This is to certify that the

thesis entitled

MODELING CONDUCTED EMISSION
TRANSIENTS DUE TO DC MOTOR SWITCHING
IN AUTOMOTIVE APPLICATIONS

presented by

Matthew Raymond Feusse

has been accepted towards fulfillment
of the requirements for

M.S. degree in Electrical Engineering

Edward J Rothwell
Dennis P. Nyquist
Major professor

Date 4-11-01

MODELING CONDUCTED EMISSION TRANSIENTS DUE TO DC MOTOR SWITCHING IN AUTOMOTIVE APPLICATIONS

By

Matthew Raymond Feusse

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Electrical Engineering

2001

ABSTRACT

MODELING CONDUCTED EMISSION
TRANSIENTS DUE TO DC MOTOR SWITCHING
IN AUTOMOTIVE APPLICATIONS

By

Matthew Raymond Feusse

Conducted emissions are currents that exit a device via its power harness. They are undesirable as they can couple to other devices or cause unwanted radiation. DC motors are common culprits for generating conducted emissions in an automobile, as they generate a large transient current when they are switched on or off. Some of this transient current is introduced onto the common power net of the vehicle.

Daimler Chrysler wishes to regulate the conducted emissions within their vehicles. They have self-imposed limits to conducted emissions and methods with which to test them. However, performing individual measurements on conducted emission transients for every DC motor in every vehicle they produce is undesirable. The ability to model unwanted emissions from a DC motor would reduce the need to individually test each motor.

This thesis examines conducted emission transients from several DC motors found in automobile applications. The conducted emission transients are measured and examined. The natural frequencies of each motor are determined and with this information the conducted emissions of the motor are modeled. Additionally, using the measured currents and impedances of each motor, a model for worst-case radiated emissions is also created.

For Grandpa

ACKNOWLEDGMENTS

First of all, I'd like to thank Dr. Dennis Nyquist and Dr. Ed Rothwell for being my advisors for this thesis and for helping me whenever I needed it. They were instrumental in the completion of this thesis, not only because of the immense amount of help and advice they provided me with, but also because they helped keep me focused on my goal and they continuously pointed me in the right direction for my research.

My parents deserve special recognition as well. They may not have always understood what I was doing, but they supported me nonetheless. Their continued encouragement and prayers are greatly appreciated. They also bailed me out financially a couple times when I found that graduate school had depleted my funds.

I would like to thank the people at the Daimler Chrysler who helped me. Particularly Andrew Shune, who helped develop the concept of this thesis, Dave Schilling, who provided me with the help I needed to do my measurements whenever I needed it, and Bob Schropshire, who helped me perform my conducted emission transient measurements.

Finally, special thanks go out to my girlfriend Kristy Mietelka. She supported me throughout my tenure as a graduate student, even though it lasted nearly a year longer than expected. She provided me with transportation to and from the research complex whenever I needed it. Most importantly she kept me focused on the goal of completing this thesis and gave me support and encouragement throughout the process. I believe that she is more excited about the completion of this thesis than I am.

LIST OF

LIST OF

CHAPT

CHAPT

CHAP

CHA

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview

Electromagnetic compatibility (EMC) has become an important part of electrical engineering. Government regulations and safety issues have forced companies to take a greater interest in the electromagnetic properties of their products. One of these regulated properties is the unintended current exiting a device via the power cable. This unwanted current is known as a conducted emission. The Federal Communications Commision (FCC) regulates conducted emissions for most products in the United States. However, the FCC does not regulate automobiles manufactured in the United States. Automobile manufacturers maintain their own standards for EMC, which are considerably more stringent than the FCC standards. This is because it is in the automobile manufacturers' best interest to produce a reliable, safe product in order to avoid lawsuits. Furthermore, automobiles manufactured in the United States are subject to regulations from the Comité International Spécial Des Perturbations Radioéletriques (CISPR) if they are to be sold in Europe.

In order to comply with the CISPR regulations, the Daimler Chrysler Corporation has dedicated a semi-anechoic chamber for the testing of electromagnetic compatibility issues outlined in the CISPR 25 electromagnetic compatibility regulations document [2]. Furthermore, they have developed a series of in-house EMC tests, such as conducted emission transient tests, in addition to those in the CISPR 25 document [3]. This thesis investigates the effects of conducted emission currents generated by the switching of DC motors used in automobiles by developing a numerical model of the testing setup and

comparing the model to results gathered using the conducted emission transients test method described in Daimler Chrysler's DCC EMC Test specifications and lab procedures [3].

This thesis is organized as follows. First, a brief explanation of the NSF GOALI project, which funded the research for this thesis, is given in Section 1.2 of this chapter. A brief description of the Michigan State University (MSU) / Daimler Chrysler interaction fostered through this project will follow.

The remainder of this thesis is divided into several chapters. Chapter 2 discusses the significance of conducted emissions and examines the measurement procedures for conducted emission transients employed by Daimler Chrysler. This includes an introduction to conducted emissions in Section 2.2, a discussion of conducted emission regulations in Section 2.3, and a more specific discussion of conducted emission transients in Section 2.4. Section 2.5 investigates the conducted emission transient measurement technique employed by Daimler Chrysler. More specifically, it investigates the use of the Broadband Artificial Network and discusses the linear model that this technique assumes.

Chapter 3 is dedicated to the measurement techniques used to acquire the necessary data for this thesis. Section 3.2 discusses the technique for acquiring the impedance characteristics of each motor and the Broadband Artificial Network. Section 3.3 discusses the conducted emission transient measurement procedures and presents the actual measured data.

Chapter 4 is devoted to the numerical modeling of conducted emission transients. Section 4.2 presents a derivation of the modeling procedure, using the extinction pulse

technique. Section 4.3 presents and discusses the numerical models generated using this technique and compares them to the measured data.

Chapter 5 discusses the implications of motor switching transients as they relate to electromagnetic compatibility concerns. Section 5.2 introduces radiated emissions and discusses their importance. A model for radiated emissions is developed in Section 5.3 that uses the previously modeled conducted emission transient data. Modeled conducted emission transient currents and radiated emissions are presented and discussed in Section 5.4.

Following the conclusion of this thesis in chapter 6, the two Fortran programs used in this thesis are included in Appendix A.

1.2 NSF GOALI Project

The research completed in this thesis was funded by the National Science Foundation (NSF) through the Grant Opportunities for Academic Liaison with Industry (GOALI) program. Additionally, this program helped fund the development of an EMC course to be taught at Michigan State University. The program was set up to foster an interaction between industry and academia. The funding provided by this program allowed graduate research assistants from MSU to help design an EMC course backed by the advice and expertise from industry provided by Daimler Chrysler. Furthermore, this project afforded graduate students the means to perform research guided by both Daimler Chrysler's industry experience and MSU's academic experience.

1.3 Michigan State University / Daimler Chrysler Interaction

As discussed in the previous section, funding through the NSF GOALI program opened the door for interaction between Michigan State University and Daimler Chrysler. Through this interaction, experts from the Electrical / Electronic Systems Compatibility (EESC) group at Daimler Chrysler identified a research project in the field of electromagnetic compatibility that would both be useful for their group and suitable for a thesis. The professors at MSU then guided the research on this project so that it would be within the scope of the academic requirements required for a Master of Science Thesis. Thus, this thesis is influenced by both industry and academia, with the goal that the research herein is both practical and scholarly.

CHAPTER 2

CONDUCTED EMISSIONS CONSIDERATIONS

2.1 Overview

This chapter discusses the concepts of conducted emissions. First, an explanation is given about what conducted emissions are and why they are important to the automobile industry. A discussion of conducted emissions transients follows, discussing what conducted emissions transients are and why they are important. A simple linear model for conducted emissions transients is presented and a discussion of the Broadband Artificial Network, which is used in this model, follows. The chapter concludes with an investigation into the limitations of the simple linear model for conducted emissions transients.

2.2 Conducted Emissions

Conducted emissions are the currents that are passed through a unit's power harness and placed on the common power net. Conducted emissions are undesirable for a couple of reasons. First, the conducted emission currents exiting a device may affect other devices on the common power network through direct coupling. That is, the current exits one device, travels through the common power net and enters another device, potentially causing interference with the second device. Conducted emissions can also cause unwanted radiation. As the conducted emissions are placed onto the wiring external to a particular unit they cause radiation to occur. This unwanted radiation can, in turn, induce currents on other wires and ultimately cause interference with other devices.

2.3 Conducted Emissions Regulations

Most products sold in the United States must comply with conducted emissions standards set forth by the Federal Communication Commision. Automobiles manufactured in the United States are exempt from FCC regulations, while automobiles imported into the United States must meet FCC standards. If an automobile manufactured in the United States is to be sold in Europe, however, it is subject to the regulations from the Comité International Spécial Des Perturbations Radioéletriques. In general, automobile manufacturers have their own self-imposed conducted emissions limits, which are more stringent than those set forth by the FCC or CISPR. Automobile manufacturers self-police themselves in this manner to avoid the necessity of government imposed regulations as well as the importance to provide a safe and reliable product for the consumer.

The emphasis of CISPR and FCC regulations falls upon devices that have ac power cords. As such, devices that require ac power are required to be tested with a Line Isolation Stabilization Network (LISN). A LISN is a circuit that filters the ac power coming into the device under test, such that at 60 Hz the device receives unpolluted power. However, at the conducted emissions test frequencies inductors in the LISN act as open circuits, isolating the device under test from the ac power supply. The LISN also provides affixed load impedance to the device under test at those radio frequencies. This insures that during testing, conducted emissions are only being tested due to the device under test and not the ac power net [1]. Automobiles do not use ac power, but are instead powered by a dc battery supply. Thus, conducted emissions testing for automobiles require something other than a LISN. Daimler Chrysler has developed a Broadband

Artificial Network (BAN), which serves much the same purpose as the LISN, but is used with a dc power supply.

2.4 Conducted Emissions Transients

DC motors are common components throughout an automobile and are subject for concern when considering conducted emissions. Whenever a DC motor is switched off a transient current is generated, which is induced onto the power cable exiting the motor. This current is very large compared to any conducted emissions generated by the motor during its normal operation. As such, these transient currents are typically worst-case scenarios for conducted emissions generated by a DC motor. Automobile companies wish to keep these conducted emissions transients below their self-imposed conducted emissions limits.

Often automobile manufacturers fabricate a DC motor to perform a specific task, without designing it to meet conducted emissions regulations. They then test the motor for conducted emission compliance and modify its design if it fails to meet the self-imposed conducted emission transient constraints. It is desirable to be able to predict if a motor will pass these tests without testing them. Thus, Daimler Chrysler has asked the author to investigate a method to model the conducted emissions transient behavior of DC motors.

2.5 Broadband Artificial Network

It is assumed that a linear model for DC conducted emission transients can be created. Figure 2.1 shows this linear model for DC motor conducted emission transients.

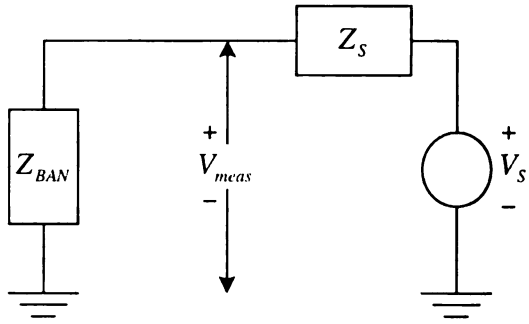

Figure 2-1: Equivalent Circuit for Linear Model of Conducted Emission Transients

V_s is the modeled source voltage. Z_s is the known source impedance. V_{meas} is the measured voltage. The modeled voltage V_s approximates V_{meas} as long as $Z_{BAN} >> Z_s$, where Z_{BAN} is the impedance of the Broadband Artificial Network (BAN). The inclusion of the BAN allows for a known impedance and repeatable results.

Daimler Chrysler developed the Broadband Artificial Network for the purpose of conducted emissions testing. Similar to the LISN, the BAN is designed to provide power to the device under test, while isolating it from the power supply at conducted emissions frequencies. This is achieved by placing several inductors in series between the source and the device under test, as well as a capacitor to ground. The BAN has three connection terminals. Terminal one is the output terminal. This terminal connects to the motor under test. Terminal two is the input terminal, which is conducted to the power supply. The BAN isolates the motor under test at terminal one from the power supply at terminal two at radio frequencies. Terminal three is a low current BNC connection which connects to an output device such as an oscilloscope.

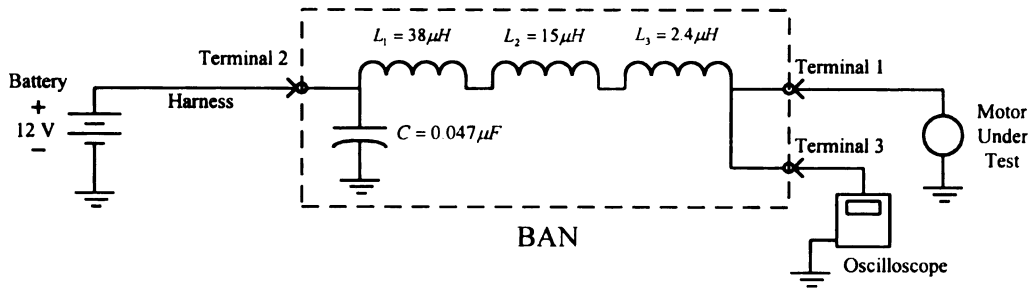

Figure 2-2: System Connected With a Broadband Artificial Network

The component values of the BAN are chosen such that the equivalent impedance of the BAN remains relatively constant over the entire conducted emissions frequency test range. This is a consequence of the variable frequency dependence of the constituent inductors. In the case of this BAN the frequency test range is 250 kHz – 500 MHz.

At dc frequencies the capacitors act as open circuits and the inductors act as short circuits. Thus, the BAN acts as a short circuit at DC frequencies and the test circuit performs as if no BAN were present.

Figure 2-3: Equivalent BAN at DC Frequencies

At conducted emissions frequencies, however, the capacitors act as short circuits and the inductors act to provide the desired, relatively high, loading impedance, which is independent of the harness impedance. This isolates the measuring equipment from the dc power source and prevents low frequency noise from affecting measurements.

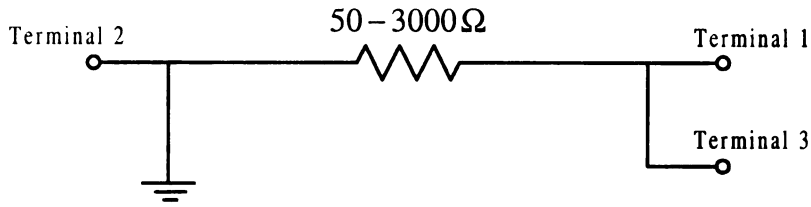

Figure 2-4: Equivalent BAN at Conducted Emissions Frequencies (250 kHz – 500 MHz)

According to the Daimler Chrysler BAN specifications this broadband isolator has impedance characteristics shown in Table 2-1.

