{\text{rad/sec}^2}$$

$$\theta = \text{Actuator Angular Position (Radians)}$$

The drive's read/write heads, which act as the servo's position transducers, are located at the ends of the flexure/arm assemblies. Their position over an arc across the disk's surface is proportional, by the actuator arm length, to the double integral of the actuator's angular acceleration. This is described by Equation 1.2.

(1.2) Actuator Position Related to Current

$$\text{Position} = l\theta = l \int \int \frac{d^2\theta}{dt^2} dt^2 = l \int \int \frac{K_{\text{torque}} I_{\text{coil}}}{J} dt^2 \xrightarrow{\text{Laplace Transform}} l \frac{K_{\text{torque}}}{Js^2} I_{\text{coil}}$$

where:

$$l = \text{Arm Length} = 80.80 \times 10^{-3} \text{ m}$$

In addition to the simple G_0/s^2 model of the actuator's dynamics, there are many mechanical modes of vibration of the coil, arm, and flexure structures. These modes of vibration tend to be at and above 1 KHz, occupying a large part of the audio spectrum. When excited, some of these modes will produce acoustic noise. Also, some of the vibrational modes result in cross-track motion of the servo head and can be detected in the measured transfer function from actuator-coil current to servo-head position, as is presented in Figure 1-3.

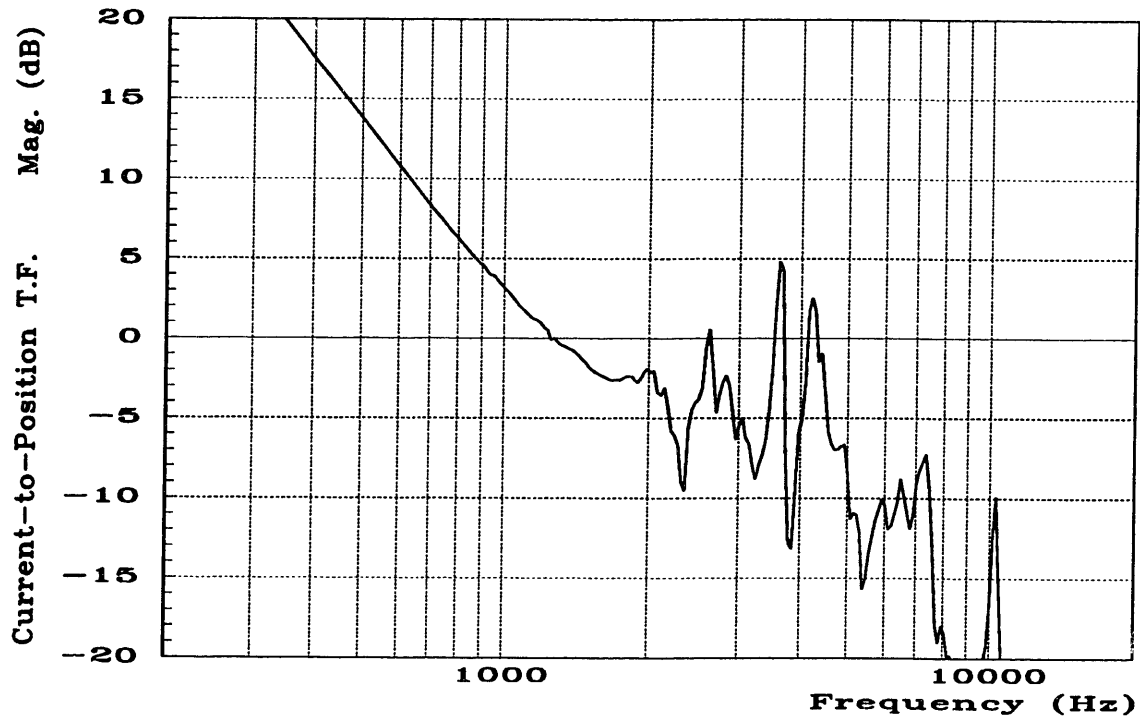


Figure 1-3 Coil Current to Servo-Head Position Transfer Function

1.3.2 The RA73's Original Seek Algorithm

Disk drives must access data at random locations on the surfaces of the platters. This requires the actuator to move, or seek, between different radial positions, or tracks, on the disk's surfaces. The commands to move the actuator can be viewed as step inputs to the drive's positioning system. However, the control loop which the production RA73 uses while maintaining its position over a track, or track-following, does not respond well to large step inputs. Therefore the RA73 employs a separate control loop when it seeks. The RA73's seek control loop is implemented as a firmware algorithm executing within a digital signal processor (DSP). This thesis develops new seek algorithms which exhibit less acoustic noise emission when compared to the original seek algorithm described here. A block diagram of the RA73's original seek control loop appears in Figure 1-4.

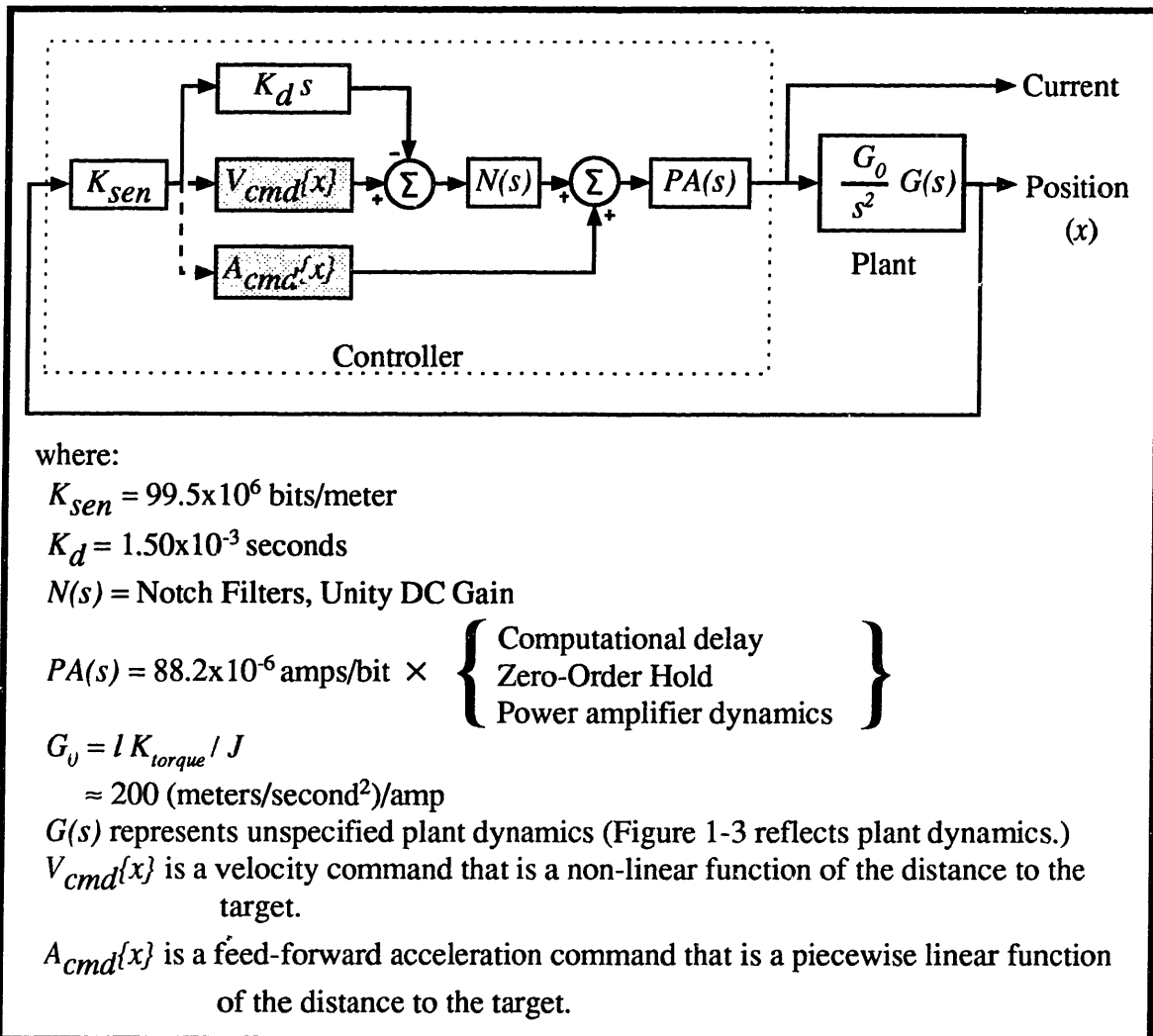


Figure 1-4 RA73 Original Seek Control Loop

The original seek algorithm in the RA73 regulates the velocity of the actuator as a function of the distance to the target; consequently, the controller's command signals, $V_{cmd}\{x\}$ and $A_{cmd}\{x\}$, are shown as functions of the position signal, x , in Figure 1-4. Because both $V_{cmd}\{x\}$ and $A_{cmd}\{x\}$ are non-linear functions of the distance to the target, or position-error, the controller is also non-linear. However, both $V_{cmd}\{x\}$ and $A_{cmd}\{x\}$ are directly proportional to the position-error when the error is less than 19.3 tracks. This causes the controller to become a linear, proportional-plus-derivative (PD) controller as the actuator approaches its destination. The control signal to the plant has both velocity feedback and acceleration feedforward components. The feedback component is the difference between a velocity command, and the first difference of the measured position (an approximation to the actuator's velocity). Appendix A describes the commands associated with the original RA73 seek algorithm in more detail.

During the first half of a seek, the controller and power amplifier saturate while accelerating the actuator to the commanded velocity. The abrupt application of the full voltage to the coil produces a rising edge in the coil current waveform (reference Figure 1-5). The sharp edge in the current adds to the high-frequency content of the current's power spectrum. After the initial application of the maximum power amplifier voltage, the coil's inductance causes the current to rise smoothly. The current causes the actuator to accelerate. As the actuator moves faster, a back EMF develops in the coil. The back EMF subtracts from the applied voltage, causing the coil current to slowly decrease. The system comes out of saturation as the actuator reaches the commanded velocity. At this point, the coil current reverses direction to decelerate the actuator. Again, the rapid change in the current waveform implies high frequency excitations are taking place in the coil. During the second half of the seek the controller causes the actuator to move at a commanded velocity which decreases as the actuator approaches the target track. The seek algorithm has been implemented so that the current, velocity, and position error signals all decay exponentially with a 640 microsecond time constant at the conclusion of the seek. Figure 1-5 shows the coil current during a 1,000-track seek. Figure 1-6 presents a comparison of the actual velocity with the commanded velocity for a 1,000-track seek.

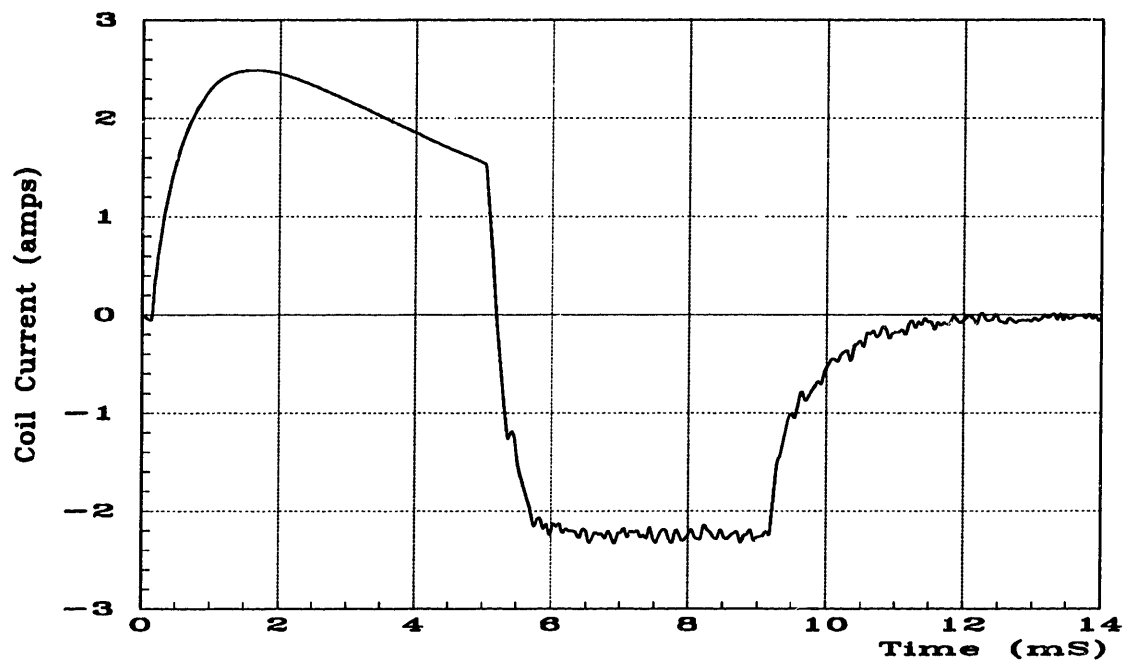


Figure 1-5 Actuator-Coil Current, 1,000-Track Seek

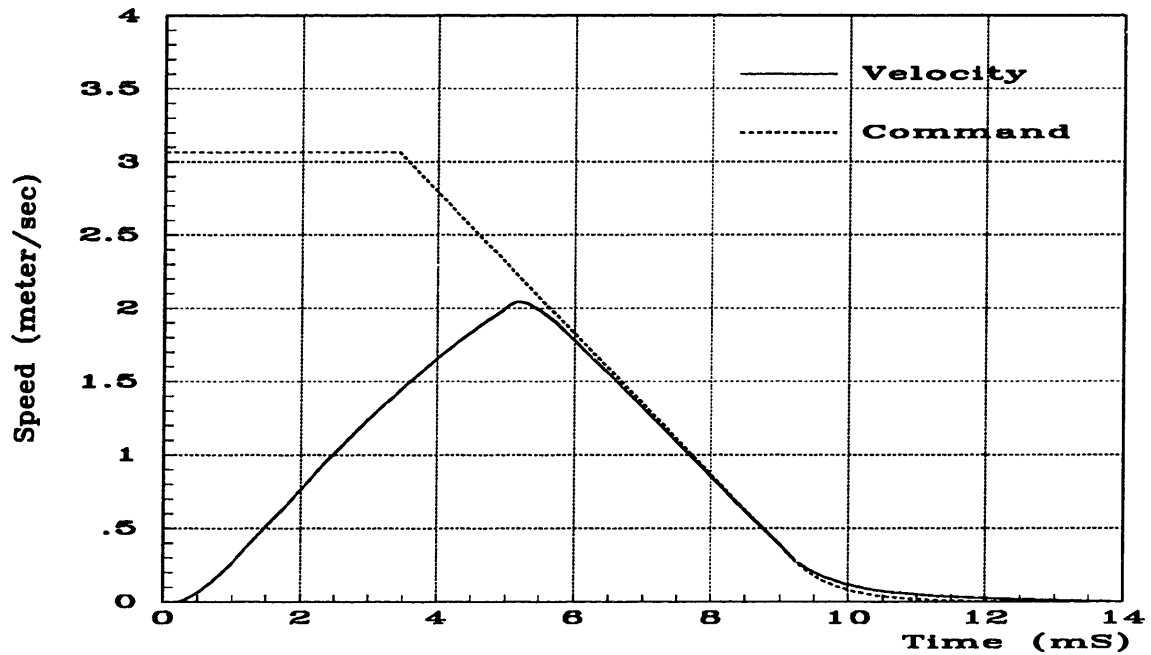


Figure 1-6 Velocity & Velocity Command, 1,000-Track Seek

1.4 Overview of disk drive noise emission

Noise produced by disk drives results from two principal sources; the spindle and the actuator. Spindle noise is generated in several ways: Imperfections in the spindle's bearing assemblies excite structural vibrations as the spindle rotates. Pulsing magnetic forces in the spindle motor also drive structural vibrations. Finally, noise is caused by the rotating airflow inside the drive's mechanical assembly. In addition to the spindle noise, there is the noise generated by the actuator as it moves. When the actuator seeks from one track to another its acceleration, and the reaction torque that drives the acceleration, excite mechanical modes of both the actuator and the housing. The noise caused by seek activity will be called 'seek noise' in this thesis. The actuator also generates perceptible noise as it makes micromotions while track following. Although the actuator's motion is very small, the spectrum of the controlling current spans the audio range and audibly excites the structure. Spindle noise and the noise the actuator generates while tracking comprise a baseline noise level. Figures 1-7 and 1-8 show the measured baseline noise of two typical RA73 disk drives operating in an anechoic chamber¹. The measurements indicate that the noise pressure is most intense between 1 and 6 KHz and is especially strong near 2 KHz. Further, two broad peaks center at 2 and 6 KHz. Each broad peak has many sharper peaks within it. Each sharp peak corresponds to a mechanical mode of vibration or an acoustic resonance of the drive enclosure.

¹The disk drive being tested was placed inside an 'AN-ECK-OIC' chamber from Eckel Industries, Inc., of Cambridge Massachusetts. A microphone was suspended a third of a meter directly above the top cover of the drive for the sound pressure measurement. The microphone fed a Brüel and Kjær Dual Channel Signal Analyzer, Type 2032. The microphone was calibrated to produce 11.8 mv/Pascal as an input to the Brüel and Kjær's pre-amplifier. The Brüel and Kjær's display and output data was referenced to 40 dB per 20 μ Pascal. The Brüel and Kjær was configured to take 500 spectrum averages of the auto-correlation of the sound pressure sampled 2,048 times at a 32.768 KHz rate. This provides both a frequency resolution and an FFT bin spacing of 16 Hz. The analyzer calculated the displayed RMS magnitude data as the square-root of one half of the average product of the spectrum and its complex conjugate.

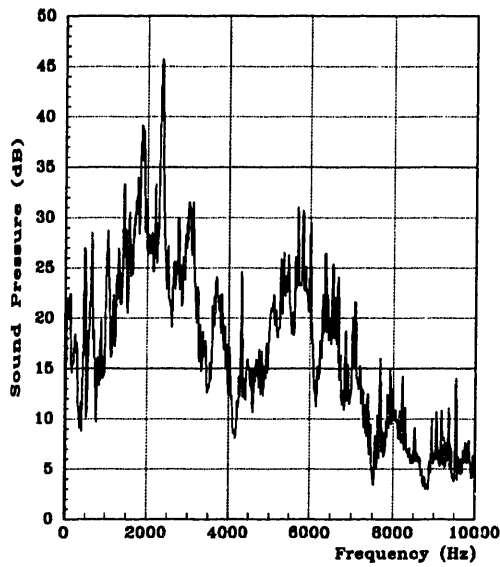


Figure 1-7 Baseline Noise, Drive #1

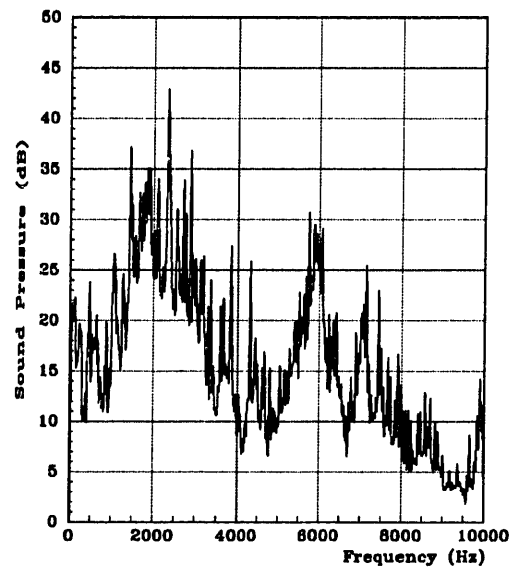


Figure 1-8 Baseline Noise, Drive #2

When a disk drive retrieves stored data, it first moves the data heads to the track location containing the data. The drive performs a move, or seek, by first accelerating the actuator with a pulse of coil current in one direction and then decelerating it with a pulse in the opposite direction. Normal operating conditions apply currents which produce an acceleration of 50 g's at the read/write heads and then reverse the acceleration several milliseconds later. These high acceleration rates are the result of large torques and forces. The high torques and forces distort the coil, the extensions holding the coil, the arms, the flexures, and the drive housing holding both the shaft and the magnet and steel assembly. Each of these structures has mechanical modes of vibration. When excited, these modes of vibration result in acoustic noise emissions. Seek noise is superimposed on the baseline noise level.

Figure 1-9 shows the total sound pressure measured while seeking at a 27 seeks per second rate compared with the baseline sound pressure measured while track following. Figure 1-10 displays the seek noise, i.e. the difference between the sound pressure while seeking and the baseline sound pressure. Figure 1-10 shows seeking causes the sound pressure to increase by 6 dB (double) over most of the audio spectrum and to increase by 15 dB (a factor of six) between 1 and 3 KHz. Seek noise is not isolated to a few frequencies. This suggests that attempts to reduce seek noise must influence the sound pressure across a broad range of frequencies.

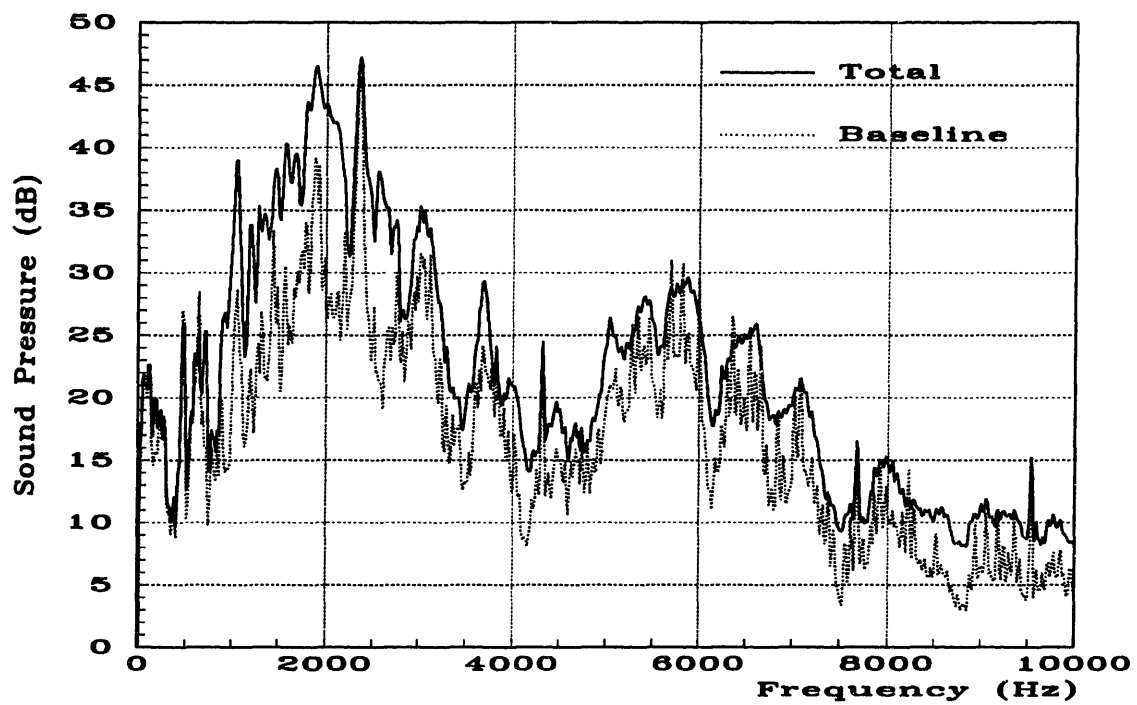


Figure 1-9 Total Noise, Drive #1

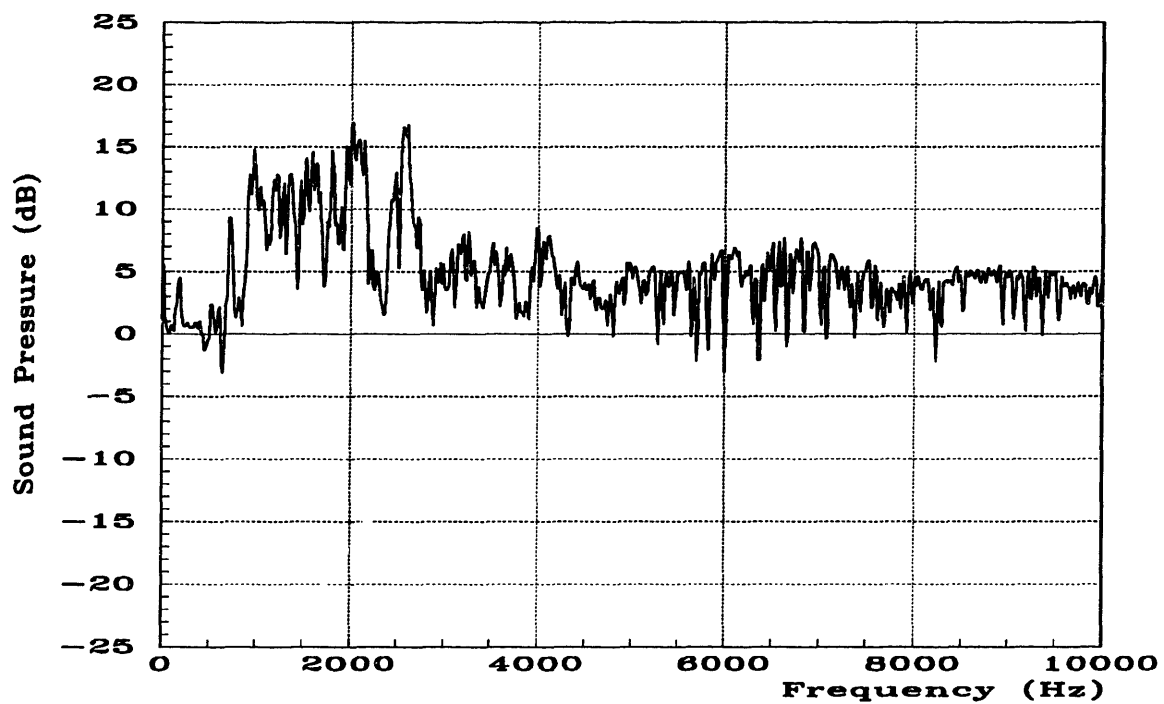


Figure 1-10 Seek Noise, Drive #1

Current flowing in the actuator coil produces a proportional torque which in turn accelerates the actuator. The torque also excites vibrational modes of the actuator and its associated structures. A portion of the drive's acoustic emission derives from a subset of the modes excited by the coil current. Consequently, there is a relationship, not necessarily linear, between the current in the coil and the emitted noise sound pressure. The methods developed in this thesis reduce acoustic noise emission by modifying the frequency content of the coil current employed while seeking. To precisely determine how to modify the coil current one needs to understand the relationship between the coil current and the resulting emitted sound pressure. Specifically, one would need to measure the ratio of the spectrum magnitude of the sound pressure to the spectrum magnitude of the coil current while the drive seeks randomly. If the system being measured were linear, this would be a transfer function measurement. However, one suspects that the large excitations inherent in seek activity cause non-linear distortions of the drive's structures. Additionally, the relationship between coil current and sound pressure likely depends upon the rotational position of the actuator within the drive cavity. Consequently, it is improbable that the system is linear; therefore the measurement is not a strict transfer function measurement in this case. Irrespective of the uncertainty concerning the system's linearity, this measurement contributes significantly to one's understanding of the relationship between coil current and sound pressure in the frequency-domain.

Figures 1-11 and 1-12 present measurements¹ of the ratio of the average sound-pressure spectrum magnitude to the average coil-current spectrum magnitude while each drive seeks randomly. The measurements display many peaks across the audio spectrum; however, they reach their highest levels between 1 and 4 KHz and between 5 and 8 KHz. They peak near 2 KHz. This implies that reducing the coil current's spectrum from 1 to 4 KHz and from 5 to 8 KHz should reduce the noise level.

¹The footnote at the bottom of page 24 describes the test set-up used for these measurements with the following modifications. Additional to the microphone which fed a sound-pressure signal to the analyzer's pre-amp channel A, a BNC cable connected the RA73's "IACT" coil-current sense signal to the analyzer's channel B. As before, the analyzer computed 500 spectrum averages of 2048-point FFT's of data spanning 62.5 milliseconds. The frequency resolution of the FFT's is 16 Hz. The analyzer calculates the data displayed in Figures 1-11 and 1-12 by computing the magnitude of the ratio of the average product of the sound-pressure FFT and its complex conjugate to the average product of both the sound-pressure FFT and the complex conjugate of the coil-current FFT. The reference for the analyzer's operation is the Brüel and Kjær, "2032 Instruction Manual, Volume 2, Operation," September 1983, pp 151-152

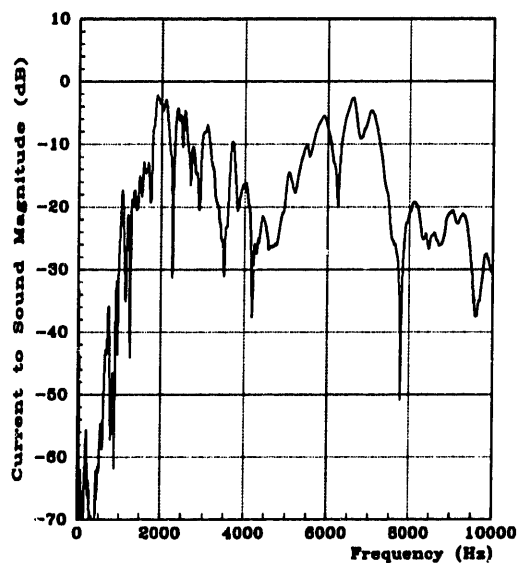


Figure 1-11 Coil-Current to Sound-Pressure Measurement,
Drive #1

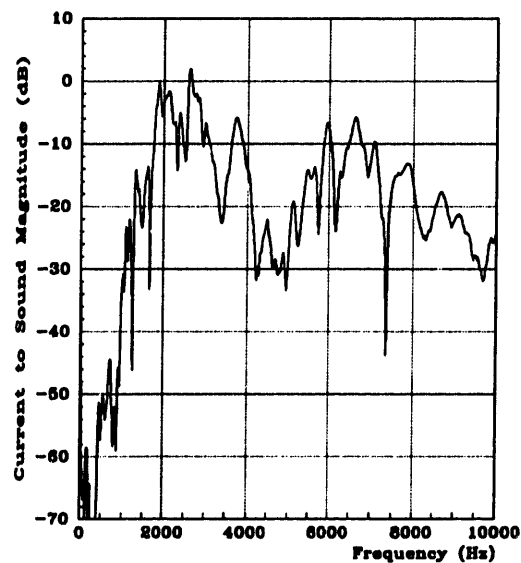


Figure 1-12 Coil-Current to Sound-Pressure Measurement,
Drive #2

Because the increase in sound pressure during seeking is highest in the frequency band between 1 and 3 KHz, it is likely the current's spectrum is higher within this range than at higher frequencies. The measurement of the coil current's spectrum, as the drives seeked randomly, shown in Figures 1-13 and 1-14 support this assumption.

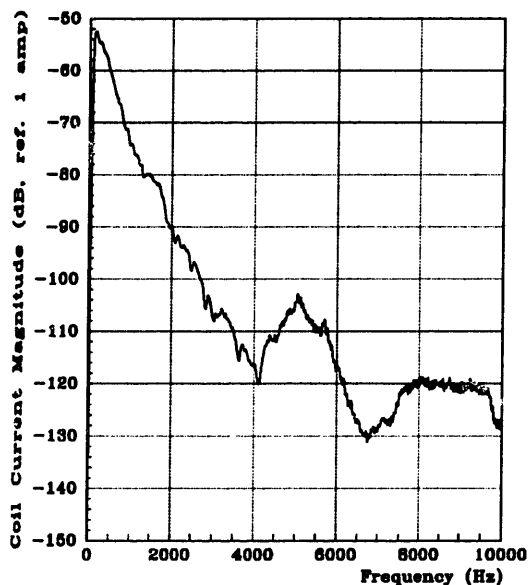


Figure 1-13 Current Spectrum, Drive #1

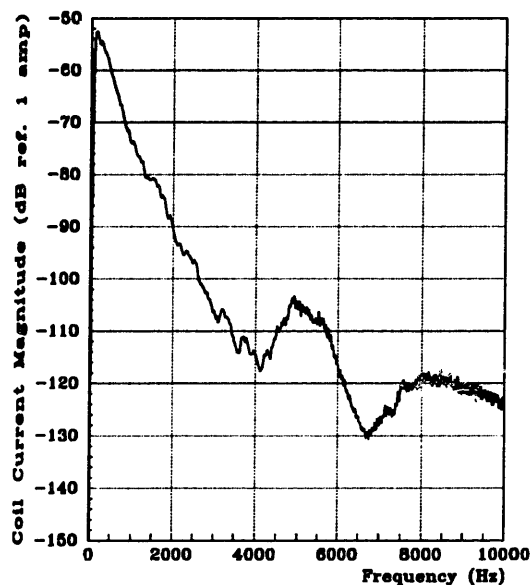


Figure 1-14 Current Spectrum, Drive #2

Note that although the coil current's spectrum is at its highest level below 1 KHz, the transfer function from coil current to sound power is at its lowest level below 1 KHz. Therefore the resulting sound pressure below 1 KHz is insignificant.

Most of the noise emitted by the RA73 disk drive is in a range of frequencies between 1 and 8 KHz. The noise is the result of many mechanical modes which are excited by the drive's operation. Although some broad peaks exist in the sound pressure spectrum, the noise is better described as 'broad band' rather than 'tonal'. Seek activity increases the sound pressure by as much as 15 dB between 1 and 3 KHz. Because the increase in the sound pressure occurs at frequencies significantly higher than the seek rate (27 seeks per second), it is reasonable to assume that the noise increase is not due to the excitation of low-frequency modes by repetitive seeks. Instead, one may conclude that the source of seek noise is the high-frequency content of the coil currents related to individual seeks. It is possible to reduce noise emissions while seeking by reducing the high-frequency content of the coil current. The use of smooth coil currents while seeking may lower a disk drive's noise emissions by as much as 15 dB (the level of the seek noise) in the 1 KHz to 3 KHz range.

Intentionally Blank

Chapter 2: Command Shapes

2.1 The Ideal Acceleration Waveform

Ignoring practical considerations, the ideal acceleration for a disk drive actuator's seek operation is described by the doublet. Double integration of the doublet results in a step; a step in position is the desired movement of the actuator.

The Fourier Transform of a signal describes its frequency content. The Fourier Transform of the ideal acceleration signal, the doublet, is $j2\pi f$ with magnitude increasing linearly with frequency f . The doublet cannot be realized in practice. An acceleration and deceleration pulse-pair, with finite duration, describes a realistic acceleration which causes a change in position. Realizable acceleration pulse-pairs exhibit finite magnitudes and finite durations, consequently their spectrum magnitudes roll off at least as fast as $1/f$.

2.2 Filtering, Convolution, and Smoothing Pulses

Commands can be shaped so that their high-frequency content rolls off faster than $1/f$. One can create shaped commands by attenuating the-high frequency components of a realizable acceleration pulse pair. There are two interrelated methods of attenuating the high-frequency components of a signal: Low-pass filtering the signal or convolving the signal with a smoothing pulse.

Filtering multiplies the input signal's spectrum by the spectrum of the filter's impulse response. Filters which have impulse responses that decrease in magnitude at high frequencies are called low-pass filters. Low-pass filters can be implemented as either infinite impulse response (IIR) filters or finite impulse response (FIR) filters. IIR filters, such as RLC filters, have exponentially (or geometrically) decaying impulse responses. Applying an IIR filter to an actuator's acceleration waveform produces a movement that asymptotically approaches the destination. This complicates quantifying the effects of filtering on move time. Applying an FIR filter to an actuator's acceleration waveform extends the move time by the duration of the FIR filter's impulse response. Shaping an acceleration-command signal with an FIR filter causes predictable changes in the move time.

An equivalent method of attenuating the high-frequency content of a signal is to convolve the signal with a smoothing pulse. Convolution of a signal with a smoothing pulse multiplies the signal's spectrum by the spectrum of the smoothing pulse. Filtering a signal is equivalent to convolving it with the filter's impulse response. In other words, a filter's impulse response is its associated smoothing pulse. In the time domain, the smoothing pulse acts to replace each point on the input signal with a weighted average of the point and neighboring points. If the area of the smoothing pulse is unity, then the area of the output signal is the same as the area of the input signal. Convolution with a smoothing pulse also extends the duration of the signal by the duration of the pulse.

To limit the high-frequency content of the command signal, smoothing pulse candidates should have most of their energy concentrated at low frequencies (below 1 KHz), but they should also be short in time duration to minimize move times. These two requirements are contradictory and must be balanced against each other. Additionally, smoothing pulse candidates should have low peak amplitudes to avoid requiring accelerations which would saturate the power amplifier. Example smoothing pulses include a half-cycle of a sine wave, one cycle of a raised cosine, and a Gaussian pulse. The trigonometric pulses are considered because they possess properties which are favorable to both time and frequency-domain analysis. The Gaussian pulse is interesting because with the exception of a time constant, its time domain and frequency domain representations are the same, as is shown by its Fourier Transform below.

$$e^{-\pi(t/\tau)^2} \Leftrightarrow \tau e^{-\pi(f\tau)^2} \quad (2.1) \text{ Gaussian Pulse Fourier Transform}$$

Time
Frequency

This thesis' application demands a pulse with its energy concentrated near zero in both domains; consequently, the Gaussian pulse is the ideal compromise because its energy content decays exponentially in both domains. However, the Gaussian pulse has infinite extent in both domains. This application requires a finite duration smoothing pulse. One engineering solution would be to truncate the Gaussian pulse's tails in the time domain. This is equivalent to multiplying the Gaussian pulse by a rectangular window in the time domain. In the frequency domain this convolves the Gaussian pulse with a sinc function.

Figure 2-1 compares the candidate smoothing pulses and their spectral magnitudes. Each pulse has unit area and a duration of 1 millisecond. The tails of the Gaussian pulse are truncated at the point where they fall below 1% of the peak value.

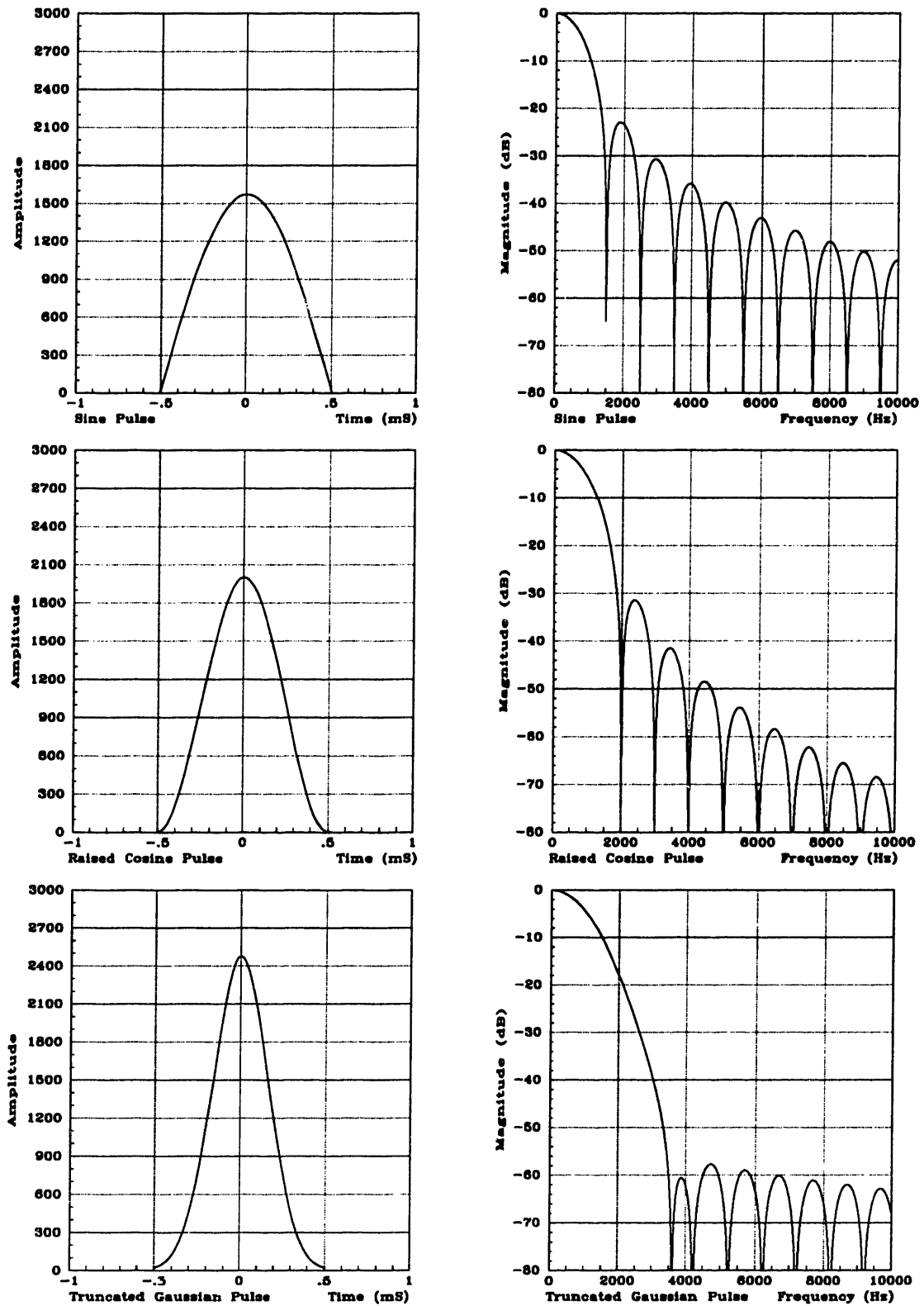


Figure 2-1 Comparison of Smoothing Pulses and Their Spectrums

Of the three pulses, the truncated Gaussian pulse has the highest peak value in the time domain, the widest first lobe in the frequency domain, and the lowest high-frequency content. The sine pulse has the lowest peak value in the time domain, the narrowest first lobe in the frequency domain, and the greatest high-frequency content. The raised cosine pulse has intermediate characteristics. In general, the more concentrated a pulse is in one domain the more dispersed it will be in the other domain. Appendix B presents the Fourier Transforms of the unit-area sine pulse and the unit-area raised-cosine pulse.

The 'best' choice of smoothing pulse is application dependent. The truncated Gaussian pulse may be preferred for an application requiring very little excitation at high-frequencies, but the sine pulse may be superior for an application demanding low levels of peak acceleration. This project uses both the sine pulse and the truncated Gaussian pulse to smooth ideal accelerations.

2.3 Anti-Symmetric Acceleration Waveforms

An anti-symmetric acceleration waveform is a function that is 'odd' about its mid-point and has equal area above and below the x-axis. The integral of such a function over its length is equal to zero. If an object at rest undergoes an anti-symmetric acceleration, the object will be at rest at the end of the acceleration independent of the scaling, duration, or shape of the acceleration. However, the position of the object will be changed by the double integral of the acceleration. An anti-symmetric acceleration waveform and the corresponding velocity and position waveforms are shown in Figure 2-2.

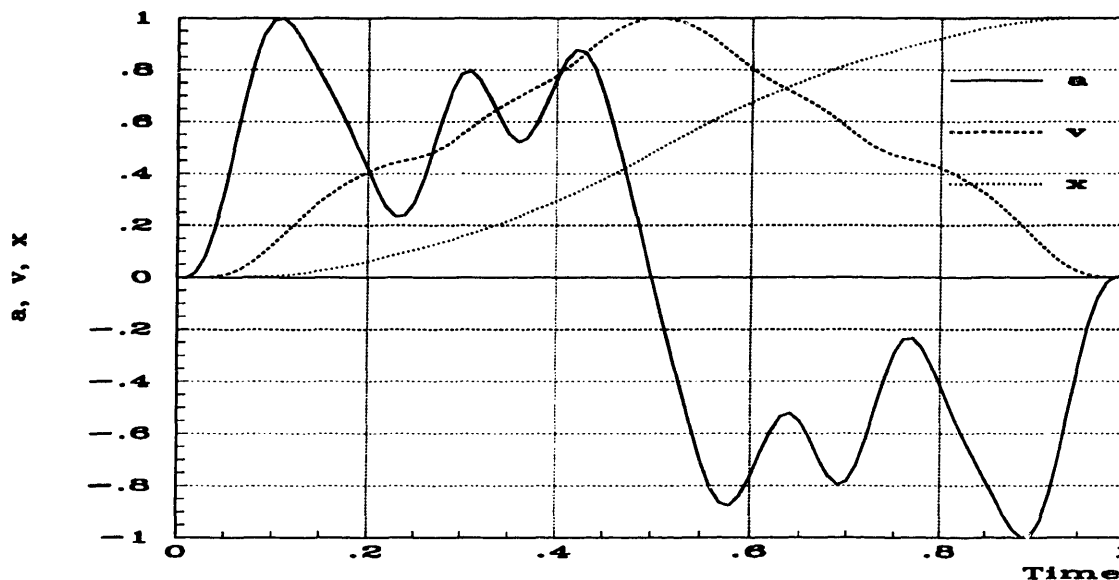


Figure 2-2 Anti-Symmetric Acceleration

Scaling an acceleration waveform also scales the corresponding velocity and position waveforms by the same factor. Therefore if a disk actuator's control system uses an anti-symmetric acceleration waveform to perform a unit move, then the control system can scale the acceleration by one half and move the actuator half the original distance. This is reflected in Figure 2-3.