Frequency Range	Minimum Magnitude of Impedance (Ω)
0.25 MHz – 0.50 MHz	50
0.50 MHz – 1.0 MHz	100
1.0 MHz – 2.0 MHz	200
2.0 MHz – 150 MHz	400
150 MHz – 500 MHz	100

Table 2-1: Impedance Specifications for Broadband Artificial Network

The measured impedance of the BAN coincides with the impedance specifications rather well, as seen in Figure 2-5. The measured impedance dips to around 350 Ω between 65-95 MHz. However, the measured impedance of the BAN is what is of true importance for this research. The impedance curve of the BAN remains the same for each motor tested. This allows for repeatable test results and makes the data easier to interpret.

W

will w

conduc

measur

compa

the fre

compa

several

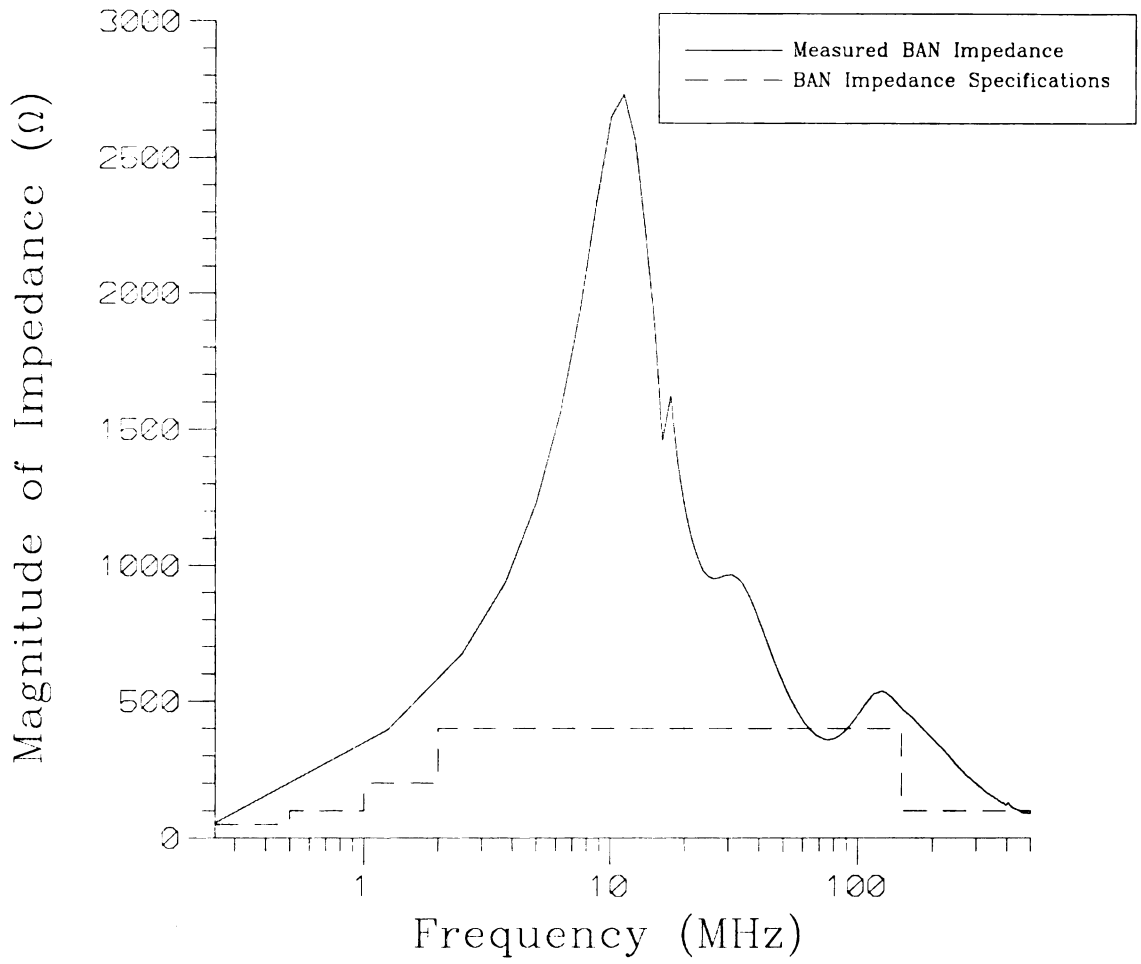

Figure 2-5: BAN Impedance Spectrum

With a known impedance spectrum it is possible to predict how well a linear model will work. It is desired to model the conducted emissions of each motor over the conducted emissions spectrum of 250 kHz-500 MHz. However, the voltage that is measured is the actual source voltage if and only if the impedance of the BAN is large compared to the source impedance. The source impedance of each motor is known, so the frequency range that this linear model is effective over can be determined by comparing the BAN impedance to the source impedance of each motor. Results for several typical motors are presented below in Figures 2-6 – 2-9.

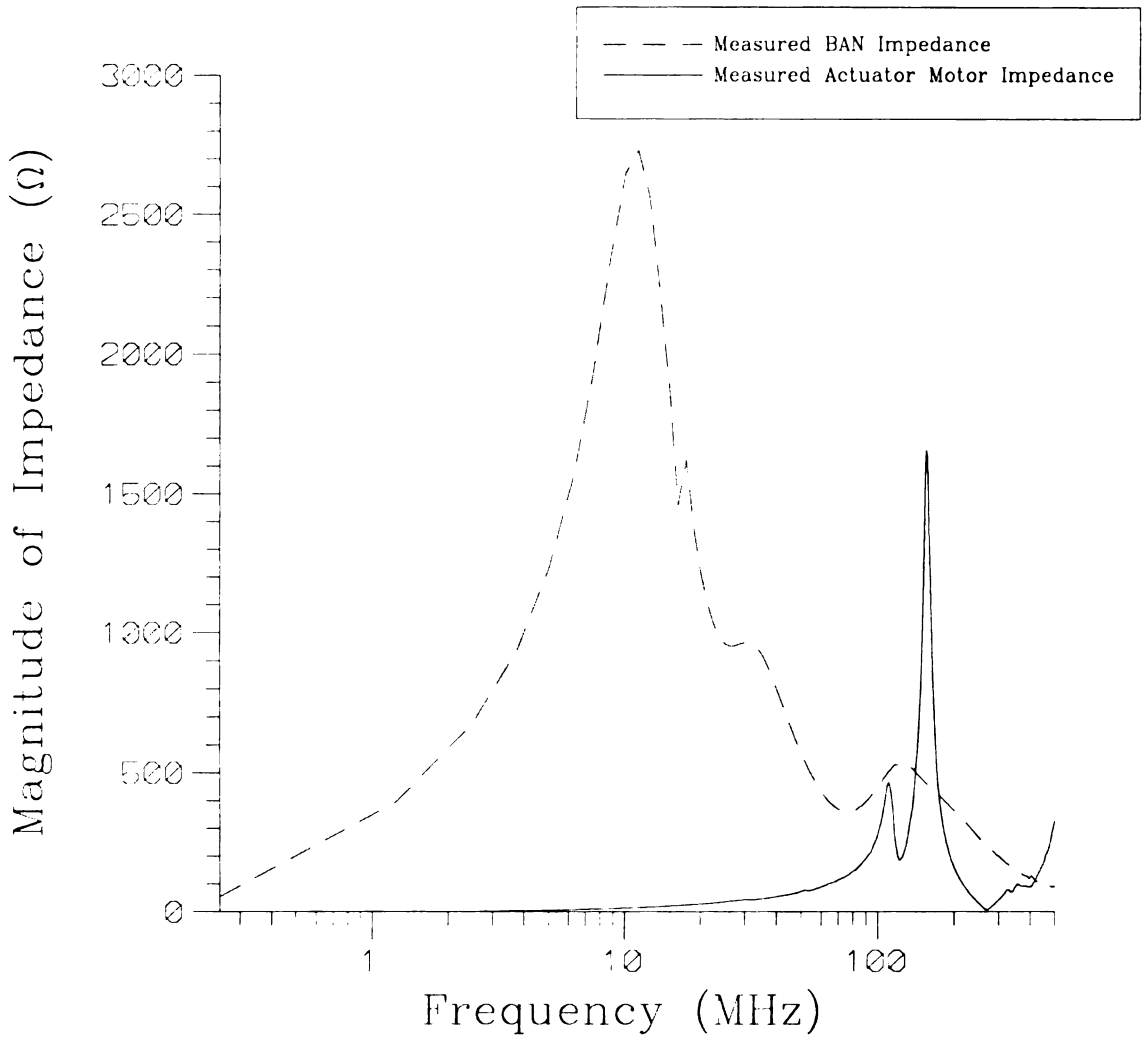

Figure 2-6: Impedance Spectrum of the Actuator Motor Compared to BAN Impedance

Figure 2-6 shows that the measured actuator motor induce radio frequency voltage, V_{meas}, can be modeled as equal to V_s over the frequency range 250 kHz-100 MHz and 200 MHz-300 MHz. In practice the measurement is not corrected for voltage division between Z_{BAN} and Z_s.

Vergrößerung des Innenbereichs

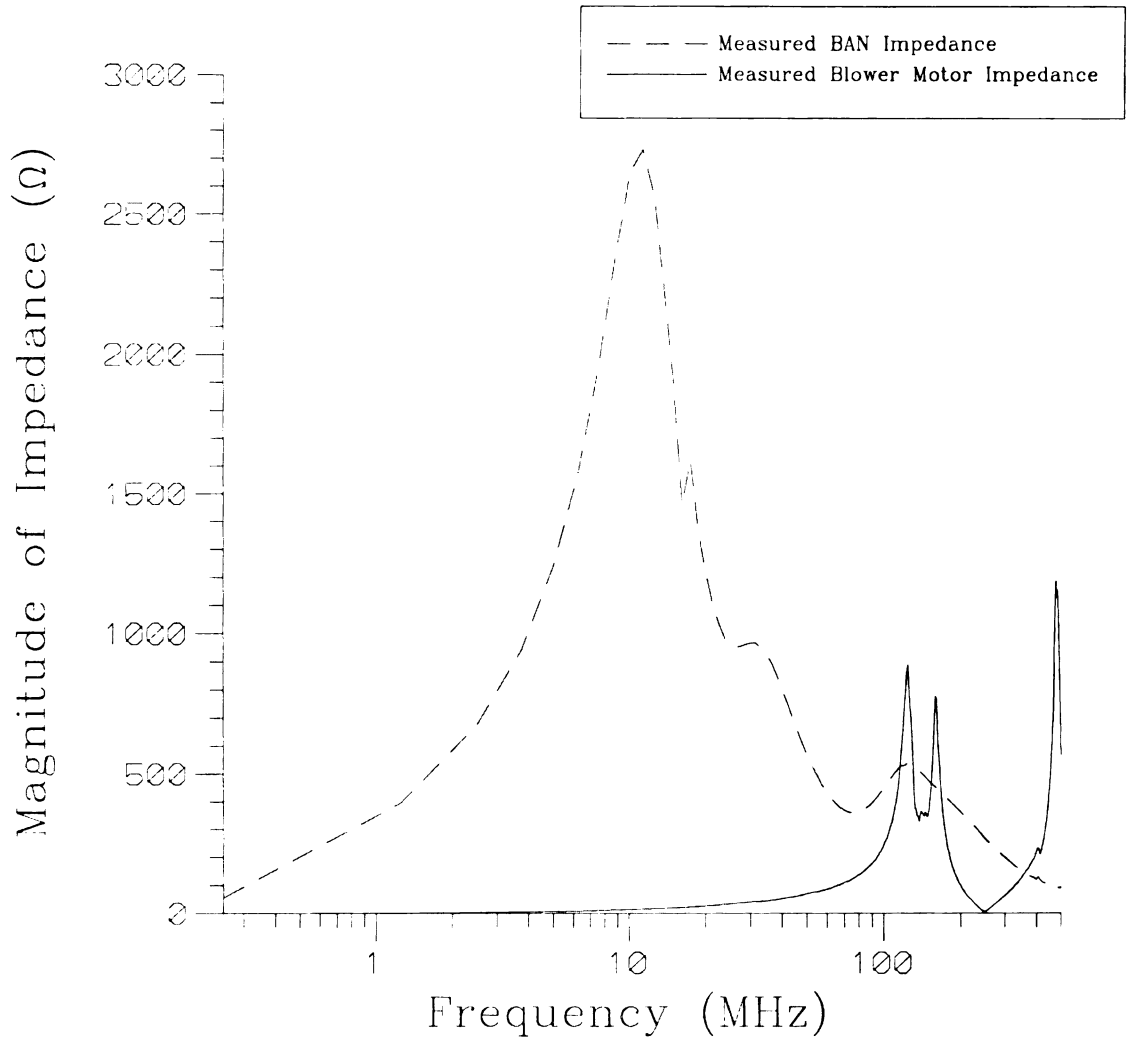

Figure 2-7: Impedance Spectrum of the Blower Motor Compared to BAN Impedance

Figure 2-7 shows that the blower motor can be modeled similarly over the frequency range 250 kHz-100 MHz and 200 MHz-300 MHz.

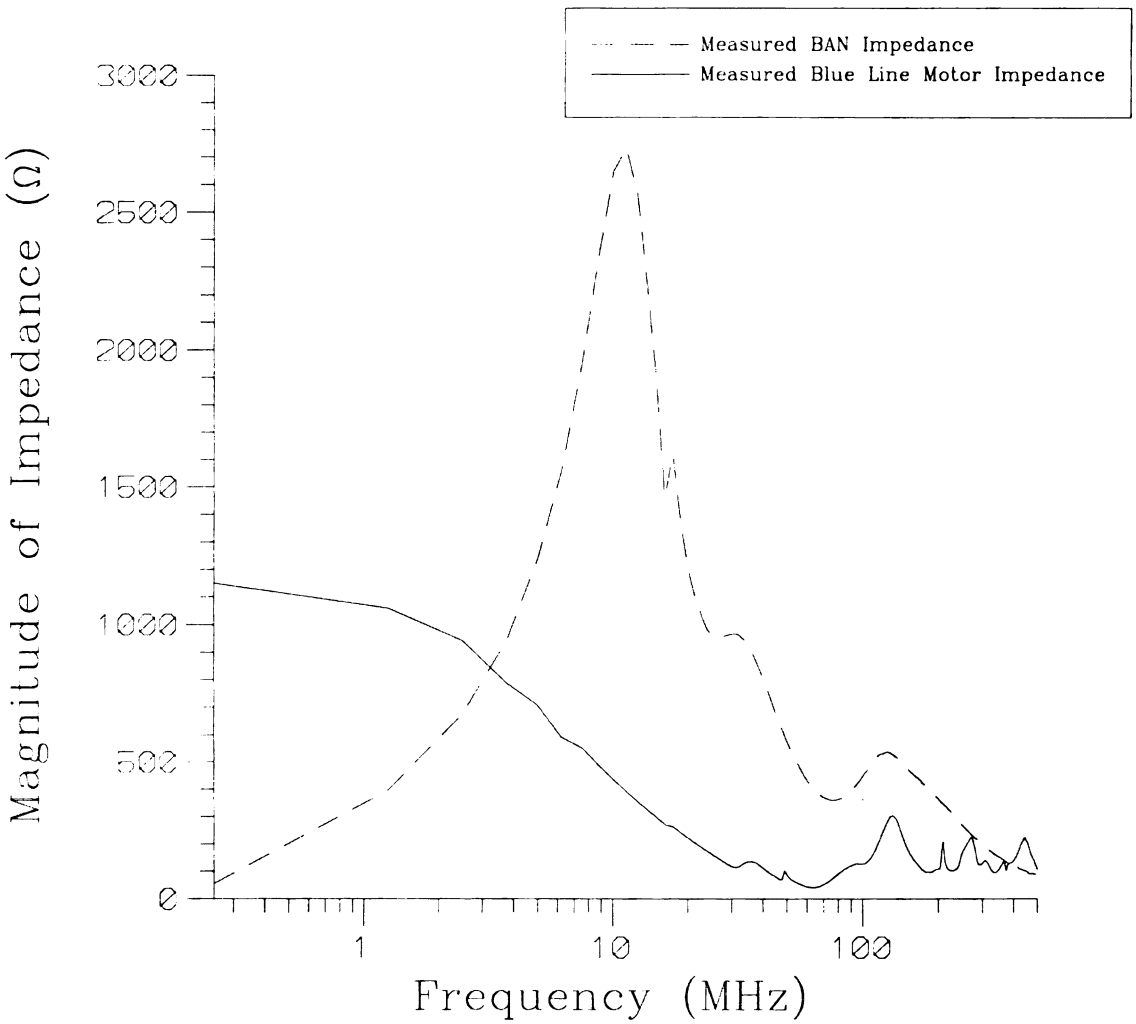

Figure 2-8: Impedance Spectrum of the Blue Line Motor Compared to BAN Impedance

Figure 2-8 shows that the induced radio frequency voltage of the blue line motor can be modeled without correction over the frequency range 4 MHz-200 MHz.

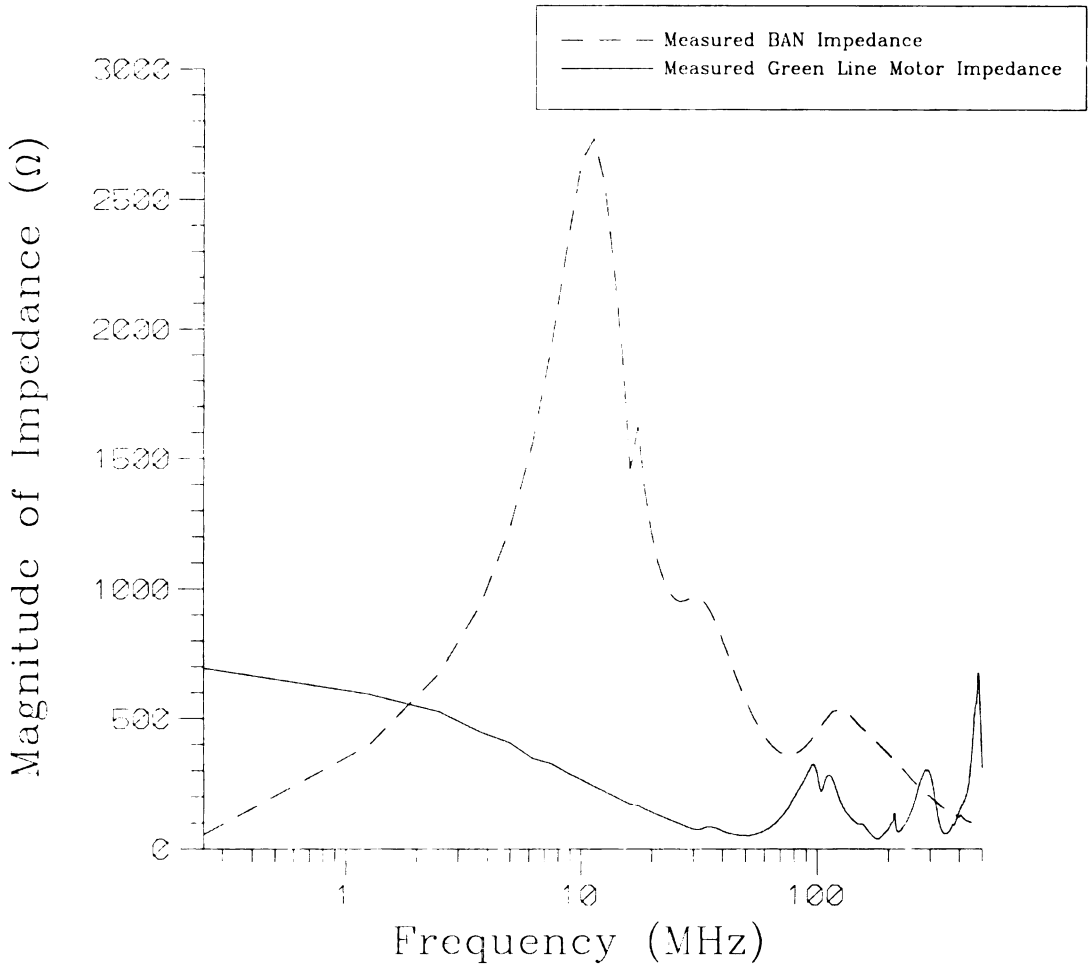

Figure 2-9: Impedance Spectrum of the Green Line Motor Compared to BAN Impedance

Figure 2-9 shows that the green line motor transient induced voltage can be modeled similarly over the frequency range 2 MHz-200 MHz.

Since the validity of the linear model falls in doubt for the four motor types over certain frequency ranges it is useful to examine the data as measured as well as adjusted to remove any frequency content that is in doubt. This is accomplished by taking the Fourier transform of the measured data and then truncating the data at the appropriate frequency. Taking the Fourier transform of this data returns the adjusted time domain

data. Figures 2-10 through 2-17 show the measured data compared to the adjusted data with frequency content removed. In each case frequency is truncated at the upper limits, but not the lower limits due to the negligible extent of the lower frequency range. The data in these plots is obtained using the methods described in Chapter 3.

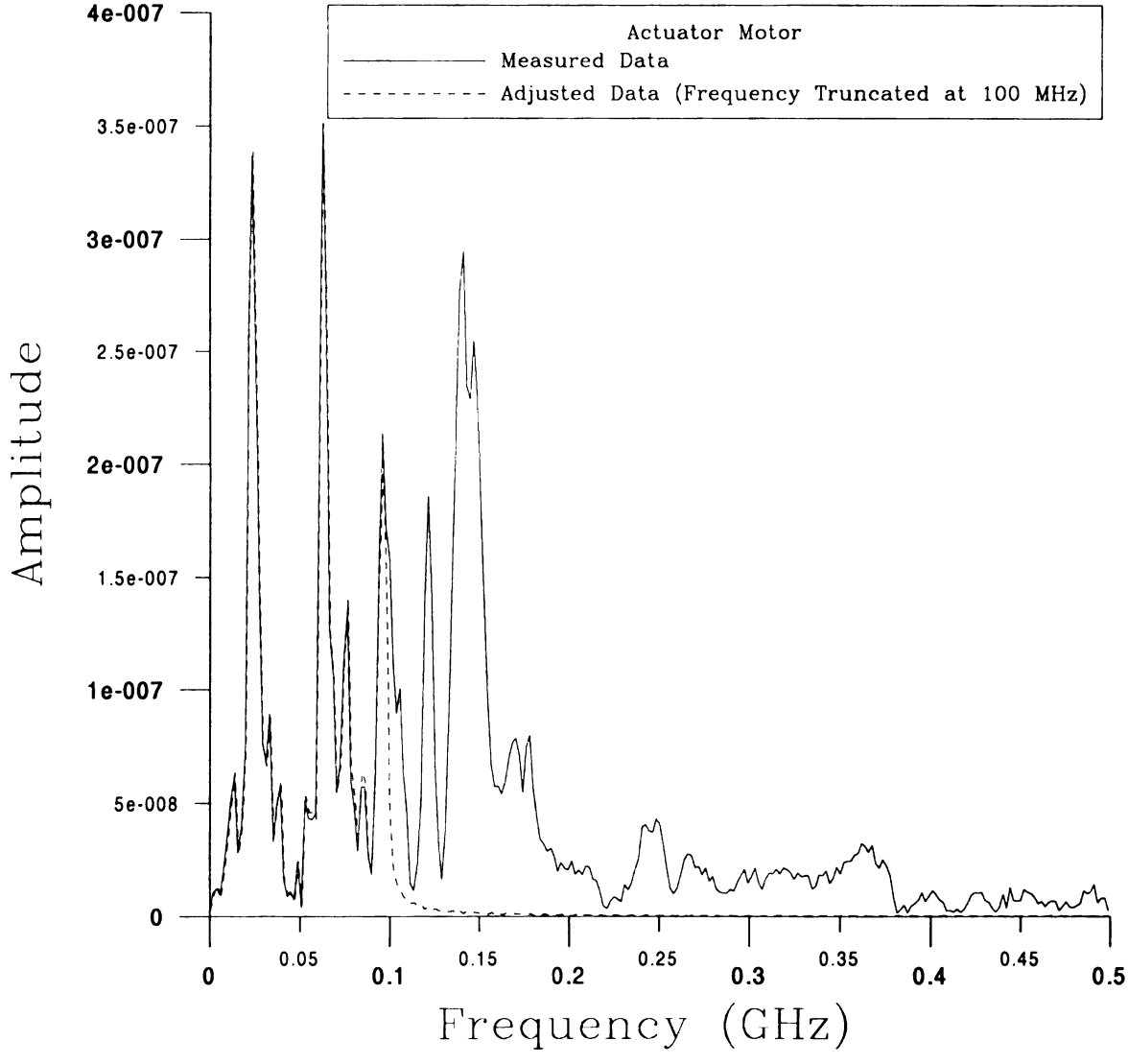

Figure 2-10: Comparison of Measured and Adjusted Frequency Content for Actuator Motor

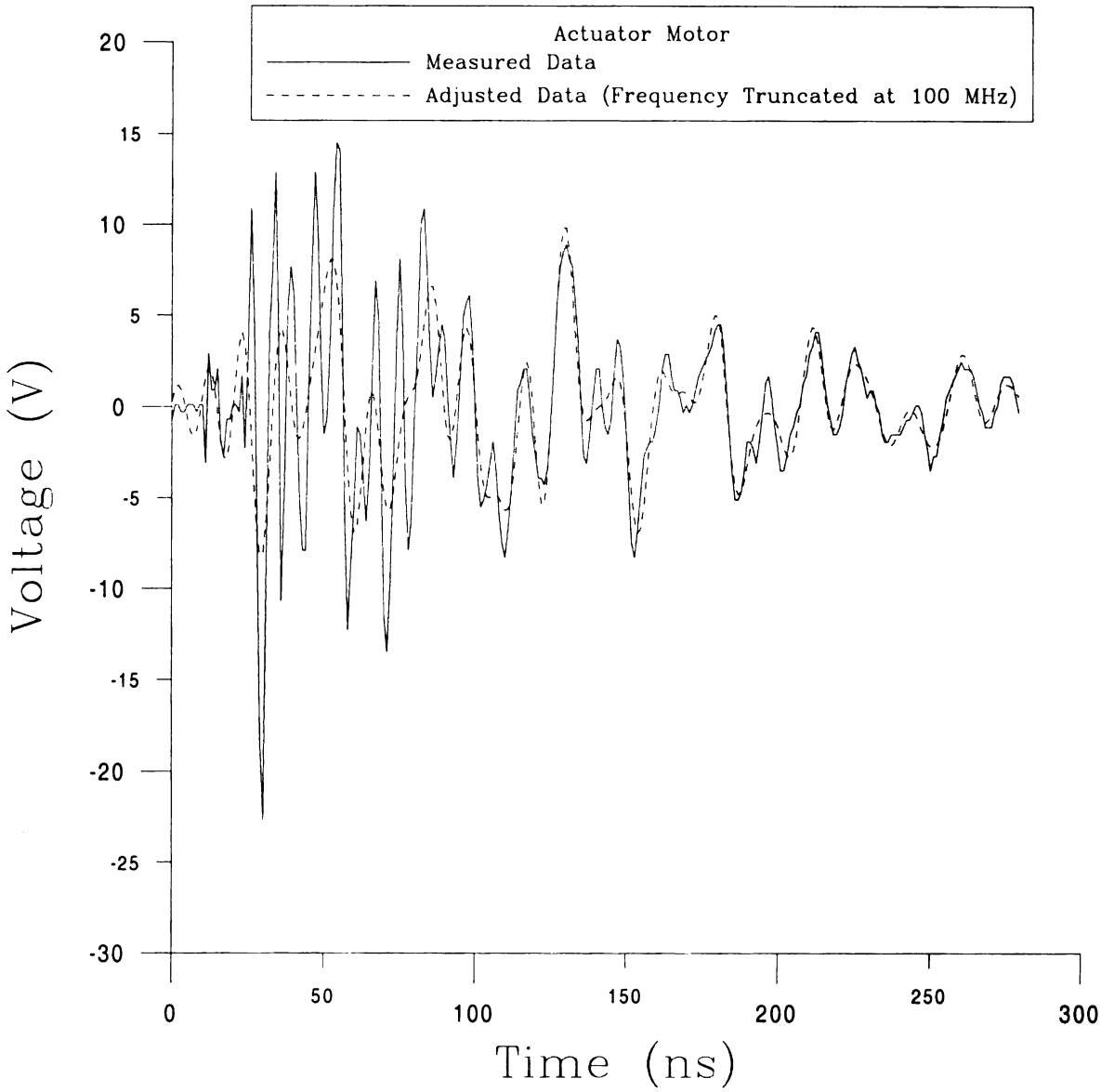

Figure 2-11: Comparison of Measured and Adjusted Time Domain Content for Actuator Motor