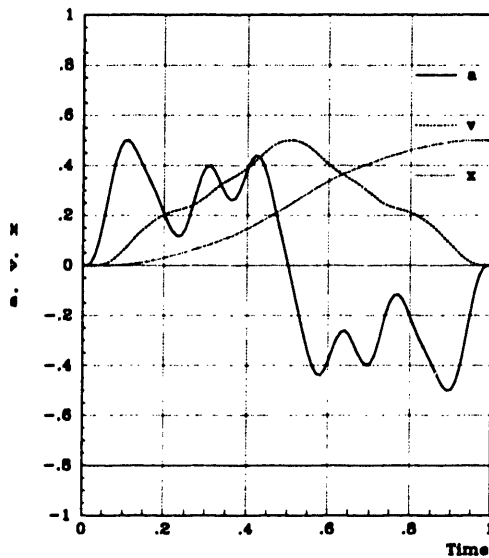


Figure 2-3 Amplitude Scaling

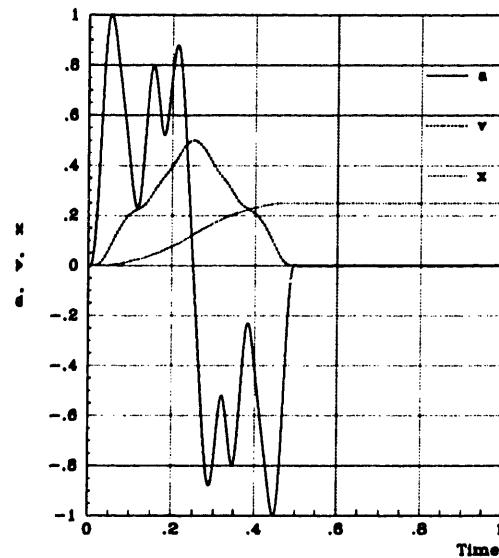


Figure 2-4 Duration Scaling

Conversely, if one scales the duration of an acceleration waveform, then the corresponding velocity waveform's amplitude is scaled by the same factor, but the position waveform's amplitude is scaled by the factor squared. If an anti-symmetric acceleration waveform results in a unit move, then a similar acceleration waveform with half the original duration will result in one fourth the original movement. This is shown in Figure 2-4.

These concepts are very simple, yet they provide a powerful method of implementing moves of arbitrary length. Any anti-symmetric acceleration waveform corresponding to a unit move can be rescaled in amplitude, duration, or both to implement moves of arbitrary length. The only restriction is that the physical system must be capable of operating at the resulting acceleration and velocity levels.

Relevant literature proposes several smooth anti-symmetric acceleration waveforms. These waveforms are similar in that most of their energy is at low frequencies.

2.3.1 Gaussian Shaped Acceleration Trajectory

In his Doctoral Thesis, Dr. Neil Singer uses a Gaussian-shaped input trajectory¹ as the acceleration command in a simulation of the Space Shuttle Remote Manipulator System². Dr. Singer compared the simulated response of the shuttle arm moved by the existing closed-loop controller with the response of the arm when an open-loop Gaussian acceleration was used. Dr. Singer reported: "The residual vibration was reduced by a factor of 25 using the Gaussian input."

$$a(t) \propto \frac{-(t-\frac{1}{2}) e\left(\frac{-(t-\frac{1}{2})^2}{2\sigma^2}\right)}{\sqrt{2\pi}\sigma^3}$$

where $\sigma = 2$

(2.2) Gaussian Shaped Acceleration Trajectory

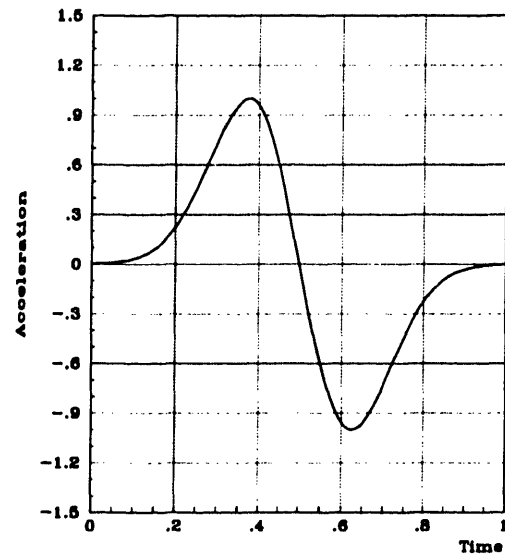


Figure 2-5 Gaussian Shaped Acceleration Trajectory

In the same thesis, Dr. Singer also discussed the use of acausal filtering to remove energy from a command signal at frequencies corresponding to mechanical modes of vibration in the system being controlled. Specifically, the time-optimal open-loop command signal (for a system that has both acceleration and velocity limits) is notch-filtered at two modal frequencies before application to the plant. The filtered command signal reduced the residual vibration at the end of a move by a factor of four.

¹Dr. Singer's Gaussian-shaped input trajectory is proportional to the derivative of the Gaussian pulse described in Equation 2.1. This trajectory results from the convolution of the doublet with the Gaussian pulse.

²Dr. Neil Singer, "Residual Vibration Reduction in Computer Controlled Machines," Massachusetts Institute of Technology, 1989, pages 7-12.

Most of Dr. Singer's thesis is devoted to discussing a method of generating vibration-free motion. His method accelerates a structure with a collection of appropriately scaled and timed impulses. The concept is easily explained if the structure is assumed to have one damped second-order mode. An initial impulse accelerates the structure and excites the mode. A second impulse, delayed by half of the mode's period in order to make its response out of phase with the original vibration, is applied to the structure. The second impulse must have an amplitude that is lower by the amount of dampening that occurs over the half cycle of delay. The second impulse again accelerates the structure and excites the mode, but the mode's response to the second impulse exactly cancels the response from the original impulse. The result is that the structure is accelerated to some velocity, but its motion is vibration-free. Since the structure is assumed to be linear, another mode would only require another impulse for cancellation. The prospects of vibration-free motion are appealing; however, this technique requires both considerably more knowledge of the plant's characteristics and more computational resources than are available in a disk drive controller implementation.

2.3.2 'SMART' Acceleration Trajectory

Susumu Hasegawa, Kazuhiko Takaishi, and Yoshifumi Miozoshita propose an anti-symmetric acceleration waveform that they call SMART (Structural Vibration Minimized Acceleration Trajectory)¹. They use optimal control theory to find an acceleration trajectory which minimizes the square of the acceleration's derivative integrated over the move duration. This is represented by the cost function, J , in Equation 2.3.

$$J = \int_0^T \left(\frac{da}{dt} \right)^2 dt \quad (2.3) \text{ SMART Cost Function}$$

Where ' a ' is acceleration.

The SMART acceleration trajectory required to execute a unit move in a time, T , is shown in Figure 2-6 on the following page.

¹ Susumu Hasegawa, Kazuhiko Takaishi, and Yoshifumi Miozoshita, "Digital Servo Control for Head-Positioning of Disk Drives," Fujitsu Scientific and Technical Journal, Volume 26, Number 4, February 1991, pages 378-390.

$$a(t) \propto \left(\frac{1}{T}\right)^2 \left[2\left(\frac{t}{T}\right)^3 - 3\left(\frac{t}{T}\right)^2 + \left(\frac{t}{T}\right) \right]$$

(2.4) SMART Acceleration Trajectory

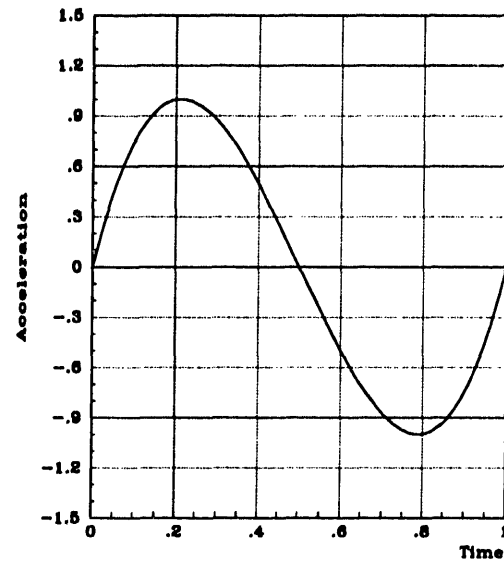


Figure 2-6 SMART Acceleration Trajectory

The SMART trajectory provides fast move times because the acceleration and deceleration levels are high near the beginning and end of the move. This makes the velocity high over a large portion of the move. Further, this trajectory is represented by a low-order polynomial in time, which may be beneficial when implementing a control system since low-order polynomials are easily computed. This waveform is not completely desirable because the abrupt edges at the ends of the trajectory must include some energy at high frequencies.

2.3.3 Piecewise Trigonometric Forcing Functions

John Scaramuzzo Jr. suggests using raised cosine segments as a method of constructing acceleration waveforms¹. His method uses half cycles of the cosine chosen with zero slope at the endpoints to join regions of constant acceleration. Mr. Scaramuzzo refers to the resulting acceleration waveforms as Piecewise Trigonometric (PWT) Forcing Functions. This method has several appealing characteristics. It results in a smooth acceleration waveform with a continuous derivative at every point. Also, the duration of the move time and the high-frequency content of the acceleration are easily traded off against each other by simultaneously varying the period of the cosine segments and the

¹ John Scaramuzzo Jr., Masters Thesis, "Band Limited Control of a Flexible Structure Using Piecewise Trigonometric Forcing Functions," Harvard University, May, 1992.

length of the constant-acceleration segments. Additionally, asymmetric acceleration waveforms, which are desirable when a plant's acceleration characteristics are also asymmetric, can be created using this method. Figure 2-7 depicts an anti-symmetric PWT forcing function, while Figure 2-8 shows an asymmetric PWT forcing function.

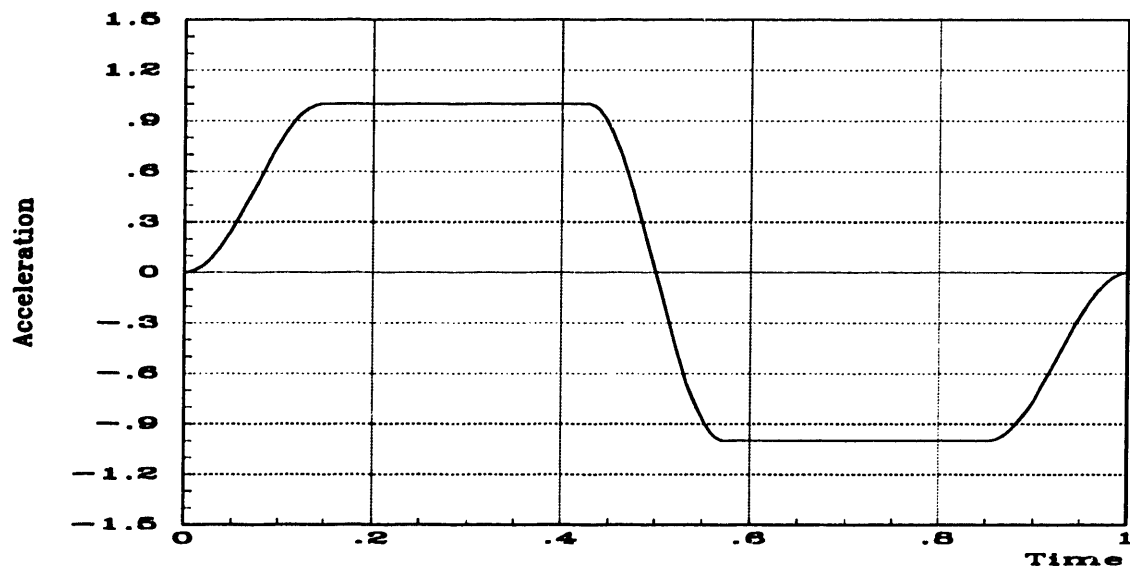


Figure 2-7 Piecewise Trigonometric Anti-Symmetric Acceleration Trajectory

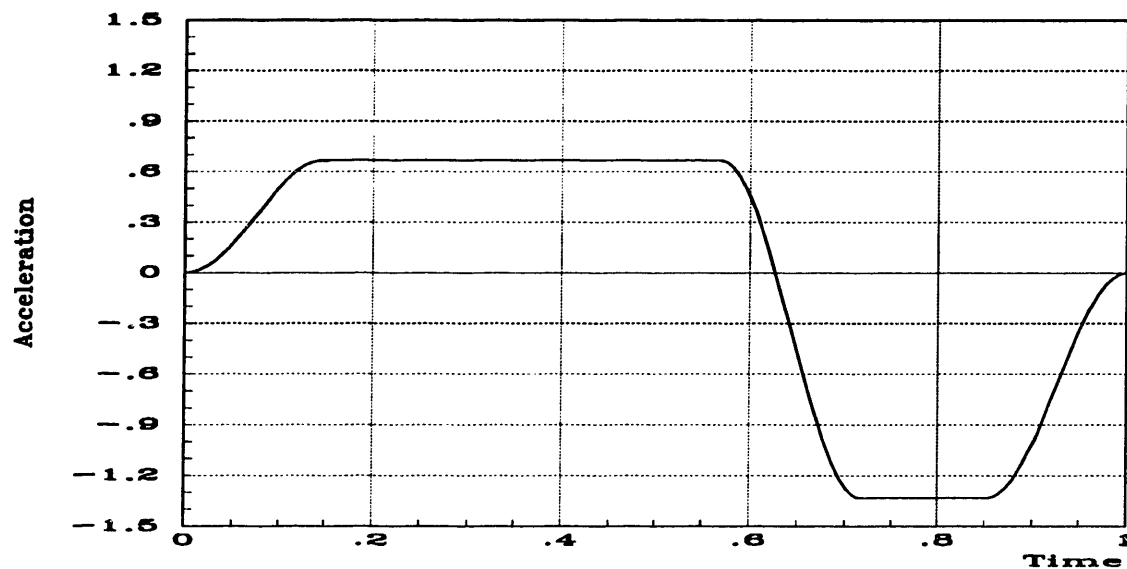


Figure 2-8 Piecewise Trigonometric Asymmetric Acceleration Trajectory

2.3.4 Ramped Sinusoid Function

Peter Meckl employs a set of functions he calls ramped sinusoids as a basis to create anti-symmetric acceleration trajectories. Ramped sinusoids are of the form shown in Function 2.5. They are linear combinations of sinusoids and a ramp function. Meckl explains how a finite sum of these functions can be fit to a square wave via a Least-Squares method in his Master Thesis¹. The resulting acceleration trajectories resemble square waves that have been processed with an ideal low-pass filter (i.e. they resemble square waves exhibiting Gibb's phenomena). This approach to generating an anti-symmetric waveform is computationally intensive and was not pursued for this project. However, it is interesting that one period of the ramped-sinusoid function shown in Figure 2-9 resembles the Gaussian-shaped trajectory presented in Figure 2-5 and the SMART trajectory shown in Figure 2-6. Further, it strongly resembles the waveform that results from omitting the constant acceleration segments in John Scaramuzzo's anti-symmetric PWT shown in Figure 2-7.

$$\frac{1}{\lambda^2} \left[\lambda \left(t - \frac{T}{2} \right) - \sin \lambda t + \frac{\lambda T}{2} \cos \lambda t \right]$$

$$\left\{ \begin{array}{l} \text{For the plot shown:} \\ T = 0.417 \\ \lambda = 21.6 \end{array} \right\}$$

(2.5) Ramped-Sinusoid
Basis Function

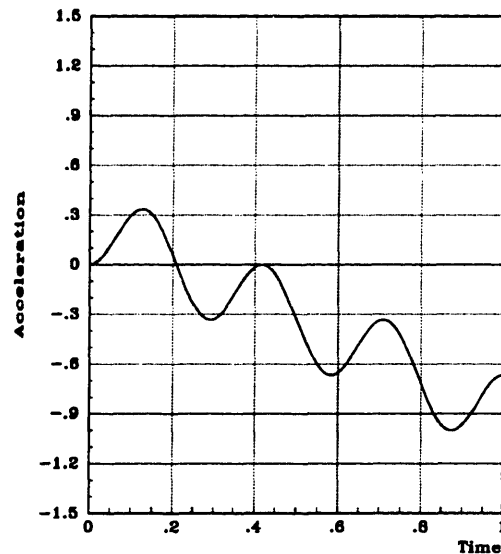


Figure 2-9 Ramped-Sinusoid Basis Function

¹Peter Meckl, Masters Thesis, "Minimizing Residual Vibration of a Linear System Using Appropriately Shaped Forcing Functions," Massachusetts Institute of Technology, May 1984, page 31.

2.4 The Smooth Anti-Symmetric (SAS) Acceleration Waveform

One of the objectives of this project was to implement a method of reducing acoustic noise on a production disk drive. Engineering experience suggested that the author use a simple, easily analyzed and easily computed function as the basic anti-symmetric acceleration waveform. The author chose the asymmetric acceleration waveform which results from the convolution of a half-cycle sine smoothing pulse with a rectangular acceleration pulse-pair. Specifically, the rectangular acceleration pulse-pair consists of a rectangular acceleration pulse immediately followed by a rectangular deceleration pulse of the same magnitude and duration. The half-cycle sine smoothing pulse has unit area and shares the duration of each of the rectangular pulses. The duration of the convolution product is three times the duration of the individual pulses. Figure 2-10 depicts the convolution of a rectangular pulse-pair with a half-cycle sine smoothing pulse. This convolution product is the smooth anti-symmetric (SAS) acceleration waveform employed by the short-seek algorithm in this thesis.

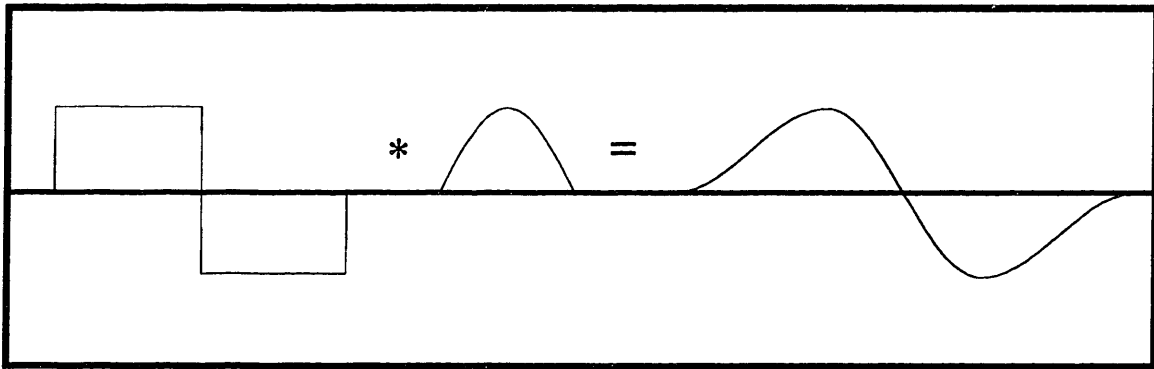


Figure 2-10 Forming the SAS Waveform by Convolution

The following formulas describe the SAS acceleration waveform for a unit move during a total move time, T :

$$\begin{aligned}
 &\frac{9}{T^2} \left(\frac{1}{2} \right) \left(1 - \cos\left(3\pi\frac{t}{T}\right) \right) && \text{for } 0 < t \leq \frac{T}{3} \\
 &\frac{9}{T^2} (-1) \left(\cos\left(3\pi\frac{t}{T}\right) \right) && \text{for } \frac{T}{3} < t \leq \frac{2T}{3} \\
 &\frac{9}{T^2} \left(-\frac{1}{2} \right) \left(1 + \cos\left(3\pi\frac{t}{T}\right) \right) && \text{for } \frac{2T}{3} < t \leq T
 \end{aligned} \tag{2.6} \text{ SAS Acceleration}$$

Equation 2.6 shows that the SAS waveform is composed of three cosine segments; the first and last being offset, or 'raised'. Therefore the SAS acceleration is a special case of a piecewise trigonometric (PWT) forcing function suggested by John Scaramuzzo in Section 2.3.3. Specifically, the SAS acceleration is a piecewise trigonometric forcing function without constant-acceleration segments.

Expression 2.7 describes the Fourier Transform of the SAS acceleration required to make a unit move in time, T :

$$\frac{9}{T^2} \frac{j e^{-j\pi f T} \sin(\frac{2\pi f T}{3}) \sin(\frac{\pi f T}{3})}{\pi f [1 - (\frac{2fT}{3})^2]} \quad (2.7) \text{ SAS Acceleration Fourier Transform}$$

The high frequency envelope of the SAS acceleration required to make a unit move is described as follows:

$$\frac{9}{T^2} \frac{\frac{4}{3^{1.5}}}{\pi f [1 - (\frac{2fT}{3})^2]} \Rightarrow \frac{3^{2.5}}{\pi f^3 T^4} \quad (2.8) \text{ SAS Acceleration Fourier Envelope}$$

The spectrum of the SAS waveform rolls off as $1/f^3$, or at 60 dB/decade. It also rolls off as a function of move time, $1/T^4$.

Figure 2-11 shows a plot of the SAS acceleration waveform corresponding to a one meter move in 3 milliseconds. Note the high acceleration requirements; a move of one track on the RA73, about 10 micrometers, demands peak accelerations of 10 meters per second-squared. Figure 2-12 shows a plot of the magnitude of the spectrum associated with the SAS acceleration depicted in Figure 2-11.

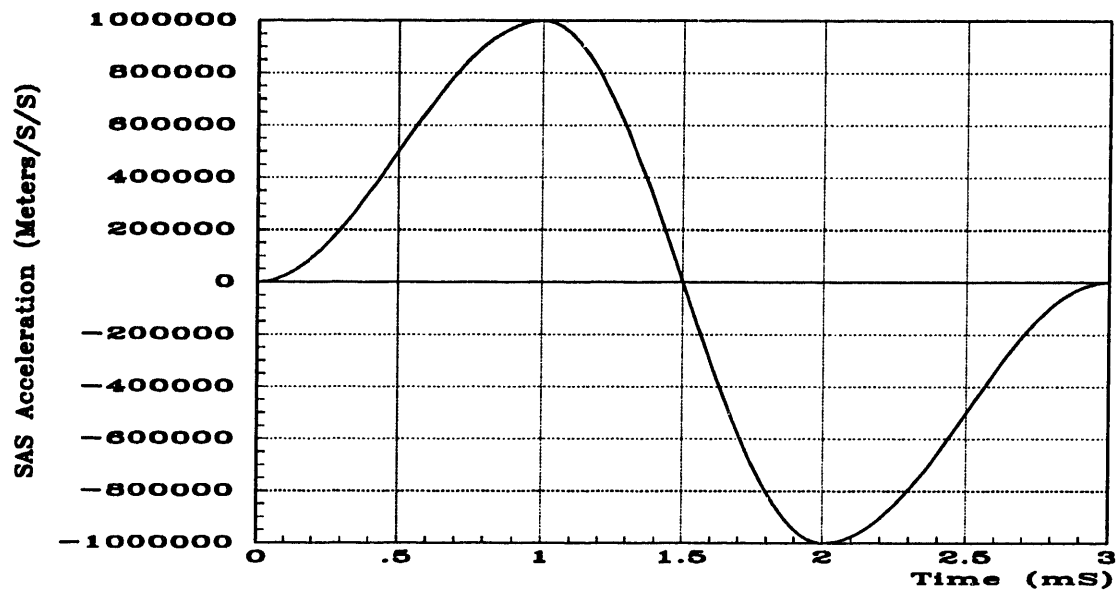


Figure 2-11 SAS Acceleration Profile, 1 Meter Move

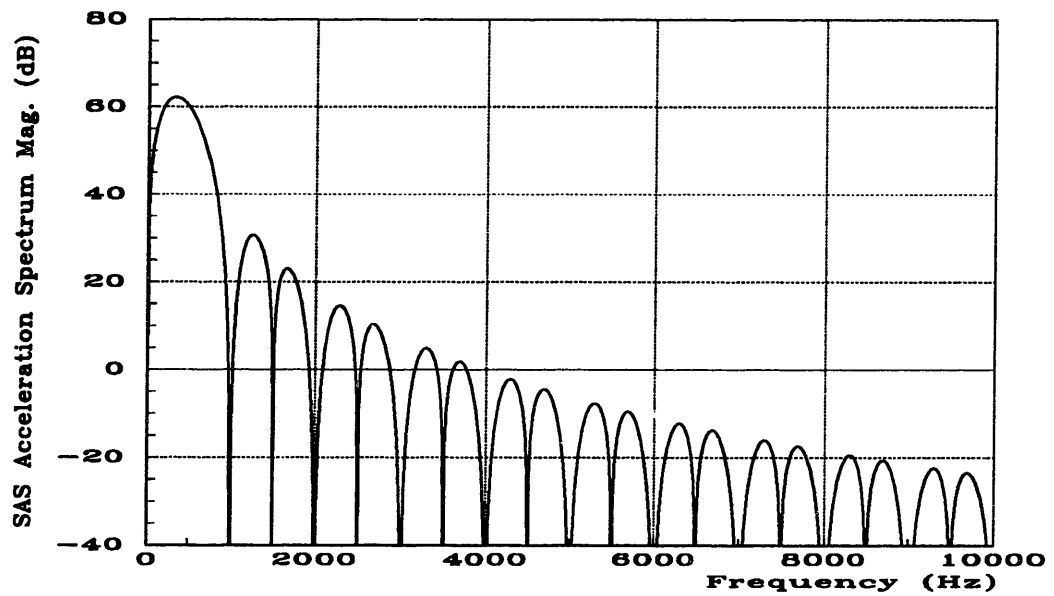


Figure 2-12 SAS Acceleration Spectrum, 1 Meter Move

Chapter 3: Implementation

This thesis develops two move algorithms to implement quiet seeks. The first algorithm described is useful for implementing short moves and is based upon the smooth anti-symmetric acceleration. This algorithm exploits time and amplitude scaling to effect seeks which do not demand the maximum acceleration current afforded by the RA73. A second algorithm is essential because long moves, of short duration, require the maximum acceleration current capabilities of the RA73. The RA73's maximum current capabilities during long moves are both asymmetric and move length dependent. Consequently, the long-seek algorithm cannot exploit time and amplitude scaling to the same extent as the short-seek algorithm. However, the long-seek algorithm is founded upon assumptions which are not valid for short moves. Therefore the long-seek algorithm cannot effect short moves. The long-seek algorithm is based upon accelerations corresponding to ideal open loop accelerations smoothed by convolution with a truncated Gaussian pulse. The RA73's differing acceleration capabilities between short and long moves precipitated the development of two algorithms.

Both algorithms use a linear position tracking loop to implement moves. A controller generates a smoothed step command to produce a position reference during a move. The accelerations required by the position tracking loop as it follows the reference are also smooth and therefore quiet.

This chapter first describes the trajectory-following loop, then discusses the short-seek algorithm, and finally explains the long-seek algorithm.

3.1 The Trajectory-Following Loop

A conceptual block diagram of the new controller structure which permits implementation of smoothed acceleration profiles appears in Figure 3-1. The block diagram shows the three reference inputs generated by the controller. The reference inputs correspond to position, velocity, and acceleration commands. The tracking loop uses proportional and derivative (PD) feedback for compensation. In the absence of input commands the loop causes the actuator to track-follow at the zero location.

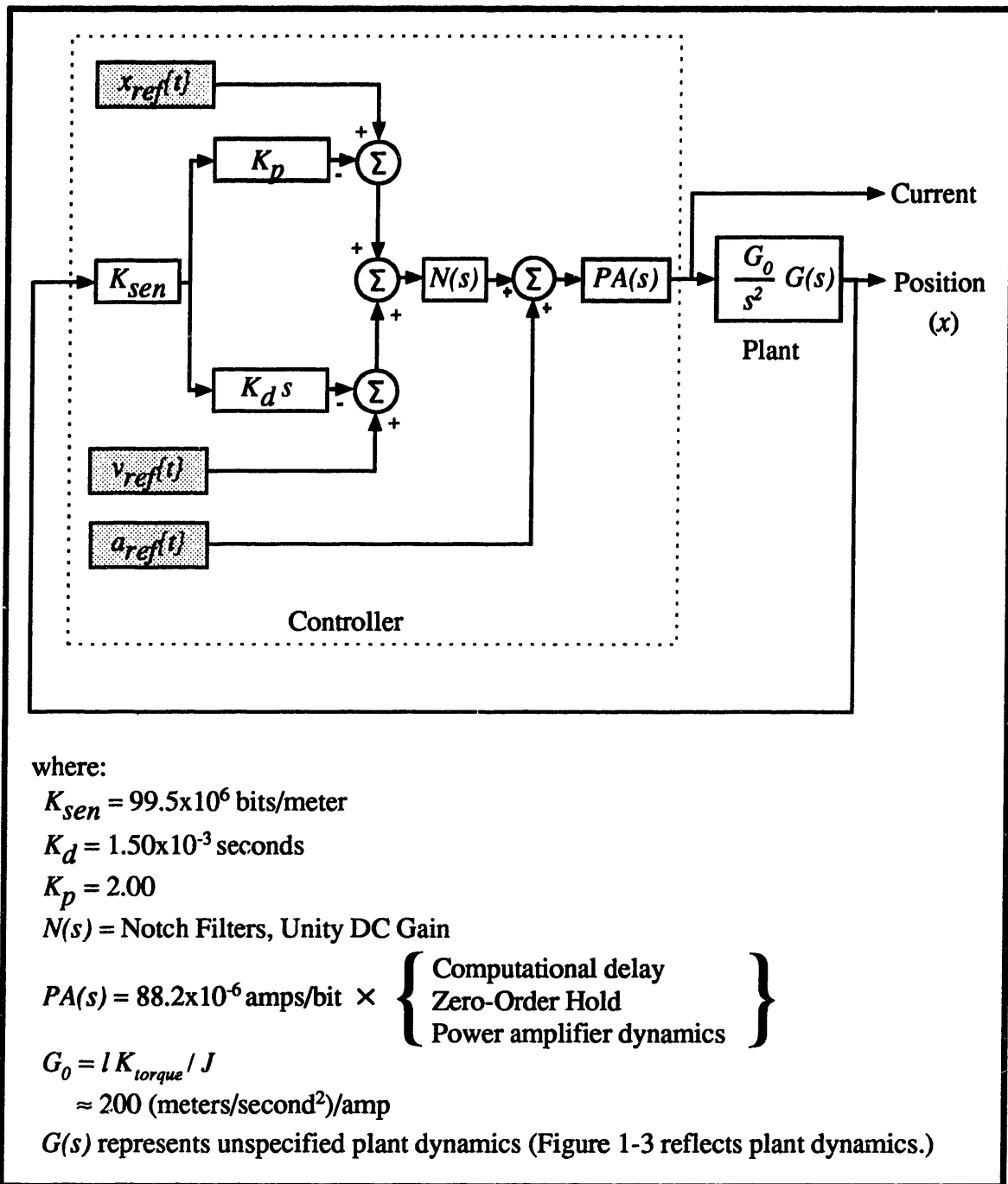


Figure 3-1 Trajectory-Following Seek Loop

The PD tracking loop used in the short-seek algorithm is similar to the RA73's PID tracking loop used for normal track following. The differences are the PID loop includes integral position-error feedback and has 17% more proportional gain. Both of these differences provide more gain at low frequencies at the expense of phase margin at

the gain crossover in the normal tracking loop. The normal PID tracking loop has an additional second-order lead-lag compensator to help recover the lost phase margin. The PD loop does not use the lead-lag compensator.

Equation 3.1 describes the PD compensator gain from the position sensor to the actuator-coil current.

$$H(s) = -K_{sen} (K_p + K_d s) N(s) PA(s) \quad (3.1) \text{ Compensator Gain}$$

The open-loop transfer function at the plant output is approximated by the expression in Equation 3.2 if all high-frequency dynamics are ignored. Equation 3.2 ignores the dynamics of the notch filters, $N(s)$, and approximates the dynamics of the power amplifier, a zero-order hold, and computational delay to $PA(0)$, or PA_0 .

$$\begin{aligned} T(s) &= K_{sen} \left[(K_d s + K_p) N(s) \right] \frac{PA(s) G_0}{s^2} G(s) \\ &\approx \frac{K_{sen} K_d PA_0 G_0}{s^2} \left[s + \frac{K_p}{K_d} \right] \\ &= \frac{470 \text{ Hz}}{(fj)^2} [(fj) + 210 \text{ Hz}] \end{aligned} \quad (3.2) \text{ Open-Loop Function}$$

This open-loop function has a gain crossover just above 500 Hz and a calculated phase margin of 67 degrees. The high-frequency dynamics, including several time delays, reduce the phase margin by about 25 degrees. The following expressions approximate the closed-loop function from x_{ref} to the servo's sensor position:

$$\begin{aligned} \frac{\text{Position}(s)}{x_{ref}(s)} &= \frac{N(s) PA(s) \frac{G_0}{s^2} G(s)}{1 + T(s)} \\ &\approx \frac{\frac{PA_0 G_0}{s^2}}{1 + \frac{K_{sen} K_d PA_0 G_0}{s^2} \left[s + \frac{K_p}{K_d} \right]} \\ &= \frac{PA_0 G_0}{s^2 + K_{sen} K_d PA_0 G_0 s + K_{sen} K_p PA_0 G_0} \\ &= \frac{(5 \times 10^{-9} \text{ m/bit}) (315 \text{ Hz})^2}{(fj)^2 + 2(0.74)(315 \text{ Hz})(fj) + (315 \text{ Hz})^2} \end{aligned} \quad (3.3) \text{ Closed-Loop Function}$$

This closed loop function shows that the response of the servo to only a position reference command (i.e. without the corresponding velocity reference command and the acceleration feedforward) deteriorates above 315 Hz. Specifically, the sensor's position will lag the reference as the reference frequency increases. Additionally, as the reference frequency increases the magnitude of the sensor's response to a sinusoidal command will continually decrease after peaking below 315 Hz. The success of this implementation depends upon the simultaneous use of the feedback and feedforward signals.

Equation 3.4 describes the output of the controller diagrammed in Figure 3-1.

(3.4) Controller Output

$$u = \left[\left((x_{ref} - K_{sen} K_p x) + (v_{ref} - K_{sen} K_d s x) \right) N(s) + a_{ref} \right] PA(s)$$

Equation 3.4 shows the controller output as a function of the three commands, x_{ref} , v_{ref} , and a_{ref} . This gives the impression that the controller has multiple inputs. This is a false impression because the inputs are related. Specifically, v_{ref} is the first derivative of x_{ref} and a_{ref} is the second derivative of x_{ref} . One can represent v_{ref} and a_{ref} as $K_v x_{ref} s$ and $K_a x_{ref} s^2$ respectively, through the appropriate choice of scale factors K_v and K_a . Therefore the controller output can also be described by the following expressions:

(3.5) Controller Output with Interdependent Reference Profiles

$$\begin{aligned} u &= \left[\left((x_{ref} - K_{sen} K_p x) + (K_v s x_{ref} - K_{sen} K_d s x) \right) N(s) + K_a s^2 x_{ref} \right] PA(s) \\ &= \left((1 + K_v s) N(s) + K_a s^2 \right) PA(s) x_{ref} - K_{sen} (K_p + K_d s) N(s) PA(s) x \end{aligned}$$

Equation 3.5 shows that the controller output is a combination of a pre-compensated version of the position reference ($[(1 + K_v s) N(s) + K_a s^2] PA(s) x_{ref}$) and a feedback signal ($K_{sen} [K_p + K_d s] N(s) PA(s) x$). Equation 3.5 also shows that if the dynamics of the notch filters, $N(s)$, and the output stage, $PA(s)$, are ignored, then the pre-compensation is second-order and has two zeros. The zero locations are determined by K_v and K_a . K_v and K_a are chosen so that the pre-compensation zeros cancel the closed-loop poles in Equation 3.3. If the pole-zero cancellation is exact, then the closed-loop function becomes a proportionality constant, as shown in Equation 3.6, and the servo-sensor position will follow the position reference without error.

(3.6) Closed-Loop Function with Interdependent Reference Profiles

$$\begin{aligned}
 \frac{\text{Position}(s)}{x_{ref}(s)} &= \frac{\left((1 + K_v s) N(s) + K_a s^2 \right) P A(s) \frac{G_0}{s^2} G(s)}{1 + T(s)} \\
 &\approx \frac{\left((1 + K_v s) + K_a s^2 \right) P A_0 \frac{G_0}{s^2}}{1 + \frac{K_{sen} K_d P A_0 G_0}{s^2} \left[s + \frac{K_p}{K_d} \right]} \\
 &= \frac{\left((1 + K_v s) + K_a s^2 \right) P A_0 G_0}{s^2 + K_{sen} K_d P A_0 G_0 s + K_{sen} K_p P A_0 G_0} \\
 &= \frac{\left(s^2 + \frac{K_v}{K_a} s + \frac{1}{K_a} \right) K_a P A_0 G_0}{s^2 + K_{sen} K_d P A_0 G_0 s + K_{sen} K_p P A_0 G_0} \\
 &= \frac{1}{K_{sen} K_p} \quad \text{when } K_a = \frac{1}{K_{sen} K_p P A_0 G_0} \text{ and } K_v = \frac{K_d}{K_p}
 \end{aligned}$$

The ability of the controller to follow signals outside the bandwidth of the closed-loop (roughly 315 Hz) degrades if the pole-zero cancellation is imperfect. The closed-loop pole locations are dependent upon plant parameters which are subject to variation. Specifically, the actuator's torque constant varies with position, is non-linear, and is not constant unit to unit. Further, the high-frequency dynamics of the notch filters, the output stage, and the plant become increasingly significant as the frequency rises above 500 Hz. Equation 3.6 ignores these high-frequency dynamics. Ideal pole-zero cancellation cannot be achieved in the disk drive system. Consequently, the controller is unable to satisfactorily follow commands with significant frequency content above approximately 500 Hz. Fortunately, in this application, the commands are shaped such that their frequency content concentrates at low frequencies.

3.2 The Short-Seek Algorithm

Actuator control in the short-seek algorithm is described by a tracking loop, following both position and velocity reference signals, with acceleration feedforward. Each of the position and velocity reference signals and the feedforward command is derived from a single set of position, velocity and acceleration trajectories stored in the servo's digital signal processor's (DSP's) memory. These trajectories correspond to the smooth anti-symmetric, SAS, acceleration. The three trajectories are each represented by 128 samples.

The drive may execute any length move by appropriately scaling the SAS trajectories provided the required accelerations do not saturate the power amplifier. If the required accelerations were to cause the power amplifier to saturate, then the controller would be unable to follow the smooth command trajectories. This could cause the controller to overshoot the destination track. Additionally, the high-frequency content of the clipped coil current could excite vibration and in turn cause acoustic noise emission. This algorithm is predicated upon accelerations which do not demand the maximum current afforded by the RA73. If the required accelerations are too high, then the time duration of the seek must be extended. The required accelerations scale as the reciprocal of the square of the move time.

Moves of up to 200 tracks can be implemented practically with this algorithm. Longer moves are also possible; however, they require excessive time because the back EMF generated in the coil at the peak velocity of the actuator limits the allowable acceleration. Subsequent sections develop a separate algorithm which considers the back EMF effect to efficiently implement long moves. Contrasting the short-seek algorithm, the algorithm developed for long moves exploits nearly the maximum current capability of the RA73.

3.2.1 The Short-Seek Continuous-Time Reference Trajectories

Consider the case in which the trajectories correspond to a unit move. The design procedure starts by first integrating the unit-move SAS acceleration trajectory (described by Equation 2.6) to obtain a velocity trajectory, and then integrating again to yield a position trajectory. The following equation describes the resulting S-shaped, unit-duration, unit-move position trajectory.

(3.7) Unit-Move Position Trajectory

$$x_{unit\ move}(\tau) = \begin{cases} \left(\frac{3\tau}{2}\right)^2 + \frac{\cos 3\pi\tau - 1}{2\pi^2} & 0 \leq \tau \leq \frac{1}{3} \\ -\frac{1}{4} + \frac{3\tau}{2} + \frac{\cos 3\pi\tau}{\pi^2} & \frac{1}{3} < \tau \leq \frac{2}{3} \\ -\frac{5}{4} + \frac{9\tau}{2} - \left(\frac{3\tau}{2}\right)^2 + \frac{\cos 3\pi\tau + 1}{2\pi^2} & \frac{2}{3} < \tau \leq 1 \end{cases}$$

The unit-move trajectory specifies a reference position command signal for the controller as the actuator moves a unit distance away from the position origin in unit time. The corresponding unit-move velocity command is shown in Equation 3.8. Equation 3.9 defines the associated acceleration command.

(3.8) Unit-Move Velocity Command

$$v_{unit\ move}(\tau) = \begin{cases} \frac{9\tau}{2} - \frac{3\sin 3\pi\tau}{2\pi} & 0 \leq \tau \leq \frac{1}{3} \\ \frac{3}{2} - \frac{3\sin 3\pi\tau}{\pi} & \frac{1}{3} < \tau \leq \frac{2}{3} \\ \frac{9}{2} - \frac{9\tau}{2} - \frac{3\sin 3\pi\tau}{2\pi} & \frac{2}{3} < \tau \leq 1 \end{cases}$$

(3.9) Unit-Move Acceleration Command

$$a_{unit\ move}(\tau) = \begin{cases} \frac{9}{2} - \frac{9\cos 3\pi\tau}{2} & 0 \leq \tau \leq \frac{1}{3} \\ -9\cos 3\pi\tau & \frac{1}{3} < \tau \leq \frac{2}{3} \\ -\frac{9}{2} - \frac{9}{2}\cos 3\pi\tau & \frac{2}{3} < \tau \leq 1 \end{cases}$$

Equation 3.10 describes the position command signal required to move from an arbitrary start position, x_0 , to an arbitrary end position, x_1 , in unit time.

$$x(\tau) = x_0 + (x_1 - x_0) x_{unit\ move}(\tau) \quad (3.10) \text{ Position Command, Unit Time}$$

The move length similarly scales the velocity and acceleration commands, as described below:

$$v(\tau) = (x_1 - x_0) v_{unit\ move}(\tau) \quad (3.11) \text{ Velocity Command, Unit Time}$$

$$a(\tau) = (x_1 - x_0) a_{unit\ move}(\tau) \quad (3.12) \text{ Acceleration Command, Unit Time}$$

The command descriptions are modified again if the duration of the move is T rather than 'unit time', as shown below:

(3.13) Reference Commands, Move Time ' T '

$$x_{ref}(t) = x_0 + (x_1 - x_0) x_{unit\ move}\left(\frac{t}{T}\right) \quad \text{where, } \tau \rightarrow \frac{t}{T}$$

$$v_{ref}(t) = \frac{(x_1 - x_0)}{T} v_{unit\ move}\left(\frac{t}{T}\right)$$

$$a_{ref}(t) = \frac{(x_1 - x_0)}{T^2} a_{unit\ move}\left(\frac{t}{T}\right)$$

The unit-move trajectories are scaled in amplitude and time to create the reference commands. Therefore all reference commands will have the same shape as shown in Figure 3-2.

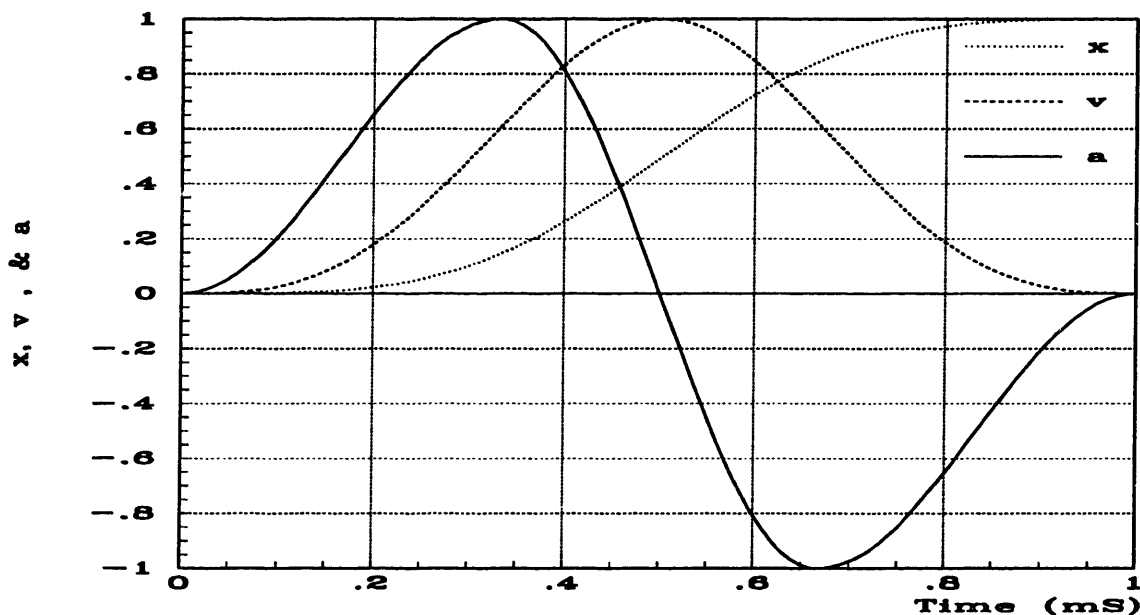


Figure 3-2 Command Shapes

The sum of the start location and a scaled unit-move defines the position command. An equivalent definition of the position command is implemented in the hardware, since it is crucial that the actuator arrives at exactly the desired end location. This definition is described by Equation 3.14.

$$x_{ref}(t) = x_1 - (x_1 - x_0) \left(1 - x_{unit\ move}\left(\frac{t}{T}\right) \right) \quad (3.14) \text{ Position Command}$$

This definition has the advantage that the term on the right will go to zero as long as the discrete-time representation of $(1 - x_{unit\ move})$ is stored as a zero at the end-point. This causes the reference command to terminate at the desired end location irrespective of the precision of the computational hardware in the DSP.