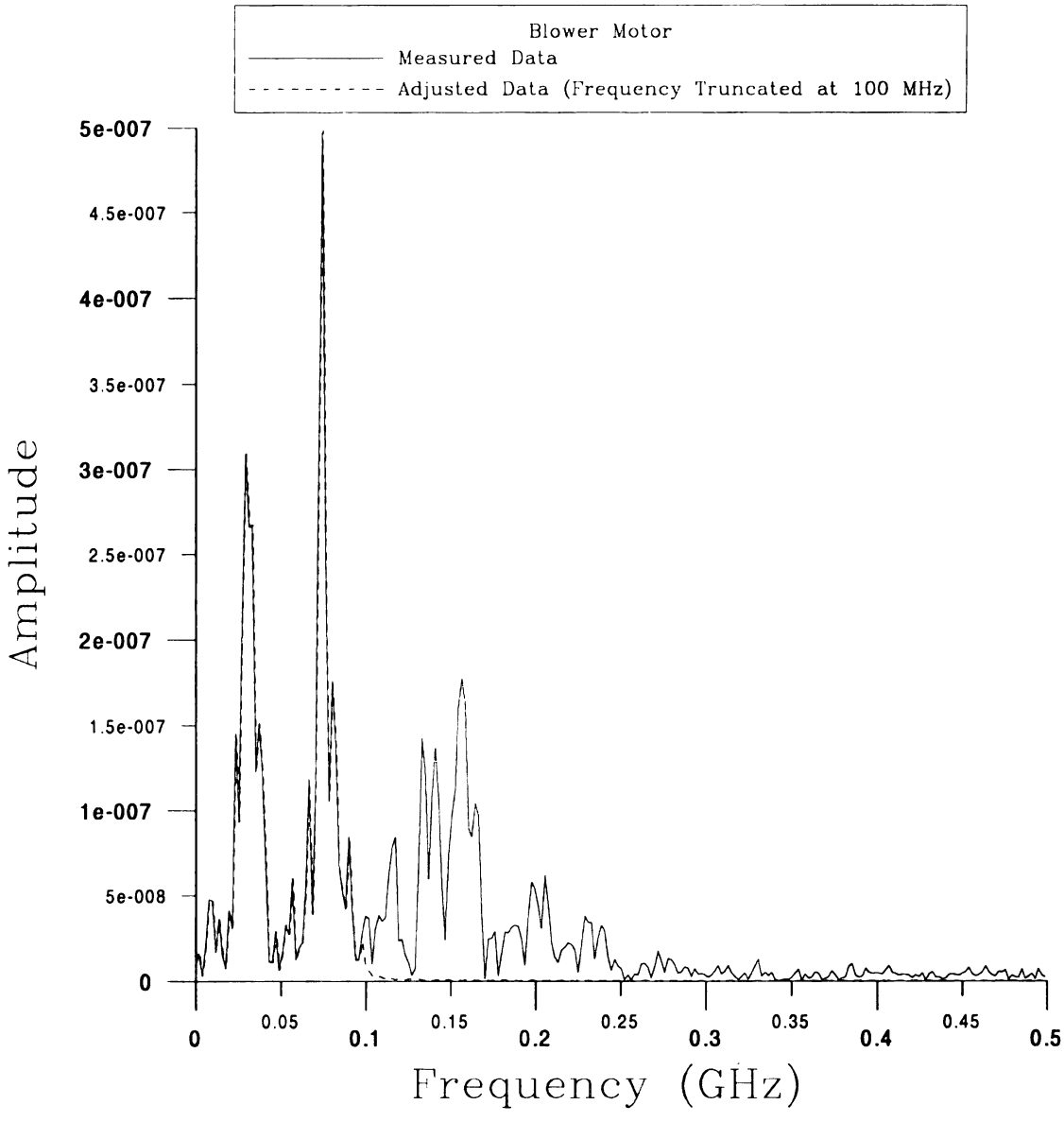

Figure 2-12: Comparison of Measured and Adjusted Frequency Content for Blower Motor

Figur

T

MHz. I

the var

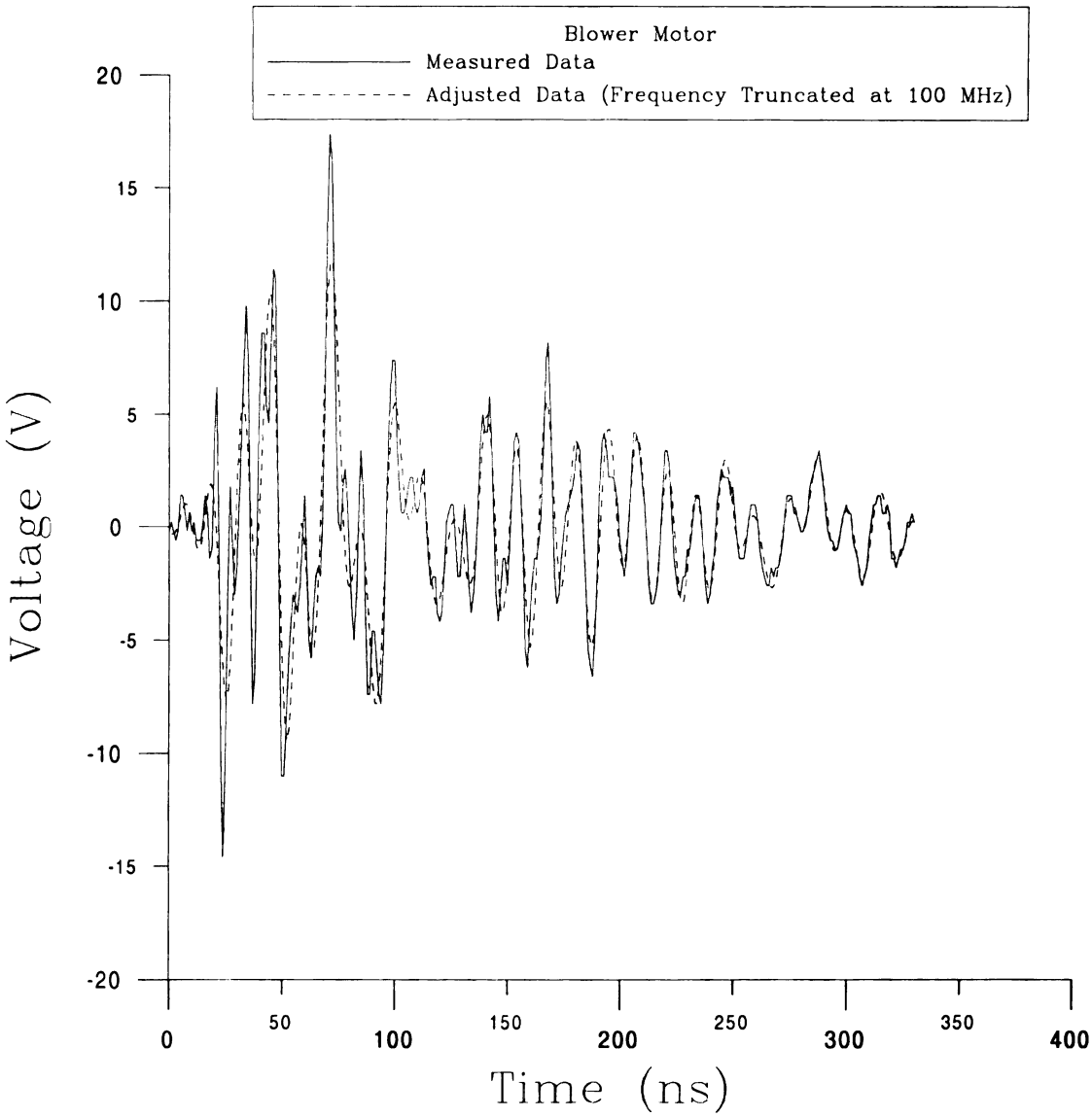

Figure 2-13: Comparison of Measured and Adjusted Time Domain Content for Blower Motor

The actuator and blower motors both are truncated in the frequency domain at 100 MHz. In both cases some significant frequency content is removed. This is reflected in the variations of the measured and adjusted time-domain plots.

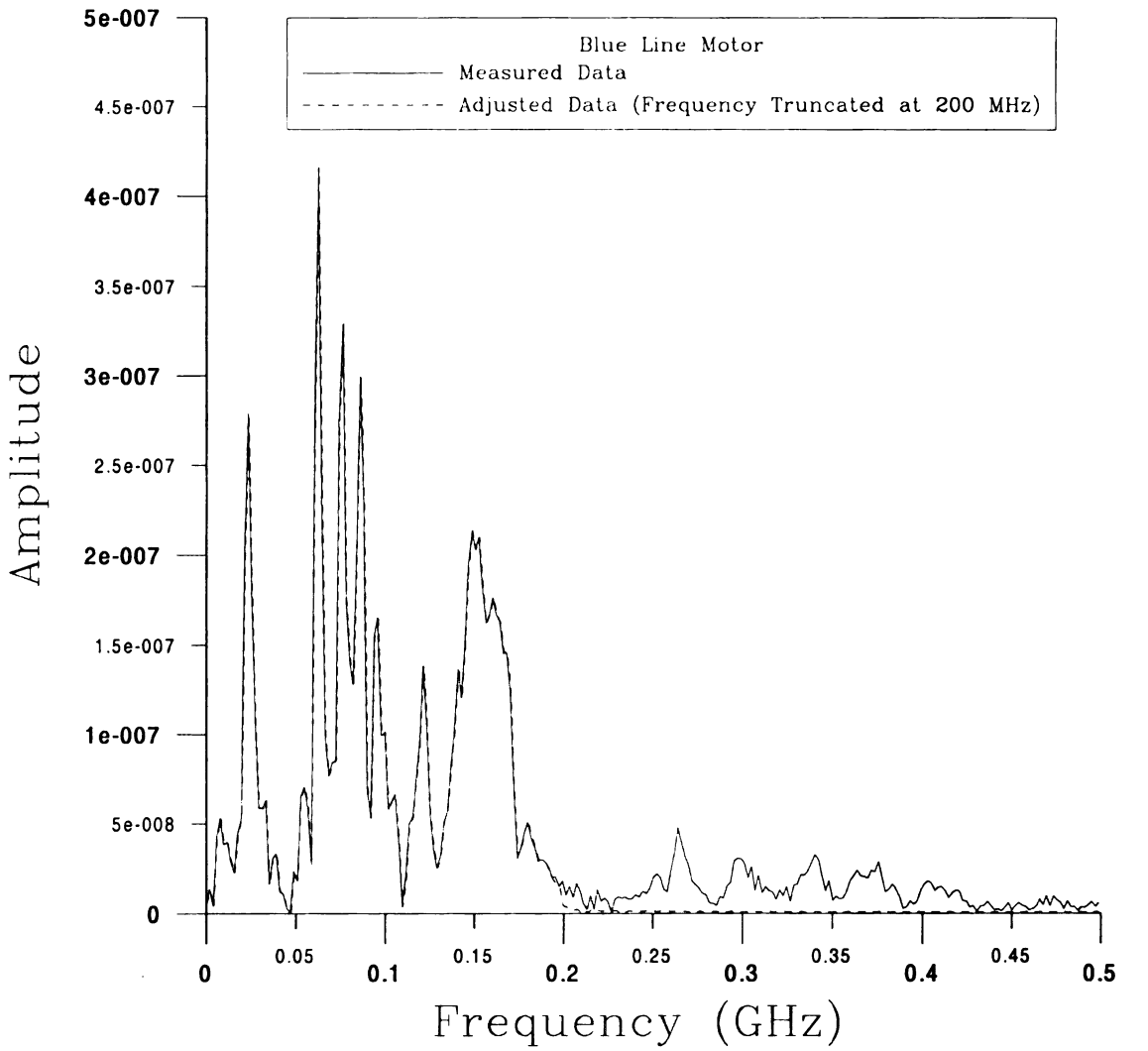

Figure 2-14: Comparison of Measured and Adjusted Frequency Content for Blue Line Motor

Figu

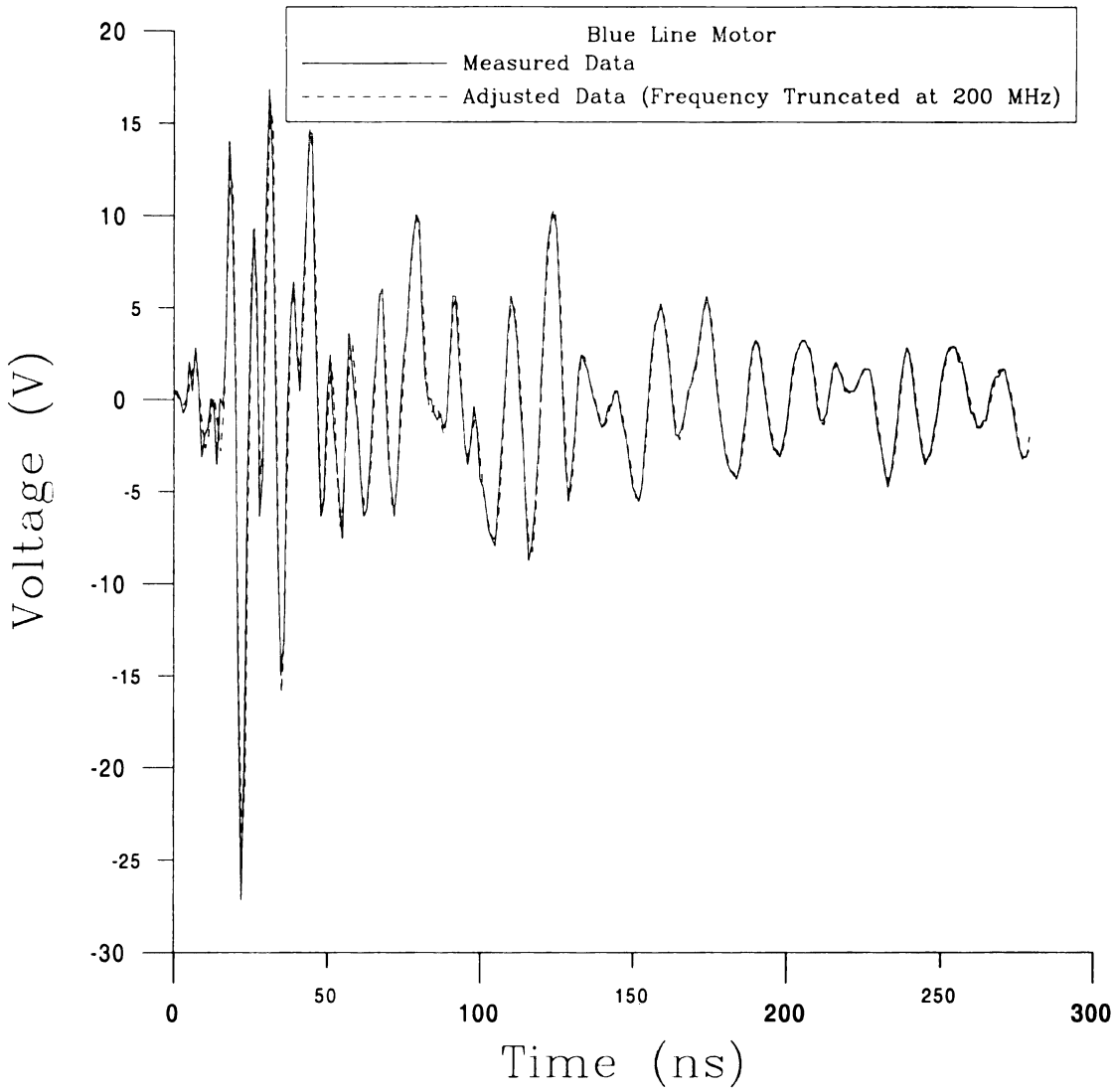

Figure 2-15: Comparison of Measured and Adjusted Time Domain Content for Blue Line Motor

4e-

3e-

2e-

1e

Figur

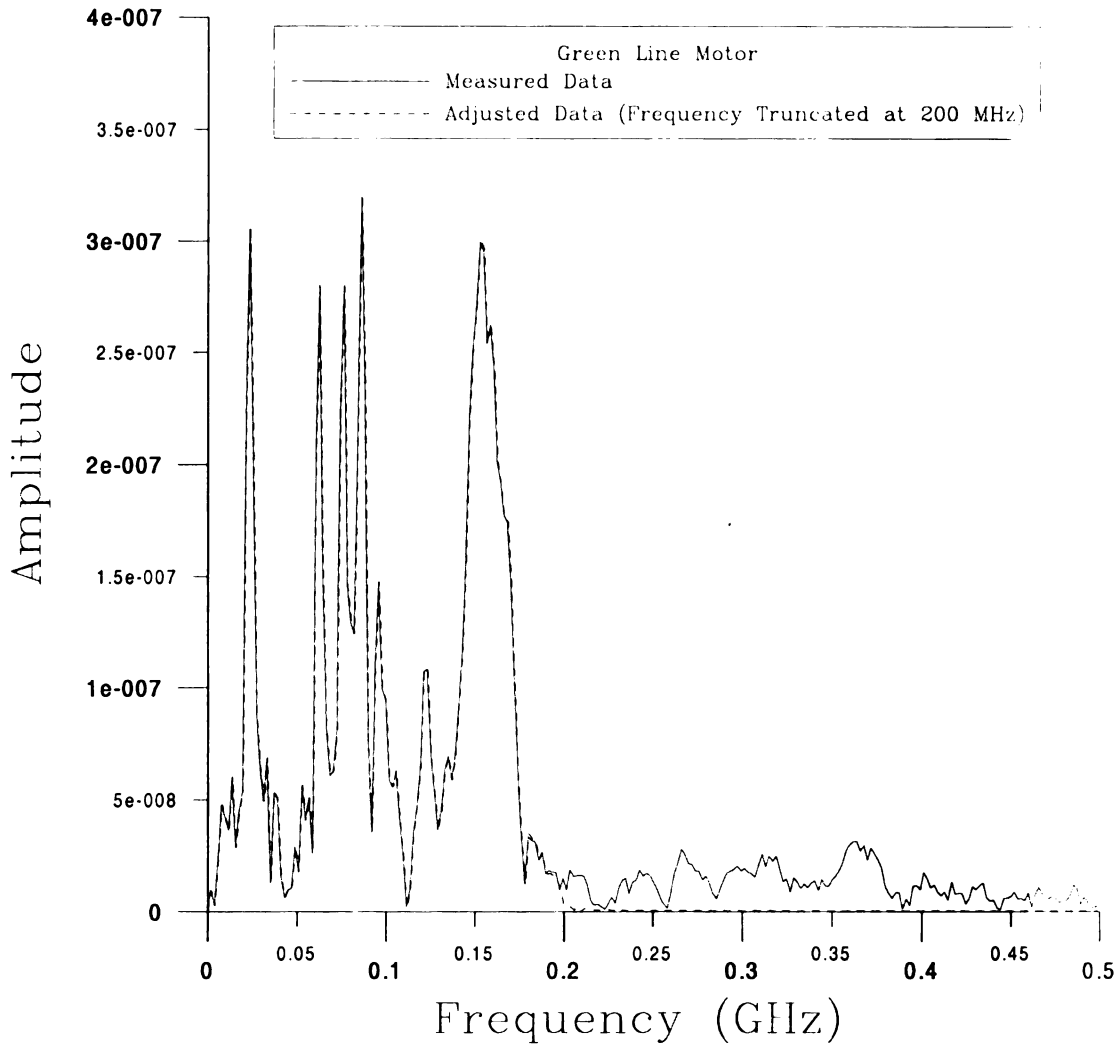

Figure 2-16: Comparison of Measured and Adjusted Frequency Content for Green Line Motor

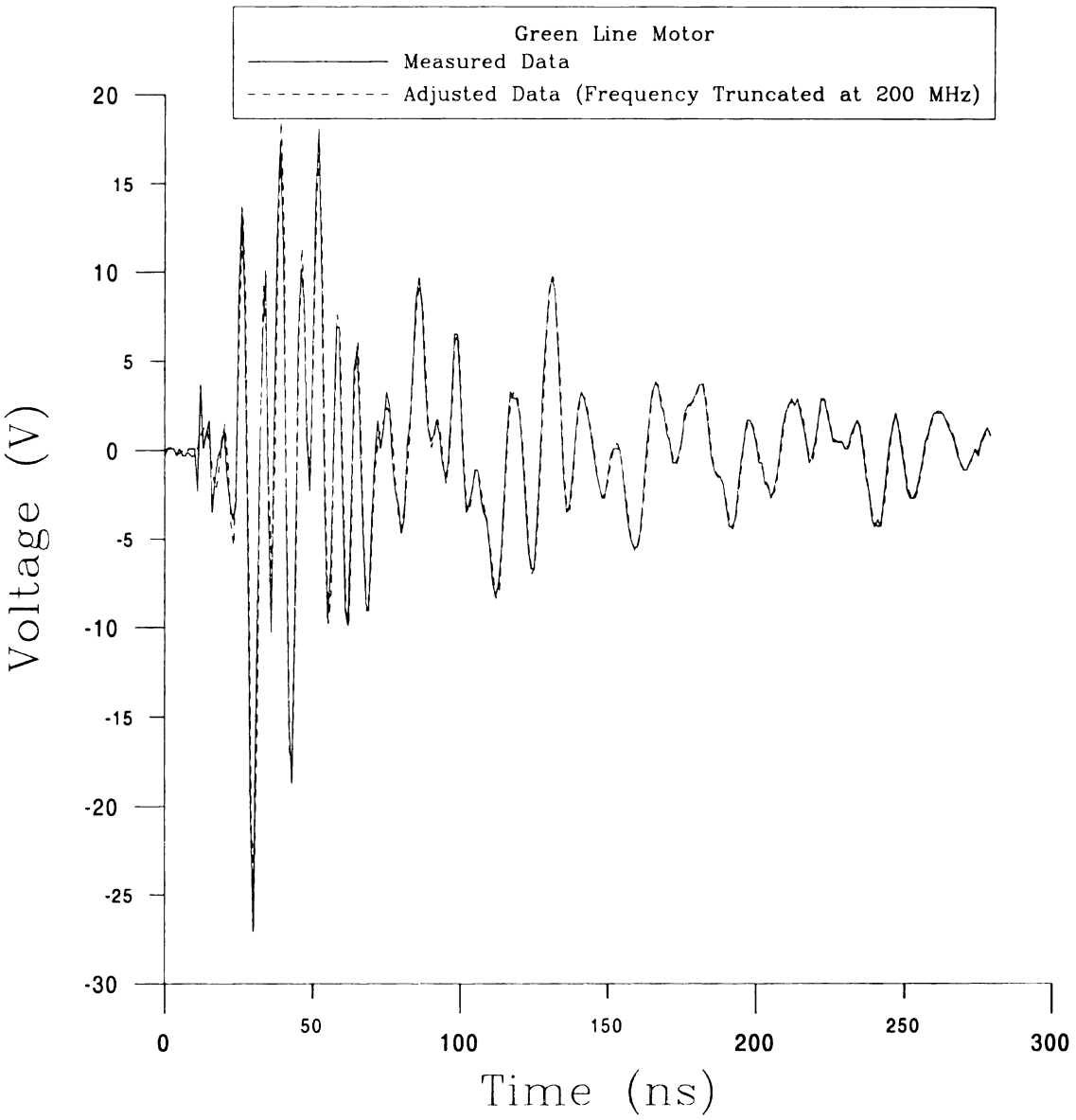

Figure 2-17: Comparison of Measured and Adjusted Time Domain Content for Green Line Motor

The blue line and green line motors, however, are both truncated at 200 MHz and no significant frequency content is removed. This also is reflected in the time domain plots, as neither motor shows significant variation between measured and adjusted data.

CHAPTER 3

MEASUREMENT OF MOTOR SWITCHING TRANSIENTS

3.1 Overview

This chapter discusses the measurement techniques used in this thesis. All measurements were performed at the Daimler Chrysler Technical Center Electromagnetic Compatibility Lab. First, a discussion of the method used to measure the impedance spectrums presented in Chapter 2 is presented. Next, the test method for measuring conducted emission transients is presented, and measured data is presented. Finally, a discussion of repeatable measurements is presented.

3.2 Measuring Impedance

Impedance measurements are performed using the HP 4195A Network Analyzer. The power for the HP 4195A is turned on and it is allowed to warm up for a period of 30 minutes. Minimum frequency is set to 250 kHz. Maximum Frequency is set to 500 MHz. All connections to the network analyzer are via a BNC connection on the impedance measurement module. Calibrations are performed by individually connecting a short-circuit termination, an open-circuit termination, and a 50 Ω load to the BNC connector and performing a calibration sweep for each termination. The HP 4195A computes the calibration coefficients and calibration is confirmed by measuring the impedance of the 50 Ω load across the entire frequency range. After the network analyzer is calibrated the BAN is connected to the BNC termination and an impedance sweep is performed across the entire frequency range. This data is saved to disk in LIF format, which is readable only by a Hewlett Packard operating system. Measurements

are similarly performed on each motor with no power applied to the motor. Daimler Chrysler's Electromagnetic Compatibility department provided software that converted the data from LIF format to a tab delimitated ASCII format, which can easily be entered into a spreadsheet. Plots of the impedance spectrums of the BAN and the motors are found in Chapter 2: Figures 2-5 to 2-9.

3.3 Measuring Conducted Emission Transients

Conducted emission transients occur when a DC motor is switched off from a running state. Thus, it is desirable to create a test procedure that measures the currents exiting the wire harness from a motor as it is switched off. However, it is also important that just the conducted transient currents are measured, and that any other noise, such as that from the DC power supply be excluded from such measurements. Furthermore, the emphasis on conducted emission transients is placed on worst-case scenarios. That is, automobile companies are only interested in the maximum transient currents generated by any given motor, as these worst-case currents cause the greatest threat to produce interference in other devices.

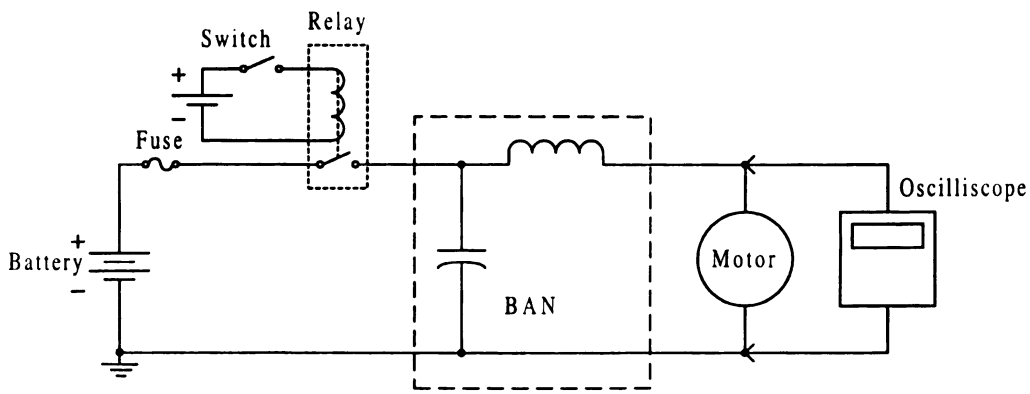

Figure 3-1: Conducted Emission Transient Test Setup

Figure 3.1 shows the test setup used to measure conducted emission transients. The entire test setup is arranged over a ground plane, which also serves as circuit ground. All wires in the circuit are as short as possible. The switch in the figure is a handheld pushbutton momentary contact switch that controls the relay. The pushbutton switch is provided power from the same battery that powers the rest of the circuit, which is fully charged to 12.7 V at the start of the testing. When the switch is activated, it energizes the relay, completing the circuit and providing power to the motor. When the pushbutton switch is depressed the circuit becomes incomplete and the motor loses power. When the motor loses power it generates a conducted emission transient current.

The oscilloscope is adjusted so that when this transient occurs the scope will trigger and capture the waveform. The circuit must be completed and broken several times to properly adjust the triggering. Triggering can be set for "positive" or "negative" triggering. Positive triggering triggers the oscilloscope only when the waveform slope is positive, and negative triggering, likewise, only triggers the scope when the waveform slope is negative. Two measurements are used for each motor state measurement, one using positive triggering and the other using negative triggering. Data for the green wire motor stalled state with forward bias measured with negative triggering was stored on a damaged disk sector and will not be presented in this paper. After the triggering is properly adjusted the circuit is again completed and broken numerous times and the user observes the waveforms displayed on the scope. After observing the waveforms, the user repeats the test several times until a representative worst-case transient can be recorded. A fast rise time and a large peak-to-peak voltage should characterize this worst-case

Data
4
4
4
4
3
3
5
3
3
5
3
3
3
3
3
0
0
0
-

transient. Table 3-1 shows the measured rise time and peak-to-peak voltage for each motor state measurement.

Data Set	Motor	Bias	Triggering	Rise Time	Peak-to-Peak Voltage
Act 19	Actuator	Forward	Positive	200 ps	31.6 V p-p
Act 18	Actuator	Forward	Negative	400 ps	37.2 V p-p
Act 21	Actuator	Reverse	Positive	200 ps	34.8 V p-p
Act 20	Actuator	Reverse	Negative	6.6 ns	33.2 V p-p
Blow 23	Blower	Forward	Positive	800 ps	32.0 V p-p
Blow 22	Blower	Forward	Negative	400 ps	31.6 V p-p
Blue 00	Blue Line, Stalled	Forward	Positive	500 ps	44.0 V p-p
Blue 01	Blue Line, Stalled	Forward	Negative	600 ps	42.8 V p-p
Blue 05	Blue Line, Stalled	Reverse	Positive	300 ps	45.2 V p-p
Blue 04	Blue Line, Stalled	Reverse	Negative	400 ps	45.2 V p-p
Blue 07	Blue Line, Traveling	Forward	Positive	500 ps	42.8 V p-p
Blue 06	Blue Line, Traveling	Forward	Negative	400 ps	44.4 V p-p
Blue 03	Blue Line, Traveling	Reverse	Positive	500 ps	44.4 V p-p
Blue 02	Blue Line, Traveling	Reverse	Negative	1.8 ns	46.0 V p-p
Green 17	Green Line, Stalled	Forward	Positive	4.2 ns	44.0 V p-p
Green 16	Green Line, Stalled	Forward	Negative	400 ps	45.6 V p-p
Green 13	Green Line, Stalled	Reverse	Positive	400 ps	46.8 V p-p
Green 12	Green Line, Stalled	Reverse	Negative	2.8 ns	48.4 V p-p
Green 15	Green Line, Traveling	Forward	Positive	2.4 ns	54.0 V p-p
Green 14	Green Line, Traveling	Forward	Negative	1.7 ns	42.8 V p-p
Green 11	Green Line, Traveling	Reverse	Positive	400 ps	48.8 V p-p
Green 10	Green Line, Traveling	Reverse	Negative	400 ps	45.2 V p-p

Table 3-1: Measured Rise Time and Peak-to-Peak Voltage for Each Motor State

Each motor is tested in each of its available states. The actuator motor was tested with both forward and reverse bias. The blue line and green line motors, both position

adjustment motors for rearview mirrors, have two states. The first state, where the motor is moving the mirror adjustment plate is called the traveling state. The stalled state is when the motor is running but the adjustment plate can go no further. These motors are tested with both forward and reverse bias in both the traveling and stalled states. The blower motor does not function with reverse bias. Thus, it is only tested in the forward bias state. Figures 3-2 through 3-22 show the measured conducted emission transients for each motor state.