3.2.2 The Short-Seek Discrete-Time Reference Trajectories

Implementing the short-seek algorithm with a DSP-based servo system required the DSP to generate three command references. Computation time had to be kept low since the control signal is updated every 47 microseconds in the RA73. The author chose to interpolate between stored trajectory samples to generate the needed references, rather than calculate the references from their trigonometric and polynomial definitions. This reduces the computation time needed to generate the references. Three 128 element tables are used; the word width of the tables is 16 bits.

There is some design freedom in creating the velocity and acceleration tables. For example, several methods can be used to create the velocity table: It may be generated as a sampled version of the continuous velocity signal. Alternatively, it may be computed as the first backward difference of the sampled position reference. Or, it may be defined as the average of the first forward and the first backward difference of the sampled position reference, etc. Any of these methods would likely provide an adequate velocity reference; however, a method which facilitates interpolation of the position reference between samples is preferred.

If the seek algorithm sequenced through the trajectory tables at the rate of one element every time the DSP updates the control signal, then interpolation would not be necessary and the duration of every seek would be fixed at 128 times 47 microseconds, or 6 milliseconds. This would unnecessarily slow short seeks and is therefore impractical. Instead, the DSP sequences through the trajectory tables at a rate which varies with seek length. The DSP uses a 15-bit counter to keep track of the time course of the seek. The counter ranges from zero to its maximum value (32,767) over the course of every seek. The algorithm increments this counter once per control update. The size of the increment depends upon the seek length. Short seeks require the least move time, consequently the increment is largest for short seeks. The counter's upper seven bits index the stored trajectory elements. The lower eight bits allow the interpolation of the position and velocity references between the stored trajectory samples. These bits represent a fraction of the time between trajectory samples. The time between stored

trajectory samples, T_{trj} , is the seek duration divided by the number of stored samples, 128. The time between stored samples, T_{trj} , multiplied by the fraction represented by the counter's lower eight bits, Δ , describes the fractional time between computed reference samples, ΔT_{trj} . The acceleration reference command is not interpolated between the stored samples. Ignoring the dynamics of the power amplifier, the actuator's acceleration may be assumed to be constant between stored samples. This simplifying assumption permits the actuator's position between trajectory samples to be described by an equation of the form: $x = x_0 + v_0 \Delta t + \frac{1}{2} a_{ref} (\Delta t)^2$. Equation 3.15 more precisely describes the actuator's position over one trajectory sample period {for $nT_{trj} \leq t < (n+1)T_{trj}$ }.

(3.15) Interpolated Position Reference

$$x(t) = x\left((n + \Delta)T_{trj}\right) = x(nT_{trj}) + v(nT_{trj})\Delta T_{trj} + \frac{1}{2}a_{ref}(nT_{trj})(\Delta T_{trj})^2$$

$$\Rightarrow$$

$$x[n, \Delta] = x[n] + v[n]\Delta + \frac{1}{2}a[n]\Delta^2$$

Matching the discrete-time representation to the continuous at $\Delta = 0, \frac{1}{2}$, and 1 provides the following three equations that can in turn be solved for the three discrete-time reference trajectories, $x[n]$, $v[n]$, and $a[n]$:

$$x(nT) = x[n] \quad (3.16) \text{ Continuous to Discrete -Time Conversion}$$

$$x\left((n + \frac{1}{2})T\right) = x[n] + \left(\frac{1}{2}\right)v[n] + \frac{1}{2}\left(\frac{1}{4}\right)a[n]$$

$$x((n + 1)T) = x[n] + v[n] + a[n]$$

\Rightarrow

$$x[n] = x(nT) \quad (3.19) \text{ Sampling } x(t)$$

$$v[n] = -3x(nT) + 4x\left((n + \frac{1}{2})T\right) - x((n + 1)T) \quad (3.17) \text{ Difference Equation for } v[n]$$

$$a[n] = 4x(nT) - 8x\left((n + \frac{1}{2})T\right) + 4x((n + 1)T) \quad (3.18) \text{ Difference Equation for } a[n]$$

Each trajectory stored in the DSP's memory is 128 points long. The discrete-time position trajectory, $x[n]$, is $x_{ref}(\tau)$ sampled 128 times as τ ranges between 0 and 1. A three step process creates the discrete velocity trajectory, $v[n]$. First, $x_{ref}(\tau)$ is sampled 256 times as τ ranges between 0 and 1. The resulting sequence is then filtered according to the difference equation for $v[n]$ (Equation 3.17). Finally, every other sample is discarded to form a 128 length sequence. A similar process creates the discrete acceleration trajectory, $a[n]$, except the difference equation for $a[n]$ (Equation 3.18) replaces the difference equation for $v[n]$. The last samples of $v[n]$ and $a[n]$, which are $v[127]$ and $a[127]$ respectively, are set to 0 since $x_{ref}(\tau) = 1$ for $\tau > 1$. The trajectory definitions described in this paragraph allow the interpolation of the position and velocity references as $x[n,\Delta] = x[n] + v[n]\Delta + \frac{1}{2}a[n]\Delta^2$, and $v[n,\Delta] = v[n] + a[n]\Delta$.

3.2.3 Time and Amplitude Scaling

The primary objective in designing a seek controller is minimizing seek time. This objective is constrained by the following limitations in the case of a controller which also attempts to minimize acoustic noise.

- The seek current cannot contain high frequencies if the drive is to operate quietly.
- The tracking loop controller must precisely follow the reference commands.
- The power amplifier must not saturate, in fact, the entire control system must remain linear.

Referring to Figure 2-12 (SAS Acceleration Spectrum, 1 Meter Move) shows that the power in a 3 millisecond duration SAS acceleration peaks near 500 Hz and is concentrated below 1 KHz. Halving the duration to $1\frac{1}{2}$ milliseconds causes the power to peak at 1 KHz and be concentrated below 2 KHz. Figures 1-11 and 1-12 show that the RA73's coil-current to sound-pressure measurement peaks near 2 KHz; however, it is 40 dB lower at 1 KHz and 60 dB lower at 500 Hz. One suspects that seeks employing SAS accelerations will be quiet if they last at least $1\frac{1}{2}$ milliseconds. The author verified that seeks as short as $1\frac{1}{2}$ milliseconds can be performed quietly. However, it is not noise emission which limits the minimum move duration. The command-following performance, rather than acoustic emissions, limits the minimum seek duration.

Section 3.1 (The Trajectory-Following Loop) showed that the control loop's ability to precisely follow a reference command depends upon the cancellation of the closed-loop poles by the zeros of a pre-compensation function. Imperfect pole-zero cancellation impairs the system's ability to follow commands with significant frequency content above 500 Hz. Accordingly, the short-seek algorithm uses 3 milliseconds as the minimum seek duration. This results in very quiet short seeks and acceptable command tracking performance.

The last factor determining scaling is the need to keep the power amplifier from saturating. Again, referring back to Figure 2-11 (SAS Acceleration Profile, 1 Meter Move), a unit move in 3 milliseconds requires one million distance units per second squared of peak acceleration. The track spacing of the RA73 is 2,468 tracks per inch, which is a distance of very nearly 10 microns (10^{-5} meters) for a single track. Accordingly, a one-track seek in 3 milliseconds requires 10 meters per second-squared peak acceleration. The plant's gain, G_0 , is roughly 220 meters per second-squared per ampere. Therefore the peak current requirement for a one-track seek is 10 meters per second-squared divided by 220 meters per second-squared per ampere, or about 50 milliamps. The design of the RA73 electronics is such that the maximum current that can be commanded is $2\frac{1}{2}$ amperes. However, during acceleration a back EMF voltage develops in the actuator coil which typically causes the power amplifier to saturate when more than 2 amps are required at the onset of a seek¹. Choosing 2 amps as the maximum limit for short seeks, it becomes apparent the drive can in 3 milliseconds perform a seek no longer than 45 tracks (2 amperes divided by 0.045 amperes per track). Performing longer seeks requires more time since move time scales as the square root of move distance when acceleration levels are held constant. Based upon these assumptions, a 200-track seek should require the square root of 200 tracks over 45 tracks times 3 milliseconds, or 6.3 milliseconds. This is longer than the time required by the long-seek algorithm to execute a 200-track seek. Consequently the long-seek algorithm, rather than the short-seek algorithm, is used to implement 200-track seeks. The short-seek algorithm is only used for seek lengths less than 144 tracks. This is because the long-seek algorithm is faster than the short-seek algorithm for 144-track and longer seeks. A 143-track seek has a duration of 6.15 milliseconds and requires 1.5 amperes of peak current. The equations relating first

¹The actuator's large velocity achieved during long seeks causes the power amplifier to saturate with less than one amp drawn; however, the velocity increase is not as significant during short seeks.

seek length to duration and then seek length to peak-current requirements (in tracks, milliseconds, and amperes) are shown as Equation 3.20. The relationships expressed in Equation 3.20 are determined by the definition relating the seek duration to the seek distance. This relationship is founded upon an empirical formula arrived at by adjusting both a scale factor and an exponent until the algorithm achieved short move times and acceptable command following performance while not saturating the power amplifier.

For Distance < 13 tracks: (3.20) Short Seek Distance, Duration, and Peak Current Relationships

$$\text{Duration} = 3 \text{ milliseconds}$$

$$\text{Peak Current (amperes)} = 0.0446 \times \text{Distance (tracks)}$$

For Distance ≥ 13 tracks:

$$\text{Distance (tracks)} = 0.332 \times \text{Duration}^{3.33}$$

$$\text{Duration (milliseconds)} = 1.39 \times \text{Distance}^{0.3}$$

$$\text{Peak Current (amperes)} = 0.402 \times \frac{\text{Distance}}{\text{Duration}^2}$$

$$\text{Peak Current (amperes)} = 0.207 \times \text{Distance}^{0.4}$$

Figure 3-3 shows seek duration plotted against seek length for the short-seek algorithm. The peak-current is plotted against seek length in Figure 3-4.

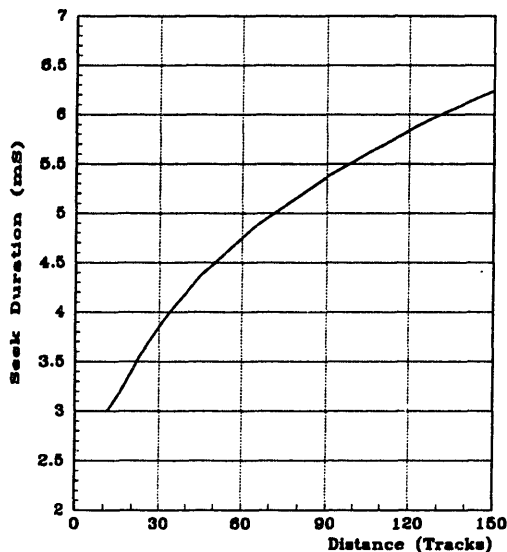


Figure 3-3 Short Seek Duration vs. Distance

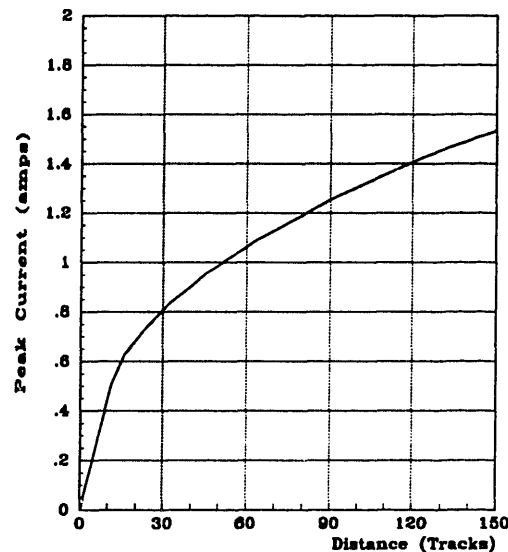


Figure 3-4 Short Seek Peak Current vs. Distance

The DSP scales the reference trajectories in both time and amplitude to implement moves of arbitrary length. The DSP first modifies the trajectories' time scale, then adjusts the amplitudes. The remainder of this section describes the time scaling implementation followed by the amplitude scaling.

A 15-bit counter determines a seek's time scale. The counter ranges from 0 to 32,767 during a seek. The DSP increments the counter every sample time (47 microseconds) by an amount dependent upon the seek length. For example, a 10-track seek executing in 3 milliseconds requires that the DSP increment the counter by 32,767 times 47 microseconds over 3,000 microseconds, or 513, every DSP sample time. Executing a seek in one half of the time requires the counter increment to be twice as large. Seek duration is inversely proportional to the counter's increment.

In addition to the three trajectory tables described earlier, the short-seek algorithm uses three tables of 16 elements each to compute the counter-increment value via interpolation. The first 16-element table contains distance entries. The last two tables represent 16 first-order equations for the counter increment. The columns of Table 3-1 represent sections of the three tables.

Index	Distance, Tracks (First Table)	Increment Base (Second Table)	Increment Slope (Third Table ÷ 32)
6	5.66	513	0
7	8.00	513	0
8	11.31	513	-6.75
9	16.00	481	-7.13
10	22.63	433	-4.53

Table 3-1 Counter Increment Interpolation Tables

The DSP compares the required seek length against the distances in the first table, and finds the index of the largest entry still less than or equal to the required seek length. The DSP uses the index to retrieve both a counter-increment base value from the second table and a counter-increment slope value from the third table. The counter-slope value is multiplied first by the difference between the seek length and the distance read from the first table; that product is then added to the counter-increment base value. Equation 3.21 illustrates the calculation of the counter increment corresponding to

a 15-track seek. The interpolated counter increment controls both the seek duration and the amplitude scaling of the velocity and acceleration trajectories.

(3.21) 15-Track Seek Counter Increment Calculation

$$\text{Increment} = (15 - \text{Distance}_8) \times \text{Slope}_8 + \text{Base}_8 = (15 - 11.31)(-6.75) + 513 = 488$$

The amplitude scaling of the interpolated unit-move position trajectory, $x[n, \Delta]$, is straight forward. The trajectory is multiplied by the required seek length. The amplitude scaling of the interpolated unit-move velocity and acceleration trajectories is only slightly more complicated. Each trajectory is also multiplied by the required seek length, but it must also be modified to account for time scaling. The unit-move trajectories are based upon executing a unit move in 6 milliseconds (128 sample times), which corresponds to a counter increment of 256. Executing the same move in half the time requires twice the velocity and twice the value of the counter increment. The counter increment, normalized by 256, is used to scale the interpolated unit-move velocity trajectory, $v[n, \Delta]$, to account for changes in seek duration. Similarly, cutting the move time in half requires 2^2 , or 4, times more acceleration, therefore the unit move acceleration trajectory, $a[n]$, is scaled by the square of the normalized counter increment. Equation 3.22 represents the algorithm's method of scaling the interpolated position and velocity references in addition to the non-interpolated acceleration command. The unscaled interpolated references have the following definitions: $x[n, \Delta] = x[n] + v[n]\Delta + \frac{1}{2}a[n]\Delta^2$, and $v[n, \Delta] = v[n] + a[n]\Delta$. Equation 3.22 is a discrete-time description of Equation 3.13.

(3.22) Scaled Interpolated Commands

$$X[n, \Delta] = x_0 + (x_1 - x_0)x[n, \Delta]$$

$$V[n, \Delta] = \left(\frac{\text{Increment}}{256}\right)(x_1 - x_0)v[n, \Delta]$$

$$A[n] = \left(\frac{\text{Increment}}{256}\right)^2 (x_1 - x_0)a[n]$$

$$\text{where } n \in [1, 2, 3, \dots, 128] \quad \text{and} \quad 0 \leq \Delta \leq 1$$

3.2.4 Short-Seek Algorithm Summary

The short-seek algorithm effects moves of length between 1 and 143 tracks, in 3 to 6.15 milliseconds respectively, by scaling both the amplitude and duration of two interpolated reference profiles and a non-interpolated acceleration command profile. The move distance of a seek determines both the time and amplitude scaling by specifying a 'counter increment value'. The unscaled interpolated position and velocity reference profiles, and the acceleration command profile, each correspond to a unit-move. The DSP constructs each of the reference profiles by interpolating between the elements of corresponding stored profiles. Each of the stored profiles consists of 128, 16-bit samples. The stored acceleration profile is a smooth anti-symmetric waveform which results from the convolution of an ideal rectangular acceleration pulse-pair with a half-cycle sine smoothing pulse. This waveform has its frequency content concentrated at low-frequencies, consequently the trajectory-following controller discussed in the previous section is capable of precisely following the references. Further, the smooth accelerations that are applied to the actuator's coil do not excite the vibrational modes responsible for acoustic noise emission.

3.3 The Long-Seek Algorithm

The long-seek algorithm, like the short-seek algorithm, follows shaped-command trajectories to implement actuator movements. Both algorithms exploit the same trajectory-following control loop.

The long-seek algorithm employs asymmetric accelerations, this contrasts the anti-symmetric accelerations employed by the short-seek algorithm. The long-seek algorithm uses accelerations corresponding to smoothed versions of the fastest accelerations possible given the RA73 hardware. The accelerations' asymmetries cause the algorithm to be both more complex and less elegant than the short-seek algorithm. The long-seek algorithm constructs nearly 200 unique command trajectories; it then rescales the amplitude of the constructed trajectories to implement over 2,500 unique seek lengths.

3.3.1 Motivation for the Development of the Long Seek Algorithm

The short-seek algorithm is based upon anti-symmetric accelerations which are impractical for long-length seeks. Anti-symmetric accelerations are impractical because they do not reflect the maximum accelerations available to the actuator, consequently the resulting move times become excessive. The root cause of the problem is the back EMF voltage created in the coil when the actuator moves at a high velocity. The back EMF in the coil adds to, or subtracts from, the available control voltage. As a consequence, the coil current, and therefore the available acceleration, must be asymmetric to minimize move time. A more time efficient long-seek algorithm considers the asymmetry of the available acceleration.

The generated back EMF is proportional to the velocity of the actuator, and acts to oppose further acceleration of the actuator. The proportionality factor, or 'back EMF constant'¹, of the RA73's actuator is approximately 0.26 volts per radian per second. During a full-stroke seek, the actuator attains an angular velocity of 37 radians per second. This produces a back EMF of 9.7 volts. The back EMF voltage subtracts from approximately 11½ volts of available control voltage and reduces the available acceleration by over 75% during the seek. The actuator-coil current of a full-stroke seek (2,680 tracks) shows the significant effect of the back EMF in Figure 3-5.

¹ In the MKS system, the back EMF constant expressed in volts per radian per second is numerically equivalent to the actuator torque constant expressed in newton-meters per amp.

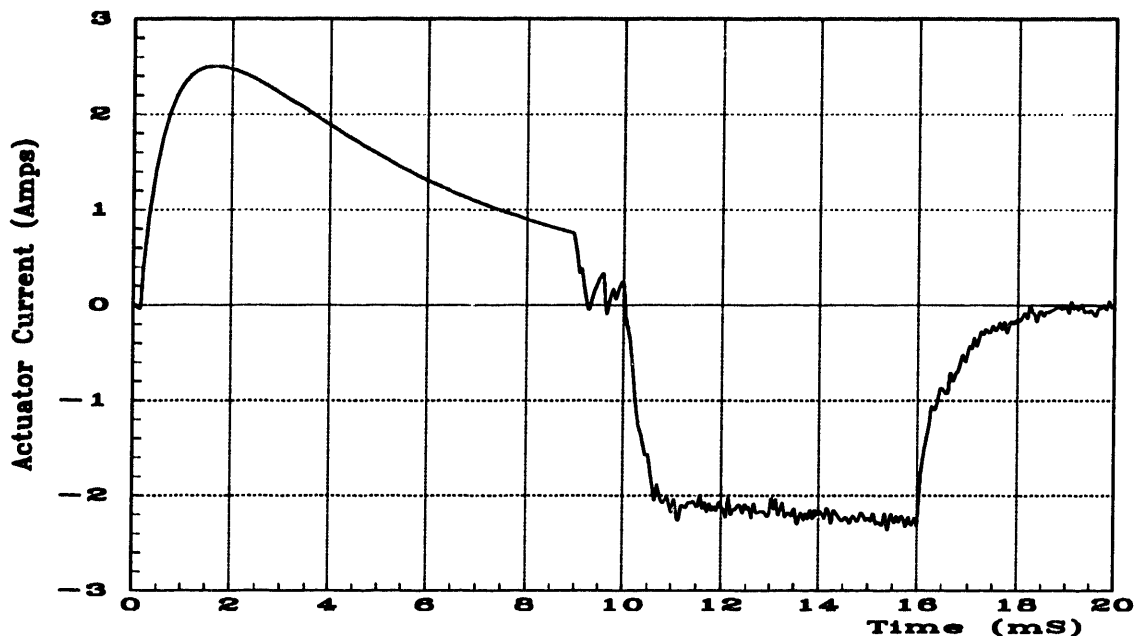


Figure 3-5 Full-Stroke Seek Current

Figure 3-5 shows that the back EMF reduces the actuator's ability to accelerate, but does not impair its ability to decelerate. Instead, the back EMF voltage adds to the available control voltage when decelerating and can aid deceleration. However, the design of the RA73's electronics limits the actuator current to $\pm 2\frac{1}{2}$ amperes. Consequently, the RA73 is unable to use the increased deceleration afforded by the back EMF.

3.3.2 Long-Seek Algorithm Overview

A seek algorithm which accounts for back EMF effects will have asymmetric, rather than anti-symmetric, acceleration. Minimizing move time results from applying the maximum acceleration and deceleration that the system can achieve, as shown in Figure 3-6. The acceleration period shows the effects of both coil inductance and back EMF. The deceleration period reflects the current limiting imposed by the RA73 electronics which constrains the maximum deceleration to a constant level. The waveform presented in Figure 3-6 is termed 'ideal' because it minimizes move times within the limitations of the RA73 disk drive.

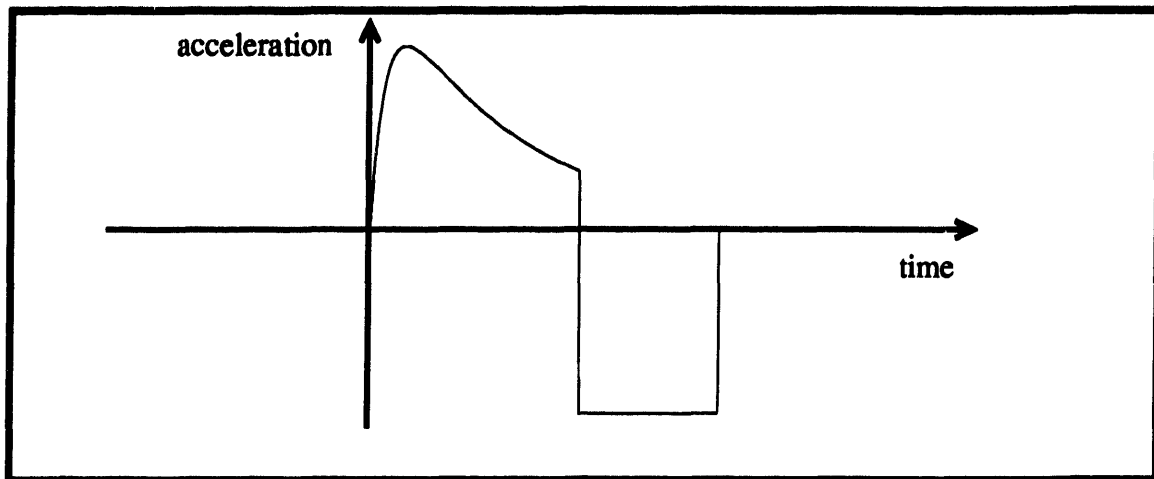


Figure 3-6 'Ideal' Acceleration

The 'ideal' waveform for minimizing seek time has sharp edges. Consequently, the 'ideal' waveform contains high-frequency components. This application demands that the waveform have most of its frequency content concentrated below 1 KHz. It is possible to reduce the high-frequency content of the 'ideal' waveform shown in Figure 3-6 by convolving it with a smoothing pulse as shown in Figure 3-7. Convolution with the smoothing pulse eliminates the sharp edges from the 'ideal' waveform; however it also increases the move time by the duration of the smoothing pulse.

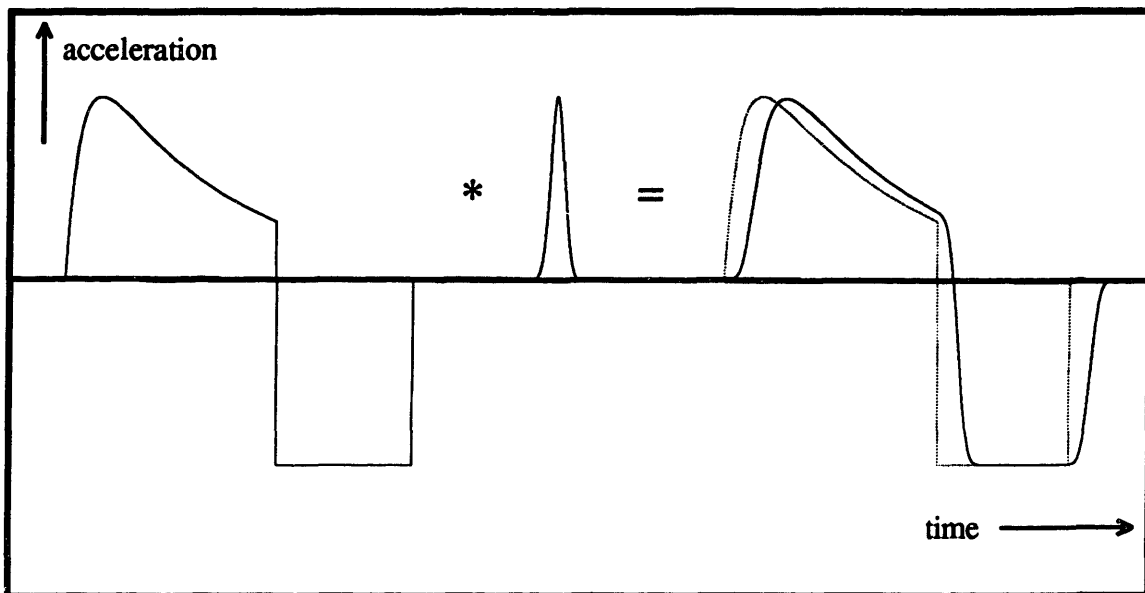


Figure 3-7 Smoothing the 'Ideal' Acceleration

Additional to requiring that the frequency content of the acceleration waveform concentrates below 1 KHz, this application also demands that the control effort never saturates. This is necessary to ensure that the trajectory following-controller can at all times precisely follow the reference profiles. The actuator's acceleration is only guaranteed to be smooth, and therefore quiet, if the actuator is able to follow the profiles. The 'ideal' waveform for minimizing seek time represents the maximum acceleration capabilities of the RA73. It corresponds to a voltage-limited acceleration, resulting from the saturation of the power amplifier's transistors, and a current-limited deceleration, resulting from the limited range of the power amplifier control signal. Because the 'ideal' acceleration waveform corresponds to saturated control effort, it is an unacceptable waveform upon which to base the reference profiles. However, a slight reduction of the amplitude of the 'ideal' acceleration yields an acceleration which does not require the control effort to saturate. Consequently, a reduced-amplitude smoothed 'ideal' acceleration is an acceptable acceleration to choose as a basis for the long-seek algorithm.

The long-seek algorithm is founded upon an 'ideal' acceleration waveform which has been both smoothed and scaled. A following section explicitly addresses the smoothing process. The scaling process reduces the acceleration level demanded by the long-seek algorithm to approximately 90% of the 'ideal', or maximum, level that would be achieved by saturating the control effort.

3.3.2.1 Constructing Profiles from Segments

Approximately 2,500 of the RA73's 2,680 possible seek lengths can be considered as 'long-length'. Each seek length requires a unique acceleration waveform. Consequently, the long-seek algorithm must create 2,500 unique acceleration commands to implement a trajectory following control approach. Fortunately, all of the acceleration waveforms have segments in common, as can be seen in the 'ideal' accelerations of Figure 3-8 and the smoothed accelerations of Figure 3-9.

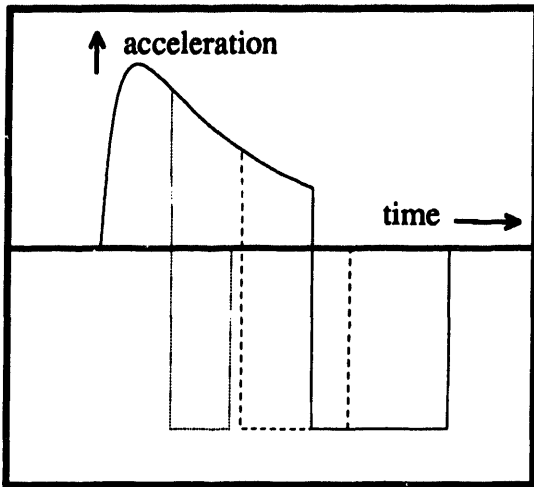


Figure 3-8 'Ideal' Accelerations

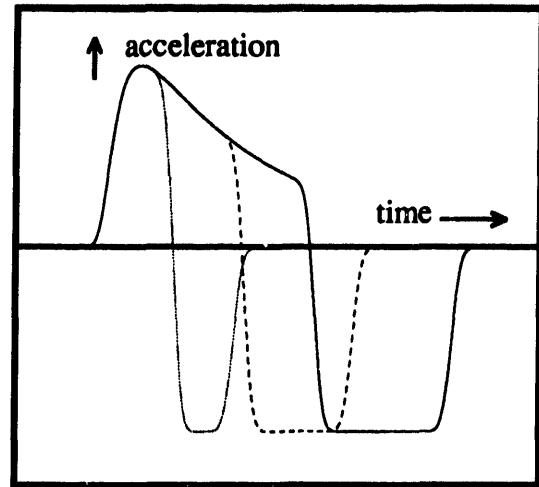


Figure 3-9 Smoothed Accelerations

Specifically, both the initial acceleration phase and the final deceleration phase are independent of seek length. Also, the shape of the small segment connecting the acceleration and deceleration phases is similar between seek lengths. Different length seeks only alter the acceleration-reversal segment's scaling and offset. The fundamental difference between the waveforms is the amount of time the actuator spends accelerating and decelerating. Figure 3-10 shows an acceleration waveform as a concatenation of three segments.

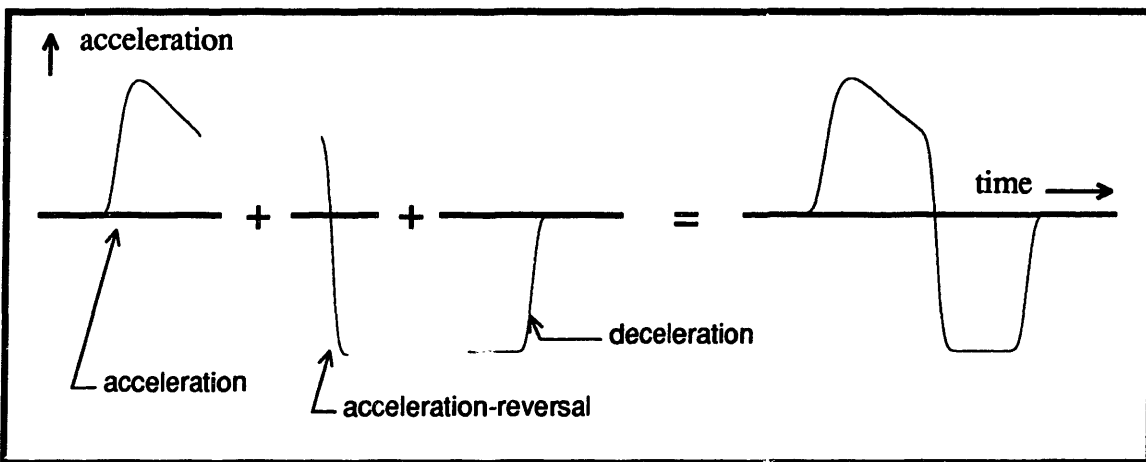


Figure 3-10 Three Acceleration Segments

Consider the acceleration waveform shown in Figure 3-10 to be associated with a mid-length seek. The first segment shown represents the entire acceleration period of

this particular seek; however, it also corresponds to a portion of the acceleration period of a full-stroke seek. Similarly, the third segment shown corresponds to a portion of the deceleration period of a full-stroke seek. The controller developed for the long-seek algorithm determines the length of both the acceleration and deceleration periods as functions of seek length. This allows the controller to construct command profiles by joining previously-computed profile segments. The previously-computed segments correspond to portions of the full-stroke acceleration and deceleration periods. These segments are sampled and stored in the controller's DSP memory.

The second segment shown in Figure 3-10 connects the acceleration period to the deceleration period; it reverses the acceleration. The optimal acceleration-reversal segment matches the levels and the derivatives of the adjoining segments at the junctures. The deceleration segment is a constant at its juncture, consequently both its amplitude and first derivative can be matched by the cosine function evaluated in the neighborhood of π . However, the acceleration segment has an exponential character, therefore its derivatives are variable. The cosine function, evaluated near zero, has derivatives which do not match the acceleration waveforms' derivatives at the juncture. Irrespective of this fact, the long-seek algorithm uses the cosine function evaluated between zero and π to model the acceleration-reversal segment. The author chose the cosine function because it is a smooth function which can be easily represented and analyzed. A cosine segment with its end points extended is shown super-imposed upon a smooth acceleration waveform in Figure 3-11.

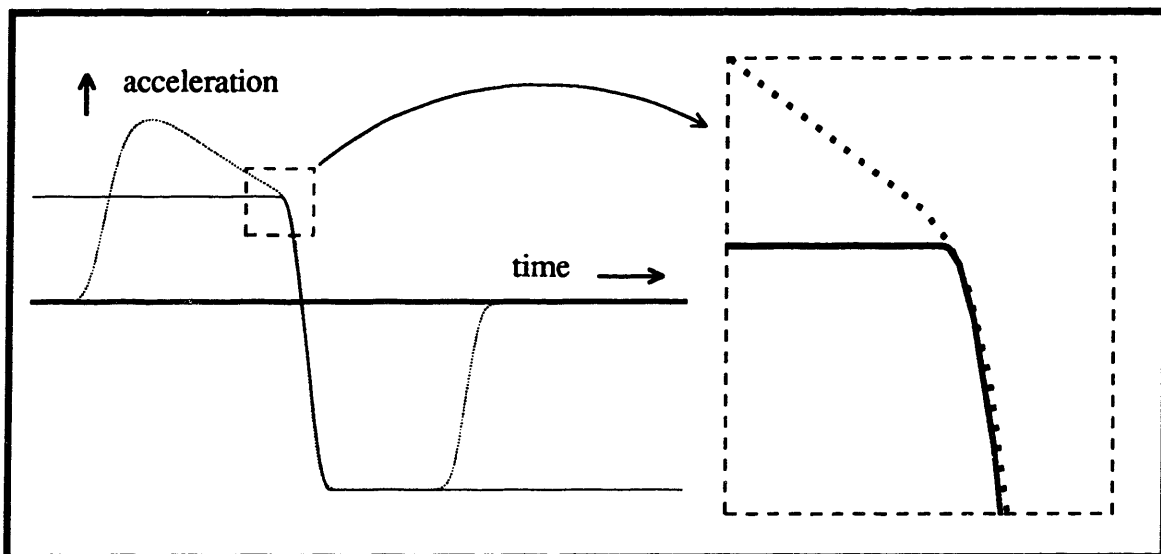


Figure 3-11 Cosine Acceleration-Reversal Segment

The long-seek algorithm constructs command and reference profiles by concatenating three segments corresponding to an acceleration period, a cosine-segment acceleration-reversal period, and a deceleration period. The deceleration period consists of two sub-sections; a variable-duration constant-level deceleration followed by a fixed deceleration. The profile segments associated with the constant-level deceleration are calculated directly by the controller; all other segments are based upon portions of previously-computed sections stored in the controller's memory.

The required seek distance determines the lengths of each profile segment. Specifically, each seek distance is matched to an index value through a look-up table. The mapping of seek lengths to index values is not one-to-one. Instead multiple seek lengths are mapped to each index value. Each index value defines all segment lengths for a particular profile set. Because seeks of multiple lengths are performed with a given profile set, the profile sets must be amplitude scaled to match the required seek lengths.

The index values correspond directly to the length of the acceleration period when the period is measured as a number of the controller's sample periods. Consequently there is a one-to-one mapping between the index values and the velocity of the actuator at the end of the discrete-time acceleration period. This fact is significant because the time course of a seek after the acceleration period is completely determined by this velocity in the implementation of this algorithm. A considerable portion of the description of this algorithm's implementation is devoted to developing the relationships between the velocity of the actuator at the end of the discrete-time acceleration period and the profile segments.

3.3.2.2 Review of Long-Seek Algorithm Concepts

The foregoing discussion explained the conceptual foundation upon which the long-seek, acoustically quiet algorithm is constructed. The discussion is summarized as follows:

- Describe the 'ideal' acceleration and deceleration waveforms. These waveforms correspond to voltage-limited acceleration and current-limited deceleration, respectively.
- Scale the 'ideal' waveforms to avoid saturating the control effort.

- Smooth the acceleration and deceleration waveforms by convolution with a smoothing pulse.
- Sample representations of the smoothed acceleration and deceleration waveforms for storage in the controller's memory.
- Model the acceleration-reversal segment as a cosine segment. Sample a representation of this segment for storage in the controller's memory.
- Develop a method of determining the lengths of the acceleration and deceleration sub-segments from the requested seek length. Also relate the scaling and offset of the cosine segment to the seek length.
- Relate the acceleration, velocity, and position commands each to the other.

The remainder of this chapter addresses these steps in a description of the long-seek algorithm's implementation. First, the chapter develops models of the actuator's accelerations resulting from saturated control efforts. These accelerations correspond to segments of the 'ideal' accelerations which minimize move times within the constraints of the RA73. Upon scaling, smoothing, and sampling, these acceleration waveform segments become the basis of the accelerations employed by the long-seek algorithm. After defining the voltage-limited acceleration and the current-limited (constant-level) decelerations, the chapter introduces the truncated Gaussian pulse that is used to smooth the 'ideal' waveforms. Following the pulse description, the chapter addresses the scaling and offset of the cosine acceleration-reversal segment. Specifically, the segment's scale and offset are related to the actuator's velocity at the end of the acceleration period. Next the deceleration segments are also related to the actuator's velocity at the end of the acceleration period. Once each segment has been defined it is possible to relate the total move time to the duration of the acceleration period and total move distance to a combination of the position and the velocity at the end of the acceleration period. Finally, the chapter culminates with a description of the discrete-time implementation of the algorithm.

3.3.3 Voltage-Limited Acceleration and Constant-Level Deceleration Models

The design process begins by modeling the maximum acceleration and deceleration capability of the disk drive's actuator. This section develops models of the actuator's

behavior under conditions of maximum control efforts to establish the limits of the algorithm's allowable accelerations. The maximum levels arrived at by these models must be scaled to ensure that the control effort required by the long-seek algorithm never saturates. Specifically, this algorithm exploits accelerations corresponding to 90% of the voltage-limited and current-limited acceleration levels.

Models which reflect only the most significant aspects of the actuator's behavior are used. These models neglect all dynamics not associated with either coil inductance or actuator inertia. Specifically, the models neglect the high-frequency dynamics of the actuator's arm, flexure, and coil structures. During the acceleration period, the model must include the effects of back EMF and coil inductance. It is sufficient to model the deceleration period as constant deceleration provided conditions defined later in this section are satisfied. Modeling the acceleration period requires a description of the coil and a description of the actuator system.

3.3.3.1 The Coil Model

The coil is described as a two-port device consisting of an inductance, L , a voltage source, EMF , and a resistance, R , connected in series. The EMF voltage is proportional to the actuator's angular velocity by the back EMF constant, K_e .

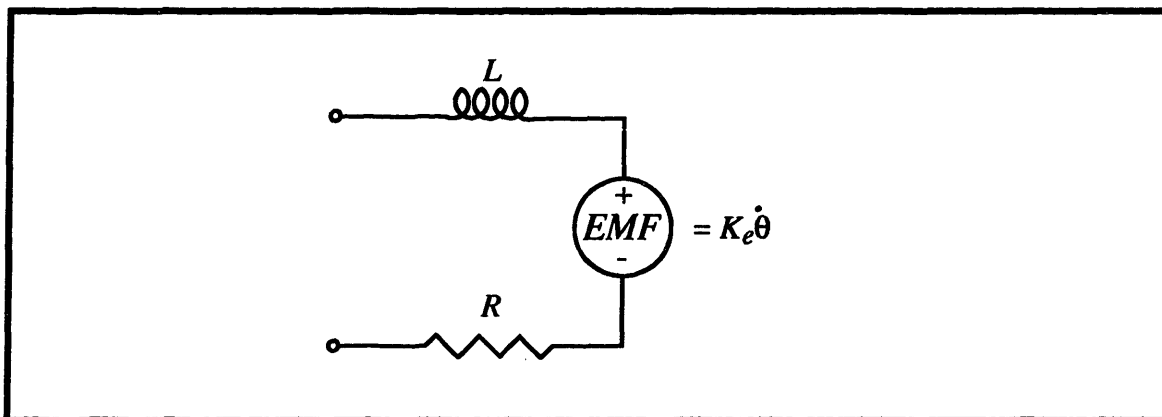


Figure 3-12 Actuator Coil Model

3.3.3.2 The Actuator Modeled with a Step of Applied Voltage

The actuator is characterized by a torque constant, K_t , an inertia, J , and an arm length, l , as shown in Figure 3-13. The actuator model is based upon the assumption that the acceleration at the servo's sensor head is proportional to the actuator's coil current.

Figure 3-13 models the actuator with a step in voltage applied to the coil. The voltage step corresponds to the power amplifier saturating at the onset of a seek. The limited voltage supplied by the power amplifier, less the back EMF voltage, causes current to flow through the coil. The coil current, scaled by the torque constant, K_t , produces a torque on the actuator. The torque accelerates the actuator's inertia, J . The arm of length l converts the actuator's rotary motion to the servo head's linear motion. An integration relates the angular acceleration to the angular velocity. The angular velocity causes the coil's back EMF voltage.

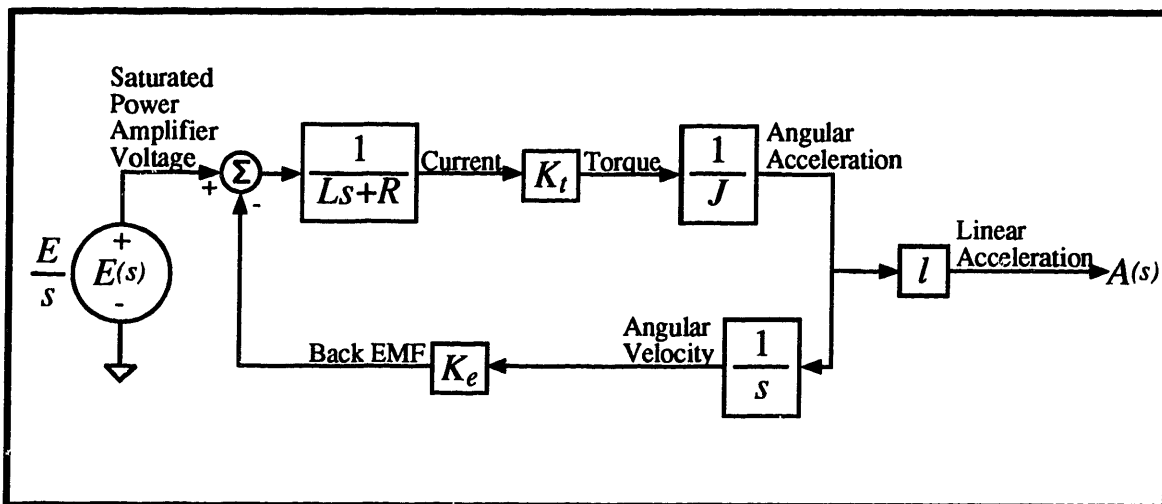


Figure 3-13 Actuator and Saturated Power Amplifier Model

3.3.3.3 The Voltage-Limited Acceleration Model

During the acceleration period of a time-optimal seek, the power amplifier saturates to its maximum output voltage. The power amplifier effectively applies a voltage step to the coil terminals. This is reflected by defining $E(s)$ as a step of magnitude 'E' in Figure 3-13. Assuming the actuator starts from rest, the acceleration, $A(t)$, of the servo's sensor head can be found as the step response of the system diagrammed in Figure 3-13. Laplace methods and a partial fraction expansion lead to a time-domain expression for the acceleration during the first phase of the seek. The derivation of this time-domain expression is shown in Appendix C, the results are shown in Equation 3.23. Appendix C also further simplifies the expression shown in Equation 3.23. This simplification is not utilized within the development of the long-seek algorithm; however, it may contribute to one's understanding of the behavior of the voltage-limited acceleration model.