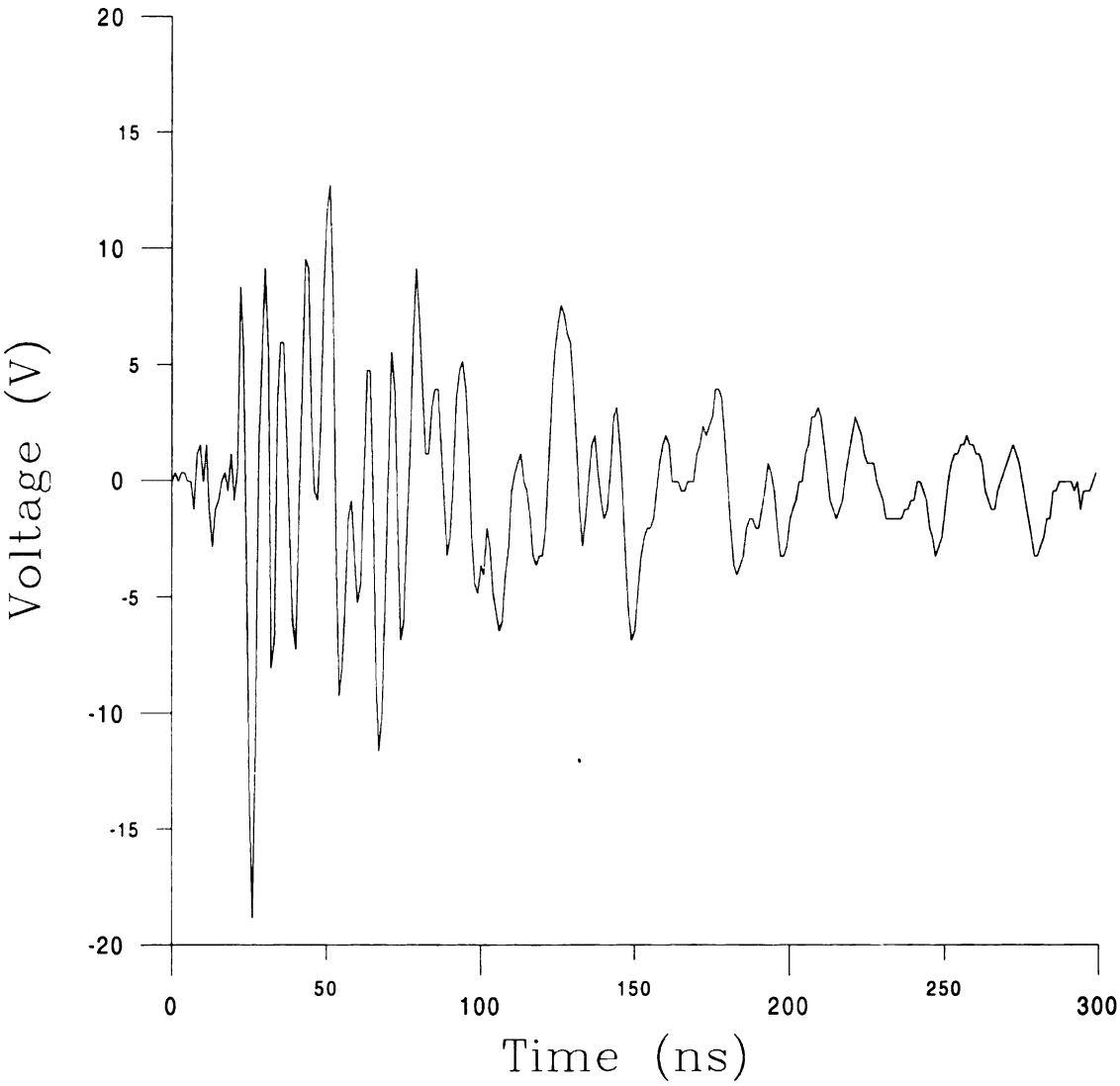

Figure 3-2: Conducted Emission Transient for Forward Biased Actuator Motor with Positive Triggering (Act 19)

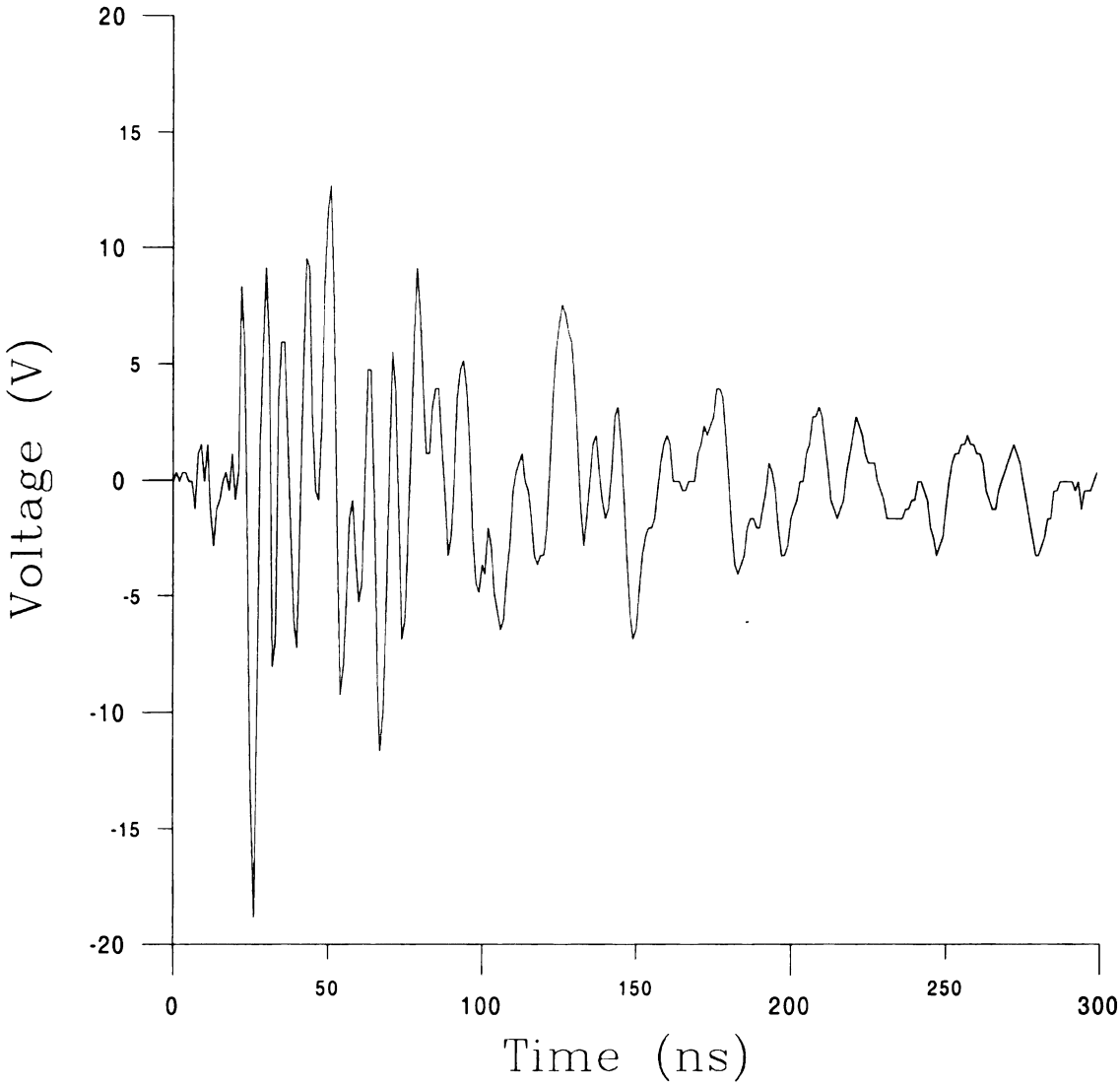

Figure 3-3: Conducted Emission Transient for Forward Biased Actuator Motor with Negative Triggering (Act 18)

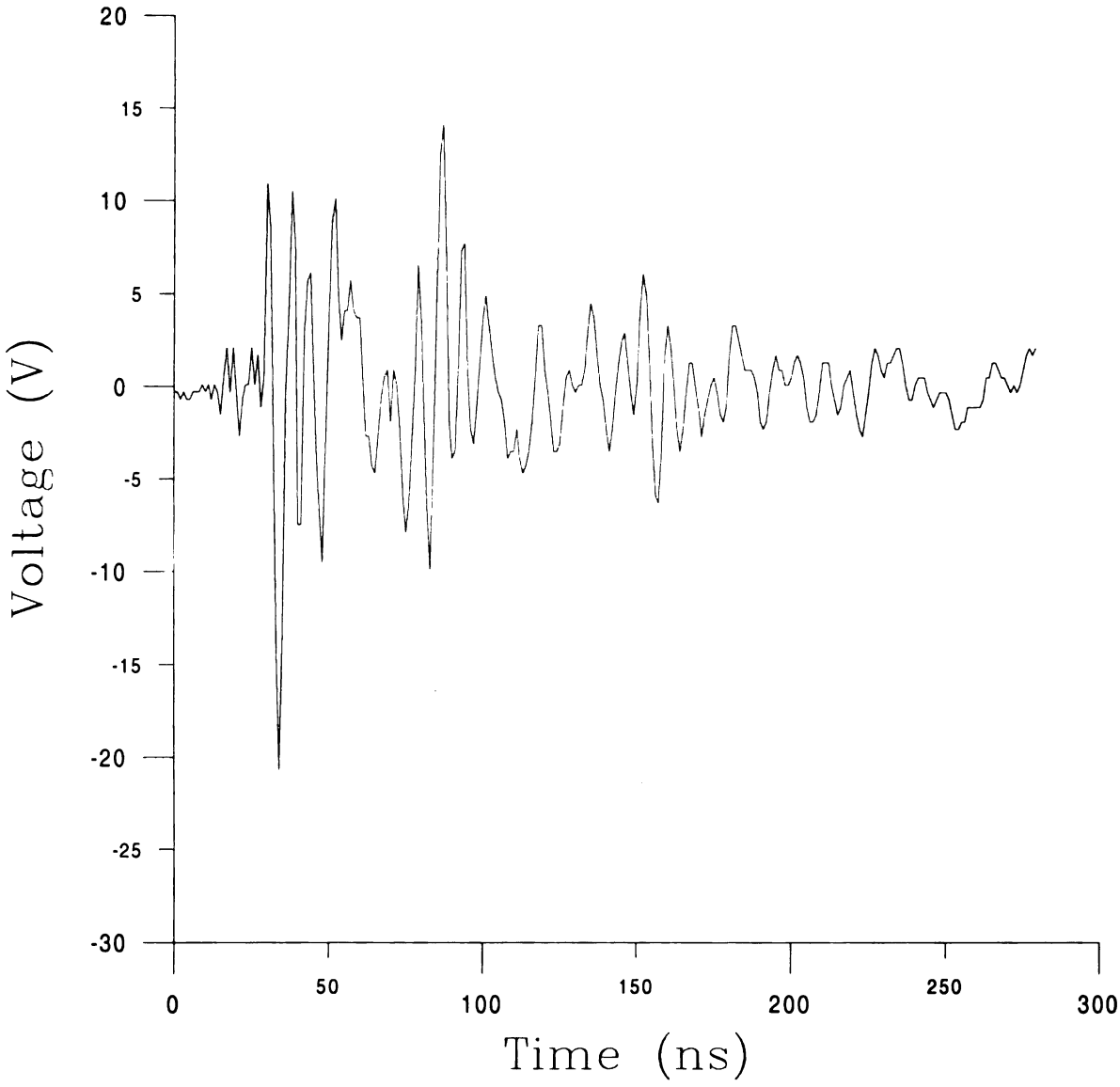

Figure 3-4: Conducted Emission Transient for Reverse Biased Actuator Motor with Positive Triggering (Act 21)