(3.23) Voltage-Limited Acceleration Model

$$A(t) = \frac{1}{\tau_m \sqrt{1-4\frac{\tau_e}{\tau_m}}} \left(e^{-\frac{t}{2\tau_e}} \left(1 - \sqrt{1-4\frac{\tau_e}{\tau_m}} \right) - e^{-\frac{t}{2\tau_e}} \left(1 + \sqrt{1-4\frac{\tau_e}{\tau_m}} \right) \right) \frac{El}{K}; \quad t \geq 0$$

Equation 3.24 evaluates the voltage-limited acceleration model, Equation 3.23, with the values corresponding to the RA73 disk drive shown in Table 3-2.

Parameter	Value	Units
J	94.34×10^{-6}	(radian/seconds ²)/(newton-meter)
K	0.22	(newton-meters)/amp
R	3.5	ohms
L	1.6	millihenries
E	10.5	volts
l	0.0808	meters
$\tau_m = JR/K^2$	6.8	milliseconds
$\tau_e = L/R$	0.46	milliseconds

Table 3-2 RA73 Parameter Values

(3.24) RA73 Voltage-Limited Acceleration Model

$$A(t) = \left(e^{-\frac{t}{6.3 \text{ milliseconds}}} - e^{-\frac{t}{0.49 \text{ milliseconds}}} \right) \times 660 \frac{\text{meters}}{\text{second}^2}; \quad t \geq 0$$

The modeled acceleration¹ is compared to the measured full-stroke current in Figure 3-14.

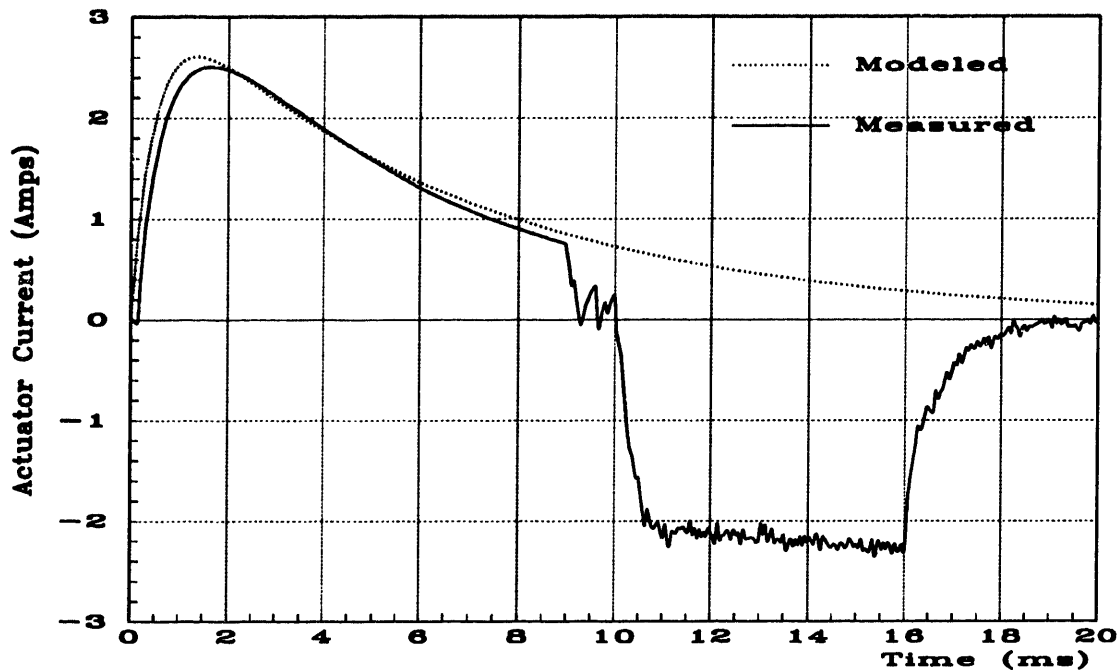


Figure 3-14 Acceleration Current, Modeled versus Measured

The matching between the modeled and the measured coil current is not exact. The actuator's inertia and arm length are defined with precision; however, the coil's resistance and inductance, the actuator's torque 'constant', and the power-supply voltage are each approximate. The coil resistance and the torque constant in particular are subject to change as the drive operates. Specifically, power dissipation in the coil during heavy-seek activity causes the coil's temperature to rise which in turn causes its resistance to rise. Further, the torque constant varies as functions of both the actuator's position and the current in the coil.

3.3.3.4 The Constant-Level Deceleration Model

The RA73's current-limited deceleration can be modeled as a constant deceleration provided the smoothed deceleration is achievable without saturating the power amplifier. The power amplifier will saturate at the end of a move if the smoothing pulse

¹ The modeled acceleration is shown scaled in amperes. Multiplication of the acceleration defined in Equation 3.24 by $J/(KI)$ converts its units to amperes.

is too narrow. This is because a narrow smoothing pulse results in a rapid transition from full deceleration to zero deceleration at the end of a move. The corresponding coil-current transition will be faster than permitted by the limited power-supply voltage. Practically, the coil's inductive time constant must be short relative to the smoothing pulse duration. In this design, the inductive time constant is 0.46 milliseconds while the smoothing pulse is 1.76 milliseconds long.

One may calculate the constant deceleration level by multiplying the maximum current available from the power amplifier by the system parameters, K/IJ , or roughly 220 meters per second-squared per ampere. The RA73 power amplifier is limited to 2½ amperes; consequently, the constant deceleration level is limited to approximately 550 meters per second-squared. The long-seek algorithm employs a constant deceleration level which is 90% of this maximum value, or 500 meters per second-squared, to ensure that the control effort never saturates.

3.3.4 The Long-Seek Algorithm Smoothing Pulse

The reduction of acoustic noise emission achieved by the long-seek algorithm results from the convolution of 'ideal' acceleration waveforms with a smoothing pulse. The short-seek algorithm is based upon a pair of rectangular acceleration pulses smoothed by a half-cycle sine pulse to form the SAS acceleration waveform. The long-seek algorithm uses the truncated Gaussian pulse as the smoothing pulse. The truncated Gaussian pulse was chosen over the other candidate pulses because it offers approximately 60 dB of attenuation beyond the first lobe. Additionally, the author chooses the truncated Gaussian pulse, rather than the half-cycle sine pulse employed by the short-seek algorithm, to gain engineering experience related to multiple pulse shapes. Figure 2-1 (Comparison of Smoothing Pulses and Their Spectrums) shows that the frequency content of a 1-millisecond wide truncated Gaussian pulse concentrates below 3½ KHz. Making the pulse 3½ times wider concentrates the pulse's frequency content below 1 KHz. Although this would lead to an acoustically very quiet system, adding 3½ milliseconds of move time significantly penalizes disk drive performance. A 1¾-millisecond pulse concentrates the pulse's frequency content below 2 KHz and continues to provide nearly 15 dB of attenuation at 1 KHz. This provides an acceptable compromise between noise reduction and move-time extension. Equation 3.25 presents

a description of the smoothing pulse employed by the long-seek algorithm. The pulse and its spectrum's magnitude are shown in Figure 3-15.

$$\text{Smoothing Pulse } (t) = \begin{cases} \frac{e^{-\pi \left(\frac{t}{0.727 \text{ ms}} \right)^2}}{0.725 \times 10^{-3}} & \text{for } -0.8805 \text{ ms} \leq t \leq 0.8805 \text{ ms} \\ 0 & \text{otherwise} \end{cases} \quad (3.25) \text{ Smoothing Pulse Definition}$$

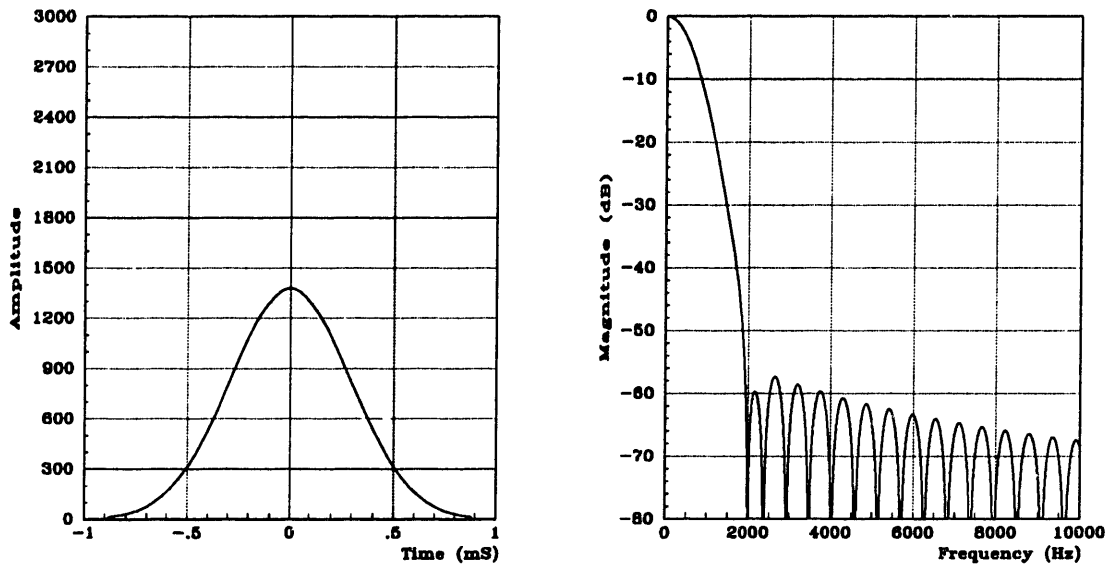


Figure 3-15 Smoothing Pulse for the Long-Seek Algorithm

3.3.5 Description of the Cosine Acceleration-Reversal Segment

The long-seek algorithm models the acceleration-reversal segment connecting the voltage-limited acceleration period to the constant-level deceleration period as a segment of a cosine waveform. Specifically, it is modeled by a scaled and offset half-cycle of a cosine waveform. Additionally, the duration of the segment is fixed at 1½ milliseconds for all seek lengths. This duration is similar to the 1¾ millisecond duration of the Gaussian smoothing pulse. Choosing the duration of the segment similar to the duration of the smoothing pulse causes the current transition to appear as though it were a step smoothed by the pulse.

One can consider the scaled and offset-cosine segment as the sum of a scaled cosine segment and a constant, $\alpha \cos \omega t + \beta$, over a time, T . Equations 3.26 through 3.32 describe the profiles corresponding to this definition during and across the acceleration-reversal segment. Note, the position and velocity profiles are dependent upon the initial conditions, X_c and V_c , at the beginning of the segment. Because the segment has a fixed duration, the changes in profile quantities across the segment are dependent only upon the scale and offset factors, α and β , and the initial velocity, V_c .

$$A(t) = \alpha \cos \omega t + \beta \quad (3.26) \text{ Cosine Segment, Acceleration}$$

$$V(t) = \frac{\alpha}{\omega} \sin \omega t + \beta t + V_c \quad (3.27) \text{ Cosine Segment, Velocity}$$

$$X(t) = \frac{\alpha}{\omega^2} (1 - \cos \omega t) + \frac{\beta}{2} t^2 + V_c t + X_c \quad (3.28) \text{ Cosine Segment, Position}$$

$$\Delta A = -2\alpha \quad (3.29) \text{ Cosine Segment, } \Delta A$$

$$\bar{A} = \beta \quad (3.30) \text{ Cosine Segment, Average } A$$

$$\begin{aligned} \Delta V &= \beta T \\ &= 1.50\beta \times 10^{-3} \text{second} \end{aligned} \quad (3.31) \text{ Cosine Segment, } \Delta V$$

$$\begin{aligned} \Delta X &= \frac{2\alpha}{\omega^2} + \frac{\beta}{2} T^2 + V_c T \\ &= (0.43\alpha + 1.06\beta) \times 10^{-6} \text{second}^2 + 1.50 V_c \times 10^{-3} \text{second} \end{aligned} \quad (3.32) \text{ Cosine Segment, } \Delta X$$

$$\text{where } T = 1.50 \times 10^{-3} \text{second} \quad \text{and} \quad \omega = \frac{\pi}{T}$$

3.3.5.1 Scaling and Offsetting the Cosine Acceleration-Reversal Segment

The cosine segment's scale factor, α , and offset, β , are such that the segment mates the the smoothed acceleration segment to the smoothed deceleration segment without a discontinuity. The scale factor, α , is equal to half the difference between the acceleration and deceleration segments at the joining points. The offset, β , is equal to the average of the acceleration and deceleration segments at the joining points (refer to Figure 3-10, Three Acceleration Segments). Determining the level of acceleration at the point where the cosine segment joins the smoothed acceleration segment, is the most

challenging piece of this section, since the smoothed deceleration segment is defined to have a constant value at the point where it joins the cosine segment.

Two exponentials describe the voltage-limited acceleration in Equation 3.24 (RA73 Voltage-Limited Acceleration Model). The exponential associated with the coil's inductive time constant ($\tau=0.49$ ms) decays quickly and becomes negligible by the time the acceleration waveform joins the cosine segment. Following the early milliseconds of a move, the actuator's acceleration and velocity can be approximated by the following equations:

(3.33) $A(t)$ and $V(t)$ Approximations

$$\begin{aligned}
 A(t) &= \left(e^{-\frac{t}{6.3 \text{ milliseconds}}} - e^{-\frac{t}{0.49 \text{ milliseconds}}} \right) \times 660 \frac{\text{meters}}{\text{second}^2}; & t \geq 0 \\
 &\approx e^{-\frac{t}{6.3 \text{ milliseconds}}} \times 660 \frac{\text{meters}}{\text{second}^2}; & t > 2 \text{ ms} \\
 V(t) &\approx 6.3 \text{ milliseconds} \times \left[660 \frac{\text{meters}}{\text{second}^2} - A(t) \right] - 0.49 \text{ milliseconds} \times 660 \frac{\text{meters}}{\text{second}^2} \\
 &\approx \left(\left(1 - e^{-\frac{t}{6.3 \text{ milliseconds}}} \right) \times 6.3 - 0.49 \right) \text{ milliseconds} \times 660 \frac{\text{meters}}{\text{second}^2} \\
 &= \left(3.8 - 4.2 \times e^{-\frac{t}{6.3 \text{ milliseconds}}} \right) \frac{\text{meters}}{\text{second}}
 \end{aligned}$$

The first line in the velocity derivation is a linear function of the acceleration, $A(t)$. Solving for the acceleration, $A(t)$, in terms of the velocity, $V(t)$, yields the following approximate relationship:

(3.34) $A(t)$ related to $V(t)$

$$\begin{aligned}
 A(t) &\approx \frac{(6.3 - 0.49) \text{ milliseconds} \times 660 \frac{\text{meters}}{\text{second}^2} - V(t)}{6.3 \text{ milliseconds}}; & t > 2 \text{ ms} \\
 &= 610 \frac{\text{meters}}{\text{second}^2} - \frac{V(t)}{0.0063 \text{ seconds}}
 \end{aligned}$$

A linear function of velocity approximates the acceleration once the transient associated with the inductive time constant diminishes. Specifically, an equation linear in V_c , the velocity at the onset of the cosine segment, describes the acceleration at the point where

the initial acceleration segment joins the cosine segment¹. This allows description of the cosine segment's scale and offset factors, α and β respectively, by equations linear in V_c . The reader should recall that the scale factor, α , is equal to one half of the acceleration's difference across the cosine segment; the offset factor, β , is equal to the average of the accelerations at the cosine segment's end-points. Equation 3.34 describes the acceleration at the onset of the cosine segment. Section 3.3.3.4 defines the constant deceleration at the segment's conclusion as approximately 500 meters per second. Equations 3.35 and 3.36 express the scale and offset factors, α and β , as linear functions of V_c , the velocity at the onset of the cosine segment.

(3.35) Cosine Segment, Scale-Factor, α

$$\begin{aligned}\alpha &= \frac{1}{2} \times \left[\left(610 \frac{\text{meters}}{\text{second}^2} - \frac{V_c}{0.0063 \text{ seconds}} \right) - \left(- 500 \frac{\text{meters}}{\text{second}^2} \right) \right] \\ &= 560 \frac{\text{meters}}{\text{second}^2} - \frac{V_c}{0.0126 \text{ seconds}}\end{aligned}$$

(3.36) Cosine Segment, Offset-Factor, β

$$\begin{aligned}\beta &= \frac{1}{2} \times \left[\left(610 \frac{\text{meters}}{\text{second}^2} - \frac{V_c}{0.0063 \text{ seconds}} \right) + \left(- 500 \frac{\text{meters}}{\text{second}^2} \right) \right] \\ &= 55 \frac{\text{meters}}{\text{second}^2} - \frac{V_c}{0.0126 \text{ seconds}}\end{aligned}$$

Substitution of the expression for β shown in Equation 3.36 into the formula describing the change in the actuator's velocity across the cosine segment, ΔV , shown in Equation 3.31, yields a linear equation relating ΔV to V_c . Equation 3.37 sums the velocity at the beginning of the cosine segment with the change in velocity across the segment to determine the velocity at the segment's conclusion.

¹ Smoothing the acceleration waveform delays the junction point, causing the component associated with the inductive time constant to be even smaller, consequently improving the linear approximation.

$$\begin{aligned}
V_c + \Delta V &= V_c + \beta T & (3.37) \text{ Cosine Segment, Final Velocity} \\
&= V_c + \left(55 \frac{\text{meters}}{\text{second}^2} - \frac{V_c}{0.0126 \text{ seconds}} \right) \times 1.50 \times 10^{-3} \text{ seconds} \\
&= 0.083 \frac{\text{meters}}{\text{second}} + 0.88 \times V_c
\end{aligned}$$

Equation 3.38 linearly relates the actuator's position at the close of the cosine segment to the velocity at the inception of the segment by substituting the expressions for α and β shown in Equations 3.35 and 3.36 into Equation 3.32.

$$\begin{aligned}
X_c + \Delta X &= X_c + \frac{2\alpha}{\omega^2} + \frac{1}{2}\beta T^2 + V_c T & (3.38) \text{ Cosine Segment, Final Position} \\
&= X_c + \left(0.43\alpha + 1.06\beta \right) \times 10^{-6} \text{ second}^2 + 1.50 V_c \times 10^{-3} \text{ seconds} \\
&= X_c + 0.43 \left(560 \frac{\text{meters}}{\text{second}^2} - \frac{V_c}{0.0126 \text{ seconds}} \right) \times 10^{-6} \text{ second}^2 \\
&\quad + 1.06 \left(55 \frac{\text{meters}}{\text{second}^2} - \frac{V_c}{0.0126 \text{ seconds}} \right) \times 10^{-6} \text{ second}^2 \\
&\quad + 1.50 V_c \times 10^{-3} \text{ seconds} \\
&= X_c + 300 \times 10^{-6} \text{ meters} + 1.38 V_c \times 10^{-3} \text{ seconds}
\end{aligned}$$

Equations 3.35 and 3.36 describe the cosine acceleration-reversal segment with first-order polynomials in V_c , the actuator's initial velocity at the commencement of the segment. Further, Equation 3.37 describes the final velocity at the conclusion of the segment with a first-order expression dependent upon V_c ; Equation 3.38 describes the final position with a first-order expression in both V_c and X_c , the segment's initial conditions. The RA73's DSP easily computes these first-order expressions which permits their implementation in the controller's hardware. The following section shows that the deceleration phase can also be described by a first-order expression in V_c . Consequently, the initial conditions upon entry to the cosine segment, V_c and X_c , completely define the actuator's movement during the remainder of the seek.

3.3.6 Determining the Duration of the Deceleration Phase

The previous section showed that both the actuator velocity and position at the end of the cosine acceleration-reversal segment are described by first-order expressions linear in the initial conditions at the onset of the cosine segment, V_c and X_c . The velocity, V_c , also determines the duration of the deceleration phase. This is most easily explained if one considers the smoothed deceleration segment as two sub-segments. The first segment has a constant deceleration level of approximately 500 meters per second-squared; however its duration is variable. The second segment is the smooth transition region where the deceleration decreases from its maximum magnitude to zero. The second segment has a constant duration equal to the width of the smoothing pulse, 1¾ milliseconds. Both the duration and amplitude of the second segment are fixed. Because the second segment is fixed, it contributes fixed amounts to the actuator's changes in velocity and position. Specifically, the actuator decelerates from 0.44 meters per second¹ to rest during the fixed segment. Consequently, during the first deceleration sub-segment the actuator must decelerate from its velocity at the conclusion of the cosine acceleration-reversal segment to 0.44 meters per second. The difference between the velocities at the sub-segment's end-points divided by the constant deceleration level yields the sub-segment's duration. The velocity at the end of the cosine segment is given by Equation 3.37. These facts are reflected in Equation 3.39 which describes the duration of the first, or constant-deceleration segment. The total time spent in the deceleration phase, T_d , is longer than the constant-deceleration time by the duration of the fixed segment, 1¾ milliseconds.

(3.39) Constant-Deceleration Time

$$T_d = \frac{\left(0.083 \frac{\text{meters}}{\text{second}} + 0.88 \times V_c\right) - 0.44 \frac{\text{meters}}{\text{second}}}{500 \text{ meters/second}^2}$$

$$= \left(-0.71 + 1.8 \times \frac{V_c}{\text{meters/second}}\right) \times 10^{-3} \text{ seconds}$$

3.3.7 Determining Total Move Time and Distance

The total move time is composed of four time slices, two of which have defined durations. The cosine acceleration-reversal segment lasts 1½ milliseconds and the fixed

¹ ½ × 1.76 ms × 500 meters/second² = 0.44 meters/second²

portion of the deceleration phase lasts $1\frac{3}{4}$ milliseconds. Equation 3.39 defines a third time section, the duration of the constant deceleration, as a linear function of V_c . The only time slice that is not either known or easily calculated is the duration of the acceleration phase, i.e. the time required to reach V_c . Unfortunately, the trajectories in this first phase of the seek are described by exponentials in time, and solving for their durations requires computing logarithms. The servo's DSP cannot compute logarithms quickly. Consequently the DSP uses a look-up table to determine the duration of the acceleration phase.

The total move distance is also composed of four pieces. Starting at the tail end of the move and then working backwards is the easiest way to proceed. The fixed-deceleration sub-segment contributes a fixed position component amounting to $21\frac{1}{4}$ tracks, or 219 micro-meters, in this implementation. The constant-level deceleration sub-segment contributes a distance described by: $\Delta x = v_0 \Delta t + \frac{1}{2}a(\Delta t)^2$. Specifically, v_0 is 0.44 meters per second, Δt corresponds to T_d as defined in Equation 3.39 as a function of V_c , and 'a' is approximately 500 meters per second-squared. Equation 3.38 shows that the cosine acceleration-reversal segment adds to the position by an amount equal to the sum of 300 micro-meters and the product of V_c multiplied by 1.38 milliseconds. Once again, exponentials in time describe the trajectories in the first segment of the move. As a result, the DSP must use a look-up table to relate the total move distance to both the acceleration phase duration and to the key parameter in the algorithm, V_c .

3.3.8 The Discrete Time Implementation

The DSP uses first-order polynomials to calculate the scale and offset factors for the cosine acceleration-reversal segment and to calculate the duration of the constant-level deceleration sub-segment. V_c is the independent variable in each calculation. However, V_c is related to the total move distance by a look-up table, rather than by a calculation, in this implementation.

The velocity at the conclusion of the initial acceleration phase, V_c , determines the durations and scalings of all acceleration segments after the initial acceleration phase. Consequently, it is possible to compute trajectories describing the actuator's desired motion following the initial acceleration phase based upon V_c . However, a representation of the trajectories associated with the initial acceleration phase must be

sampled and stored in table form inside the DSP's memory. If the DSP had sufficient memory it would be convenient to store individual position, velocity, and acceleration trajectories for the first phase of the move. Since the DSP does not have sufficient memory, only one of the trajectories, the velocity trajectory, is stored while the other two are synthesized via integration and differentiation. Two reasons demand that the velocity trajectory be stored -- it makes V_c directly available in a table and it allows practical synthesis of the other two trajectories.

Synthesizing one trajectory from another requires either integration or differentiation. When implemented with a microprocessor, this amounts to performing either repetitive additions or subtractions, respectively. The results of these operations are limited in both range and precision by the hardware architecture of the microprocessor. The RA73's servo microprocessor, a Texas Instruments TMS320C25 DSP, is a fixed-point integer arithmetic device with a 16-bit data path and a 32-bit accumulator. This DSP represents its data as 16-bit numbers. The DSP represents data as numbers between 0 and $2^{16}-1$, or 65,535. This gives it a resolution of one part in 65,536. The resolution of a stored trajectory impacts the DSP's ability to practically synthesize corresponding trajectories. Examples given in the following passage illustrate this point.

The DSP operates with 16-bit integer words. Storage of a position reference which allows the representation of over 1,023 tracks during the acceleration phase requires 11 bits of track address, leaving 5 bits to represent the distance of one track. This leads to a scaling of 32 bits per track, or 0.32 micro-meters per bit. The first difference of the position reference, sampled every 47 microseconds, permits a velocity resolution of 0.0068 meters per second per bit¹. The second difference of the position reference permits an acceleration resolution of 145 meters per second-squared per bit². A resolution of only 145 meters per second-squared per bit is insufficient because the full-scale acceleration is on the order of 500 meters per second-squared, or only four bits. Conversely, storing an acceleration reference scaled at 500 meters per second-squared per 2^{16} bits leads to an acceleration resolution of 0.0076 meters per second-squared per bit³. Half of one bit of round-off error at the input to a double integrator at the beginning of a seek lasting 20 milliseconds leads to a final position

¹ $(0.32 \mu\text{m/bit}) / (47 \mu\text{s}) = 0.0068 \text{ m/s/bit}$

² $(0.0068 \text{ m/s/bit}) / (47 \mu\text{s}) = 145 \text{ m/s}^2/\text{bit}$

³ $(500 \text{ m/s}^2/\text{bit}) / (2^{16}) = 0.0076 \text{ m/s}^2/\text{bit}$

error of about 8% of a track¹. In this case the limited computational precision may cause an error to build within the position reference. The scheme selected for implementation stores the velocity reference scaled at 512 bits per inch per second, or nearly 0.000050 meters per second per bit. This permits an acceleration resolution of 1.06 meters per second-squared per bit², and leads to a final position error of roughly 5% of a track³ given a one-half-bit error at the input to a single integrator at the beginning of a seek lasting 20 milliseconds. Storing the velocity trajectory reduces the effects of round-off error only slightly when compared to storing the acceleration trajectory. These effects show that it is important to avoid potential sources of round-off error when coding the algorithm.

Several steps describe the process which creates the sampled, smoothed velocity trajectory that is stored in the DSP's memory. First, compute the voltage-limited acceleration described by Equation 3.24 (RA73 Voltage-Limited Acceleration Model) and sample it at the DSP's sample rate, or every 47 microseconds. One must also, sample the smoothing pulse described in Equation 3.25 (Smoothing Pulse Definition) at the DSP's sample rate. Next, convolve the two sampled signals to produce a sampled version of the smoothed acceleration. The duration of the signal must be long enough to describe the acceleration phase of a full-stroke seek. Finally, integrate, or more precisely, sum the sampled smoothed acceleration according the following difference equation:

(3.40) Difference Equation Relating $v[n]$ to $a[n]$

$$v[n] = v[n-1] + a[n-1] \times 47 \text{ microseconds}$$

$a[n]$ is *acceleration*

The resulting signal is a sampled version of the velocity trajectory, scaled in meters per second, which can be rescaled by a factor of 20,000 to produce a version for storage in the DSP's 16-bit memory. The DSP recovers the acceleration trajectory, delayed one sample, by taking the first difference of the stored velocity trajectory.

¹ $\frac{1}{2} (\frac{1}{2} \text{ bit} \times 0.0076 \text{ m/s}^2/\text{bit}) \times (0.02 \text{ s})^2 = 0.76 \times 10^{-6} \text{ m} \approx 8\% \text{ of a track}$

² $(0.000050 \text{ m/s/bit})/(47 \text{ } \mu\text{s}) = 1.06 \text{ m/s}^2/\text{bit}$

³ $(\frac{1}{2} \text{ bit} \times 0.000050 \text{ m/s/bit}) \times 0.02 \text{ s} = 0.5 \times 10^{-6} \text{ m} \approx 5\% \text{ of a track}$

The DSP calculates the smoothed position trajectory from the stored velocity trajectory according to the following difference equation:

(3.41) Difference Equation Relating $v[n]$ to $x[n]$

$$\begin{aligned} x[n] &= x[n-1] + \left(v[n-1] + \frac{1}{2} a[n-1] \times 47 \text{ microseconds} \right) \times 47 \text{ microseconds} \\ &= x[n-1] + \frac{1}{2} (v[n] + v[n-1]) \times 47 \text{ microseconds} \end{aligned}$$

Equations 3.40 and 3.41 define the desired actuator position, velocity, and acceleration at any sample time during the initial acceleration period given the stored velocity trajectory, $v[n]$.

The elements of the velocity trajectory, $v[n]$, are possible values of V_c . For this implementation, they are the only possible values of V_c . Much of this section has been devoted to showing that V_c determines both the duration of the seek after V_c has been reached and the distance covered during the seek after V_c has been reached. Each potential value of V_c , $v[\tilde{n}]^1$, has an associated total move distance. The total move distance is the sum of the distance covered before the \tilde{n} 'th sample, $x[\tilde{n}]$, and the distance covered after the \tilde{n} 'th sample, which has been shown to be a function of $v[\tilde{n}]$. This defines a total move distance sequence, $d[\tilde{n}]$, corresponding to seeks having an initial acceleration phase with duration equal to \tilde{n} samples. Not all values of \tilde{n} correspond to seek lengths implemented by the long-seek algorithm. In particular, for $\tilde{n} < 50$, the seek lengths become short (defined as less than 144 tracks) and are better implemented with the short-seek algorithm². The maximum seek length of the RA73 is 2,680 tracks. This roughly corresponds to $\tilde{n} = 243$. Therefore, the only values of $d[\tilde{n}]$ having significance to the long-seek algorithm are the values between $d[50]$ and $d[243]$, inclusive.

¹The character \tilde{n} is introduced to denote a final value of a sequence denoted by n , i.e., $n = 1, 2, 3, \dots \tilde{n}$.

²There are several problems associated with using the long-seek algorithm to implement short seeks. One problem is that the fixed deceleration at the end of the seek provides more than the deceleration required by the seek, which leads to a calculation of a negative duration for the constant-level deceleration sub-segment. An additional problem is that an acceleration discontinuity forms at the junction of the acceleration segment and the cosine acceleration-reversal segment. This is because the assumption that inductive effects have decayed away by the time the cosine acceleration-reversal segment is encountered is invalid for short seeks.

At this point, the reader might be concerned that the long-seek algorithm allows the execution of only 194 different seek lengths¹, most of which are not an integer number of tracks. What is needed is an algorithm capable of executing 2,537 seek lengths², each length an integer number of tracks. This problem is not as difficult as it may seem. Consider the trajectories of two seeks of identical duration, but one seek is 1,000 tracks long, and the other is 999 tracks long. The second seek will have exactly the same trajectories as the first, but scaled by a factor of 0.999. The long-seek algorithm rescales the 194 different trajectories it can synthesize, to span the 2,537 seek lengths it must implement. The DSP performs the rescaling by first normalizing the synthesized trajectories of a 'discrete' seek with length as long as, or slightly longer than, the required seek length, and then multiplying them by the required seek length. The normalization of the trajectories can be thought of as dividing the trajectories by $d[\tilde{n}]$, the 'discrete' seek length corresponding to the index, \tilde{n} . This is not completely correct; the subtleties will be revealed following the next paragraph.

Implementing this algorithm in discrete time causes a problem in the calculation of the length of the constant-level deceleration sub-segment by Equation 3.39. The equation expresses the duration as a continuous, linear function of V_c . The discrete-time implementation forces the DSP to round the duration to a nearby multiple of the sample time. This can lead to the constant-level deceleration sub-segment being either longer or shorter than desired by up to one half of one sample period. Consequently, a discontinuity forms in both the velocity and the position at the junction between the constant-level deceleration sub-segment and the fixed deceleration sub-segment. The velocity mismatch can be as large as 2.6% of the desired velocity³. The position discontinuity can be as large as one track⁴. The DSP eliminates the discontinuities by the following method. The DSP calculates the desired deceleration time at the beginning of a seek and rounds the result to the nearest integer multiple of sample times. However, the DSP saves the remainder. The fractional remainder represents 'missing' deceleration. The missing deceleration is equal to the product of the deceleration level, 500 meters per second-squared, and the fraction. The DSP spreads the 'missing' deceleration evenly over many (80) samples in advance of the junction between the

¹ $243 - 50 + 1 = 194$

² $2,680 - 144 + 1 = 2,537$

³ $500 \text{ m/s}^2 \times \frac{1}{2} 47 \mu\text{s} = 0.012 \text{ m/s} = 2.6\% \text{ of } 0.44 \text{ m/s}.$

⁴ $\frac{1}{2} 500 \text{ m/s}^2 \times (\frac{1}{2} 47 \mu\text{s})^2 + 0.44 \text{ m/s} \times (\frac{1}{2} 47 \mu\text{s}) = 10 \mu\text{m} \approx 1 \text{ track}.$

constant-level and fixed deceleration sub-segments. This eliminates the velocity mismatch at the junction. The DSP implements this method by adding a ramp to the $v[n]$ trajectory. The ramp is described by Equation 3.42.

(3.42) Velocity Ramp Correction

$$ramp[k] = k \times \frac{\text{"fraction"}}{80} \times 47 \text{microseconds} \times 500 \frac{\text{meters}}{\text{second}^2}$$

k ranges from 1 to 80 over the 80 samples immediately preceding the junction.

Although the velocity discontinuity is eliminated by this solution, the position discontinuity is generally made larger. The problem disappears if the 'position' associated with the fixed deceleration sub-segment is adjusted by the amount of the discontinuity. The solution is to change the length of the move associated with the index, \tilde{n} , i.e. adjust $d[\tilde{n}]$ by the amount of the position discontinuity.

The reciprocals of the adjusted discrete seek lengths are stored as double precision¹, or 32-bit, words in the DSP's memory. The DSP normalizes the synthesized trajectories by the reciprocals of the adjusted discrete seek lengths before rescaling them by the desired move length.

In addition to the reciprocals of the adjusted discrete seek lengths the DSP also possesses a record of the adjusted discrete seek lengths, rounded to the nearest track, stored in its memory. The DSP uses this table to determine the index, \tilde{n} , for any required seek length. When the DSP receives a seek command, it compares the required move distance with the distances in the table. It finds the element in the table that is closest to the required move distance, without being shorter than the required move distance. The index of the table element is related to the index, \tilde{n} , by a constant. Once \tilde{n} is determined, the DSP uses it to both index into the $v[n]$ table to find $v[\tilde{n}]$, or V_c , and to index into the table of adjusted discrete-seek-length reciprocals. Having found V_c , the DSP can calculate the scale and offset factors, α and β , for the cosine segment as well as the duration of the constant-level deceleration segment, T_d , and its associated 80-sample correction. The DSP has sufficient information to determine the seek trajectories up to the point of the fixed deceleration sub-segment.

¹Normalizing a 2500 track seek with a precision of 1% of a track requires calculations good to 1 part in 250,000, more precision than is possible with 16 bits.

The fixed deceleration sub-segment, the final phase of the seek, is only 39 samples long and is stored in the DSP's memory as both velocity and position trajectories. This last piece of the position trajectory is referenced to the seek's end-point. The sub-segment position trajectory is stored as a sequence of negative displacements, decreasing in magnitude. Referencing the sub-segment to the end-point allows the position reference to be created by adding the sub-segment to the end-point position. This causes discontinuities, resulting from computational errors during the earlier segments, to form in the references at the beginning of the sub-segment, not at the end of the move. Referencing the sub-segment to the end-point gives the system the duration of the sub-segment, nearly 2 milliseconds, to eliminate errors formed during the early phases of the seek. With the addition of the fixed deceleration trajectories, the DSP is able to completely describe the seek trajectories for a $d[\tilde{n}]$ length seek, which it then rescales to the required seek length.

3.3.9 Long-Seek Algorithm Summary

The long-seek algorithm effects moves of lengths between 144 and 2,680 tracks by amplitude scaling 194 sets of smooth position, velocity, and acceleration profiles. Each profile set is constructed by concatenating segments of several previously-computed profiles and one immediately-calculated segment. The previously-computed segments correspond to an acceleration period, an acceleration-reversal segment, and a fixed deceleration segment. The immediately-calculated segment is a variable-duration constant-level deceleration. The required move distance determines the lengths of each component segment. Specifically, each of the possible 2,537 move lengths is matched to 1 of 194 index values through a look-up table. Each index value defines all segment lengths for a particular profile set. Because the required seek lengths do not generally correspond to the move distances associated with the constructed profile sets, the profile sets must be amplitude scaled to match the required seek lengths.

The long-seek algorithm employs asymmetric waveforms to implement moves requiring nearly the maximum acceleration capabilities of the RA73. The accelerations used by this algorithm account for the effects of the coil's back EMF and inductance. Like the short-seek algorithm's accelerations, the long-seek algorithm's accelerations have their frequency content concentrated at low frequencies. Consequently, the smooth coil currents exploited by the algorithm do not excite the vibrational modes responsible for acoustic noise emission.

Chapter 4: Testing and Results

The author implemented the short-seek and long-seek algorithms in the RA73's DSP microcode so that they could be resident with the RA73's original seek algorithm. Further, the author developed special commands for the RA73's servo to allow it to switch between the original and the quiet algorithms for testing. This enabled comparison of the original and the quiet algorithms on the same mechanics and electronics, thereby eliminating the possibility that differences in the test data are related to differences in the set-up as opposed to the algorithms. The author tested two RA73 disk drives.

Several aspects of a disk drive's servo performance were characterized and compared between the original and quiet algorithms, including the following:

- 1) Acoustic Noise Emission
- 2) Average Coil Current Power Spectrum
- 3) Coil Current Shapes and Power Spectrums for Selected Seek Lengths
- 4) Average Seek Times
- 5) Post Seek Settling Characteristics

The following sections address these aspects in the order set forth above.

The test equipment used while acquiring data included an anechoic chamber, two signal analyzers, and an anti-aliasing filter. Specifically, an 'AN-ECK-OIC' chamber from Eckel Industries, Inc., of Cambridge Massachusetts and a Brüel and Kjær Dual Channel Signal Analyzer, Type 2032 were used for the acoustic measurements. Coil current measurements and measurements related to the post-seek settling of the actuator's position were acquired with a Hewlett-Packard 3563A Control System Analyzer and a dual channel Krohn-Hite model 3342 filter. The Krohn-Hite performed anti-aliasing filtering. The measurement 'states' of the Hewlett-Packard analyzer are documented in Table 4-1. These 'states' represent the analyzer's configuration while acquiring data for sections 4.2, 4.3, and 4.5. More detailed descriptions of the configurations accompany the discussion of each test.

State Parameter	Section		
	4.2	4.3	4.5
Resolution	Linear	Linear	Linear
Measurement, Channel 1	Power Spectrum	Power Spectrum	Power Spectrum
Measurement, Channel 2	Power Spectrum	Power Spectrum	Power Spectrum
Window, Channel 1	Hanning	Uniform	Uniform
Window, Channel 2	Hanning	Uniform	Uniform
Average Type	Stable Mean	Average Off	Average Off
Number of Averages	300	1	1
% Overlap	0	0	0
Time Averaging	Off	Off	Off
Frequency Center	5.1 Khz	12.5 Khz	5 Khz
Frequency Span	10 Khz	25 Khz	10 Khz
Frequency Resolution (BW)	18.7 Hz	31.2 Hz	12.5 Hz
Record Length	80 ms	32 ms	80 ms
Sample Time (dt)	78.1 μ s	15.6 μ s	39.1 μ s
Trigger	Freerun	Channel 1	Channel 1
Trigger Level	3.8 V	3.8 V	3.8 V
Trigger Slope	Negative	Negative	Positive
Input Range, Channel 1	6.32 Vpeak	6.32 Vpeak	6.32 Vpeak
Input Range, Channel 2	2.51 Vpeak	2.51 Vpeak	2.0 Vpeak
Scaling, Channel 1	1 V/(Eng. Unit)	1 V/(Eng. Unit)	1 V/(Eng. Unit)
Scaling, Channel 2	1 V/(Eng. Unit)	1 V/(Eng. Unit)	1 V/(Eng. Unit)
Input Coupling, Channel 1	DC	DC	DC
Input Coupling, Channel 2	DC	DC	DC
Delay, Channel 1	No Delay	-1.99 ms	-1.99 ms
Delay, Channel 2	No Delay	-1.99 ms	-1.99 ms

Table 4-1 Hewlett-Packard 3563A Measurement States

4.1 Acoustic Noise Emission

The author tested two RA73 disk drives for acoustic noise emission. A comparison of the noise emission between the disk drives operating with the original seek algorithm and the disk drives operating with quiet-seek algorithm¹ was the test's objective. In particular, the author wished to compare the increase in the sound pressure power spectrum above a baseline level as a result of seek activity. The two disk drives were tested independently. Testing included measuring the baseline sound pressure power spectrum as the disk drives operated in their track following mode and measuring the sound pressure power spectrum as the disk drives performed random seeks. The baseline measurements were acquired with the servo sensor head positioned over a track in the center of the platter's surface, track #1,340. The author tested each unit while executing random seeks at a 'normal' rate and at a faster 'worst case' rate. The 'normal' seek rate, 44 seeks per second, corresponds to a standard seek rate used for acoustic testing by companies within the disk industry. The faster seek rate, 79 seeks per second, corresponds to a 'worst case' measurement where the drive seeks continuously. The slower seek rate dilutes the measurements by taking data while the drive is in its track following mode, but the data is valuable for comparison with other results.

The acoustic noise emission section begins with a description of the measurement equipment and its configuration. A presentation of the sound pressure power spectrum measurement data follows the test set-up description. The baseline measurements are introduced first. They are followed by data acquired while the drives sought at the 'normal' rate. The 'worst case' data is presented last. The data for both the 'normal' and the 'worst case' rates is shown in a sequence which facilitates the comparison of the differences between the quiet-seek algorithm and the original seek algorithm. The plots comparing the algorithms are presented in the following order:

- Total Noise while Randomly Seeking
- Seek Noise (Total Noise less Baseline Noise) while Randomly Seeking
- Seek Noise Difference (Quiet-Seek Algorithm less Original Seek Algorithm) while Randomly Seeking

¹The phrase 'quiet-seek algorithm' implies the combination of both the short-seek algorithm and the long-seek algorithm.

The A-weighted sound pressure of a disk drive was measured in an anechoic chamber while the drive performed random seeks at a defined rate. The disk drive being tested was placed inside an 'AN-ECK-OIC' chamber from Eckel Industries, Inc., of Cambridge Massachusetts. A microphone was suspended a third of a meter directly above the top cover of the drive for the sound pressure measurement. The microphone fed a Brüel and Kjær Dual Channel Signal Analyzer, Type 2032. An anti-aliasing filter at 25.1 KHz is embedded in the Brüel and Kjær analyzer. The microphone was calibrated to produce 11.8 mv/Pascal as an input to the Brüel and Kjær's pre-amplifier. The Brüel and Kjær's display and output data was referenced to 40 dB per 20 μ Pascal.

The Brüel and Kjær was configured to take 500 spectrum averages of the autocorrelation of the sound pressure sampled 2,048 times at a 32.768 KHz rate. The autocorrelation function is implemented as the product of the 2,048-point FFT of the measurement data with its complex conjugate. This provides both a frequency resolution and an FFT bin spacing of 16 Hz. The presented data may be considered to represent either sound power, scaled as 10 times the logarithm of the autocorrelation average, or sound pressure magnitude, scaled as 20 times the logarithm of the autocorrelation average. The presentation is based upon the simplifying assumptions that the anti-aliased measurements are bandlimited to less than 16 KHz and that the applied Hanning window results in minimal spectral smearing. Actual peak heights may be greater than shown in the plots because of the smearing effects of the sampling window.

4.1.1 Baseline Noise

The baseline noise (Figures 4-1 and 4-2) is similar between the two tested drives. The spectrum contains dozens of individual peaks, but also shows concentrations of power around 2 KHz and again near 6 KHz, with the largest concentration being around 2 KHz. A 2.4 KHz tone produces the highest peak for both drives. The logarithmic data presentation masks the magnitude of the power concentration near the broad peaks. Approximately 82%¹ of the sound power lies between 1 and 3 KHz. There is an additional 4% between 3 and 4 KHz and approximately 10% between 5 and 7 KHz. The broad peaks near 2 and 6 KHz account for 95% of a drive's sound power; 85% of a drive's sound power lies between 1 and 4 KHz.

¹ This analysis assumes the measurement data, spanning 0 to 12.8 KHz, represents all of a drive's sound power. The percentages reflect ratios of sums of sound powers across a frequency band to the sum of sound powers across the total, 0 to 12.8 KHz, frequency band. The percentages are an average of the values for both drives measured.

Baseline Noise, Track Following

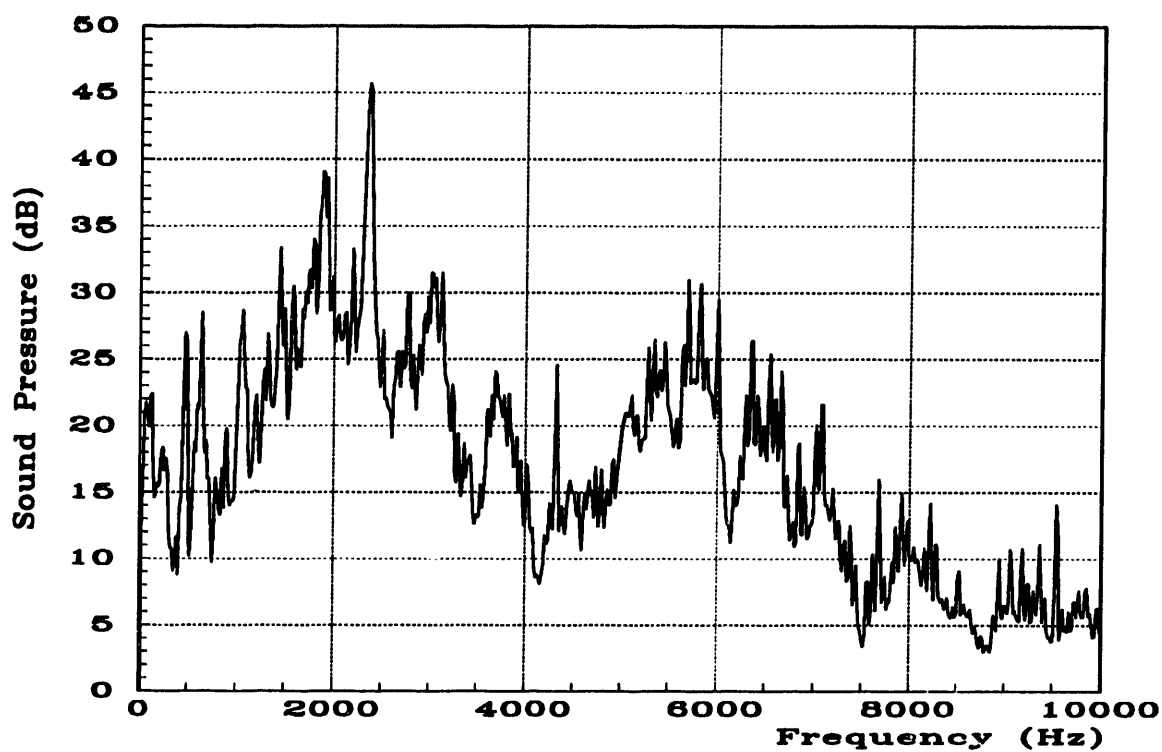


Figure 4-1 Baseline Noise, Drive #1

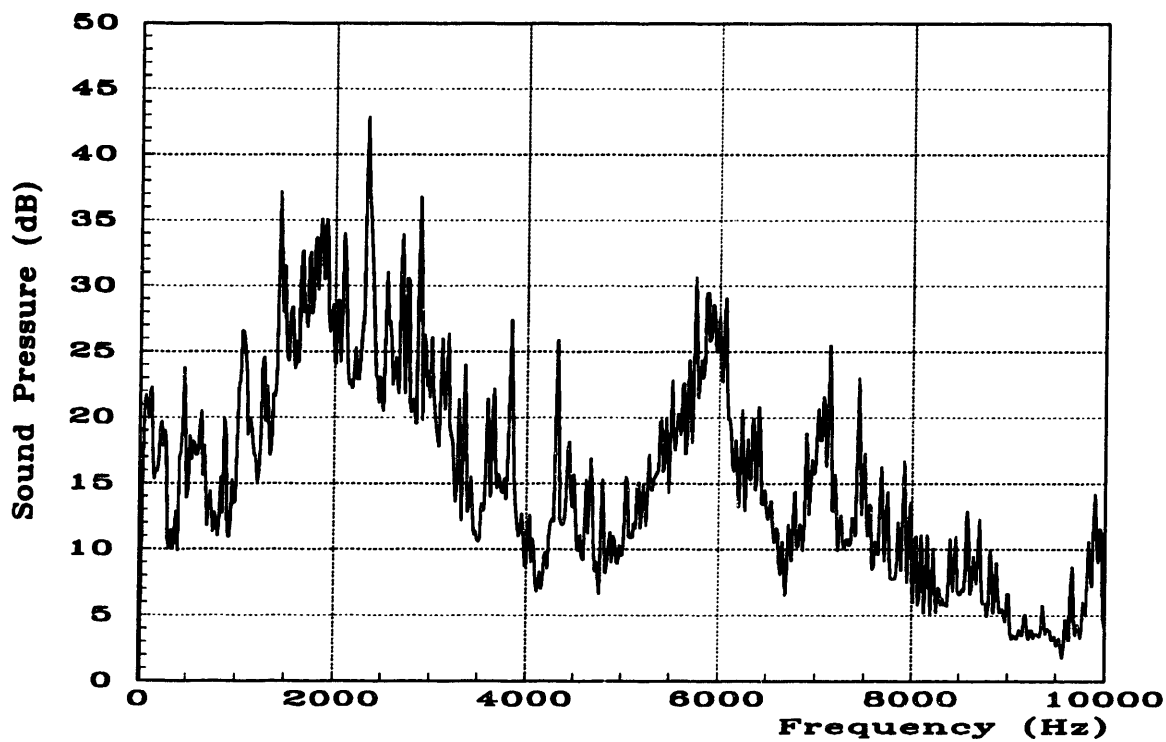


Figure 4-2 Baseline Noise, Drive #2

4.1.2 Total Noise

The total noise (Figures 4-3 and 4-4) reflects the sound power emitted by a drive as it randomly seeks. It has a character similar to the baseline noise. Again, the noise exhibits broad peaks at 2 and 6 KHz. The plots compare the sound pressure between the original seek algorithm and the quiet-seek algorithm. The quiet-seek algorithm has sound pressure that is lower than the original seek algorithm by approximately 10 dB between 1 and 3 KHz. However, below 1 and above 4 KHz the sound pressure traces show little difference between the two algorithms. Despite the quiet-seek algorithm's lack of effectiveness above 4 KHz, its reduction in sound-power output is significant. This is because the algorithm is most effective over the band of frequencies where the sound power concentrates. The frequency band between 1 and 3 KHz accounts for 94% of the sound-power output for the original seek algorithm. The remaining 6% of the sound power is evenly split between the 3-to-4 KHz band and all other frequencies. The quiet-seek algorithm's sound power is more distributed. It has only 80% of its sound power between 1 and 3 KHz and about 6% of its sound power between 3 and 4 KHz. The quiet-seek algorithm's sound-power distribution is also broader than the sound-power distribution when the drive is in its track-following mode.

Total Noise - 'Normal' Random Seek Rate

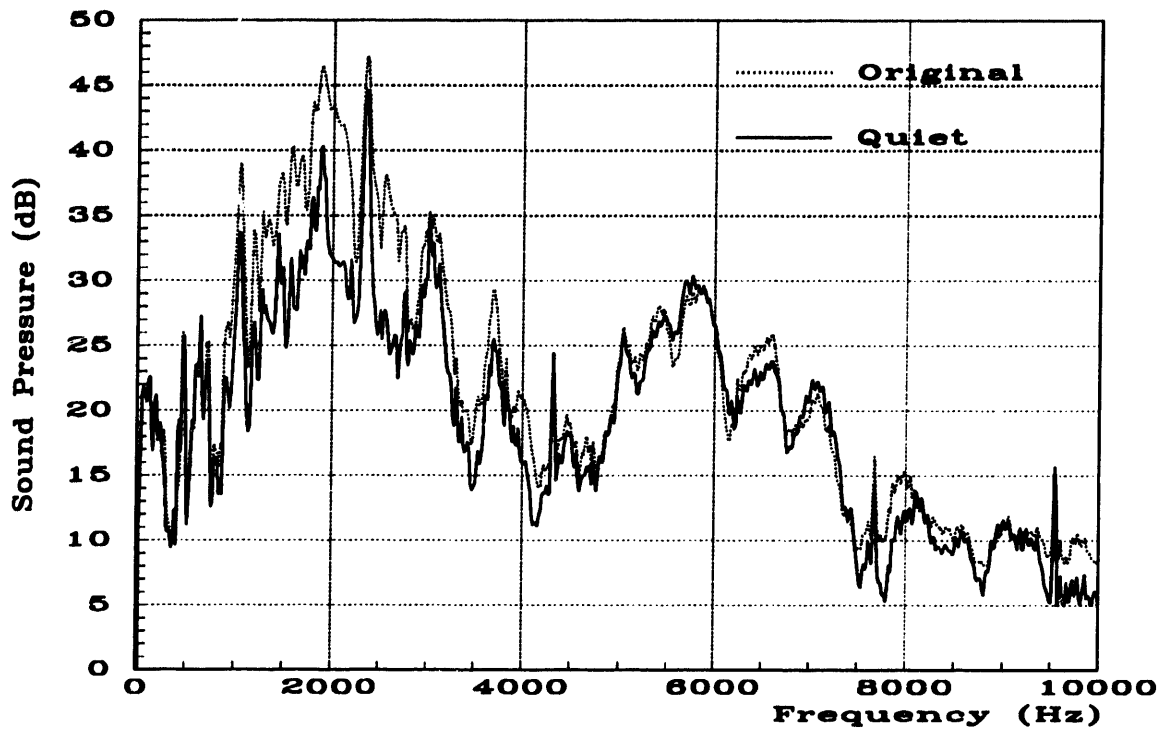


Figure 4-3 Total Noise, 'Normal' Random Seek Rate, Drive #1

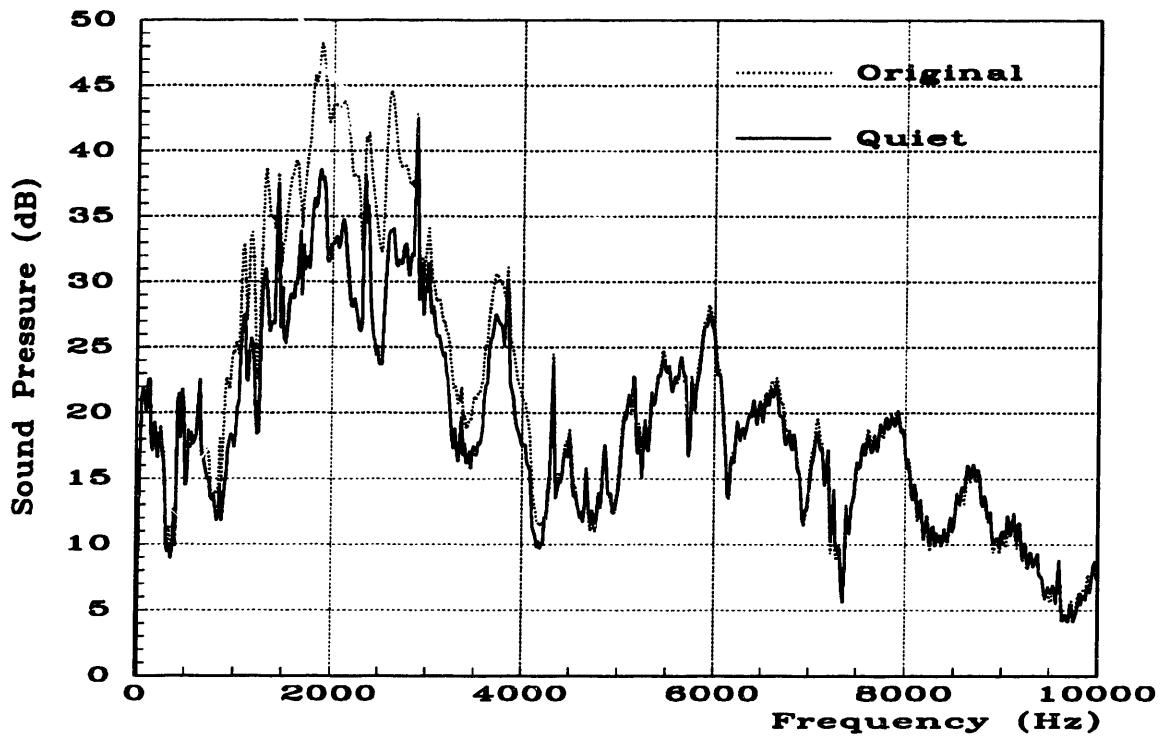


Figure 4-4 Total Noise, 'Normal' Random Seek Rate, Drive #2

4.1.3 Seek Noise

The seek noise plots (Figures 4-5 and 4-6) show the ratio of the total sound power while seeking to the baseline sound power while track following, i.e. they show the increase in sound power due to seeking. The presented data is calculated by subtracting the baseline sound-power data, expressed in decibels, from the total-sound-power data, also expressed in decibels. These plots indicate that the quiet-seek algorithm is most effective between 1 and 4 KHz. This frequency band contains 94% of the drive's sound-power output when the original seek algorithm is active. In the region between 1 and 3 KHz the increase in sound-power output is 710% for the original seek algorithm and 44% for the quiet-seek algorithm. The over-all increase in sound-power output across the measurement band is 600% for the original seek algorithm, but only 47% for the quiet-seek algorithm. Therefore the quiet-seek algorithm reduces the increase in emitted sound power due to seek activity by a factor of twelve.

Seek Noise, 'Normal' Random Seek Rate

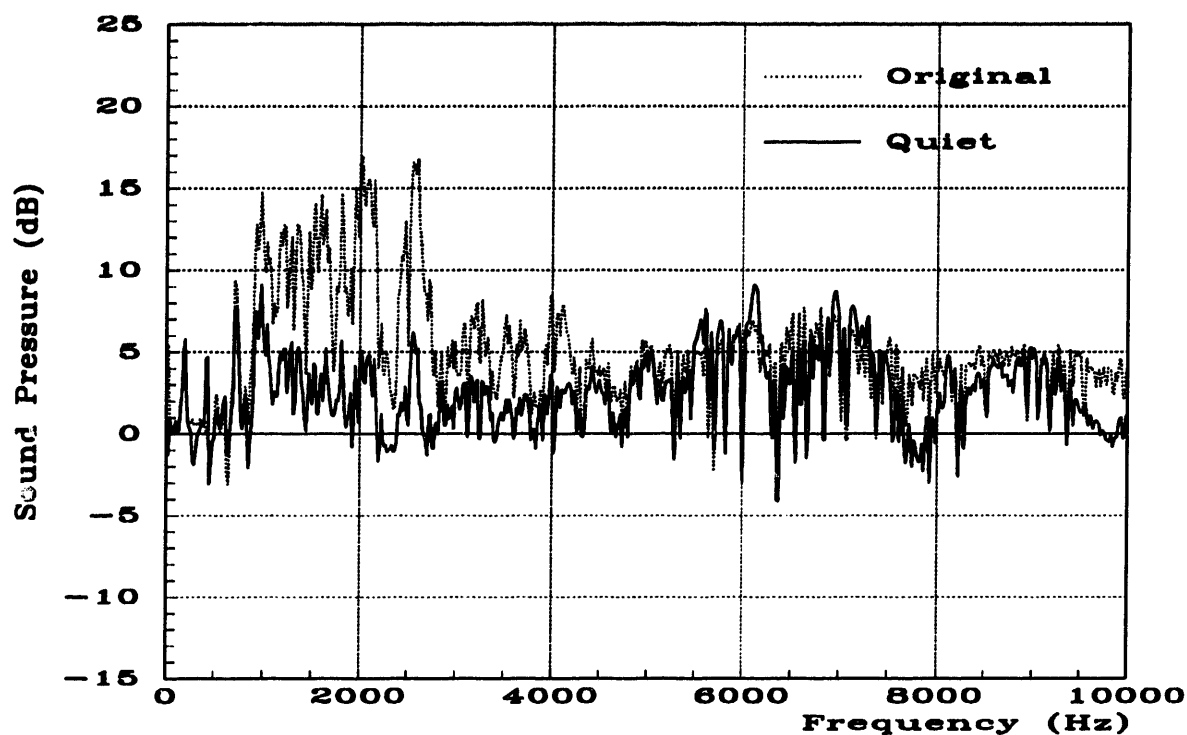


Figure 4-5 Seek Noise, 'Normal' Random Seek Rate, Drive #1

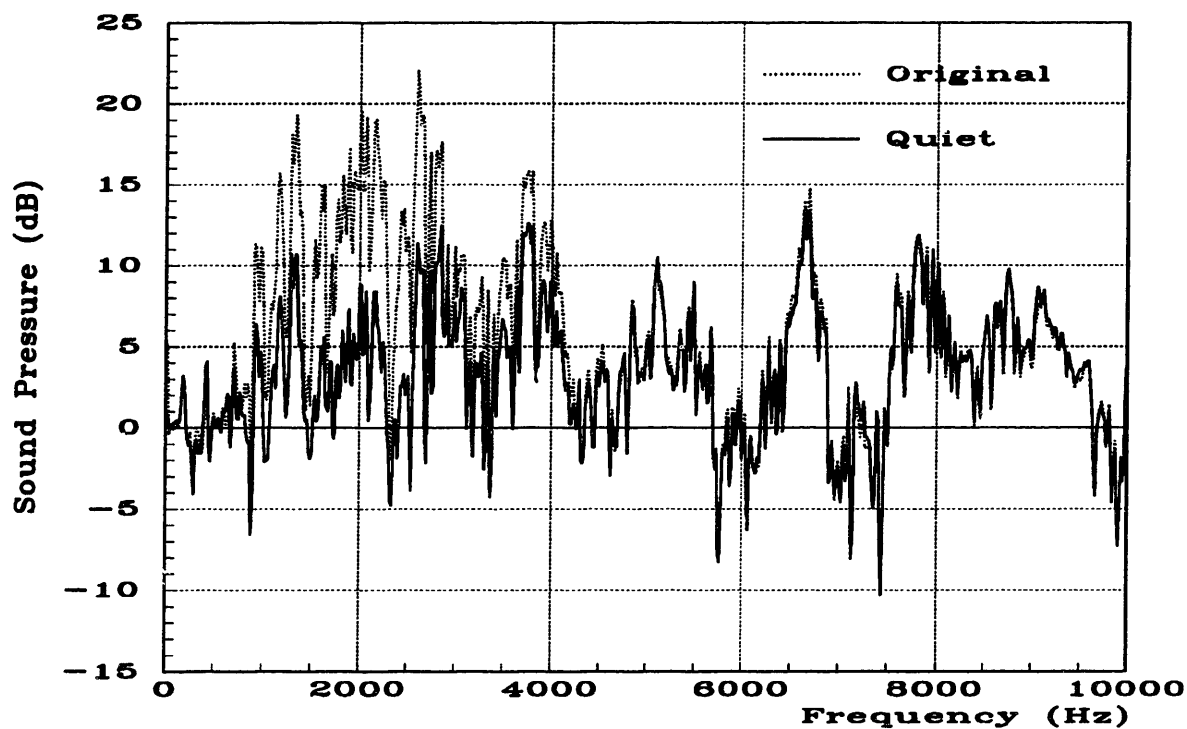


Figure 4-6 Seek Noise, 'Normal' Random Seek Rate, Drive #2

4.1.4 Seek Noise Difference

The seek noise difference plots (Figures 4-7 and 4-8) reflect the reduction in sound-power as a result of using the quiet-seek algorithm over the original seek algorithm. The presented data is a ratio of the total sound power emitted while seeking with the quiet algorithm to the total sound power emitted while seeking with the original RA73 algorithm. The data is calculated by subtracting the total-sound-power data corresponding to the original RA73 algorithm, expressed in decibels, from the total-sound-power data corresponding to the quiet algorithm, also expressed in decibels. The quiet-seek algorithm lowers the sound-pressure power spectrum by 5 to 10 dB between 1 and 3 KHz and by about 3 dB between 3 and 4 KHz. An interesting feature of this data is the peaking visible between 1 and 3 KHz. The peaks correspond to regions where the sound power is less affected by the quiet-seek algorithm. Referring to Figures 4-1 and 4-2 (Baseline Noise) the peaks correspond to tones that are present while track following. It is likely that these strong tones are the result of the spindle's operation because they are unaffected by the actuator's activity.

Seek Noise Difference, 'Normal' Random Seek Rate

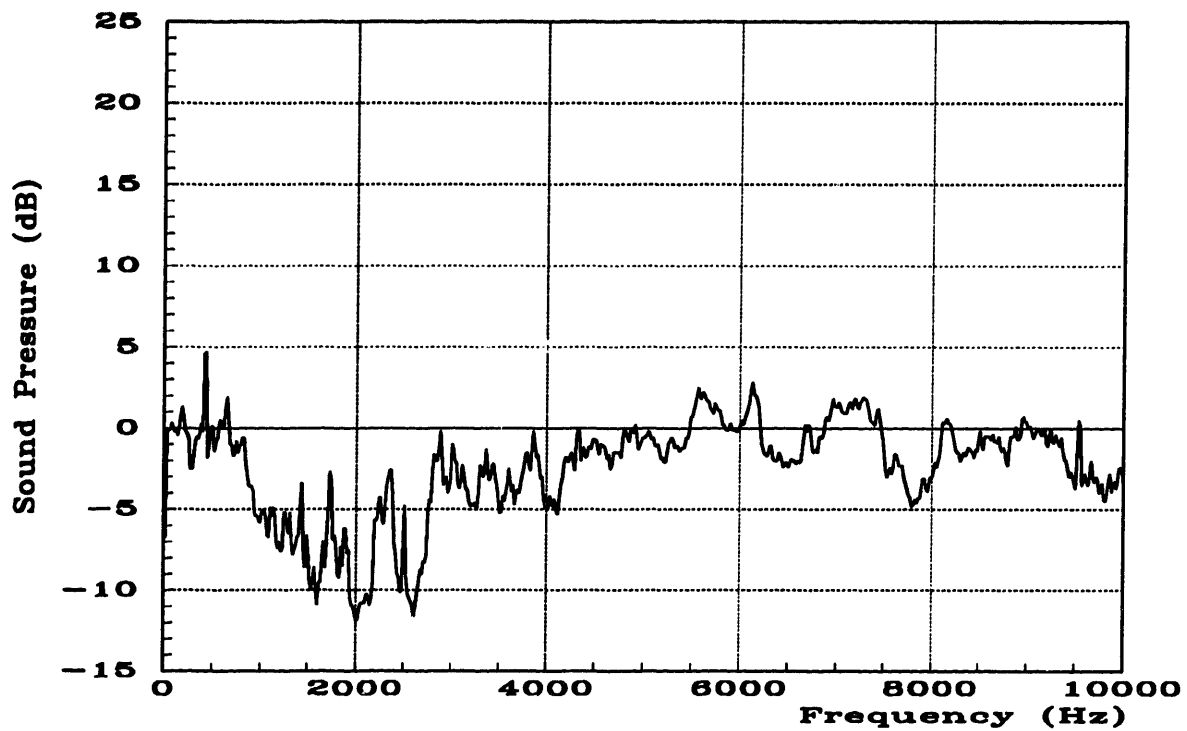


Figure 4-7 Seek Noise Difference, 'Normal' Random Seek Rate, Drive #1

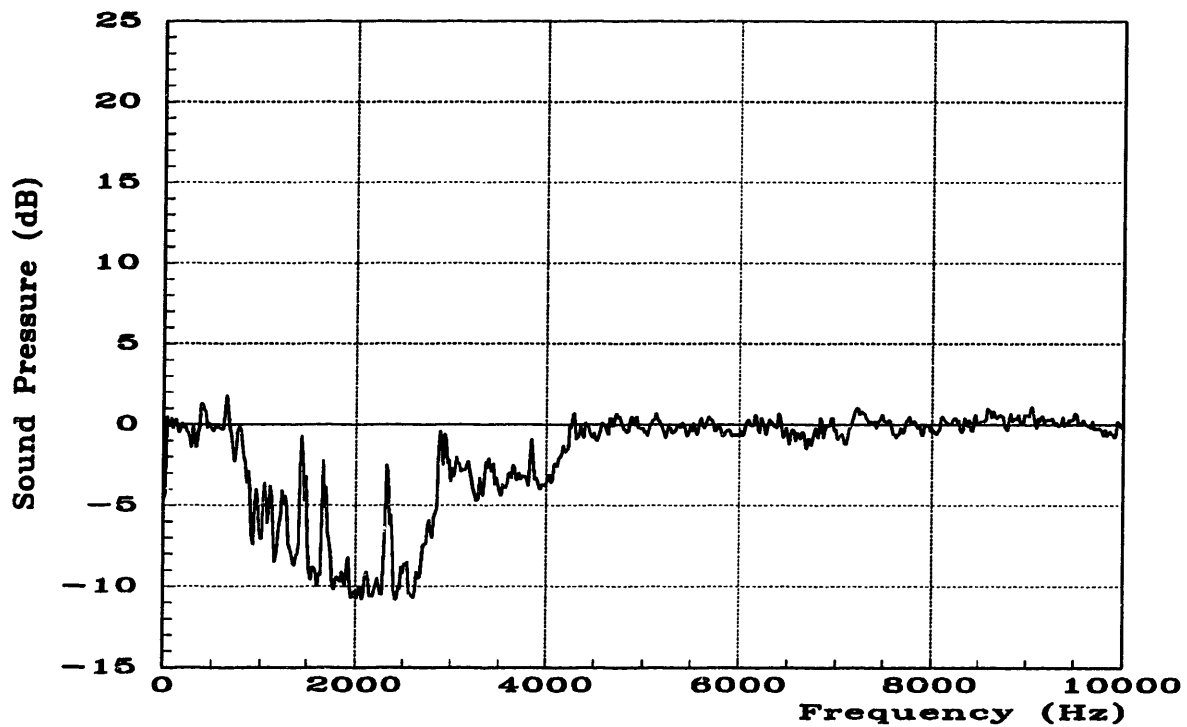


Figure 4-8 Seek Noise Difference, 'Normal' Random Seek Rate, Drive #2

4.1.5 'Worst Case' Measurements

Seek activity causes a drive's noise output to increase. Measurement data was acquired at two operating points corresponding to 44 seeks per second and 79 seeks per second. One expects the sound power due to seeking to scale directly with the seek rate. The data acquired at the 79 seeks per second, or 'worst case', seek rate should show an increase in sound power by the ratio of 79 to 44, or 2½ dB, above the data acquired at the 44 seeks per second, or 'normal', seek rate. The data presented shows this to be the case. This is the only significant difference between the 'normal' and the 'worst case' data. The discussion of the 'normal' data generally applies to the 'worst case' data, consequently and the 'worst case' data is not discussed further.

Total Noise, 'Worst Case' Random Seek Rate

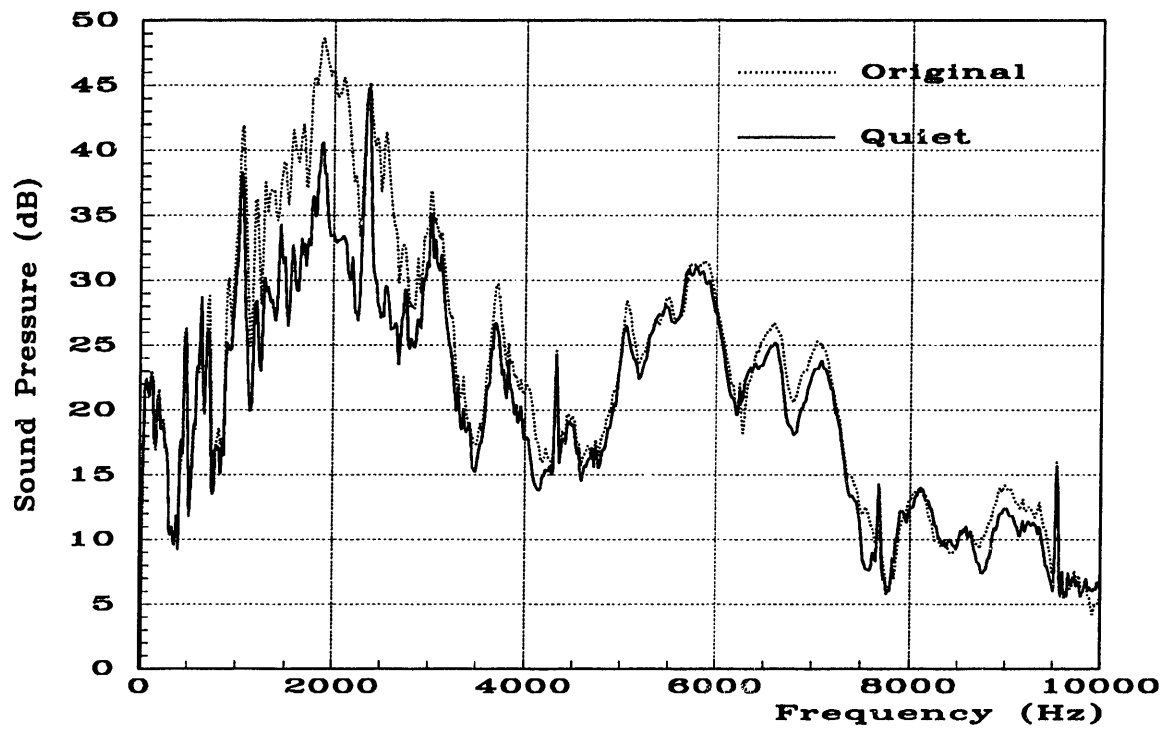


Figure 4-9 Total Noise, 'Worst Case' Random Seek Rate, Drive #1

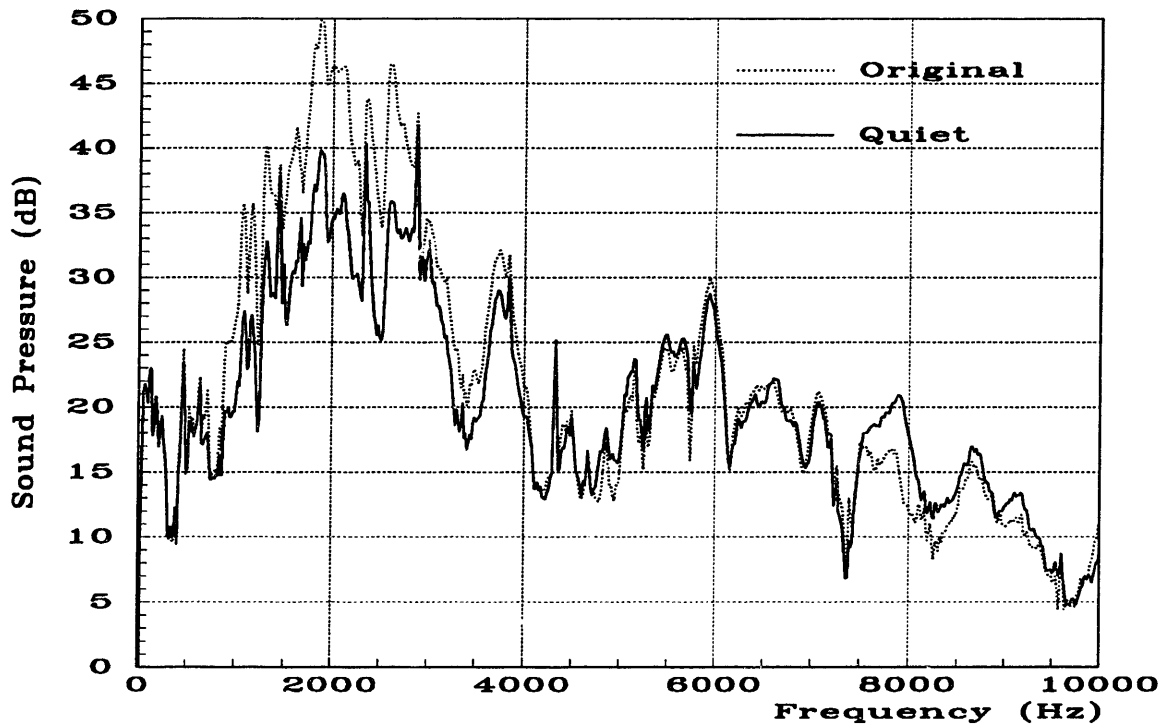


Figure 4-10 Total Noise, 'Worst Case' Random Seek Rate, Drive #2

Seek Noise, 'Worst Case' Random Seek Rate

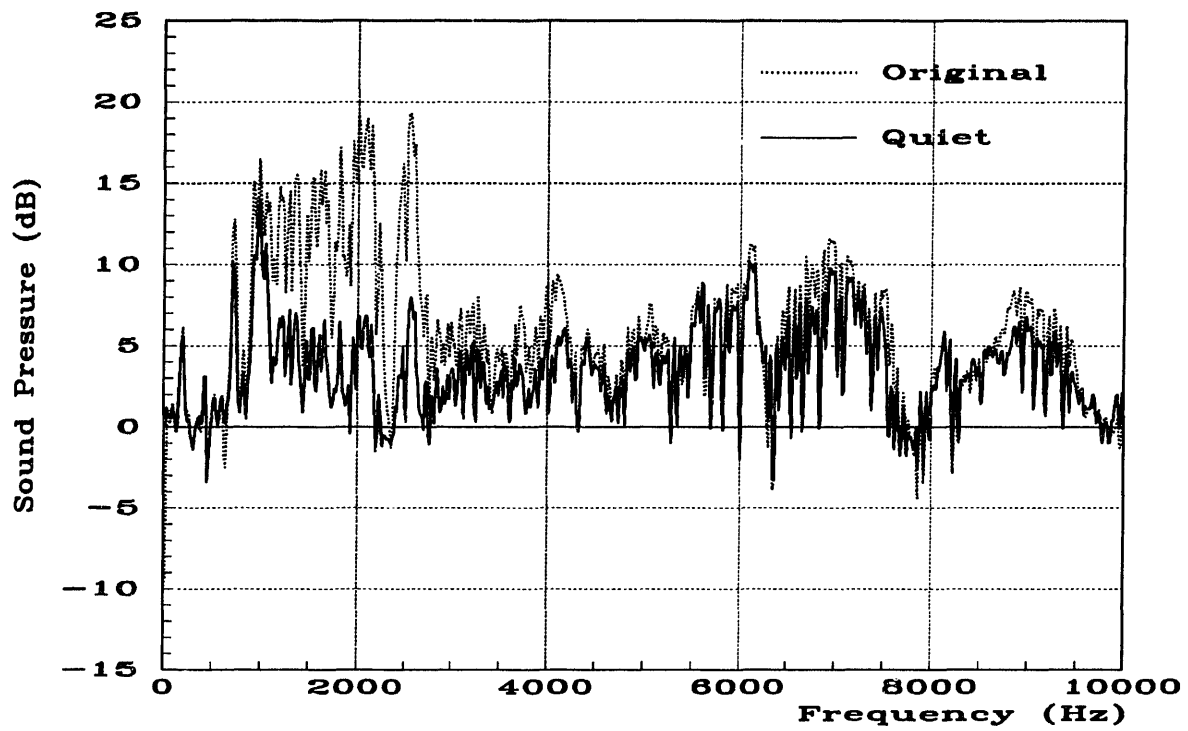


Figure 4-11 Seek Noise, 'Worst Case' Random Seek Rate, Drive #1

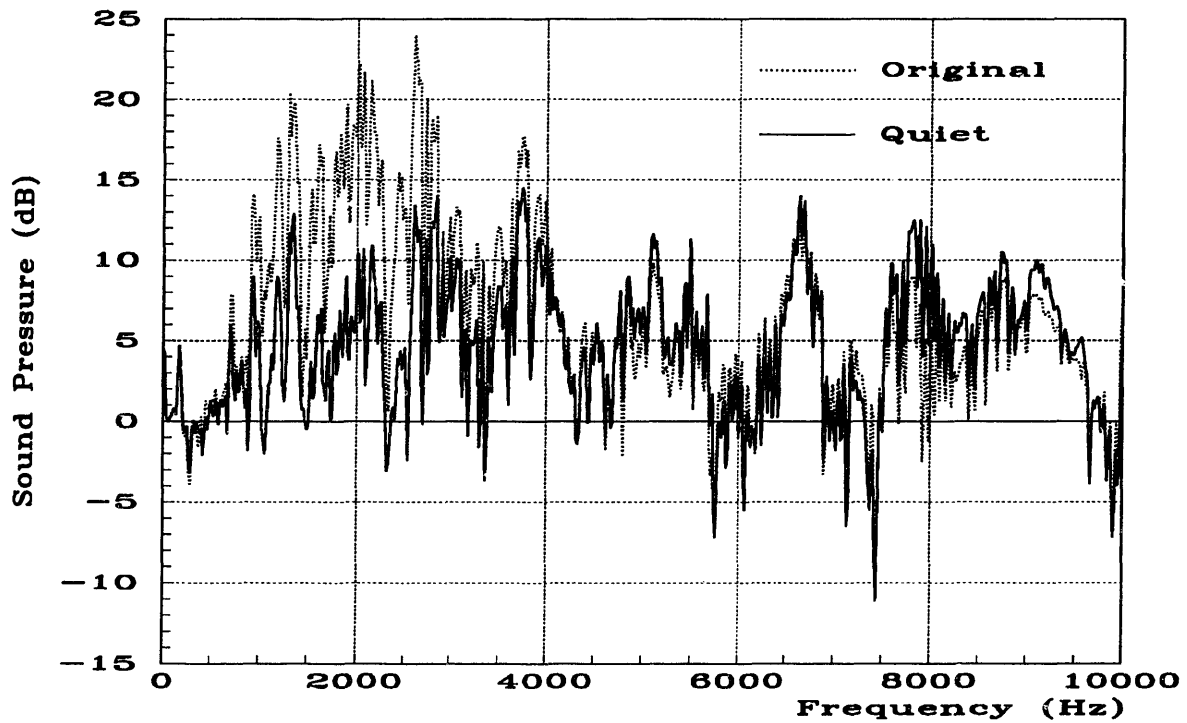


Figure 4-12 Seek Noise, 'Worst Case' Random Seek Rate, Drive #2

Seek Noise Difference, 'Worst Case' Random Seek Rate

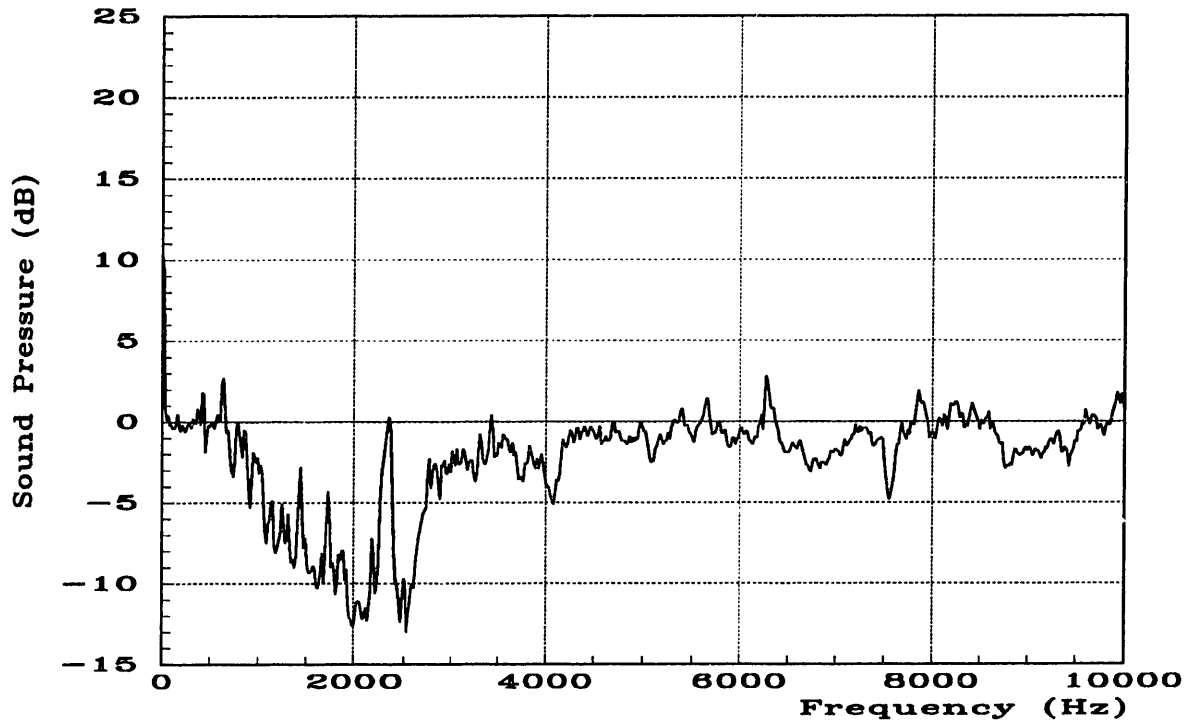


Figure 4-13 Seek Noise Difference, 'Worst Case' Random Seek Rate, Drive #1

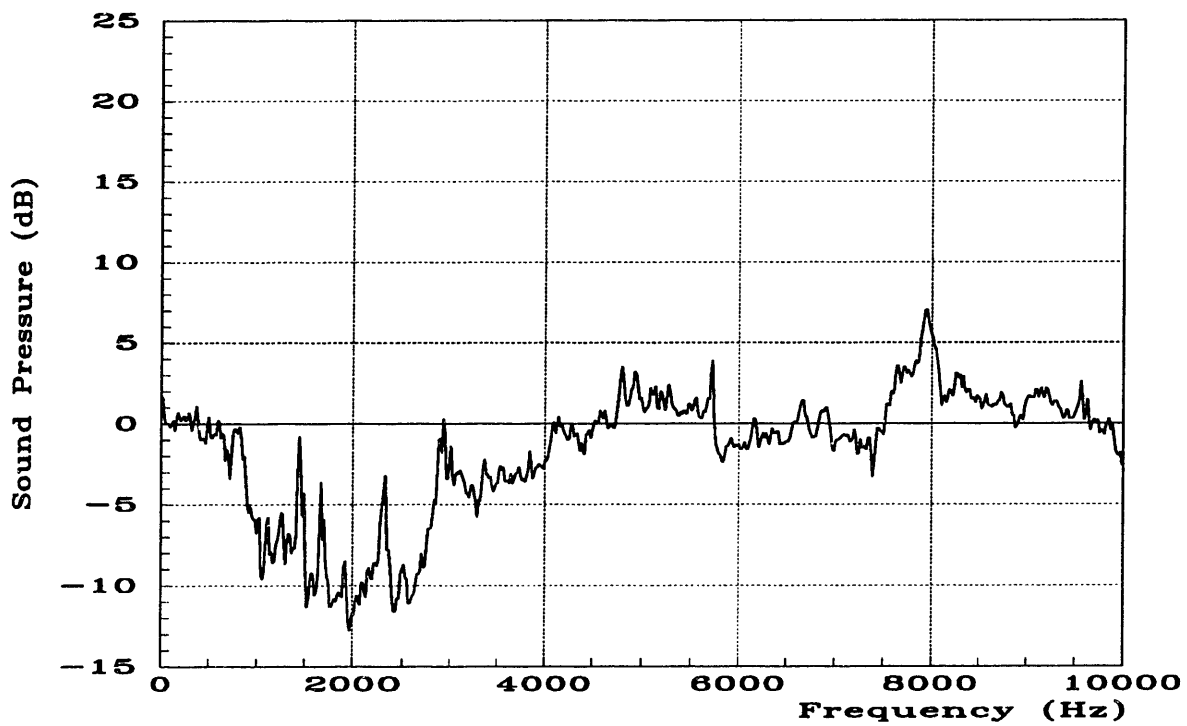


Figure 4-14 Seek Noise Difference, 'Worst Case' Random Seek Rate, Drive #2

4.2 Average Coil Current Power Spectrum

Testing included measuring the average coil current power spectrum as the drive performed random seeks. The objective of the test was to determine the quiet algorithm's effectiveness at reducing high-frequency components of the coil current. Additionally, this test allows verification of the original assumption that altering the frequency content of the coil current results in a corresponding change in the sound pressure. This is accomplished by comparing the changes in the coil current power spectrum with the changes in the acoustic noise emission presented in the last section.

The coil current power spectrum was measured as an RA73 performed random seeks at the 'normal' rate of 44 seeks per second. The test signal was the RA73's "IACT" actuator-current sense signal. The IACT signal is scaled at 1 volt per amp. The test equipment included a Hewlett-Packard 3563A Control System Analyzer and a dual channel Krohn-Hite, which acted as an anti-aliasing filter, before being input to the analyzer's channel #2. The Krohn-Hite was configured as a 4-pole, maximally-flat, 30.0 KHz, low-pass filter with unity gain. The Hewlett-Packard analyzer's state is documented in Table 4-1.