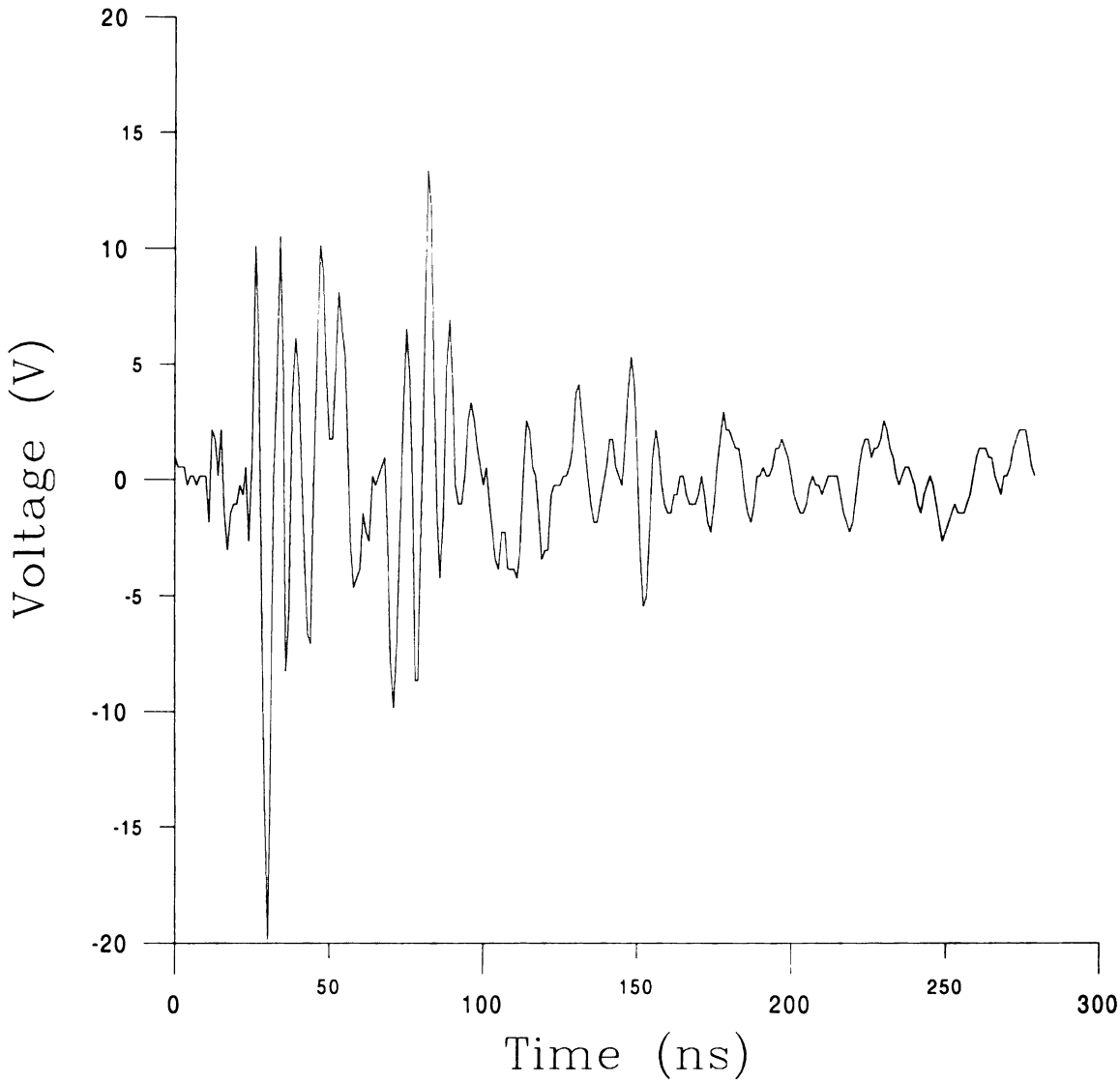

Figure 3-5: Conducted Emission Transient for Reverse Biased Actuator Motor with Negative Triggering (Act 20)

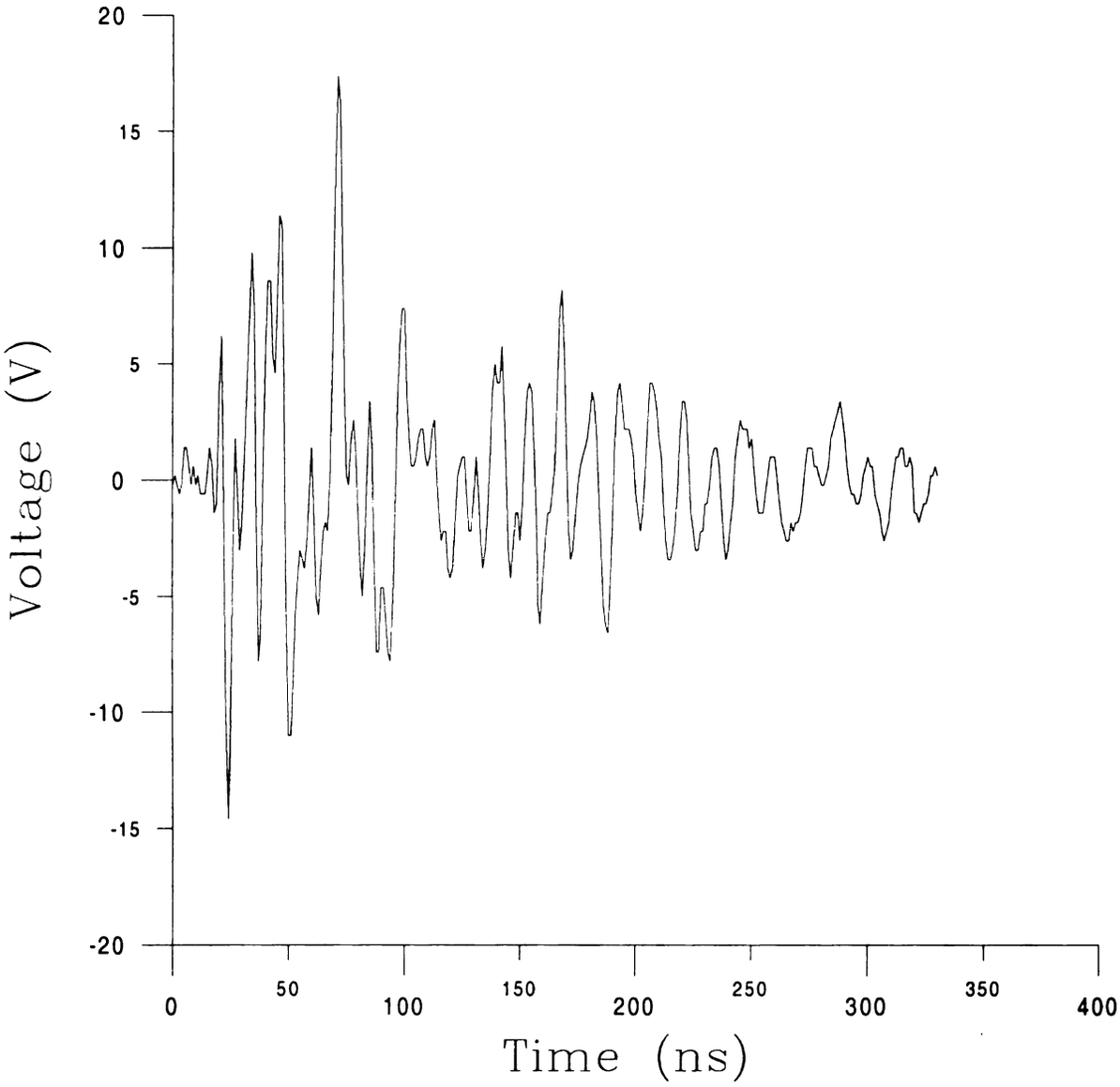

Figure 3-6: Conducted Emission Transient for Forward Biased Blower Motor with Positive Triggering (Blow 23)

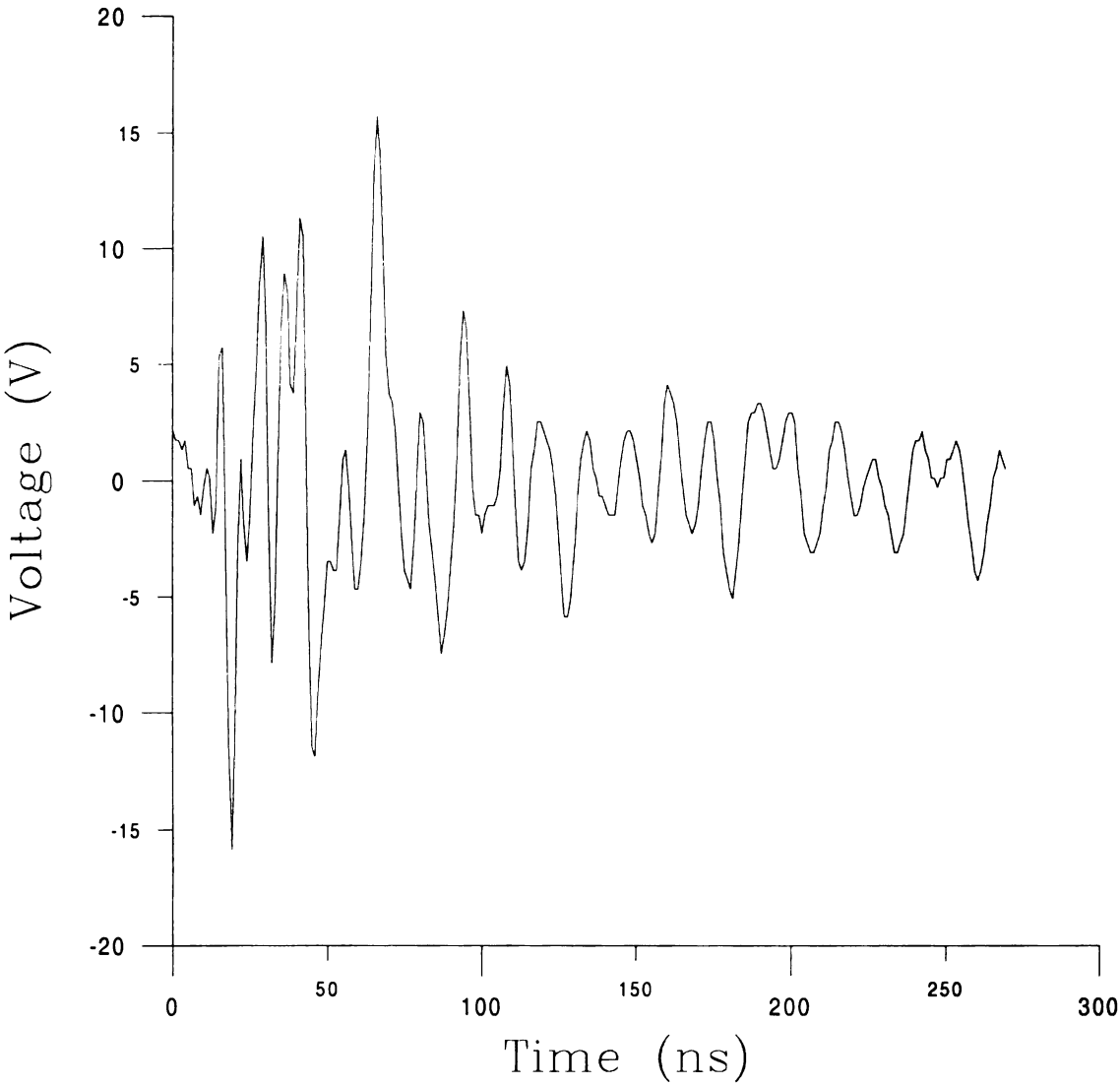

Figure 3-7: Conducted Emission Transient for Forward Biased Blower Motor with Negative Triggering (Blow 22)

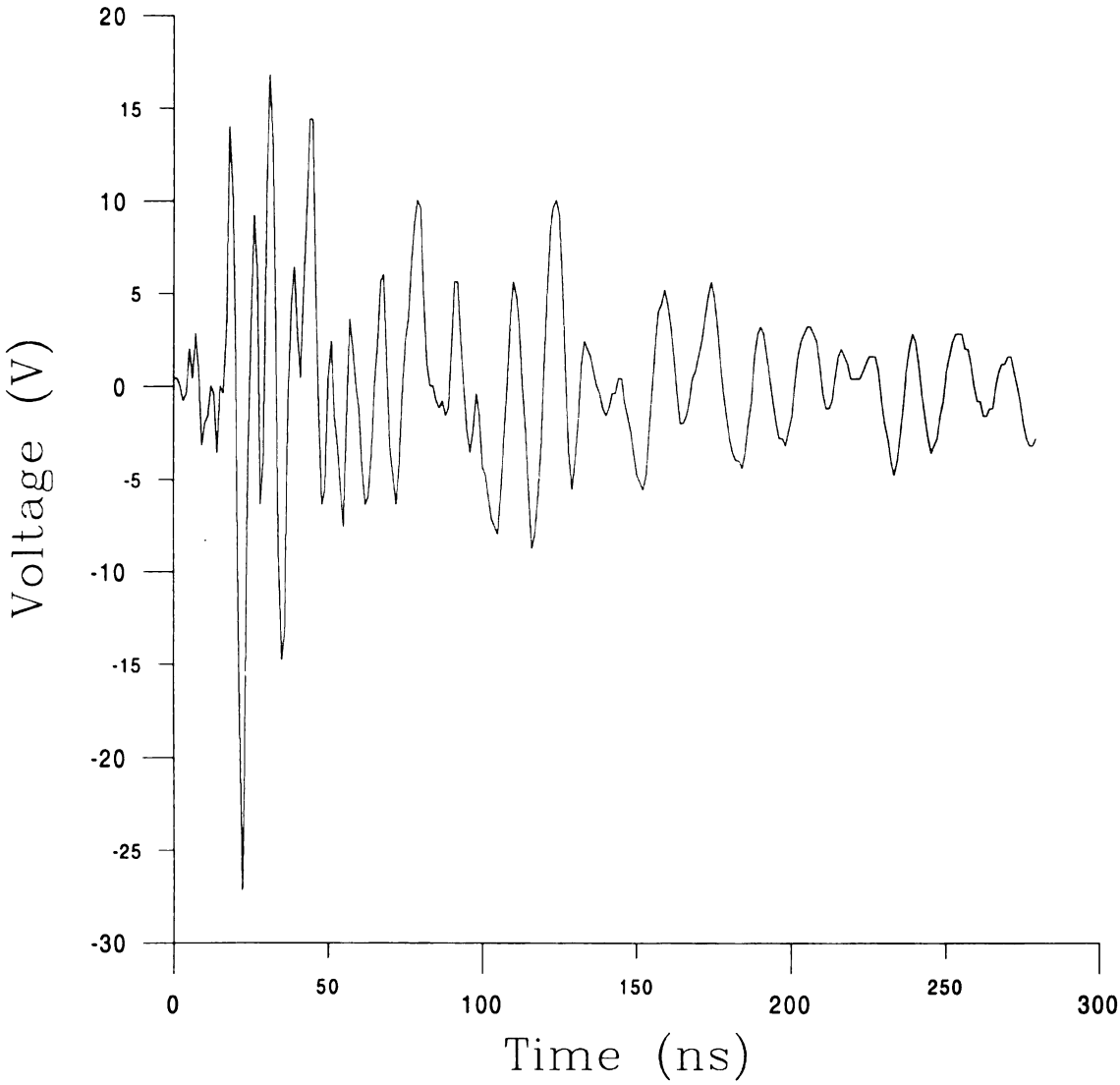

Figure 3-8: Conducted Emission Transient for Forward Biased, Stalled, Blue Line Motor with Positive Triggering (Blue 00)