The results for Drive #2 are presented in Figures 4-15 and 4-16 on the following page.¹ The first plot (Figure 4-15) shows a comparison between the original seek algorithm and the quiet-seek algorithm. The average coil-current power spectrums for both algorithms are shown. The second plot (Figure 4-16) shows the difference between the average coil-current power spectrums. The second plot also shows the difference between the corresponding sound-pressure power spectrums. The results support the assumption that the sound pressure is linearly related to the coil current.

¹ The coil current measurements show little variation between drive #1 and drive #2.

Coil Current Power Spectrums, 'Normal' Random Seek Rate

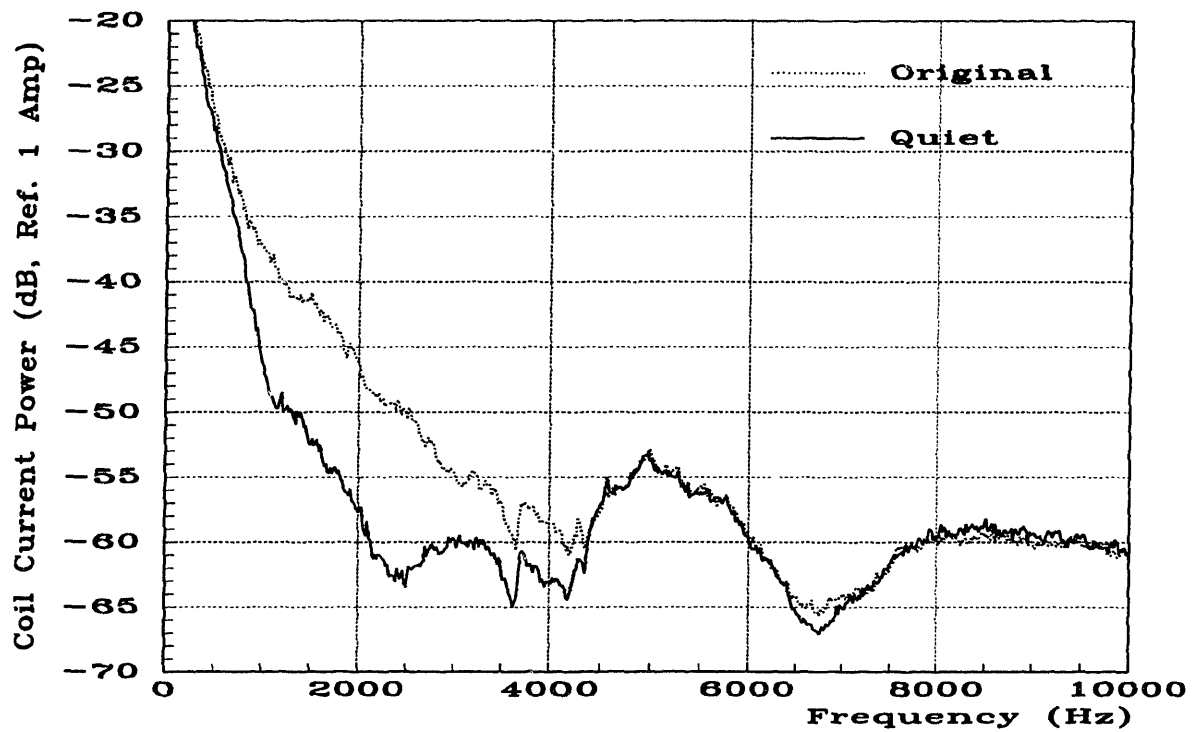


Figure 4-15 Coil Current Power Spectrum, 'Normal' Random Seek Rate, Drive #2

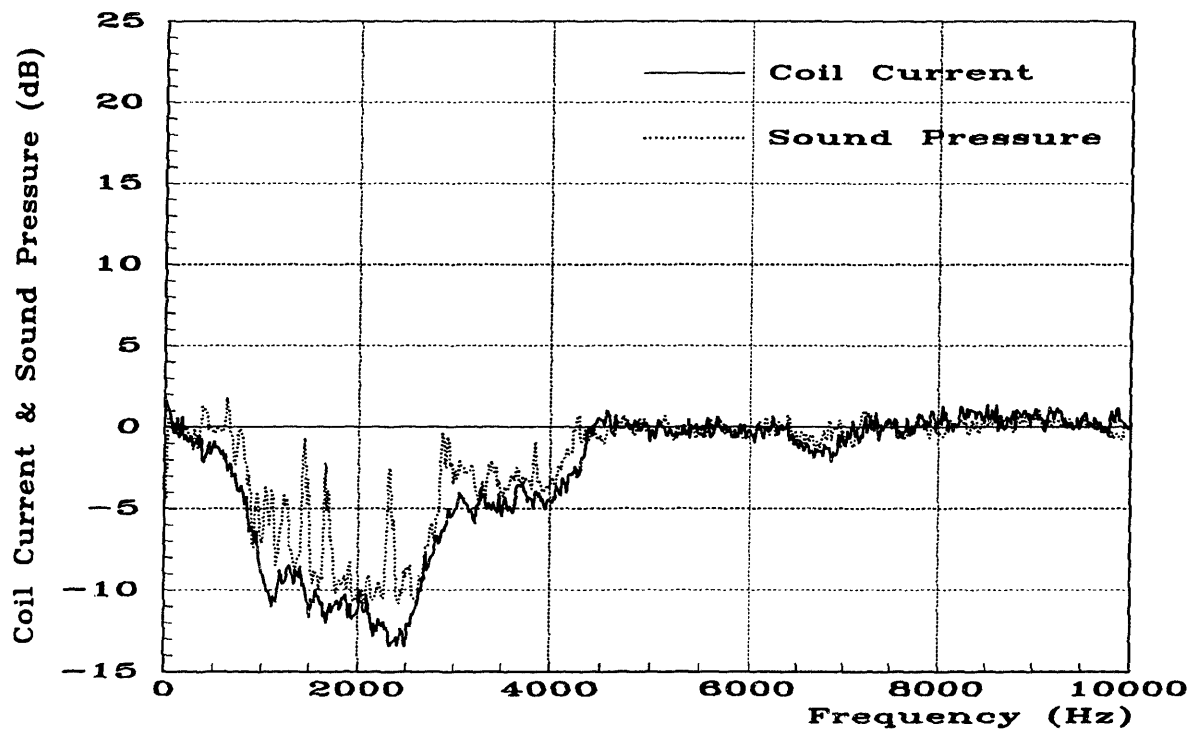


Figure 4-16 Coil Current & Sound Pressure Differences, 'Normal' Seek Rate, Drive #2

Intentionally Blank

4.3 Coil Current Shapes and Power Spectrums for Selected Seek Lengths

The author measured the actuator-coil current for seeks of the following lengths: Single-track, 8-track, 64-track, 160-track, 1000-track, and full-stroke (2,680 track). Figures 4-17 through 4-21 compare the current waveforms between the original seek algorithm and the quiet-seek algorithm. Each figure contains three plots. The first plot compares the time-domain coil-current waveforms over a 20-millisecond period. The coil-current waveforms of the single-track and 8-track seeks are shown on a one ampere magnitude full-scale grid; all other seek lengths are presented on a three ampere magnitude full-scale grid. The second plot compares the coil-current power spectrums between zero and 10 KHz.¹ The last plot shows a smoothed version of the difference (a power ratio) between the current spectrum of the quiet-seek algorithm and the current spectrum of the original seek algorithm. The smoothing permits viewing the general effect of the quiet-seek algorithm on the coil-current power spectrums. The author used a 33-point moving average to smooth the data. Since the coil current data is similar between the two drives, only the data for drive #2 is presented.

The test configuration used for measuring the coil-current data was similar to the test configuration described in section 4.2 (Average Coil Current Power Spectrum). Again, the RA73's "IACT" actuator-current sense signal was used for the measurements; and again, the test equipment included a Hewlett-Packard 3563A Control System Analyzer and a dual channel Krohn-Hite model 3342 filter. The IACT signal was passed through the Krohn-Hite, which acted as an anti-aliasing filter, before being input to the analyzer's channel #2. The Krohn-Hite was again configured as a 4-pole, maximally-flat, 30.0 KHz, low-pass filter with unity gain. Table 4-1 documents the state of the Hewlett-Packard analyzer.

The analyzer was triggered to capture a single record (2048 samples) of time domain data at the beginning of a seek. The trigger signal was the RA73's "HI SLEW L" line which switches from +5.0 volts to ground at the beginning of every seek. Sample records were captured while seeking forward between known tracks. The tracks used for testing and the associated seek lengths are presented in Table 4-2.

¹The power spectrum plots were created by applying an FFT to the time-domain data. The 20 millisecond sample of data contains 1281 points sampled at an interval of 15.625 μ S. The samples were zero padded to 4096 points. The resulting spectrum has a frequency resolution of 50 Hz and a bin resolution of 15.6 Hz. The presented data is $20 \times \log_{10}$ of the magnitude of the FFT output.

Seek Length	Source Track	Destination Track
Single-Track	1,339	1,340
8-Track	1,332	1,340
64-Track	1,276	1,340
160-Track	1,180	1,340
1,000-Track	340	1,340
Full-Stroke	0	2,680

Table 4-2 Coil Current Measurement Tracks

The results, while not surprising, merit discussion. In general, the data reaffirms that sharp edges in time-domain current waveforms lead to broad coil-current power spectrums. The data also shows that the individual current waveforms produced by the quiet-seek algorithm have lower power in the 1-to-4 KHz frequency band than the same waveforms resulting from the original seek algorithm.

The long-seek algorithm replaces the short-seek algorithm as the seek length increases beyond 144 tracks. Consequently, the short-seek algorithm applies to the three short seek lengths, the long-seek algorithm applies to the three long seek lengths.

The short-seek algorithm uses accelerations exhibiting a smooth anti-symmetric, SAS, shape. This is reflected in both the 8-track and the 64-track seek measurements. The short-seek algorithm scales both the amplitude and the duration of the SAS acceleration. This is also reflected in the 8-track and the 64-track measurements. The low current levels associated with the single-track seek limit ones ability to distinguish features within the data.

The long-seek algorithm produces time-domain current waveforms similar to the waveforms produced by the original seek algorithm. The notable difference is that the long-seek algorithm waveforms are generally smoother. This holds true at all times other than at the end of the seek, where the original seek algorithm exhibits a slow exponential decay of the deceleration current. The exponential decay of the original seek algorithm's deceleration current produces excellent settling characteristics; however, it increases the move time by 1½ to 2 milliseconds over the 'ideal' move time.

The convolution of the long-seek algorithm's smoothing pulse with the 'ideal' acceleration waveform delays the 'ideal' move time by the duration of the smoothing pulse, 1¾ milliseconds. These features combine to make the long-seek algorithm waveforms follow the original seek algorithm waveforms by approximately 1¾ milliseconds over most of a seek; however, during the last milliseconds the waveforms converge and the move times become nearly the same. This is the underlying reason for the similar seek times reported in the next section.

The most significant observations from the individual seek-current measurements are summarized as follows:

- The short seeks exhibit coil currents with a smooth anti-symmetric, SAS, waveform. The SAS waveform is scaled in amplitude and duration, dependent upon seek length, as described by the relationships in Equation 3.20 (Short Seek Distance, Duration, and Peak Current Relationships).
- The short-seek algorithm produces current waveforms with considerably less power than the original seek algorithm in the range of acoustically active frequencies. This statement is dramatically supported by Figure 4-19 (Coil Current Comparison, 64-Track Seek, Drive #2).
- The long seeks have coil currents resembling the smoothed 'ideal' accelerations shown in Figure 3-9 (Smoothed Accelerations). All three seek lengths shown exhibit the smoothing of the Gaussian pulse. They also exhibit the variable duration of the acceleration and deceleration phases and the fixed-duration of the cosine acceleration-reversal segment.
- The scaling and offset of the cosine acceleration-reversal segment agrees with values predicted by Equations 3.35 (Cosine Segment, Scale-Factor, α) and 3.36 (Cosine Segment, Offset-Factor, β).
- The long-seek algorithm consistently produces current waveforms with less power than the original seek algorithm in the 1-to-4 KHz range.

The following sub-sections discuss the data for each seek length in detail.

4.3.1 Single-Track Seek

The most noteworthy feature of the single-track coil currents is that they are difficult to discern. In the time-domain, the original seek algorithm produces a 200 milliamp pulse, while the quiet-seek algorithm scarcely disturbs the steady-state tracking current. The current power spectrums show low power levels at all frequencies for both algorithms. The single-track seek data implies that the low power levels used during a single-track seek may not be sufficient to generate significant audible noise. This is true because the RA73 performs single-track seeks quietly, independent of the seek algorithm.

Coil Current, Single-Track Seek

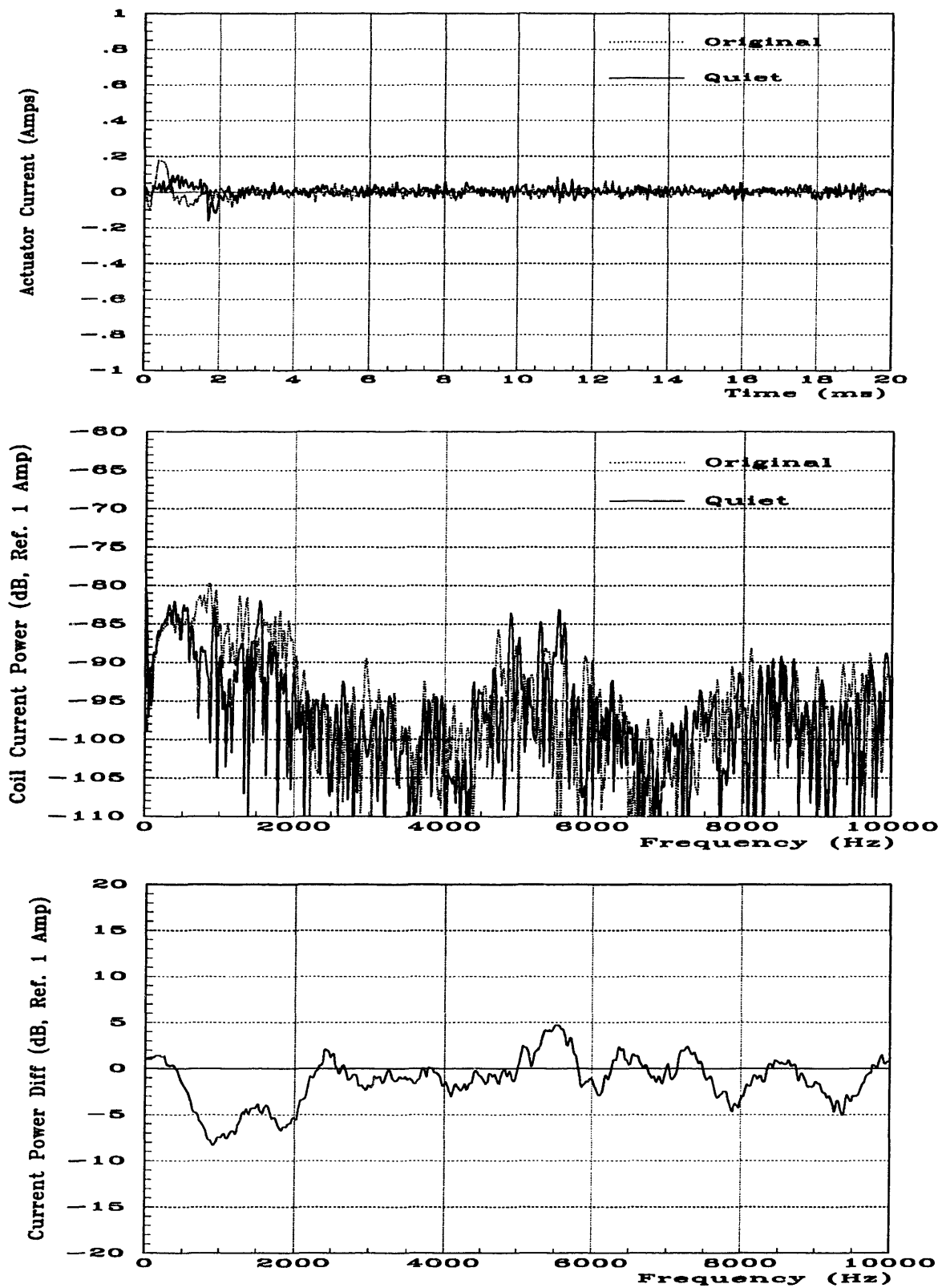


Figure 4-17 Coil Current Comparison, Single-Track Seek, Drive #2

4.3.2 8-Track Seek

The 8-track seek data clearly shows that the characteristics of both algorithms for short seek lengths. The original seek algorithm saturates the power amplifier and produces an initial spike in the coil current. The spike's amplitude is almost one ampere and its duration is approximately 700 milliseconds. The fact that the spike resembles an impulse is reflected in the frequency domain as a broad lobe extending to $2\frac{1}{2}$ KHz. The quiet-seek algorithm does not saturate the power amplifier and produces an anti-symmetric current waveform resembling the SAS waveform. The waveform's amplitude is approximately 400 milliamperes and its duration is three milliseconds, which is the minimum move duration implemented by the short-seek algorithm. The quiet-seek algorithm's spectrum has its first lobe ending at 1 KHz. This agrees with Figure 2-12 (SAS Acceleration Spectrum, 1 Meter Move) which shows that the first major valley of a 3-millisecond SAS spectrum is at 1 KHz. The last plot of the data indicates that the quiet-seek waveform has approximately 15 dB less power than the original seek waveform between 1 and 2 KHz. It also reveals that the quiet-algorithm is moderately effective between 500 Hz and 2,500 Hz.

Coil Current, 8-Track Seek

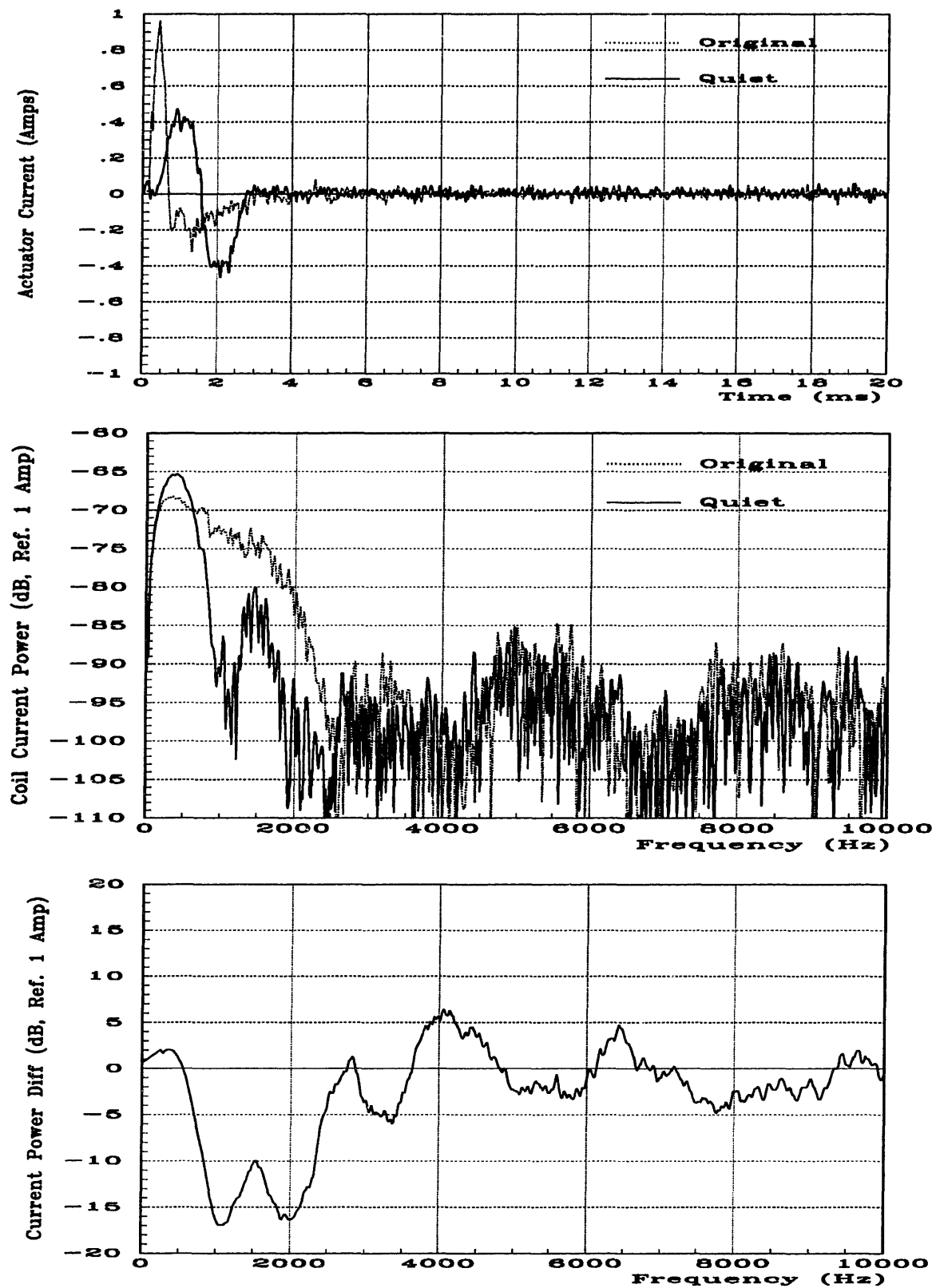


Figure 4-18 Coil Current Comparison, 8-Track Seek, Drive #2

4.3.3 64-Track Seek

The 64-track seek data, like the 8-track seek data, exhibits the characteristics of the two algorithms. Again, the original seek algorithm produces a time-domain current signal with sharp edges which translates into a frequency-domain waveform with several large lobes in the current power spectrum. The quiet-seek algorithm's time-domain signal has the expected SAS appearance. It's duration is nearly five milliseconds, which agrees with a calculation of 4.84 milliseconds from Equation 3.20 (Short Seek Distance, Duration, and Peak Current Relationships) and also agrees with the graphical data presented in Figure 3-3 (Short Seek Duration vs. Distance). The 64-track quiet-seek time-domain waveform is almost five-thirds as long as the 8-track quiet-seek waveform. Since the shapes of the quiet-seek signals are similar, the 64-track seek's spectrum resembles the 8-track seek's spectrum, but with three-fifths scaling of the frequency axis. This is seen in Figures 4-18 and 4-19, where the first valley of the current power spectrum moves from 1 KHz to 600 Hz when the seek length increases from 8 to 64 tracks. Finally, the current-power difference plot shows that the current power of the quiet-seek algorithm is lower than the current power of the original seek algorithm by 5 to 18 dB between 400 Hz and 4,500 Hz.

Coil Current, 64-Track Seek

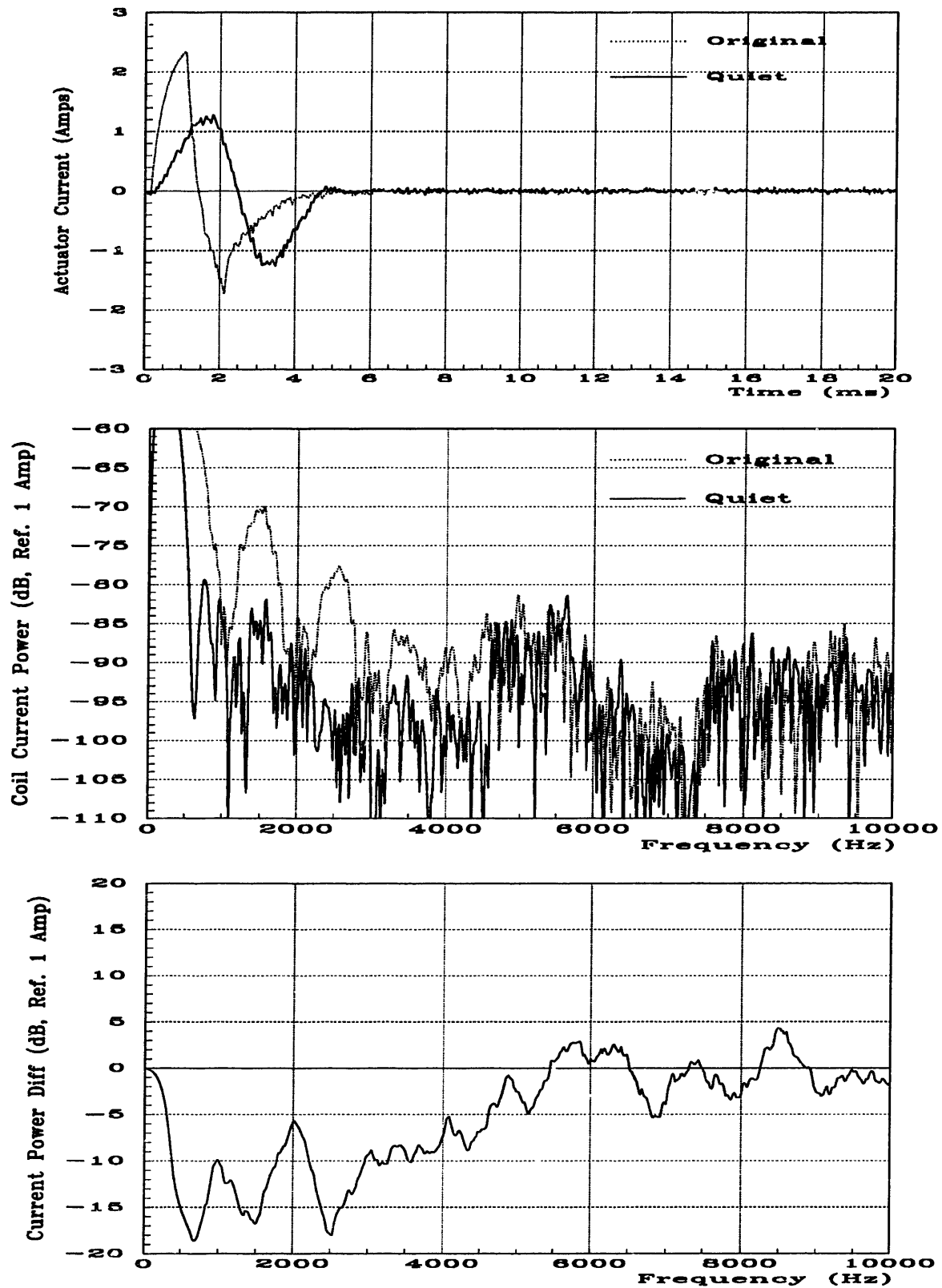


Figure 4-19 Coil Current Comparison, 64-Track Seek, Drive #2

4.3.4 160-Track Seek

The 160-track seek current signal associated with the original seek algorithm has abrupt transitions at three points -- at the beginning of the seek, at the end of the acceleration period, and at the beginning of the exponential decay period. The abrupt transitions are expressed as high-frequency power in the power spectrum plot. The current-power difference plot shows that the original seek algorithm's current has several times the power of the quiet-seek algorithm's current between 1 and 4 KHz. This is consistent with the measured differences in acoustic noise emission.

Coil Current, 160-Track Seek

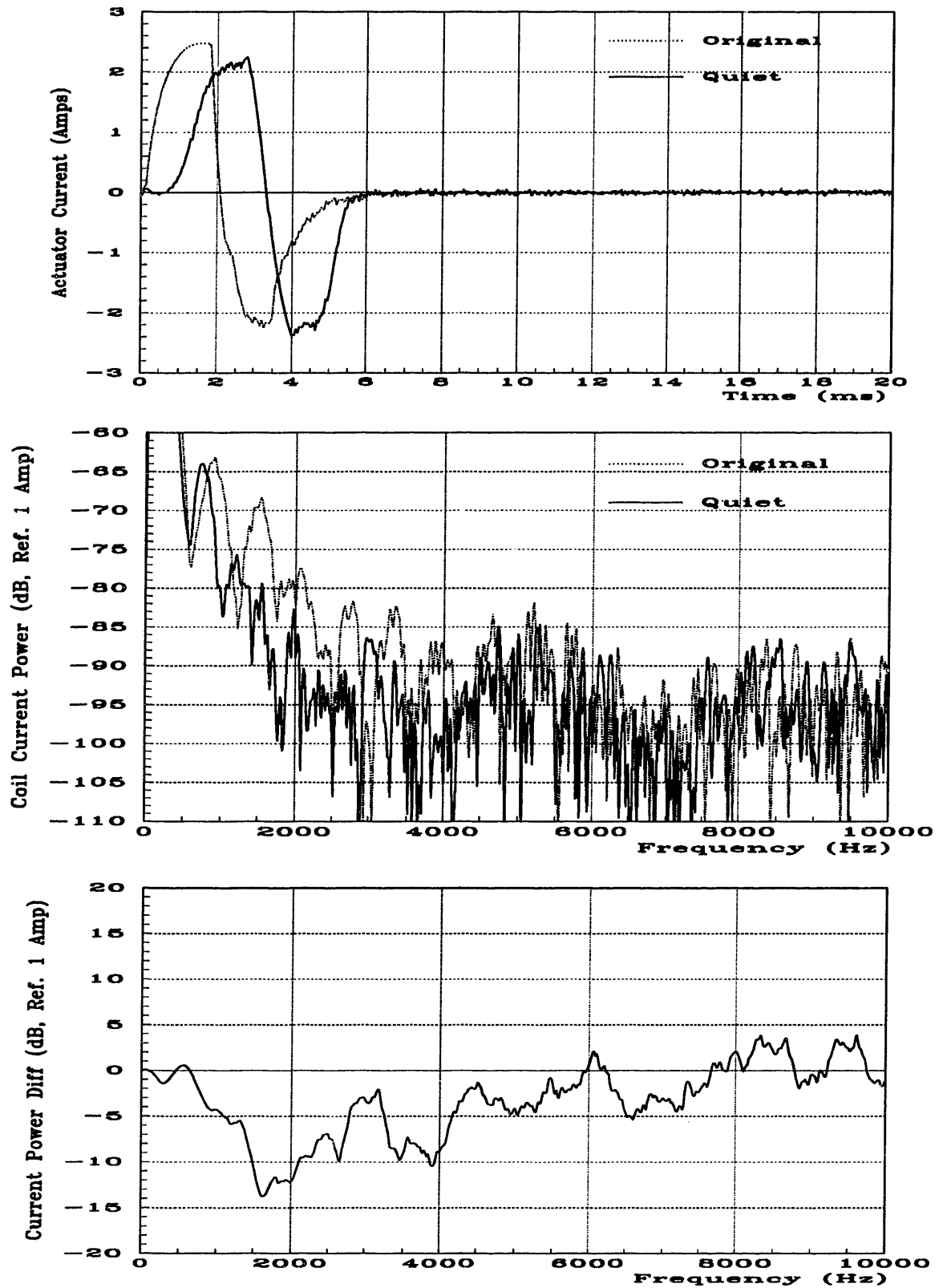


Figure 4-20 Coil Current Comparison, 160-Track Seek, Drive #2

4.3.5 1,000-Track Seek

The 1,000-track seek current waveforms more clearly illustrate the differences illuminated by the 160-track seek. All three plots in Figure 4-21 show the quiet-seek algorithm's current waveform to be smoother than the original seek algorithm's current waveform. The current power difference plot again shows improvement resulting from the quiet-seek algorithm between 1 and 4 KHz. Additionally, the shape of the waveform shown in Figure 3-10, which typifies the intended shape of long-seek algorithm accelerations, strongly resembles the 1,000-track quiet-seek current waveform; they both exhibit smoothing.

Coil Current, 1,000-Track Seek

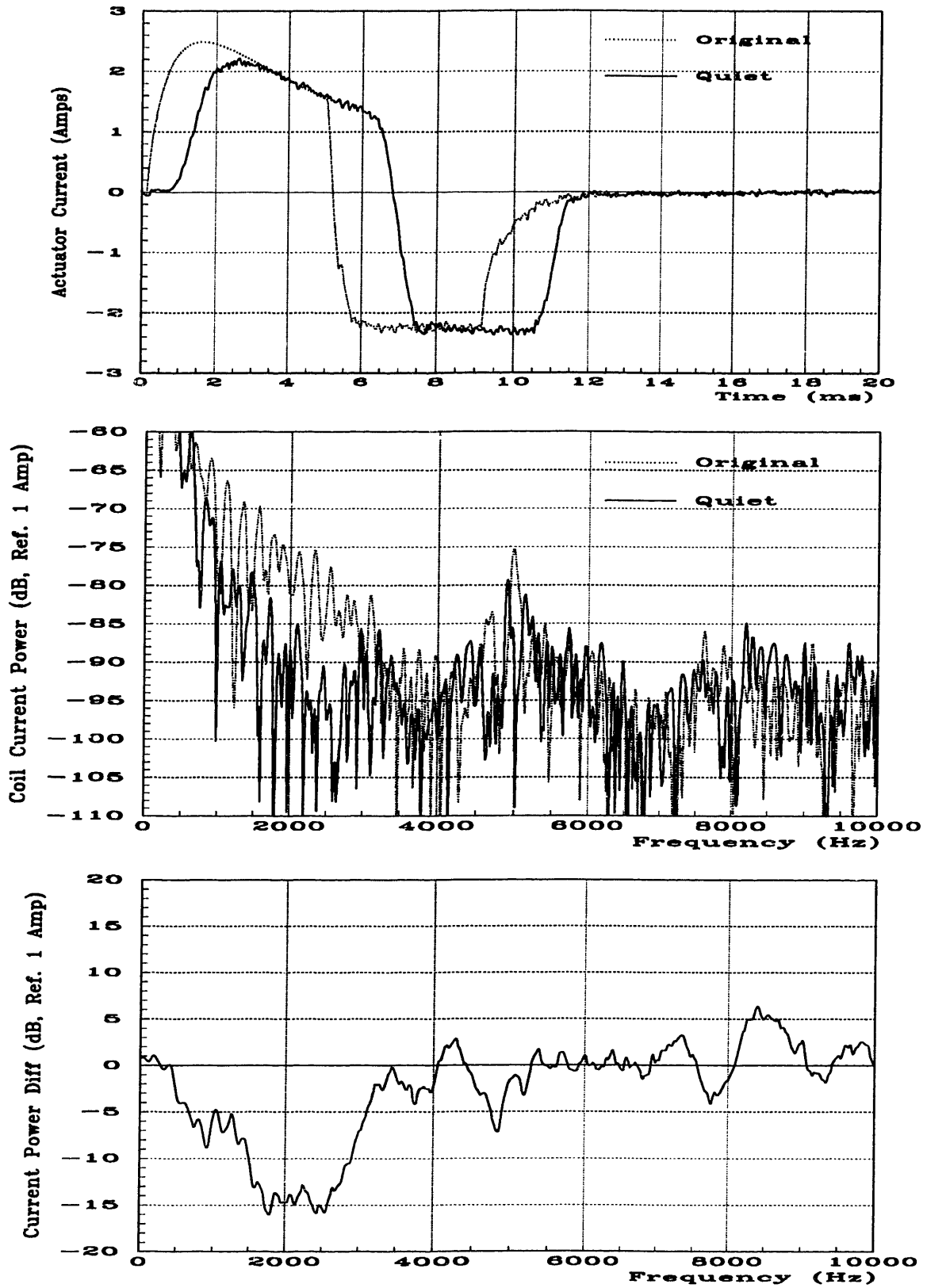


Figure 4-21 Coil Current Comparison, 1000-Track Seek, Drive #2

4.3.6 Full-Stroke Seek

The full-stroke original seek current signal shows a one-millisecond period in the middle of the seek where the coil current is near zero. The original seek algorithm regulates the actuator's velocity to a maximum speed near 3 meters per second. The quiet-seek algorithm does not share this feature. The full-stroke quiet-seek current waveform illustrates the scaling and level shifting of the cosine acceleration-reversal segment. The servo sensor's velocity is near 2.75 meters per second at the onset of the cosine acceleration-reversal segment. Substituting 2.75 meters per second for V_c in Equation 3.35 (Cosine Segment, Scale-Factor, α) leads to a scale factor of 340 meters per second-squared. Assuming the actuator's gain is 220 meters per second-squared per ampere, the calculated scale factor corresponds to a cosine segment with amplitude of 1.6 amperes. The measured signal appears to have a peak amplitude near 1.5 ampere. The measured signal also appears to have an offset of -0.7 amperes at the cosine segment. This is close to -0.74 ampere, which is the value calculated by substituting 2.75 meters per second for V_c in Equation 3.36 (Cosine Segment, Offset-Factor, β) while assuming the actuator's gain is 220 meters per second-squared per ampere. The time-domain trace also evidences the 1½ millisecond duration of the cosine segment. The tail end of the trace reflects the duration of the Gaussian smoothing pulse in the width of the transition from full deceleration current to zero current. This transition area is 1.76 milliseconds wide, although the low amplitude at the edges of the smoothing pulse causes the transition zone to appear slightly narrower. The frequency-domain characteristics of the full-stroke seek are similar to those of the 1,000-track seek. Once again, the quiet-seek algorithm results in a lower current power spectrum between 1 and 4 KHz as compared with the spectrum of the original seek algorithm.

Coil Current, Full-Stroke Seek

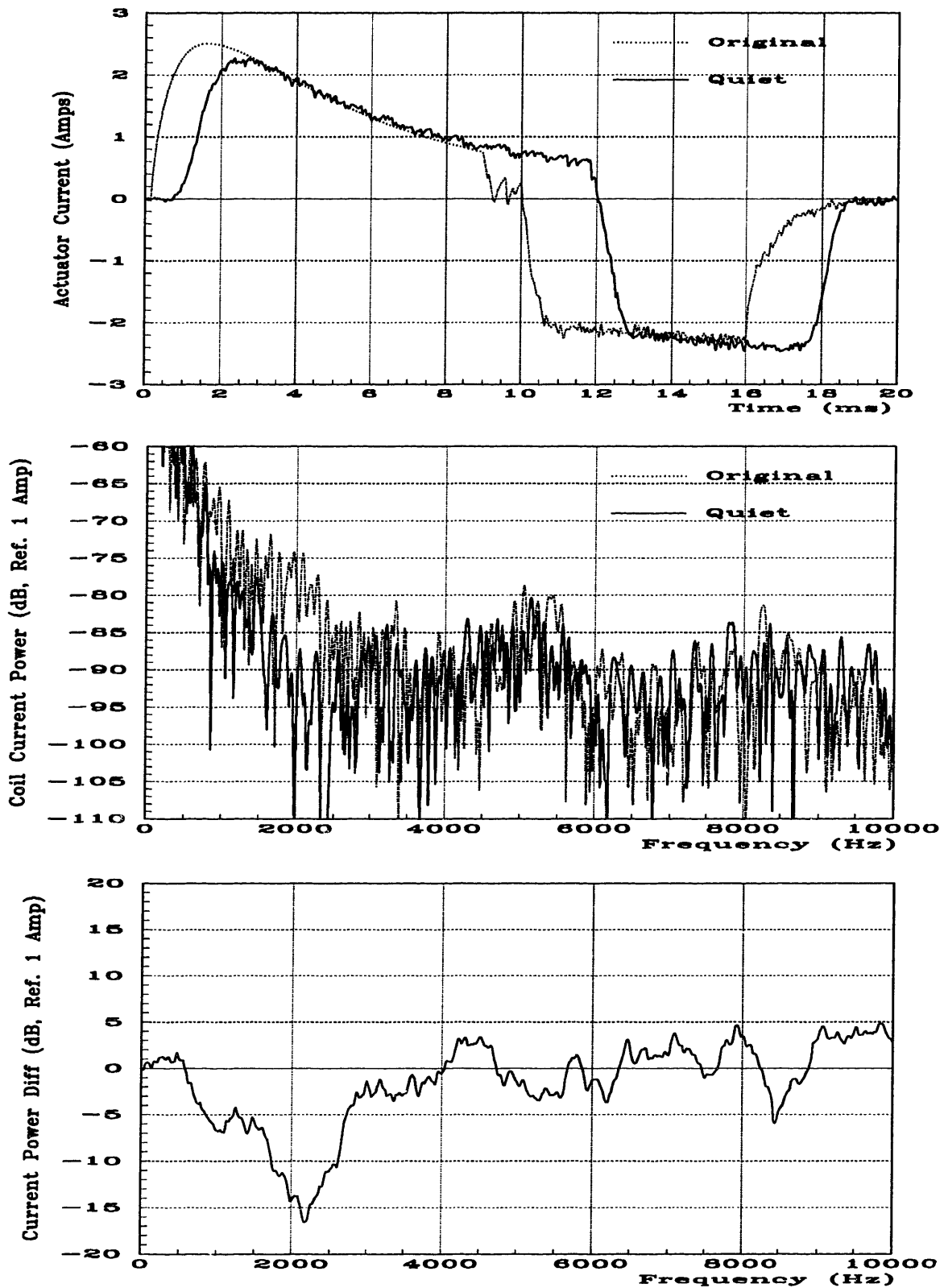


Figure 4-22 Coil Current Comparison, Full-Stroke (2,680 Track) Seek, Drive #2

4.4 Average Seek Times

The RA73's seek-time performance while employing the quiet-seek algorithm was compared with the seek-time performance while using the original seek algorithm. The performance was compared for four seek lengths (single-track, eight-track, third-stroke, and full-stroke) and for a pattern of 'random' seek lengths. The testing employed a utility existing inside the RA73's servo microcode. The utility measures the average duration of 1,020 individual seeks. The duration of a seek is defined as the time between when the servo's DSP decodes a seek instruction and when the servo's DSP declares that the actuator has settled on track with little residual motion.

The measured seek time has three components -- seek-initialization time, move time, and settle time. The seek-initialization time includes both the time required to access various look-up tables and the time to calculate several initial values (assorted scale, offset, and correction factors). The seek-initialization time is typically on the order of 100 microseconds. For most seeks move time comprises the largest portion of the total seek time. However, for short seeks the settle time may be the dominant component. The original seek algorithm's move time is defined as the time required to move the actuator from a start track to within 12½% of a destination track centerline. The quiet-seek algorithm's move time is simply the time required to sequence through the seek trajectories. The move time of both algorithms averages approximately 12 milliseconds; however it ranges between one and 19 milliseconds. The last component of the seek time is settle time. Settle time varies from roughly one to two milliseconds depending upon seek length. Long seeks are more likely to induce mechanical disturbances and have the longest settle times.

The RA73's seek-time measurement utility measures the duration of individual seeks by counting sample periods. This implies that the resolution of an individual measurement is one sample period, or 47 microseconds. The utility accumulates 1,020 individual measurements and then calculates the average seek time. A 256-element table, stored in the DSP's memory, determines the 1,020 seek lengths that are measured. Each entry in the table corresponds to a destination track. The utility sequences through the table and seeks from destination to destination while measuring and accumulating individual seek times. The utility runs through the table four times, accumulating 255 seek times with every iteration; ultimately acquiring 1,020 measurements. The lengths of the seeks being measured depend upon the differences between the elements of the table. When making single-track seek-time measurements the table elements ascend sequentially

from 1,278 to 1,406 and then descend sequentially back to 1,279. This provides an average of forward and reverse single-track seek times. The utility uses the same technique to measure the average eight-track seek-time, except the table runs from 829 to 1,853 and then back to 837 in increments of eight. Third-stroke and full-stroke measurements use tables that result in 'toggle' seeks. The third-stroke table causes repetitive seeking between tracks 894 and 1,788, while the full-stroke table produces seeks between tracks 0 and 2,682. The table used to measure the 'average' seek-time consists of 256 uniformly-distributed integers between 0 and 2,682¹.

The tables below present the seek-performance data. Each table corresponds to the measurements from one the two RA73's that were tested. The tables present the average seek-time and their standard deviations resulting from repeating each test sixteen times. The presented seek averages derive from 16,320 individual seek times; however, the standard deviations derive from 16 averages of 1,020 individual seek times. It is interesting that the standard deviations of the quiet-seek algorithm times are roughly one third the size of the standard deviations of the original seek algorithm times. The greater predictability of the quiet-seek algorithm times may be an advantage in computer systems using seek ordering to optimize storage-systems' performance.

Performance Comparison:	Single Track (ms)	Eight Track (ms)	Third Stroke (ms)	Full Stroke (ms)	Average Seek (ms)
Drive #1					
Original Average	2.563	4.138	13.465	20.811	13.253
Quiet Average	3.800	3.932	13.185	20.445	13.044
Improvement	(1.237)	0.206	0.280	0.367	0.209
Original Standard Deviation	0.003	0.004	0.037	0.054	0.028
Quiet Standard Deviation	0.001	0.001	0.012	0.010	0.004

Table 4-3 Seek-Time Performance Comparison, Drive #1

¹The table was generated by taking the integer portion of 2682 times a 256-element random vector returned by the MatrixXTM commands, "rand('uniform')" and "vector=rand(256,1)".

Performance Comparison: Drive #2	Single Track (ms)	Eight Track (ms)	Third Stroke (ms)	Full Stroke (ms)	Average Seek (ms)
Original Average	2.509	4.131	13.773	21.119	13.208
Quiet Average	3.797	3.940	13.241	20.514	12.921
Improvement	(1.288)	0.191	0.532	0.605	0.288
Original Standard Deviation	0.002	0.005	0.055	0.041	0.023
Quiet Standard Deviation	0.001	0.003	0.013	0.012	0.008

Table 4-4 Seek-Time Performance Comparison, Drive #2

The seek-time results shown in the above tables require interpretation. The results indicate the quiet-seek algorithm slightly improves performance over the original seek algorithm. This is correct; however, the accelerations employed by the original seek algorithm deviate from the 'ideal' accelerations upon which the quiet-seek algorithm is based.

The following analysis explains why the quiet-seek algorithm is approximately one half of one millisecond faster than the original seek algorithm for the third-stroke and full-stroke seek lengths. The original seek algorithm, like a time-optimal solution, saturates the power amplifier during the early part of the seek. This provides the fastest possible, or 'ideal', acceleration. The original seek algorithm also employs nearly all the available current over a constant-level deceleration phase; however, it uses an exponentially decaying current for the final 19.3 tracks. This exponential segment is not represented in the 'ideal' accelerations. The exponential-decay period allows high-frequency vibrations to diminish and consequently produces smooth settles. The duration of the exponentially decaying current is approximately 3½ milliseconds for long seek lengths (greater than 80 tracks). A controller implementing the 'ideal' accelerations would require significantly less time to cover the last 19.3 tracks. Ignoring the inductance of the actuator's coil, such a controller would require nearly 0.9 milliseconds¹ to move the final 19.3 tracks while decelerating at 500 meters per second-squared. Accounting for the actuator's inductance raises the time to roughly 1.2 milliseconds. The original seek algorithm's exponential-decay period causes the

¹ $x = \frac{1}{2}at^2 \Rightarrow t = \sqrt{(2x/a)}$, $x = 19.3 \text{ tracks} = 200\mu\text{m}$, $a = 500\text{m/S}^2 \Rightarrow t = 0.89\text{mS}$

algorithm to be at least 2.3 milliseconds slower than an algorithm using saturated control signals. The original seek algorithm uses an exponential-decay period to reduce post-seek vibration; the long-seek algorithm uses smoothed 'ideal' accelerations to limit vibration. The smoothing process employed in the long-seek algorithm adds 1.76 milliseconds to the duration of the 'ideal' moves. This is in contrast to the original seek algorithm's 2.3 millisecond penalty. Consequently, the long-seek algorithm performs seeks approximately one half of one millisecond faster than the original seek algorithm.

The short-seek algorithm's implementation limits the minimum move duration to three milliseconds. This limitation is imposed by the controller's inability to track faster trajectories. Consequently, the quiet-seek algorithm consistently produces three millisecond single-track and eight-track move times. Additional to move time, seek time includes seek-initialization time and settle time. The measured settle time for short seeks is primarily determined by the number of servo samples used by the DSP to verify that the actuator has settled, rather than the actual time required for settling. This verification time, which is included in the measured seek times, is typically 0.70 to 0.95 milliseconds. The measured single-track and eight-track quiet-seek times match the predicted times based upon the algorithm's minimum time requirements. The single-track and eight-track seek times associated with the original seek algorithm are not as easily explained; however, the measurements in the tables show that the original seek algorithm out-performs the quiet-seek algorithm by $1\frac{1}{4}$ milliseconds for single-track seeks and under-performs the quiet-seek algorithm by one fifth of a millisecond for eight-track seeks.

The 'average' seek-time measurements indicate that the quiet-seek algorithm is approximately one quarter of one millisecond faster than the original seek algorithm, corresponding to a 2% performance improvement. Comparing the quiet-seek algorithm's performance to a theoretical 'ideal' performance reveals that the quiet-seek algorithm is approximately $1\frac{3}{4}$ milliseconds slower than the fastest possible algorithm (given the RA73's current limiting feature). However, an algorithm which produces 'ideal' move times will cause settle times to increase. The settle times increase to allow the decay of mechanical vibrations excited by switched saturated control signals. The following section discusses this problem.

4.5 Post-Seek Settling Characteristics

The reduction of the RA73's acoustic noise emission by the quiet-seek algorithm results from the fact that the quiet-seek algorithm does not excite mechanical vibrations. One might presuppose reducing general vibration would reduce the vibrations which disturb the servo's sensor head at the end of a seek. Accordingly, one would expect the quiet-seek algorithm to have smoother settles than the original seek algorithm. In particular, the position-error signal after a seek implemented with the quiet-seek algorithm should have a lower power spectrum over the 1-to-4 Khz frequency range. The author tested this theory by comparing the position-error signals of both the original and quiet-seek algorithms after seeks of lengths of one-track, 8-tracks, 64-tracks, 160-tracks, and 1,000-tracks.

4.5.1 Comparing the Post-Seek Vibration of the Quiet and the Original Seek Algorithm

The RA73's position-error signal, "DED C-D," was used to measure settling characteristics. The position-error signal is an analog voltage corresponding to the servo sensor's distance from an intended position (a track centerline). This signal is scaled at approximately 2.25 volts per track. The test equipment included a Hewlett-Packard 3563A Control System Analyzer and a dual channel Krohn-Hite model 3342 filter. The position-error signal was passed through the Krohn-Hite, which acted as an anti-aliasing filter, before being input to the analyzer's channel #2. The Krohn-Hite was configured as a 4-pole, maximally-flat, 10.5 Khz, low-pass filter with unity gain. The Hewlett-Packard analyzer's state is shown in Table 4-1.

The analyzer was triggered to capture a single record (2048 samples) of time domain data at the end of a seek. The rising edge of the RA73's external flag signal (XF, pin #56 of the TMS320C25 DSP) was used to post-trigger the analyzer. As in the measurement of the coil current shapes, the sample records were captured while seeking forward between defined tracks. The tracks used for testing are presented in Table 4-5.

Seek Length	Source Track	Destination Track
Single-Track	1,339	1,340
8-Track	1,332	1,340
64-Track	1,276	1,340
160-Track	1,180	1,340
1,000-Track	340	1,340

Table 4-5 Post-Seek Settling Measurement Tracks

Four records were acquired for each seek length. Each record contains 80 milliseconds of sampled information. The settle period lasts only a few milliseconds, consequently the data was edited to contain only a six-millisecond sample. The data was also edited to remove the constant irregularities exhibited by the position-error signal as it follows a track. The constant irregularities were found by time averaging many (100) records of position-error data that was acquired synchronously with the disk's revolution. This 'average' position-error signal was subtracted from the settle measurements to isolate the components of the position-error signal produced by the seek. The data was truncated to contain only six milliseconds of information (154 samples) after the 'average' position-error signal was removed. The author attempted to present the data such that the origin of the time scale coincides with the moment when the servo's sensor arrived at the desired track. The plots show two milliseconds of data prior to the servo sensor's arrival at the desired track, and four milliseconds of settle time after the arrival.

Although four settles were captured for each seek length, the differences between settles for a given seek length do not warrant the presentation of the entire data set. Instead, only one time capture per seek length is presented. The settle data shown in Figure 4-23 was acquired while testing disk drive #2.

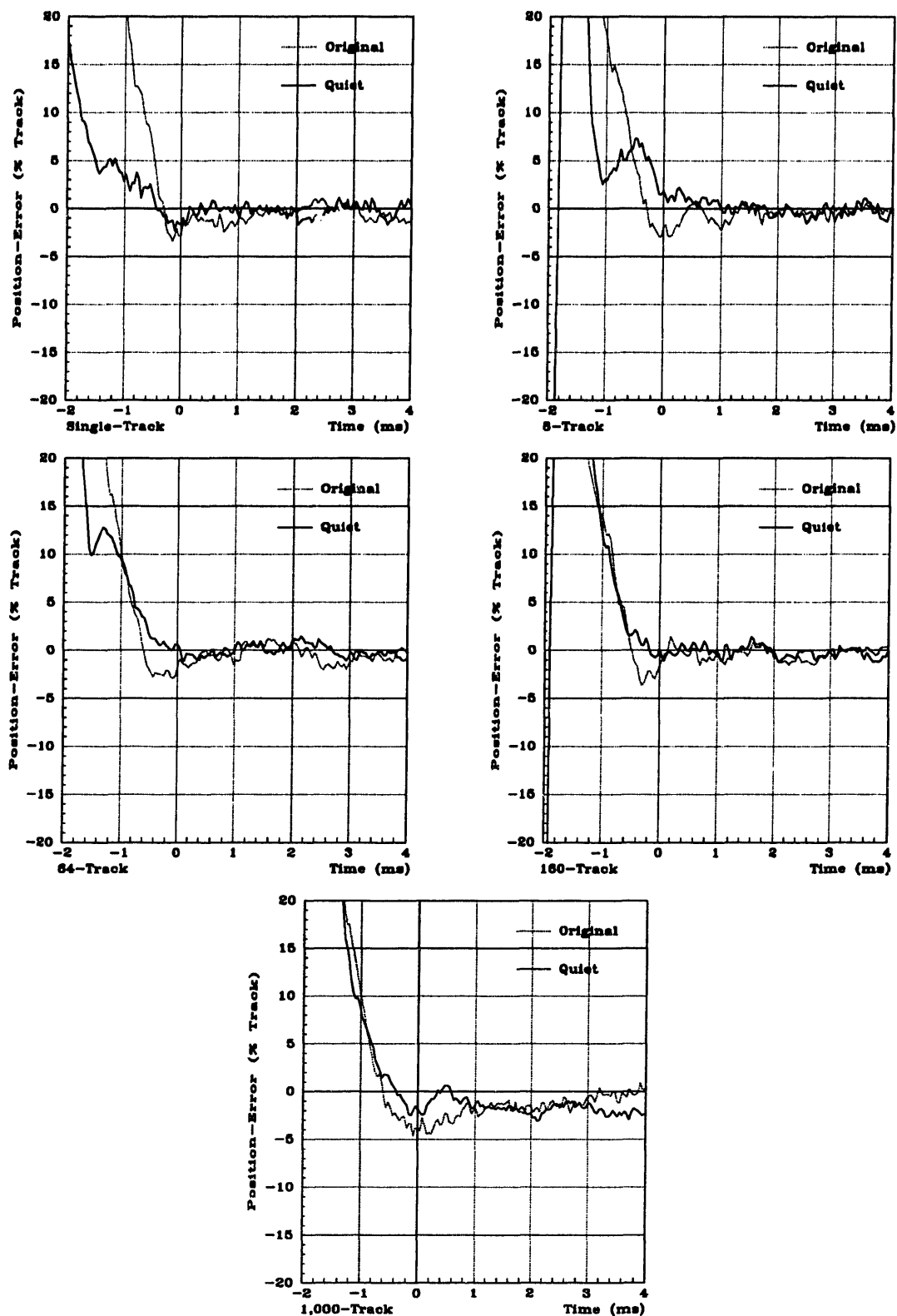


Figure 4-23 Post-Seek Settles, Position-Error Signal, Drive #2

The author expected the time-domain data presented in Figure 4-23 to show that the quiet-seek algorithm produces less post-seek vibration than the original seek algorithm. The fact that the quiet-seek algorithm excites the coil with less high-frequency power is the basis of this expectation. However, contrary to expectation, the settles of both algorithms appear similar in Figure 4-23.

Because subtle differences in vibration can be difficult to discern in time-domain data, the author proceeded to examine the data in the frequency domain. The author prepared the data by first applying a 128-point FFT to the 102 samples of data corresponding to the first four milliseconds of position-error data after each track arrival¹. The author then averaged the position-error power spectrums of four settles for each measured seek length. The resulting frequency-domain data is presented in Figure 4-24.

The frequency-domain analysis, like the time-domain data, does not conclusively support the hypothesis that the quiet-seek algorithm produces less post-seek vibration. Although the 64-track and 1,000-track seek settles show some benefit at frequencies near 2 KHz, the single-track, 8-track, and 160-track seeks appear to be unaffected. The data suggests that the quiet-seek algorithm is far less effective at reducing post-seek vibration than it is at reducing acoustic noise emission.

One possible explanation of the results is that because both algorithms employ smooth decelerations for several milliseconds before track arrival, they both exhibit little post-seek vibration. The original seek algorithm uses an exponentially decaying deceleration ($\tau=0.64$ milliseconds) at the end of a move; this ensures smooth settles. The exponentially decaying current lasts $3\frac{1}{2}$ milliseconds at the end of most seeks. Even short seeks have the same exponentially decaying current over most of their duration. Because this current is represented by a real exponential, as opposed to a complex exponential, it does not excite vibration. This implies that vibration is allowed to decay over a period lasting several milliseconds at the conclusion of an original seek algorithm move. Consequently, the original seek algorithm, like the quiet-seek algorithm, exhibits little post-seek vibration.

¹ The data was sampled at 25.6 KHz, or every 39.06 microseconds. The sample length of 102 yields a frequency resolution of 253 Hz, whereas the 128-point FFT yields a bin resolution of 200 Hz. The average position-error power spectrums are presented in decibels referenced to one track per kilohertz, squared ($[1 \text{ track-second} \times 10^{-3}]^2$).

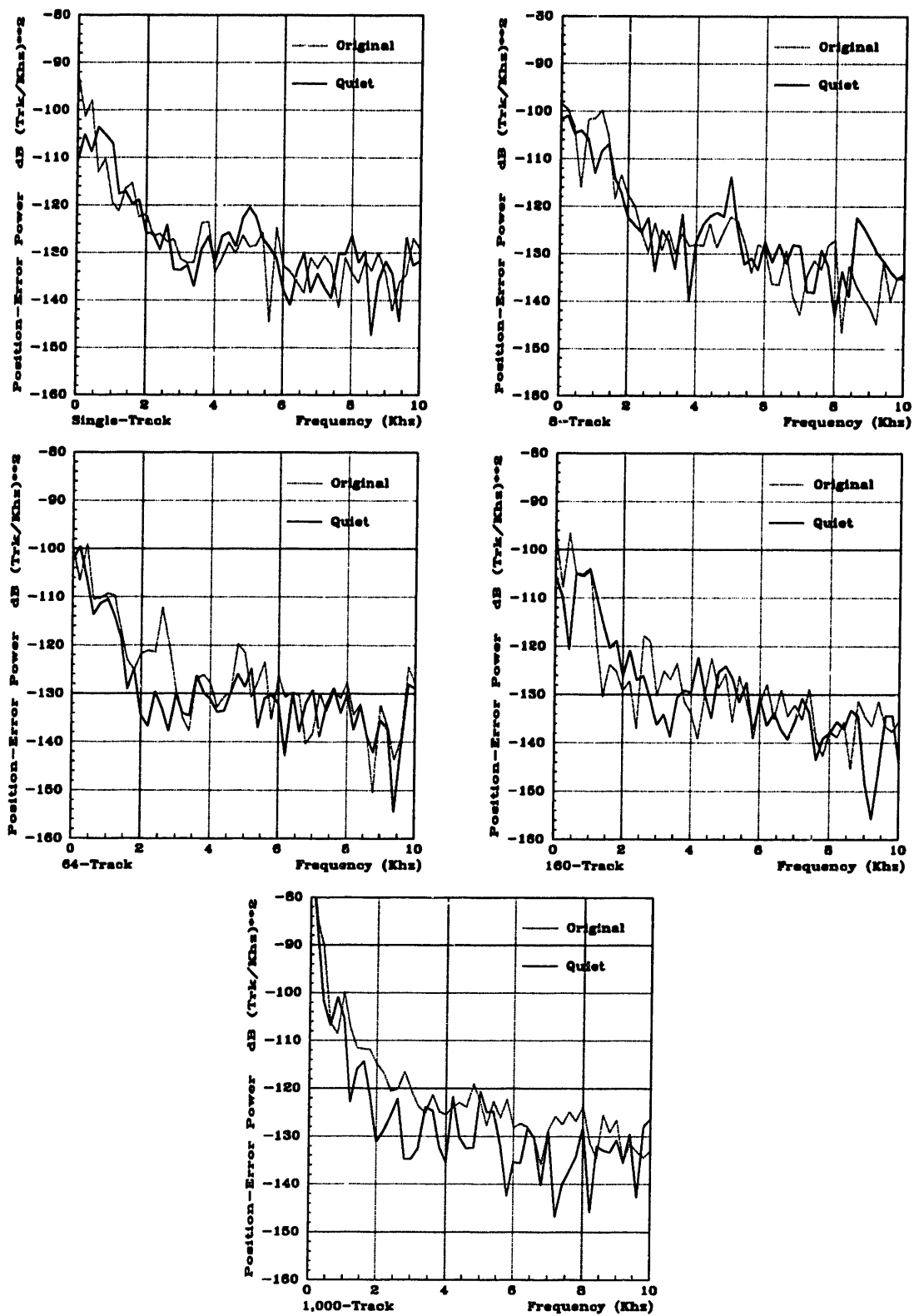


Figure 4-24 Post-Seek Settles, Position-Error Average Power Spectrums, Drive #2

4.5.2 Comparing the Post-Seek Vibration of the Quiet Algorithm and a Surrogate Bang-Bang Controller

A time-optimal, or bang-bang, controller should exhibit faster move times than either of the original or the quiet-seek algorithms; however, it should also result in post-seek vibration. The author attempted to demonstrate this behavior by implementing a bang-bang controller and collecting data. Unfortunately, the RA73's 47 microsecond sample period does not provide sufficient time resolution to switch the control signal at the precise times required to implement a true third-order bang-bang controller¹. Consequently, the author implemented an open-loop controller which displays the significant features of a bang-bang controller. In particular, the open-loop controller drives the power amplifier toward saturation to accelerate and decelerate the actuator. It also drives the power amplifier toward saturation to remove the current from the actuator coil at the end of the seek. Given that the open-loop controller produces move times and post-seek vibration characteristics similar to a bang-bang controller, one may infer that the following results for the open-loop controller apply equally well to a theoretical bang-bang controller. The open-loop controller move times are compared with the quiet-seek algorithm move times in table 4-6.

Move-Time Comparison: Drive #2	Single Track (ms)	Eight Track (ms)	64 Track (ms)
Open-Loop Controller	1.2	1.7	3.2
Quiet-Seek Algorithm	3.0	3.0	4.8
Difference	(1.8)	(1.3)	(1.6)

Table 4-6 Open-Loop vs. Quiet-Seek Move-Time Comparison, Drive #2

The next three pages show comparisons of the actuator voltage, the actuator current, and the position error settle characteristics for single-track, 8-track, and 64-track seeks. Both time-domain data and frequency-domain power spectrums are presented.

¹ A bang-bang controller for a disk-drive actuator applies the full power amplifier voltage to accelerate the actuator, reverses the applied voltage once to decelerate the actuator, and reverses the applied voltage a second time to bring the current in the actuator coil to zero. Current and velocity are zero at the end of the seek, but the position is changed. The times at which the applied voltage is reversed are critical and generally require a continuous control system for an ideal implementation of a bang-bang controller.

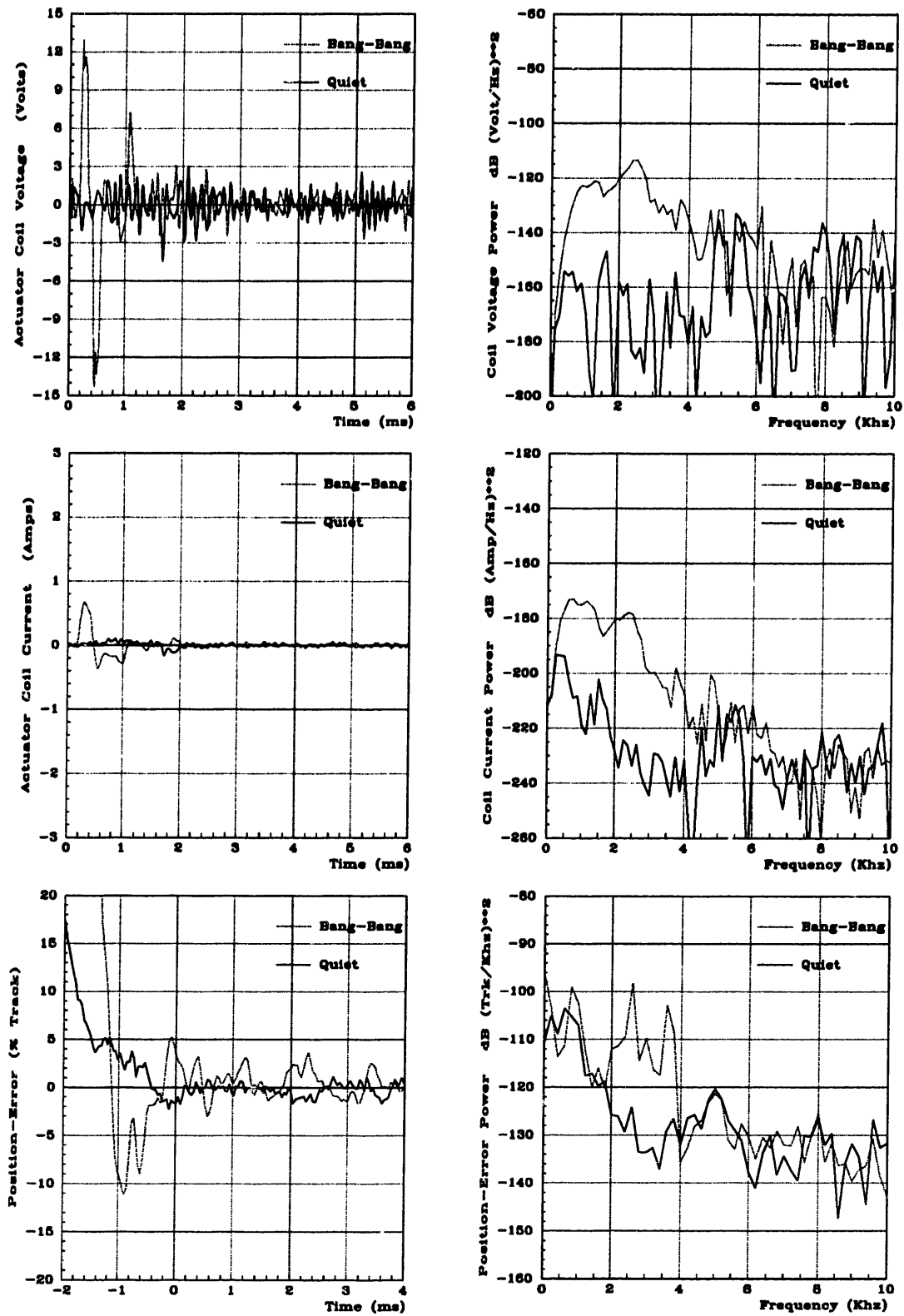


Figure 4-25 Open-Loop vs. Quiet-Seek Algorithm, Single-Track Seek, Drive #2

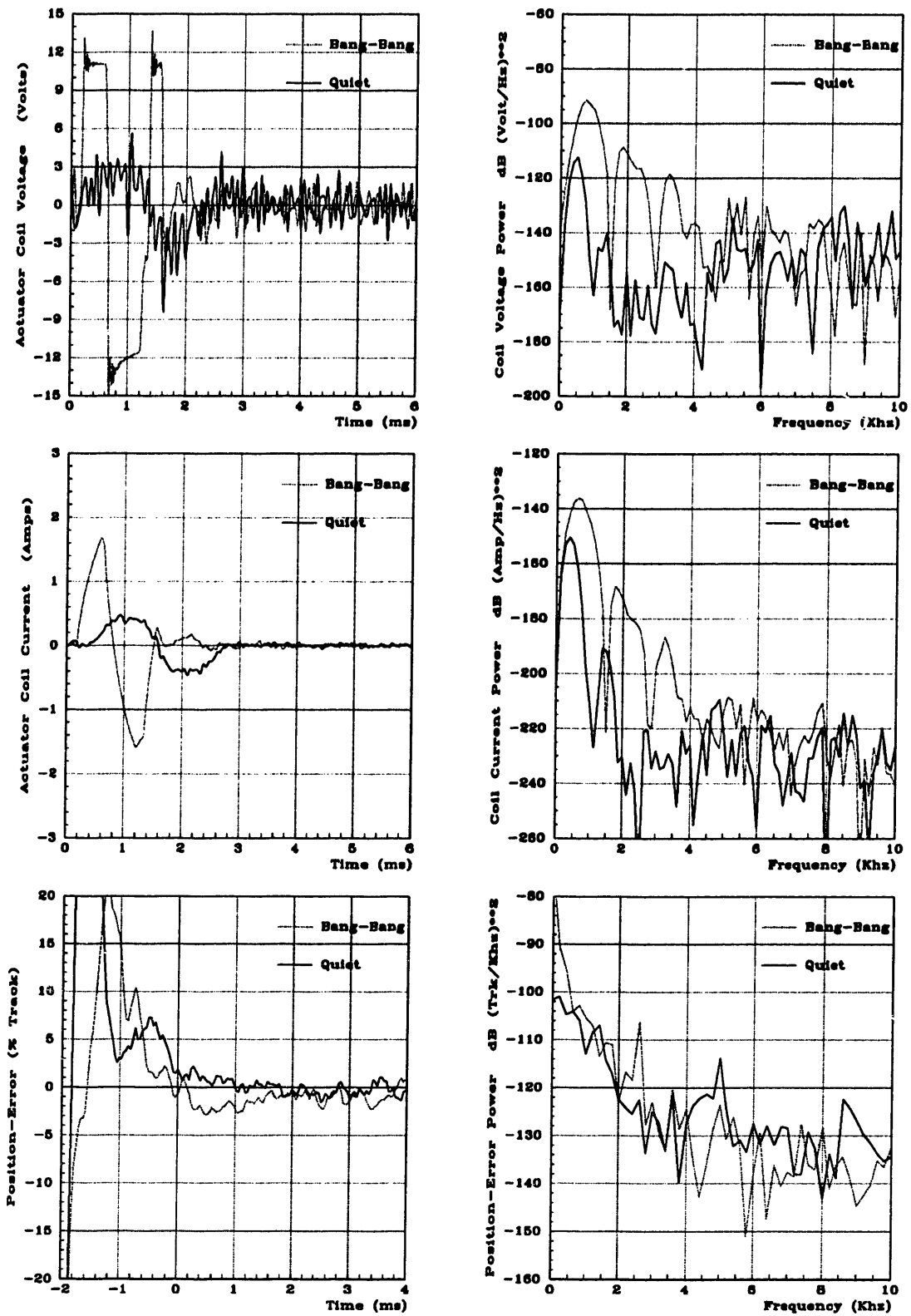


Figure 4-26 Open-Loop vs. Quiet-Seek Algorithm, 8-Track Seek, Drive #2

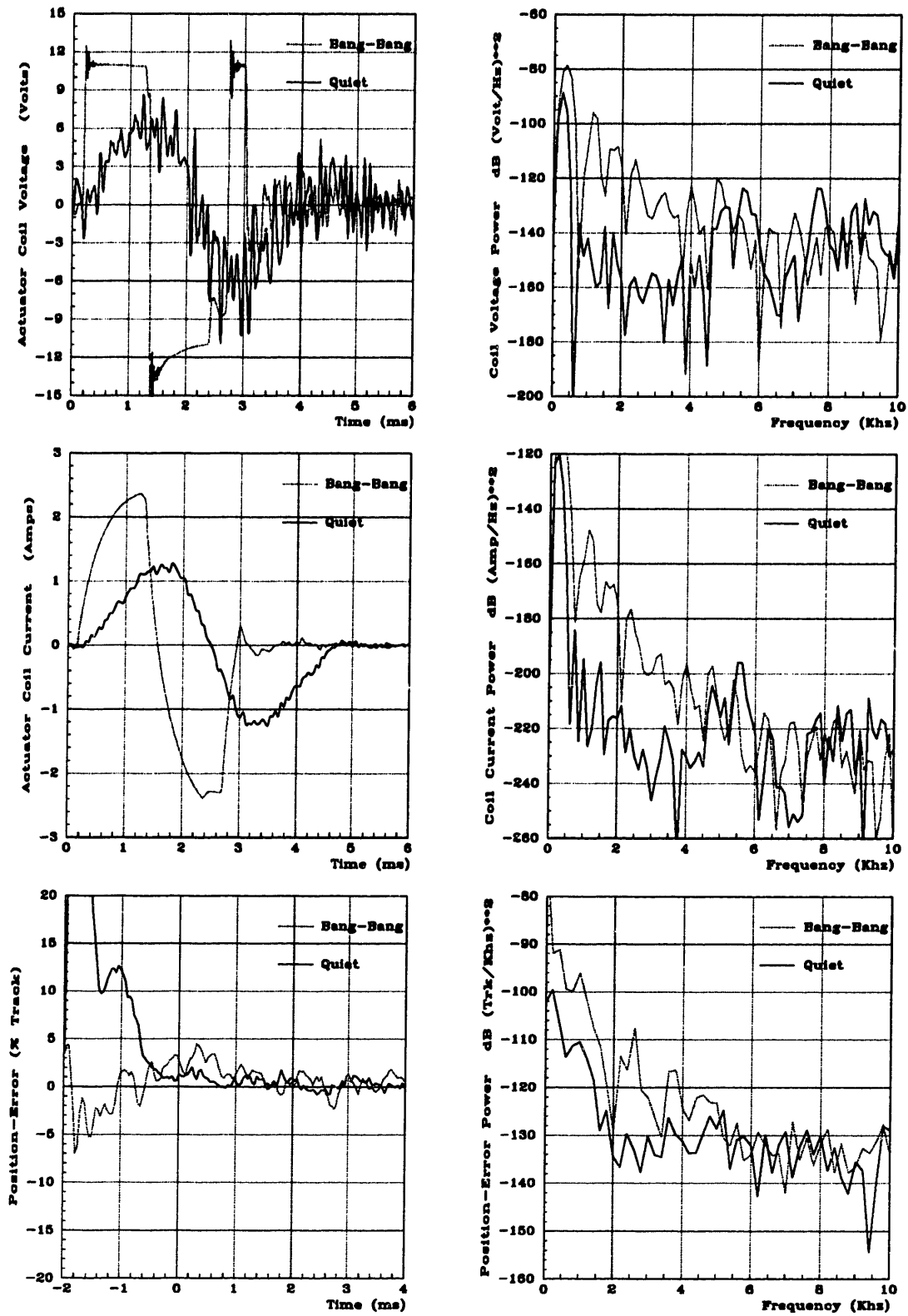


Figure 4-27 Open-Loop vs. Quiet-Seek Algorithm, 64-Track Seek, Drive #2

Comparing the data from the open-loop controller to the data from the quiet-seek algorithm verifies that a controller employing a switched saturated voltage can produce post-seek vibration. The actuator-coil-voltage plots shown in Figures 4-25 through 4-27 show that the open-loop controller employs a switched saturated voltage. The plots also show that the open-loop controller, like a bang-bang controller, causes the voltage to switch polarity twice during the seek. Sharp edges occur in the voltage waveforms as the polarity switches; these sharp edges correspond to broad power spectrums in the frequency-domain. Between 1 and 4 KHz the open-loop voltage power spectrums have 20 to 80 dB more power than the quiet-seek voltage power spectrums. The coil-current spectrums show similar differences. These differences are also reflected in the position-error power spectrums -- increased high-frequency power in the voltage leads to more high-frequency power in the coil current, which in turn leads to more vibration during the settle for the single-track and 64-track seeks.

The single-track settle corresponding to the open-loop controller exhibits significant residual vibration in the time-domain position-error plot (Figure 4-25). Peaks in the time-domain signal are visible for several milliseconds following the track arrival. The position-error power spectrum in Figure 4-25 shows that the bang-bang spectrum is 20 dB higher than the quiet-seek spectrum between 2 and 4 KHz. The same data for the 64-track seek settle (Figure 4-27) indicates a 20 dB difference between the two algorithms in the position-error power from low frequencies up to 5 KHz. Again, the time-domain position-error signal shows peaks lasting several milliseconds following the track arrival. Despite a significant difference in the coil-current power spectrums for the 8-track seek (Figure 4-26), the position-error power spectrums are similar, contrary to expectation. The author found no clear explanation for this similarity.

Comparing the seek-time performance and post-seek vibration of the quiet-seek algorithm to an approximation to a time-optimal algorithm shows that, although the time-optimal algorithm may produce shorter move times by $1\frac{1}{2}$ to 2 milliseconds, it may also produce more residual vibration and longer settle times. The lengthening of the settle times may out-weigh the advantage of the shorter move times.

4.6 Testing and Results -- Conclusions

The quiet-seek algorithm reduces seek-related noise by 11 dB, or a factor of 12, when compared against the RA73's original seek algorithm. This improvement is gained without sacrificing the seek-time performance of the original seek algorithm. The reduced vibration afforded by the quiet-seek algorithm does not improve the post-seek settling characteristics over those obtained using the original seek algorithm.

The author measured the sound pressure produced by the RA73 both as it track-followed and as it sought randomly. The measurements include data corresponding to both the original seek algorithm and the quiet-seek algorithm. The data indicates that seeks implemented with the original seek algorithm increase the RA73's emitted sound power by 600% above the base level produced while track-following. However, the RA73's emitted sound power increases by only 47% above the base level when the quiet-seek algorithm is used to perform the seeks. The data presented in Figures 4-1 through 4-8 supports these conclusions.

The improved noise emission results from the quiet-seek algorithm's attenuation of the average coil-current power between 1 and 4½ KHz, as shown in Figures 4-15 and 4-16. Figure 4-16 shows that the quiet-seek algorithm reduces the average coil-current power by between 10 and 13 dB in the range of 1 to 2½ KHz, and by roughly 5 dB between 2½ and 4½ KHz. This is consistent with the sound-pressure improvement, also shown in Figure 4-16.

Figures 4-17 through 4-21 show that the improvement of the average coil-current power spectrum can be traced to the spectrums of individual current waveforms, which are in turn related to the current waveforms' shapes. The fact that the improvements in the average power spectrum can be traced to the individual spectrums supports a claim put forth in the thesis' introduction: "...the noise increase is not due to the excitation of low-frequency modes by repetitive seeks. Instead, one may conclude that the source of seek noise is the high-frequency content of the coil currents related to individual seeks." This verifies that a successful approach to noise reduction must consider accelerations within individual seeks.

Section 4.4 presents seek-time measurements indicating that the quiet-seek algorithm is slightly faster than the original seek algorithm. Although the original seek algorithm executes single-track seeks in 1¼ milliseconds less time than the quiet-seek algorithm,

it performs all other measured seek lengths 0.2 milliseconds to 0.6 milliseconds more slowly than the quiet-seek algorithm. This implies that the production RA73 disk drive can be made to operate more quietly while still meeting its seek-performance specifications. This does not imply that limiting the frequency content of a command signal results in faster moves. In general, smoothing a command signal to limit its frequency content extends the command's duration by the duration of the smoothing pulse¹. The smoothing pulse used to smooth the 'ideal' waveforms in the development of the long-seek algorithm is 1¾ milliseconds long. Consequently, the quiet-seek algorithm is approximately 1¾ milliseconds slower than an algorithm based upon the 'ideal' accelerations. The difference between 'ideal' move times and the quiet-seek algorithm move times is supported by data presented in table 4-6.

Disk drives demand good post-seek settling characteristics to ensure that data is written in the correct locations on a platter's surface. This requirement led to the original seek algorithm having a smooth, 3½ millisecond, exponential 'tail' at the conclusion of each seek. The smooth tail allows the original seek algorithm to minimize post-seek vibration, as documented in Figure 4-23. However, the tail also causes the algorithm to be 2¼ milliseconds slower than an 'ideal' implementation. The quiet-seek algorithm exhibits smooth movement throughout a seek. Consequently, the quiet-seek algorithm shares the original seek algorithm's excellent settle characteristics. This is also documented in Figure 4-23.

Section 4.5.2 compares both the move times and the post-seek vibration resulting from the quiet-seek algorithm to the same data resulting from an open-loop controller that exhibits the salient features of a time-optimal controller. Specifically, a time-optimal, or bang-bang, controller applies full-scale control voltage to an actuator coil to accelerate an actuator at the onset of a seek. During a seek, a bang-bang controller reverses the polarity of the applied voltage twice; once to decelerate the actuator and then again to eliminate current from the coil. The author compared the quiet-seek algorithm against an open-loop controller exhibiting a switched full-scale control voltage. Figures 4-25 through 4-27 display coil voltages and currents measured while seeking and the position measured while settling. The high-frequency components in the open-loop controller's applied voltage are reflected in the resulting coil current's power spectrum. The coil

¹ Smoothing a signal by convolution with a smoothing pulse extends the signal by the length of the smoothing pulse. If a signal is smoothed by low-pass filtering, then the signal is extended by the length of the filter's impulse response.

current in turn excites vibrations in the actuator which are visible in the position signal as the actuator settles following a seek. The comparisons show that although the open-loop controller results in faster move times, it produces signals containing much more high-frequency power which may in turn cause more post-seek vibration.

Chapter 5: Summary, Conclusions, and Suggestions for Future Work

Acoustic noise emission by disk drives is a distracting and objectionable aspect of today's office environment. This thesis develops a method of reducing the acoustic noise emission from a hard disk drive by shaping the control current to the actuator. State variables trajectories, including position, velocity, and acceleration are shaped so that their audio-frequency content is reduced. This thesis develops two move algorithms; both trajectory following. Measurement data shows that the methods developed in this thesis reduce the seek-related noise power of an RA73 disk drive by a factor of twelve.

The algorithms developed in this thesis have been demonstrated on a Digital Equipment Corporation RA73 disk drive. This disk drive is described in the thesis Introduction. The Introduction also discusses conventional techniques of reducing disk drive noise before proposing command shaping as a method of reducing noise. An overview of disk drive noise emission concludes the Introduction.

The topic of the second chapter is acceleration command shapes. After a brief description of an ideal acceleration waveform, the doublet, the second chapter reviews two methods of shaping a signal to reduce its high-frequency content; low-pass filtering and convolving with a smoothing pulse. Several smoothing pulses are examined. Next, the chapter introduces anti-symmetric acceleration waveforms. Anti-symmetric accelerations can be predictably scaled in both amplitude and duration to implement movements of arbitrary length. The chapter reviews several acceleration waveforms proposed in relevant literature for applications requiring low vibrational excitation. The proposed waveforms are smooth and generally anti-symmetric. The chapter closes by introducing a particular smooth anti-symmetric (SAS) waveform. The SAS waveform describes the accelerations employed by one of the algorithms developed in Chapter 3.

Chapter 3 describes the implementation of two trajectory-following algorithms. The chapter first discusses the trajectory-following controller employed by both algorithms. This controller simultaneously follows both a position and a velocity reference with the aid of a feedforward acceleration command. Simultaneous use of the three command profiles causes an approximate pole-zero cancellation of the closed loop poles of the position-reference to position-sensor function. The resulting controller is capable of

precisely following the references generated by the trajectory-following algorithms. The first algorithm discussed employs the smooth anti-symmetric waveform introduced in Chapter 2 to implement moves which do not require the maximum accelerations afforded by the RA73. This algorithm exploits amplitude and time scaling to modify stored position, velocity, and acceleration profiles, each corresponding to a unit-move. This algorithm is used to execute seeks of lengths between 1 and 143 tracks. The second algorithm presented uses asymmetric waveforms to implement moves requiring nearly the maximum acceleration capabilities of the RA73. The accelerations used by this algorithm account for the effects of the coil's back EMF and inductance. This algorithm constructs unique position, velocity, and acceleration profiles by concatenating segments of several previously-computed profiles and one immediately-calculated segment. This algorithm executes 2,537 seeks of lengths between 144 and 2,680 tracks by amplitude scaling 194 unique profile sets.

The developed algorithms, dubbed the 'quiet-seek algorithm', were tested on a pair of 5¼ inch, RA73 disk drives. Chapter 4 describes the testing and associated results. Specifically, the testing showed that the quiet-seek algorithm reduces the increase in emitted sound power resulting from seek activity from 600% of the drive's baseline noise level to only 47% of the baseline level. This represents a reduction of seek related sound power to one twelfth the original value. It also represents a significant reduction of noise as perceived by the human ear. This improvement was realized without compromising either the seek-time performance or the settling characteristics of the original RA73 seek algorithm. Data showed that the changes in the drive's emitted sound power could be directly related to the modification of the actuator-coil current's power spectrum as a result of command shaping.

Additional to comparing the quiet-seek algorithm with the RA73's original seek algorithm, Chapter 4 compares the quiet algorithm with an open-loop controller which exhibits the salient features of a bang-bang controller. Specifically, the open-loop controller uses a switched full-scale control voltage to drive the actuator current. Measurements indicate that a bang-bang type controller may be capable of achieving shorter average move times, by 1½ to 2 milliseconds. However, increased excitation of vibrational modes resulting from a bang-bang controller's switched voltage waveforms may increase settle times by a similar amount.

Chapter 5 begins with a summary of the thesis. Following the summary, the chapter proceeds to recount the success of the quiet-seek algorithm before identifying the costs

associated with the implementation of the algorithm. Specifically, implementation of the algorithm demands a digital signal processor and 4 kilobytes of memory for microcode and profile storage. Because these requirements may not be met by low-cost disk drive controllers, Chapter 5 submits several suggestions aimed at reducing the resources needed to implement a quiet-seek algorithm.

5.1 Conclusions

The quiet-seek algorithm developed in this thesis and implemented in two RA73 disk drives proves that command-shaping techniques can be successfully employed to reduce the acoustic noise emissions from a hard disk drive. Specifically, the thesis presents measurements establishing that the methods reduce seek-related acoustic noise power emissions by a factor of twelve, which is equivalent to 11 dB. In other terms, the increase in emitted noise power due to seeking is reduced from 600% of the base level of noise generated while the drive track-follows, to 47% of the noise base level. Further, the techniques have been shown to compromise neither the disk drive's seek performance nor the actuator's settling qualities.

The methods introduced in this thesis are founded upon shaping the acceleration profiles of individual seeks. The author pursued this approach based upon the following assumption which was put forth in the Introduction: "Because the increase in the sound pressure occurs at frequencies significantly higher than the seek rate, it is reasonable to assume that the noise increase is not due to the excitation of low-frequency modes by repetitive seeks. Instead, one may conclude that the source of seek noise is the high-frequency content of the coil currents related to individual seeks." The success of the quiet-seek algorithm verifies that the frequency content of accelerations associated with individual seeks must be considered when attempting to reduce seek-related acoustic noise.

The human ear perceives the acoustic noise reduction achieved by the presented methods as a major reduction of the drive's noise output. To demonstrate the improvement realized by the quiet-seek algorithm, the author caused a drive to alternately employ both the original RA73 seek algorithm and the quiet-seek algorithm while randomly seeking. The drive alternated between the two algorithms every 256 seeks; it used one algorithm for several seconds, then switched to the other for several seconds. Approximately one dozen people witnessed the demonstration. All were impressed; both with the dramatic reduction of the drive's acoustic emissions, and by

the fact that such a reduction could be realized without mechanical modifications and without compromising the drive's seek times. The perceived improvement of the drive's noise output may be in part due to the redistribution of the sound power resulting from the use of the quiet-seek algorithm. Figures 4-5 and 4-6 show that the original seek algorithm concentrates seek noise between 1 and 3 Khz; however, the quiet-seek algorithm distributes seek noise more evenly across the audible spectrum. Consequently, the seek noise generated by the quiet-seek algorithm is less tonal than the noise generated by the original seek algorithm, therefore it may also be perceived as less offensive.

The primary penalty paid to reduce the acoustic noise emissions of the RA73 was that the servo controller became more complex. Additional to the increased controller complexity, the quiet-seek algorithm may also have increased the power dissipation within the power amplifier. Most of the remainder of the thesis addresses the issue of controller complexity, however the potential for increased power dissipation is also briefly discussed.

The original seek algorithm is implemented in 450 bytes of microcode instructions. The quiet-seek algorithm requires twice as many instructions, 920 bytes. This figure does not include the tables of sampled trajectories and related information that the quiet-seek algorithm requires. The original seek algorithm uses 256 bytes for tables whereas the quiet-seek algorithm requires 2,750 bytes. The quiet-seek algorithm increases the servo controller's memory requirements by 3 kilobytes. Additionally, the quiet-seek algorithm demands a digital signal processor.

The quiet-seek algorithm is implemented as firmware in a Texas Instrument 320C25 digital signal processor (DSP). This DSP executes a 16-by-16-bit fixed-point multiply/accumulate cycle in 100 nanoseconds. It updates the RA73's power amplifier control signal every 47 microseconds. This DSP has been marketed since 1986 and presently commands a price of approximately \$10 to \$20 when purchased in volume. This device can be classified as a mid-range DSP microprocessor. It is an appropriate selection for the implementation of the quiet-seek algorithm because a low-cost 8-bit controller would lack the instructions and processing speed required by the algorithm. Conversely a 32-bit floating-point DSP would certainly be capable of implementing the short-seek algorithm; however, it would also be significantly more expensive.

Many high-end disk drives possess the resources necessary to implement a form of the quiet-seek algorithm. However, less-expensive disk drives found in PC's and desktop computers have more limited controllers and may be incapable of implementing the quiet-seek algorithm. Unfortunately, the disk drives in PC's and desktop computers are the drives contributing most to office noise. The author offers suggestions in the following section which may reduce the resources required to implement a 'quiet-seek algorithm' and therefore permit its implementation in low-cost disk drives.

5.2 Suggestions for Future Work

There are two issues which must be addressed before the methods developed in this thesis can be applied broadly to solving the acoustic noise problem in low-cost disk drives. The primary concern is that the required controller complexity must be reduced. Additionally, the potential increase in the power dissipation within the power amplifier resulting from non-saturated operation must be studied. These issues are discussed in the order mentioned.

5.2.1 Reducing Controller Complexity

The quiet-seek algorithm developed in this thesis uses a truncated Gaussian pulse and a half-cycle sine pulse to smooth acceleration trajectories. Smoothing the trajectories reduces their high-frequency content and results in quieter drive operation. Quieter drive operation is achieved at the expense of both increased move time and increased controller complexity. Using a rectangular smoothing pulse, rather than the Gaussian or sine pulse, would increase both move time and controller complexity to a lesser extent. This may sacrifice a portion of the noise improvement gained by using the other pulses; however, it may also enable low-cost controllers to implement the quiet-seek algorithm. This section discusses these possibilities.

This section first explains how a rectangular smoothing pulse can reduce the short-seek algorithm's memory requirements. It then suggests a method of computing voltage-limited acceleration signals which, when smoothed, may replace some of the long-seek algorithm's tables. Following the suggestion, it explains how a filter can provide smoothing equivalent to the smoothing achieved by convolution with a rectangular pulse. Finally, it compares the spectrums of a rectangular pulse to the spectrums of both a half-cycle sine pulse and a truncated Gaussian pulse.

Replacing the half-cycle sine pulse with a rectangular pulse in the development of the short-seek algorithm causes an anti-symmetric triangular waveform to replace the SAS waveform. This is shown in Figure 5-1.

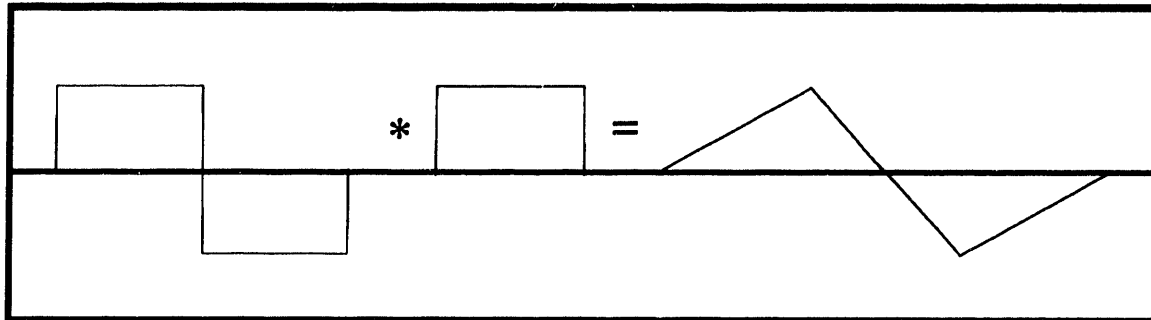


Figure 5-1 Smoothing the Rectangular-Pulse Pair with another Rectangular Pulse

Three segments comprise the triangular acceleration waveform; each segment can be represented by a linear equation. The corresponding velocity and position segments can be represented by second and third-order equations, respectively. Nine low-order polynomials describe the unit-move command references required by the trajectory-following loop. Again, any length seek can be implemented by appropriately scaling the unit-move commands. One may conclude that the short-seek algorithm's trajectory tables can be replaced by nine polynomials. This reduces the quiet-seek algorithm's memory requirements by the size of the tables less the storage required for the polynomial coefficients, or approximately 700 bytes.

Eliminating the short-seek algorithm's tables significantly reduces the quiet-seek algorithm's memory requirements; however the long-seek algorithm requires twice the memory of the short-seek algorithm. Synthesizing the acceleration commands of the long-seek algorithm would lead to substantial memory savings.

The voltage-limited acceleration described in the long-seek algorithm's development is represented by two exponentials in Equation C.3. Each continuous-time exponential can be sampled to produce a discrete-time sequence as shown below:

(5.1) Continuous to Discrete-Time Conversion

$$x(t) = e^{-\frac{t}{\tau}} \rightarrow e^{-\frac{nT}{\tau}} = \left(e^{-\frac{T}{\tau}}\right)^n \rightarrow (X)^n \rightarrow x[n]$$

$$n = 0, 1, 2, \dots$$

The long-length seek algorithm performs seeks lasting several hundred samples. Consequently, 'n' ranges from 0 to several hundred and, as a result, direct calculation of $x[n]$, or X^n , would require several hundred multiplications. The DSP cannot execute several hundred multiplications and perform other functions within its sample period. However, X^n is the product of factors such as $X^0, X^1, X^2, X^4, X^8, \dots, X^{2^k}$, etc. The DSP can readily execute the multiplications required to form X^n from factors with form X^{2^k} within one sample period. Implementing this method requires the DSP to conditionally multiply a sub-product by a factor, X^{2^k} , on the condition that the k'th bit of n's binary representation is a one and not a zero. For example, if 'n' equals five, then n's binary representation is '000101,' which shows bit-0 and bit-2 as ones. Therefore the DSP must compute the product of the factors X^{2^0} and X^{2^2} , or the product of X^1 and X^4 . The product of X^1 and X^4 is X^5 , or $x[5]$, as demanded. This algorithm requires that roughly ten factors be stored in memory for each exponential modeled. This method allows the DSP to compute the discrete-time values corresponding to an acceleration trajectory described by exponentials. It also allows the computation of the velocity and position trajectories as explained in the following paragraph.

The voltage-limited acceleration is described by a pair of exponentials, consequently the corresponding velocity and position are described by the integrals of a pair of exponentials. Linearity permits integration of each exponential separately. The first integration of an exponential results in the same exponential scaled by its time constant and a constant term. Therefore each component of the velocity is related to its corresponding acceleration component by a scale factor and a constant. A second integration produces position components. Each component of the position is related to its corresponding acceleration component by a scale factor, a constant, and a linear time term. Calculating the voltage-limited reference commands associated with Equation C.3 (Simplified RA73 Acceleration Model) involves only scaling and addition. Exponentials are scaled by time constants, linear coefficients are scaled by 'n', and constant terms are added to the former terms. Most microprocessors are capable of these computations.

The deceleration segment is modeled as a constant-level deceleration in the long-seek algorithm. Constant-level deceleration results in a linear velocity expression and a position described by a quadratic function. Again, these expressions can be evaluated by most microprocessors.

The author has shown that a microprocessor may be able to synthesize the reference commands corresponding to the voltage-limited accelerations put forth in section 3.3.3 without using tables. However, an algorithm for quiet seeks demands smoothed references. Smoothing can be accomplished by either convolving with a smoothing pulse or filtering.

Smoothing a signal by convolution with a rectangular smoothing pulse is equivalent to integrating the difference between the signal and a delayed version of itself. If the smoothing pulse has unit area, then the integration is necessarily scaled by the reciprocal of the delay time. The delay time equals the duration of the smoothing pulse. Varying the delay effectively varies the width of the smoothing pulse. Consequently, varying the delay also varies the degree of smoothing and the spectrum of the smoothed signal. This technique permits an engineer to balance move-time extension against noise emission by varying a single delay within an algorithm.

Equation 5.2 is a continuous-time representation of this smoothing filter. The reader may recognize the representation as the Laplace description of a zero-order hold.

$$H(s) = \frac{1-e^{-sT}}{sT} \quad (5.2) \text{ Continuous-Time Rectangular Smoothing Filter}$$

Where 'T' is the delay time.

Equation 5.3 shows a discrete-time representation of this filter.

$$H(z) = \frac{1}{N} \frac{1-z^{-N}}{1-z^{-1}} \quad (5.3) \text{ Discrete-Time Rectangular Smoothing Filter}$$

Where 'N' is the number of samples of delay.

Methods discussed in this section may permit a DSP to synthesize the reference commands required by the long-seek algorithm, rather than construct the commands from stored trajectories. This would significantly reduce the memory required to implement a quiet-seek algorithm. Further, smoothing command signals by filtering with a variable delay time may permit seek time to be balanced against noise production in a 'programmable' fashion. Although smoothing with a rectangular pulse (or equivalent filter) offers several advantages, it does so at the expense of increased high-frequency power in the command signals. This is illustrated in the remainder of the chapter.

Figure 2-1 compares both a truncated Gaussian and a half-cycle sine pulse and their spectrum magnitudes. Figure 5-2 presents a rectangular pulse and its spectrum magnitude. Each pulse has a one millisecond duration and unit area. Of the compared pulses, the rectangular pulse has both the lowest peak amplitude and the greatest high-frequency content.

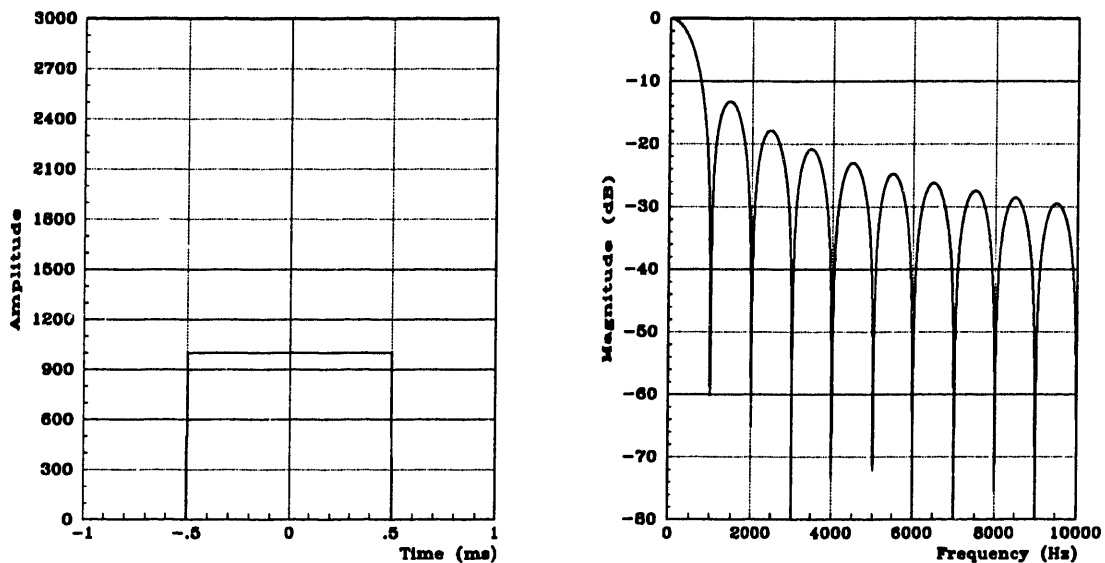


Figure 5-2 One Millisecond Rectangular Pulse and Spectrum Magnitude

Figure 5-3 compares the spectrum magnitudes of the pulses employed by the quiet-seek algorithm (a 1 millisecond duration half-cycle sine and a $1\frac{3}{4}$ millisecond duration truncated Gaussian) with the spectrum of a $\frac{1}{2}$ millisecond rectangular pulse. The rectangular pulse has the greatest high-frequency content. However, it offers low energy near 2 KHz. Frequencies near 2 KHz are strongly represented in the RA73's sound-power spectrum. A half-millisecond rectangular smoothing pulse may yield acceptably quiet operation while reducing the quiet-seek algorithm's move times by $\frac{1}{2}$ to $1\frac{1}{4}$ milliseconds.

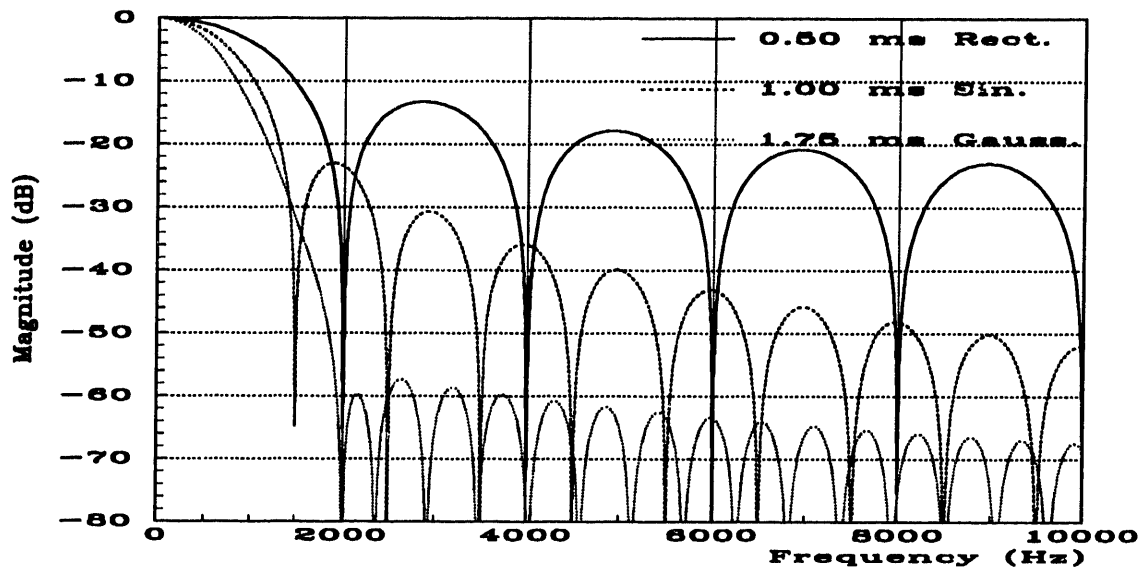


Figure 5-3 Smoothing-Pulse Spectrum Comparison

Several suggestions for reducing the required controller complexity have been put forth. Although controller complexity may be the main factor restricting the implementation of the quiet-seek algorithm, increased power dissipation within the power amplifier is also a concern.

5.2.2 Power Amplifier Dissipation

The quiet-seek algorithm generally demands that the control effort never saturates. Specifically, the power amplifier transistors must not saturate. This is necessary to ensure that the trajectory following-controller can at all times precisely follow the reference profiles. The actuator's acceleration is only guaranteed to be smooth, and therefore quiet, if the actuator is able to follow the profiles. Because the quiet-seek algorithm avoids saturating the power amplifier transistors, it may increase the power dissipation and heating within these devices. Both the quiet-seek algorithm and the original RA73 seek algorithm avoid saturating the power amplifier during the deceleration period of a seek; however, during the acceleration period the original algorithm generally saturates the power amplifier. The saturated power amplifier applies the maximum control voltage to the actuator coil. This is shown in Figure 5-4 which presents the actuator-coil voltage corresponding to a 1,000 track seek as executed by the original seek algorithm. Figure 5-5 presents the same information corresponding to a seek executed by the quiet-seek algorithm.

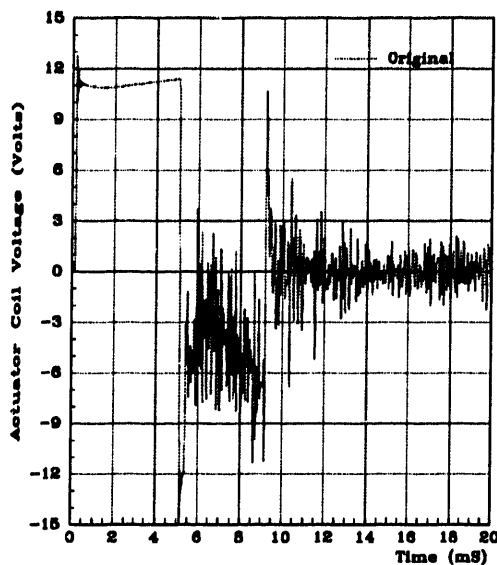


Figure 5-4 Original Algorithm
Coil Voltage, 1,000-Track Seek

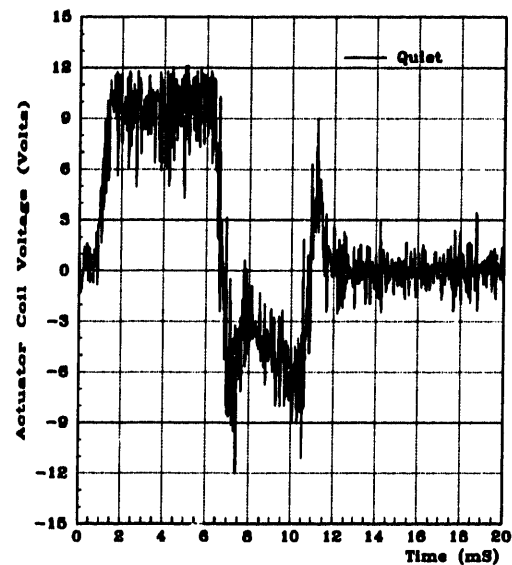


Figure 5-5 Quiet-Seek Algorithm
Coil Voltage, 1,000-Track Seek

Saturating the power amplifier transistors minimizes the voltage across them. The voltage across the power devices may be inferred from the voltage across the coil. Specifically, one may subtract the coil voltage from the voltage supplying the power amplifier's transistors to determine the voltage across the transistors. Figures 5-6 and 5-7 compare the total voltage magnitude across the transistors given a supply voltage at the amplifier of $11\frac{3}{4}$ volts.

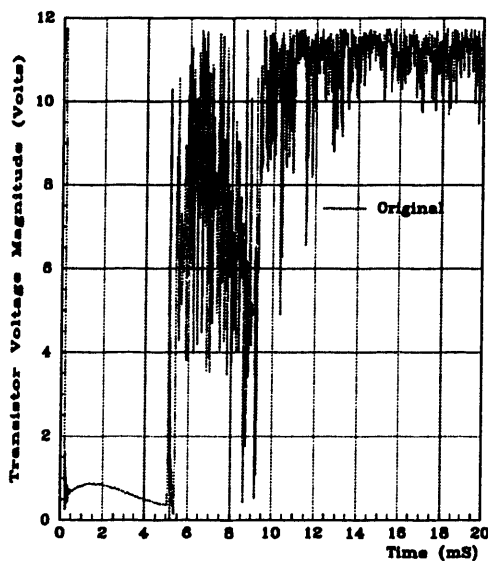


Figure 5-6 Original Algorithm
Transistor Voltage, 1,000-Track Seek

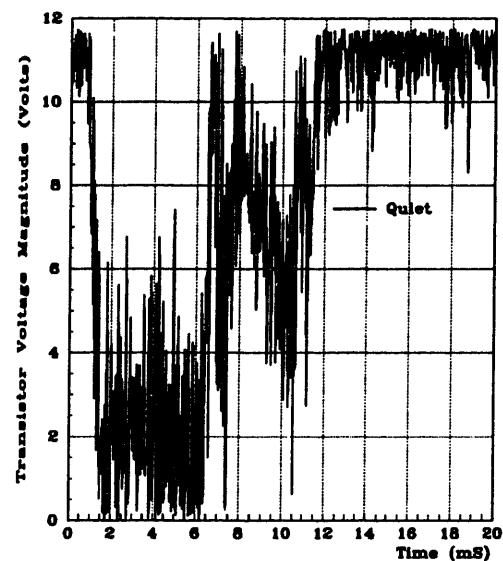


Figure 5-7 Quiet-Seek Algorithm
Transistor Voltage, 1,000-Track Seek

Note that the voltage across the saturated transistors, shown in Figure 5-6, never exceeds 1 volt. However, the voltage across the transistors during the corresponding time period is nearly 2 volts when the quiet-seek algorithm is employed, as shown in Figure 5-7. Consequently, one may at first suspect that the quiet-seek algorithm causes the transistors' power dissipation to double during the acceleration period, and therefore increase by approximately 50% when the entire seek is considered. The following analysis shows this assumption to be incorrect.

Minimizing the voltage across the transistors also minimizes the power dissipated by them at a given current. However, the current waveforms are different between the two algorithms. A calculation of the power dissipated by the transistors must account for the differences between the coil currents. The power dissipated by the transistors is determined by the product of the voltage across the transistors and the current through them. Figures 5-8 and 5-9 present the power dissipated by the transistors calculated as the product of the magnitude of the voltage across the transistors with the magnitude of the current through the transistors, for data corresponding to each algorithm.

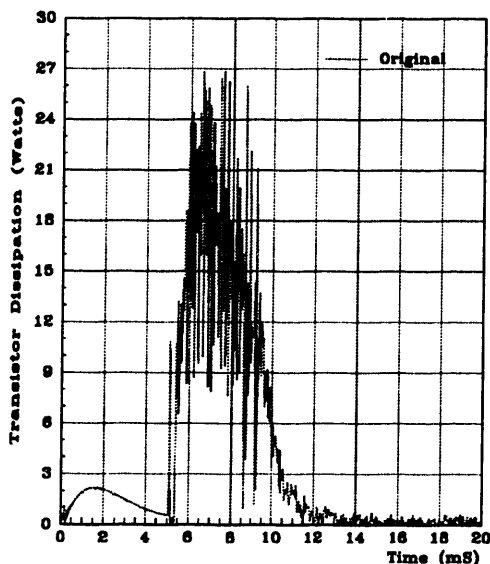


Figure 5-8 Original Algorithm
Transistor Power Dissipation,
1,000-Track Seek

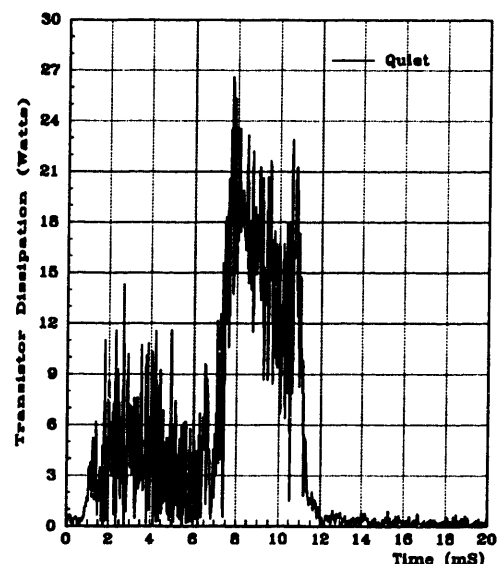


Figure 5-9 Quiet-Seek Algorithm
Transistor Power Dissipation,
1,000-Track Seek

The figures show that there are significant differences in the power dissipation during the acceleration period between the two algorithms. However, the figures also show that the power dissipated during the acceleration period is dominated by the power dissipated during the deceleration period. Consequently, the difference resulting from

the algorithms between the average power dissipation over the seek is not as significant as one might at first suspect. The average transistor power dissipation during the 12 millisecond period of the seek is 6.4 watts when the original seek algorithm is employed and 7.0 watts when the quiet-seek algorithm is employed. This analysis has shown that the quiet-seek algorithm increases the power dissipation of the transistors during a 1,000-track seek by 10 percent.

The foregoing analysis showed that the quiet-seek algorithm marginally increases the power dissipation within the power amplifier as a result of the requirement that the power devices not saturate. However, the analysis considered only one seek length. Further, the fact that the power dissipation is dominated by the dissipation during the deceleration period results from the current-limiting feature of the RA73 electronics. This feature may be peculiar to the RA73, consequently the power dissipation may not be dominated by the dissipation during the deceleration period when other drives are considered. The quiet-seek algorithm may cause a greater percentage increase in the power devices' dissipation in other drives. This issue merits further study

Appendix A

The original seek algorithm in the RA73 regulates the velocity of the actuator as a function of the distance to the target; consequently, the controller's command signals, $V_{cmd}[x]$ and $A_{cmd}[x]$, are shown as functions of the position signal, x , in Figure 1-4. Because both $V_{cmd}[x]$ and $A_{cmd}[x]$ are non-linear functions of the distance to the target, or position-error, the controller is also non-linear.

The velocity reference, $V_{cmd}[x]$, is implemented as a function which; is constant when the actuator is greater than 946 tracks from the target track, varies as the square root of the distance to the target for distances between 946 and 19.3 tracks, and is directly proportional to the remaining distance for distances less than 19.3 tracks. Figure A-1 shows the original RA73 algorithm's velocity reference as a function of the distance to the target track.

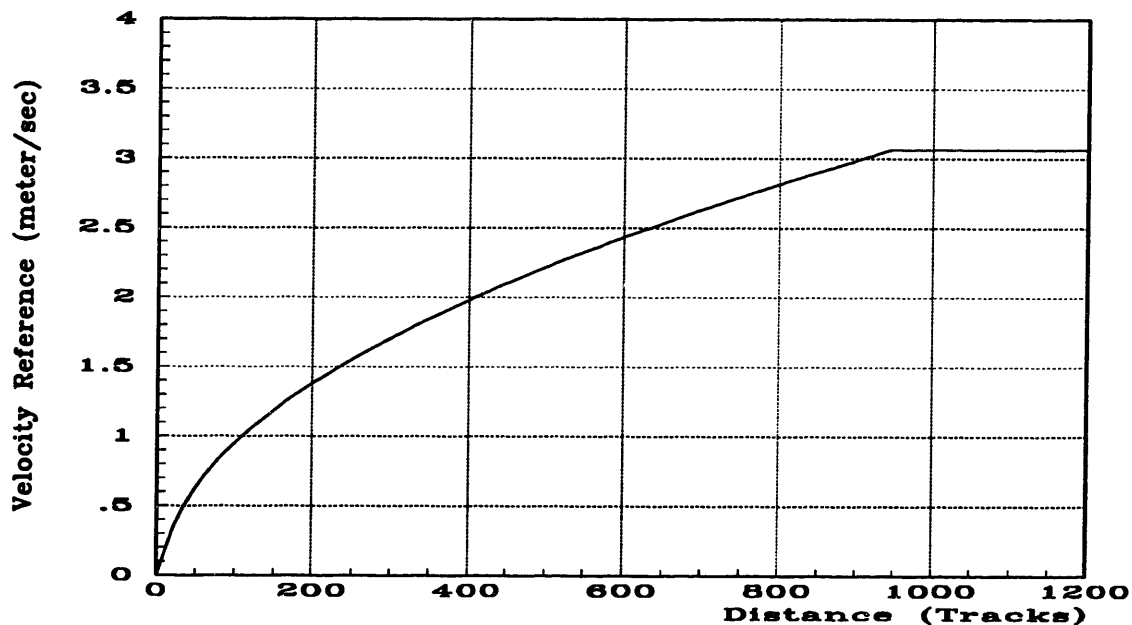


Figure A-1 RA73 Velocity Reference as a Function of Distance to Target

The velocity reference is computed from one of a series of 32, second-order polynomials. Each polynomial defines the velocity reference over a limited range of distances from the target track. The servo controller selects the polynomial coefficients based upon the distance remaining to the target track. The three coefficients for the polynomial are selected from three-32 element tables. Effectively, the tables contain $v|_{x_n}$, $dv/dx|_{x_n}$, and $d^2v/dx^2|_{x_n}$, where x_n is one of 32 stored distances.

Equation A.1 shows the second-order interpolation used to define the velocity reference.

(A.1) Original RA73 Seek Algorithm Velocity Reference

$$V_{cmd} = \left(v|_{x_n} \right) + \left(\frac{dv}{dx} \Big|_{x_n} \right) (x - x_n) + \frac{1}{2} \left(\frac{d^2v}{dx^2} \Big|_{x_n} \right) (x - x_n)^2$$

For distances to the target greater than 946 tracks, the commanded velocity is a constant 3.065 meters per second. Accordingly, dv/dx and d^2v/dx^2 are both zero at the corresponding table entries. This is reflected in Equation A.2.

(A.2) Velocity Reference, Position > 946 Tracks

$$V_{cmd} = \left(v|_{946} \right) = 3.065 \frac{\text{meters}}{\text{second}}$$

For distances to the target less than 19.3 tracks, the velocity command is directly proportional to the distance to the target by 16.1×10^{-3} meters per second per track. For short distances, both v and d^2v/dx^2 are zero. This is shown in Equation A.3.

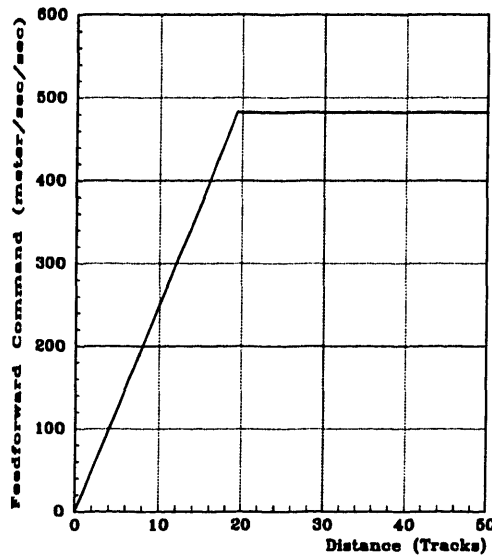
(A.3) Velocity Reference, Position < 19.3 Tracks

$$V_{cmd} = \left(\frac{dv}{dx} \Big|_0 \right) (x - 0) = \left(16.1 \times 10^{-3} \frac{\text{meters / second}}{\text{track}} \right) (x)$$

The proportionality constant, $dv/dx|_0$, can be expressed as 1,567 meters per second per meter, or 1,567 radians per second which is equivalent to 249 Hz. This proportionality constant effectively provides proportional position feedback within the control loop. Therefore the loop is a second-order, PD compensated, type-II loop. Further, although it is not apparent from this analysis, the $dv/dx|_0$ proportionality constant sets a zero location in the open loop transfer function at 249 Hz.

The proportionality constant, $dv/dx|_0$, which relates the actuator's commanded velocity to its position, can also be viewed as the reciprocal of the time constant in the following first-order differential equation: $dx/dt = (-1/\tau) x$, where τ equals 0.64 milliseconds. This differential equation implies that the actuator's velocity and position are both described by a decaying exponential with time constant equal to 0.64 milliseconds when the actuator is within 19.3 tracks of the target and the actuator is moving at the commanded velocity.

Feedforward is the second component of the control signal. The feedforward control produces a coil current which nominally corresponds to the deceleration required by the changing velocity command. The required deceleration rate is a constant level of 482 meters per second-squared between distances to the target of 19.3 and 946 tracks. This figure decreases linearly to zero for shorter distances to the target. The RA73's feedforward command is described by Figure A-2 and Equation A.4.



$$|a| = 482 \frac{m}{sec^2} \quad \text{for } x > 19.3 \text{ tracks}$$

$$|a| = 25.0 \frac{m/sec^2}{track} \quad \text{for } x \leq 19.3 \text{ tracks}$$

(A.4) Feedforward Definition

Figure A-2 Feedforward Profile

Deceleration does not occur for distances greater than 946 tracks; irrespective of this fact, the controller applies constant-level feedforward control. This is because switching feedforward 'on' abruptly causes large transients in the control loop. The abrupt application of the constant-level feedforward is equivalent to applying a large step in torque to the velocity regulator. Additionally, leaving the feedforward active during the first half of a long seek does not slow the initial acceleration. This is because the feedback component dominates the feedforward component at large distances.

The proportionality constant of 25.0 meters per second-squared per track can also be expressed as the quantity, 1,567 radians per second, squared. This is equivalent to 249 Hz squared. Like the velocity command, the feedforward signal is related to the position signal by a differential equation; $d^2x/dt^2 = (-1/\tau)^2 x$. Consequently, both the deceleration and the actuator coil current are described by the same exponential decay that describes the actuator position and velocity. Specifically, both the deceleration and the current exhibit exponential decay with a time constant equal to 0.64 milliseconds. This behavior is shown for a 1,000 track seek in Figures A-3 and A-4.

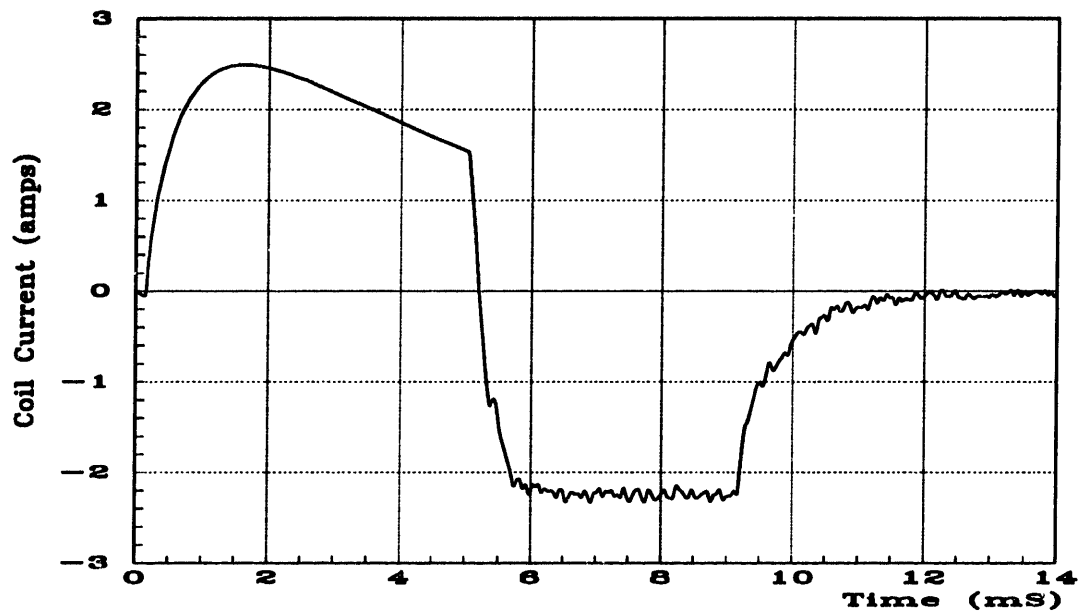


Figure A-3 Original Algorithm, Actuator-Coil Current, 1,000 Track Seek

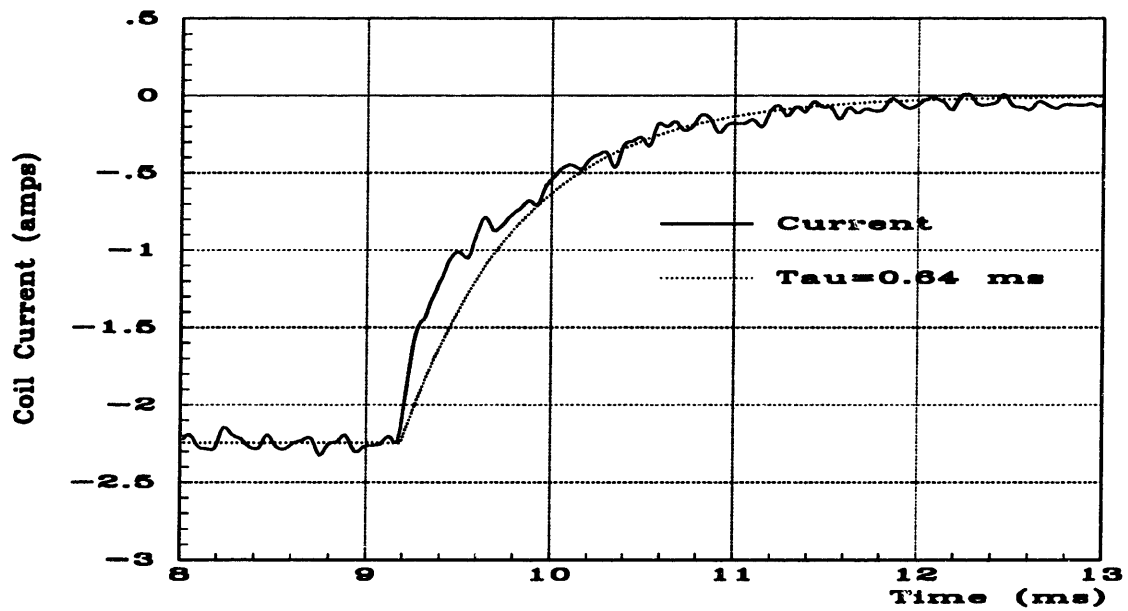


Figure A-4 Original Algorithm, Tail Current, 1,000 Track Seek

Appendix B

Equations B.1 and B.2 show the Fourier Transforms of the unit-area sine pulse and the unit-area raised-cosine pulse. In both cases the variable ' T ' represents the pulse width. The smoothing properties of these pulses are compared with the smoothing properties of a truncated Gaussian pulse in Section 2.2.

$$\frac{\cos(\pi fT)}{1 - (2fT)^2} \quad \text{(B.1) Fourier Transform of Unit Area Sine Pulse}$$

$$\frac{\sin(\pi fT)}{\pi fT (1 - (fT)^2)} \quad \text{(B.2) Fourier Transform of Unit Area Raised Cosine Pulse}$$

Appendix C

Equation C.1 derives a Laplace expression for the acceleration response of the servo sensor to a step of E volts applied to the actuator coil at the time origin. The equation is based upon the assumptions that the actuator is initially at rest and that the coil is initially discharged. This situation corresponds to the acceleration of the servo sensor at the onset of a movement in which the accelerating current is limited only by the power amplifier's maximum voltage, the coil's inductance, and the back EMF generated within the moving coil.

$$\begin{aligned}
 A(s) &= \frac{A(s)}{E(s)} \times E(s) & (C.1) \text{ Voltage-Limited Acceleration Derivation} \\
 &= \frac{\left(\frac{1}{Ls+R}\right) K \left(\frac{1}{j}\right)}{1 + \left(\frac{1}{Ls+R}\right) K^2 \left(\frac{1}{js}\right)} l \times \frac{E}{s} \\
 &= \frac{Ks}{JLs^2 + JRs + K^2} l \times \frac{E}{s} \\
 &= \frac{\left(\frac{K^2}{JR}\right) \left(\frac{R}{L}\right)}{s^2 + \frac{R}{L}s + \left(\frac{K^2}{JR}\right) \left(\frac{R}{L}\right)} \frac{El}{K} \\
 &= \frac{\frac{1}{\tau_m \tau_e}}{s^2 + \frac{s}{\tau_e} + \frac{1}{\tau_m \tau_e}} \frac{El}{K} \quad \text{where } \tau_m = \frac{JR}{K^2}, \text{ and } \tau_e = \frac{L}{R} \\
 &= \frac{\frac{1}{\tau_m \tau_e}}{\left[s + \frac{1}{2\tau_e} \left(1 - \sqrt{1 - 4\frac{\tau_e}{\tau_m}}\right)\right] \left[s + \frac{1}{2\tau_e} \left(1 + \sqrt{1 - 4\frac{\tau_e}{\tau_m}}\right)\right]} \frac{El}{K} \\
 &= \left(\frac{\frac{1}{\tau_m \sqrt{1 - 4\frac{\tau_e}{\tau_m}}}}{\left[s + \frac{1}{2\tau_e} \left(1 - \sqrt{1 - 4\frac{\tau_e}{\tau_m}}\right)\right]} - \frac{\frac{1}{\tau_m \sqrt{1 - 4\frac{\tau_e}{\tau_m}}}}{\left[s + \frac{1}{2\tau_e} \left(1 + \sqrt{1 - 4\frac{\tau_e}{\tau_m}}\right)\right]} \right) \frac{El}{K}
 \end{aligned}$$

Equation C.2 is a time-domain representation of the results that appear in Equation C.1.

(C.2) Voltage-Limited Acceleration Model

$$A(t) = \frac{1}{\tau_m \sqrt{1 - 4\frac{\tau_e}{\tau_m}}} \left(e^{-\frac{t}{2\tau_e}} \left(1 - \sqrt{1 - 4\frac{\tau_e}{\tau_m}}\right) - e^{-\frac{t}{2\tau_e}} \left(1 + \sqrt{1 - 4\frac{\tau_e}{\tau_m}}\right) \right) \frac{El}{K}; \quad t \geq 0$$

Approximating the square-root quantity in Equation C.2 further simplifies the voltage-limited acceleration model. The simplified model is shown in Equation C.3. The values used to evaluate the expressions correspond to the RA73 and are listed in Table 3-2. This model's simplicity may contribute to one's understanding of the relationship between the acceleration behavior and the system's parameters.

Assuming, $1 \gg 4\frac{\tau_e}{\tau_m}$,

Then, $\sqrt{1 - 4\frac{\tau_e}{\tau_m}} \approx 1 - 2\frac{\tau_e}{\tau_m} \approx 1$

For: $\tau_m = \frac{JR}{K^2} = 6.8 \text{ milliseconds}$, and $\tau_e = \frac{L}{R} = 0.46 \text{ milliseconds}$

$\Rightarrow 4\frac{\tau_e}{\tau_m} = 0.268$, $\sqrt{1 - 4\frac{\tau_e}{\tau_m}} = 0.856$, $1 - 2\frac{\tau_e}{\tau_m} = 0.866$

$$A(t) = \frac{1}{\tau_m \sqrt{1 - 4\frac{\tau_e}{\tau_m}}} \left(e^{-\frac{t}{2\tau_e}} \left(1 - \sqrt{1 - 4\frac{\tau_e}{\tau_m}} \right) - e^{-\frac{t}{2\tau_e}} \left(1 + \sqrt{1 - 4\frac{\tau_e}{\tau_m}} \right) \right) \frac{El}{K}; \quad t \geq 0$$

$$= \left(e^{-\frac{t}{6.3 \text{ milliseconds}}} - e^{-\frac{t}{0.49 \text{ milliseconds}}} \right) \times 660 \frac{\text{meters}}{\text{second}^2}$$

$$A(t) \approx \left(e^{-\frac{t}{\tau_m}} - e^{-\frac{t}{\tau_e}} \right) \frac{EKl}{RJ}; \quad t \geq 0 \quad \text{(C.3) Simplified RA73 Acceleration Model}$$

$$= \left(e^{-\frac{t}{6.8 \text{ milliseconds}}} - e^{-\frac{t}{0.46 \text{ milliseconds}}} \right) \times 570 \frac{\text{meters}}{\text{second}^2}$$

The simplified model shows that the acceleration behavior of the actuator is governed by an electrical time constant (L/R) and a mechanical time constant (JR/K^2) when a constant voltage is applied to the actuator-coil terminals. The acceleration is proportional to the maximum control voltage, E , the torque constant, K , and the arm length, l . It is inversely proportional to the coil resistance, R , and the actuator inertia, J .

The simplified model shown in Equation C.3, based upon two time constants and a scale factor, is easily adjusted to reflect either measured or desired accelerations. It can be used for development without precise measurements of the underlying system parameters.

References

Leo L. Beranek and Istvan L. Ver, "Noise and Vibration Control Engineering, Principles and Applications," John Wiley & Sons Inc., 1992.

Brüel and Kjær, "2032 Instruction Manual, Volume 2, Operation," September 1983, pp. 151-152.

European Computer Manufacturers Association (ECMA), "Standard ECMA-109, Declared Noise Emission Values of Computer and Business Equipment," 2nd Edition - December 1987.

R. W. Hamming, "Digital Filters," Third Edition, Prentice Hall Signal Processing Series, Prentice Hall Inc., Englewood Cliffs, New Jersey 07632, 1989.

Cyril M. Harris, "Handbook of Acoustical Measurements and Noise Control," Third Edition, McGraw-Hill Inc., 1991.

Susumu Hasegawa, Kazuhiko Takaishi, and Yoshifumi Miozoshita, "Digital Servo Control for Head-Positioning of Disk Drives," Fujitsu Scientific and Technical Journal, Volume 26, Number 4, February 1991, pages 378-390.

L. F. Luttrell and E. K. Dunens, "Noise Sources in Disk Drives," Proceedings of the 1987 National Conference on Noise Control Engineering, Jiri Tichy, Ed. (Institute of Noise Control Engineering, Poughkeepsie, NY, 1987) pp. 141-146.

Peter Heinrich Meckl, "Minimizing Residual Vibration of a Linear System Using Appropriately Shaped Forcing Functions," Massachusetts Institute of Technology, May 1984, pp. 31-32.

Alan V. Oppenheim and Ronald W. Schaffer, "Discrete-Time Signal Processing," Prentice Hall Signal Processing Series, Prentice Hall Inc., Englewood Cliffs, New Jersey 07632, 1989.

M. G. Prasad, D. Quinlan and G. Elko, "Noise and Vibration Studies of Small Winchester Disk Drives," Proceedings of the 1987 National Conference on Noise Control Engineering, Jiri Tichy, Ed. (Institute of Noise Control Engineering, Poughkeepsie, NY, 1987) pp. 147-152.

John A. Scaramuzzo Jr., "Band Limited Control of a Flexible Structure Using Piecewise Trigonometric Forcing Functions," Harvard University, May, 1992.

William M. Siebert, "Circuits, Signals, and Systems," The MIT Press, and McGraw-Hill Book Company, 1986.

Neil Cooper Singer, "Residual Vibration Reduction in Computer Controlled Machines," Massachusetts Institute of Technology, 1989.

Gilbert Strang, "Introduction to Applied Mathematics," Wellesley-Cambridge Press, Wellesley, Massachusetts, 1986.

W. Tempest, "The Noise Handbook," Academic Press Inc., Orlando Florida, 1985.

Texas Instruments Incorporated, "TMS320C25 User's Guide, Preliminary," 1986.

Larry Wittig and Chuck Jokel, PAG #4872, (an interoffice memorandum to Don Arthur concerning RA72 noise levels), May 17, 1991.

C. M. Woldemar and A. Kumano, "Noise Control of Disk Drives for Desk-Top Computers," Proceedings of the 1987 National Conference on Noise Control Engineering, Ilene J. Busch-Vishniac, Ed. (Institute of Noise Control Engineering, Poughkeepsie, NY, 1990) pp. 89-94.