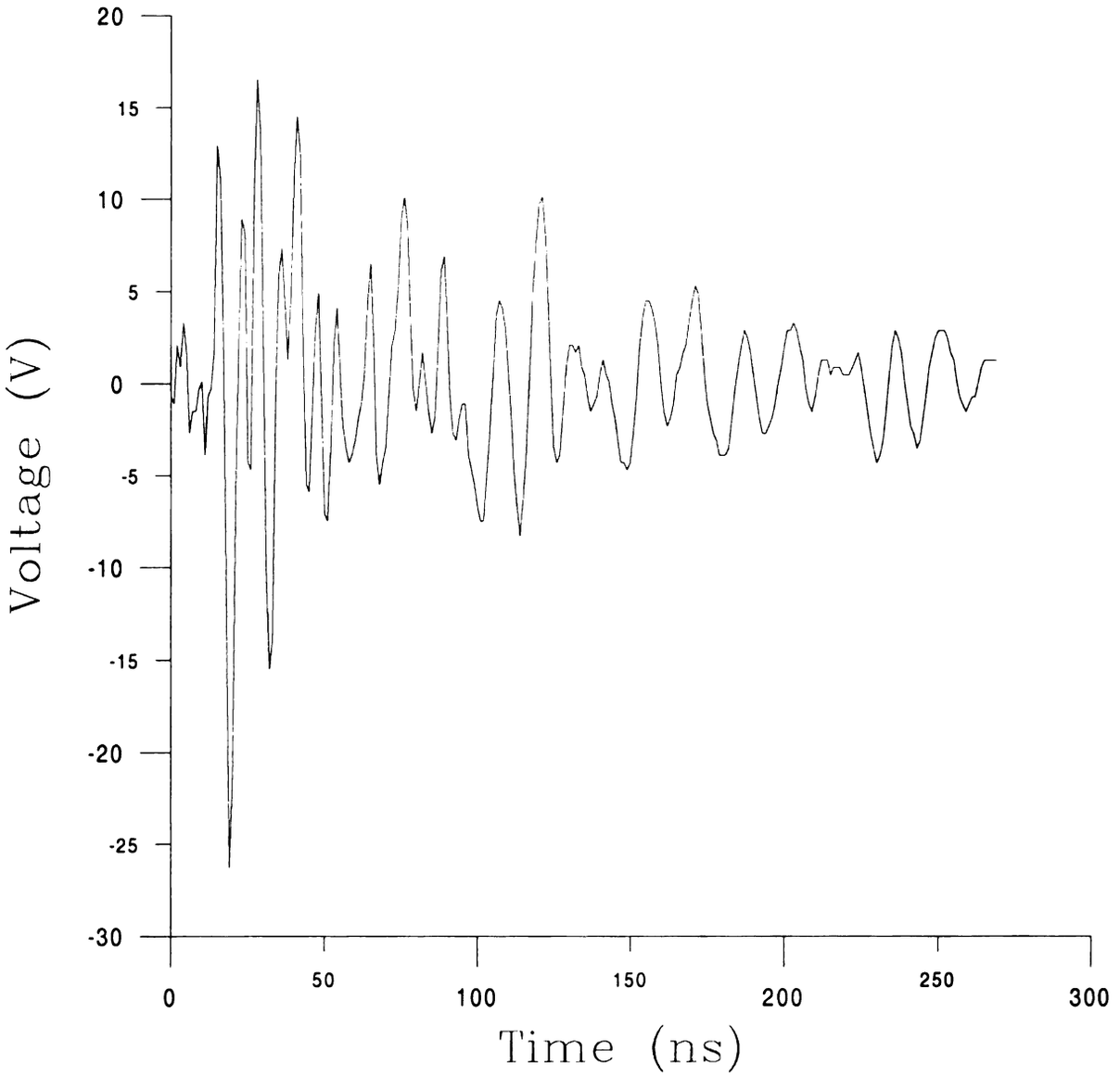

Figure 3-9: Conducted Emission Transient for Forward Biased, Stalled, Blue Line Motor with Negative Triggering (Blue 01)

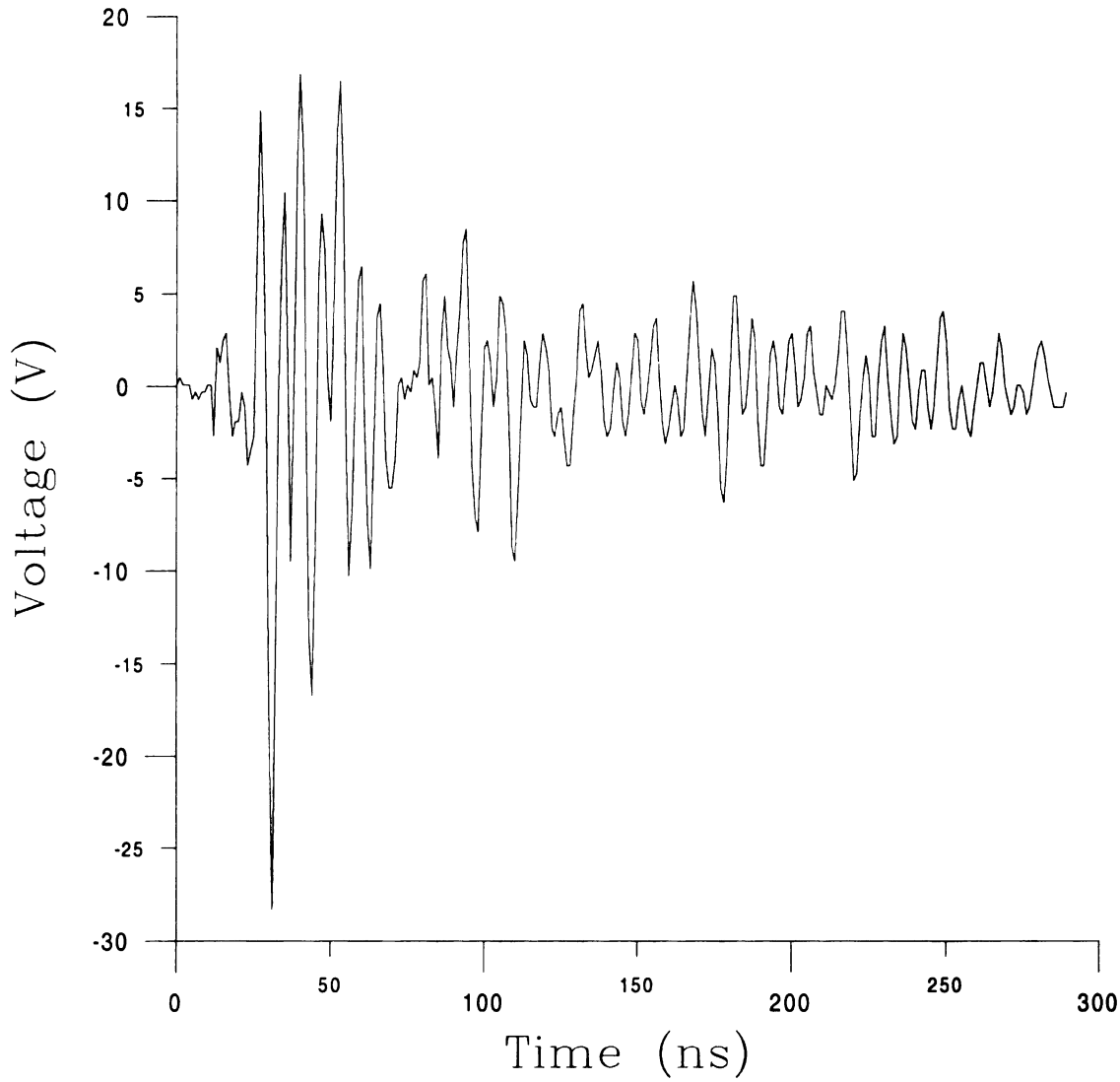

Figure 3-10: Conducted Emission Transient for Reverse Biased, Stalled, Blue Line Motor with Positive Triggering (Blue 05)

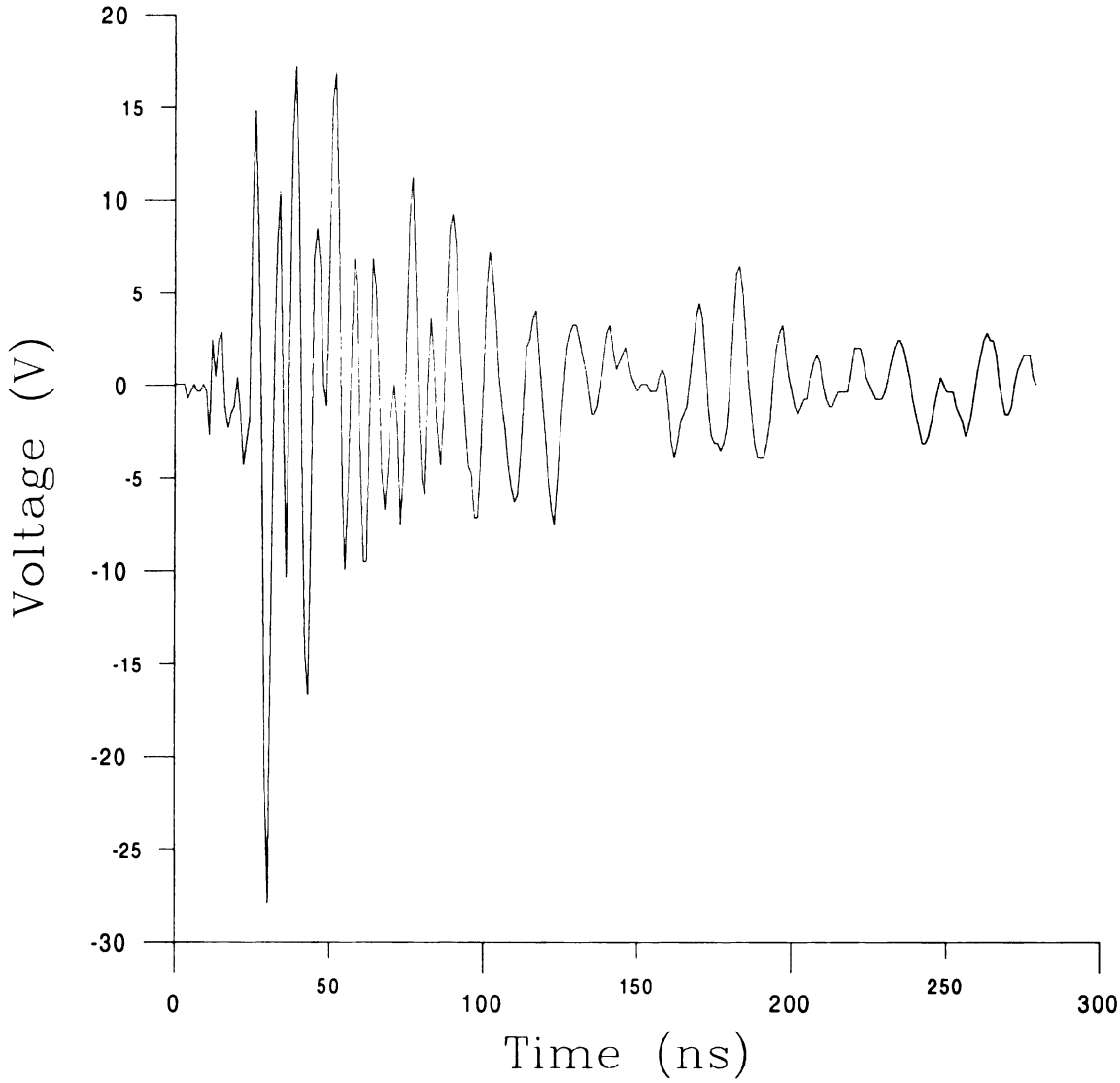

Figure 3-11: Conducted Emission Transient for Reverse Biased, Stalled, Blue Line Motor with Negative Triggering (Blue 04)

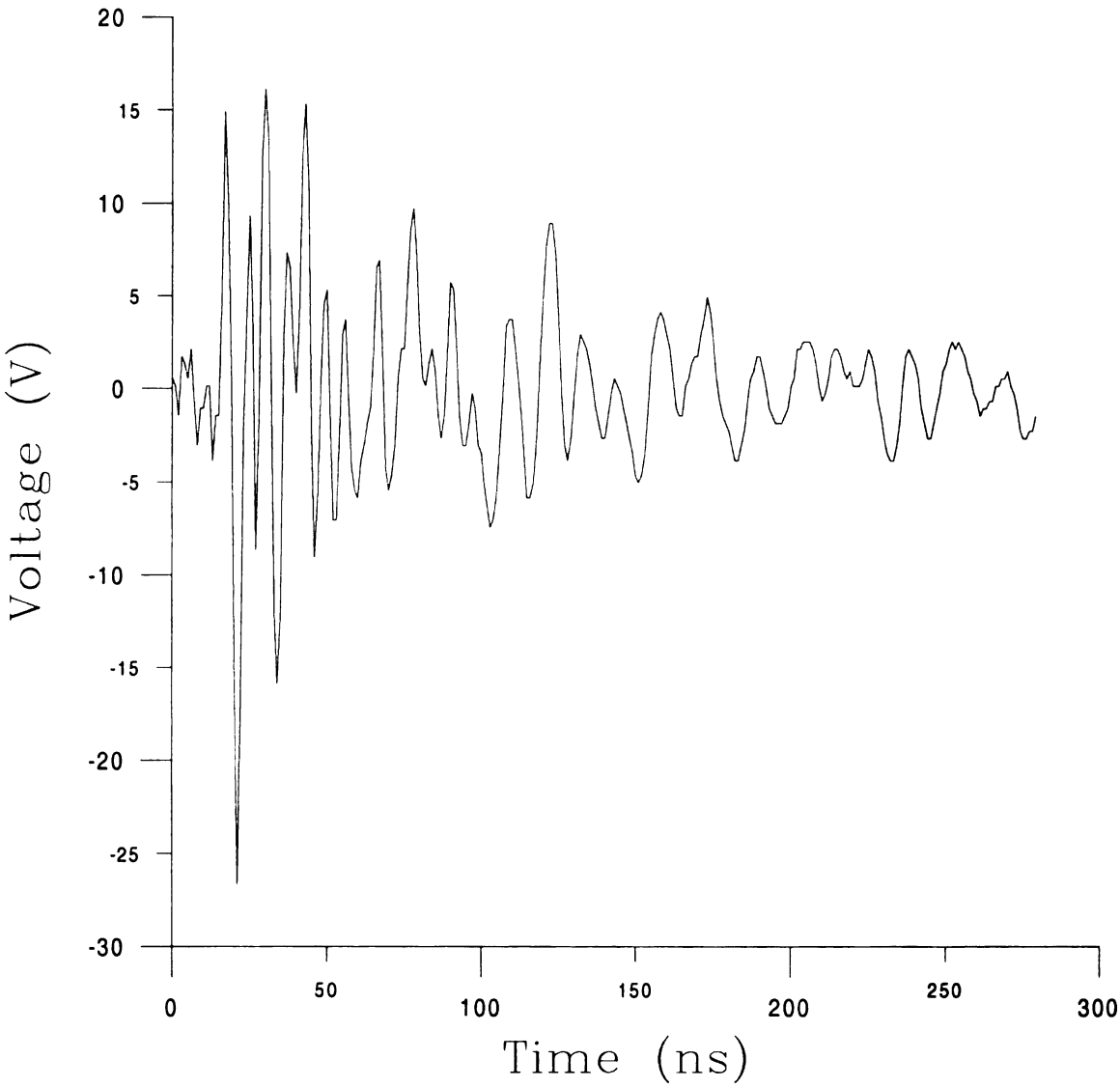

Figure 3-12: Conducted Emission Transient for Forward Biased, Traveling, Blue Line Motor with Positive Triggering (Blue 07)

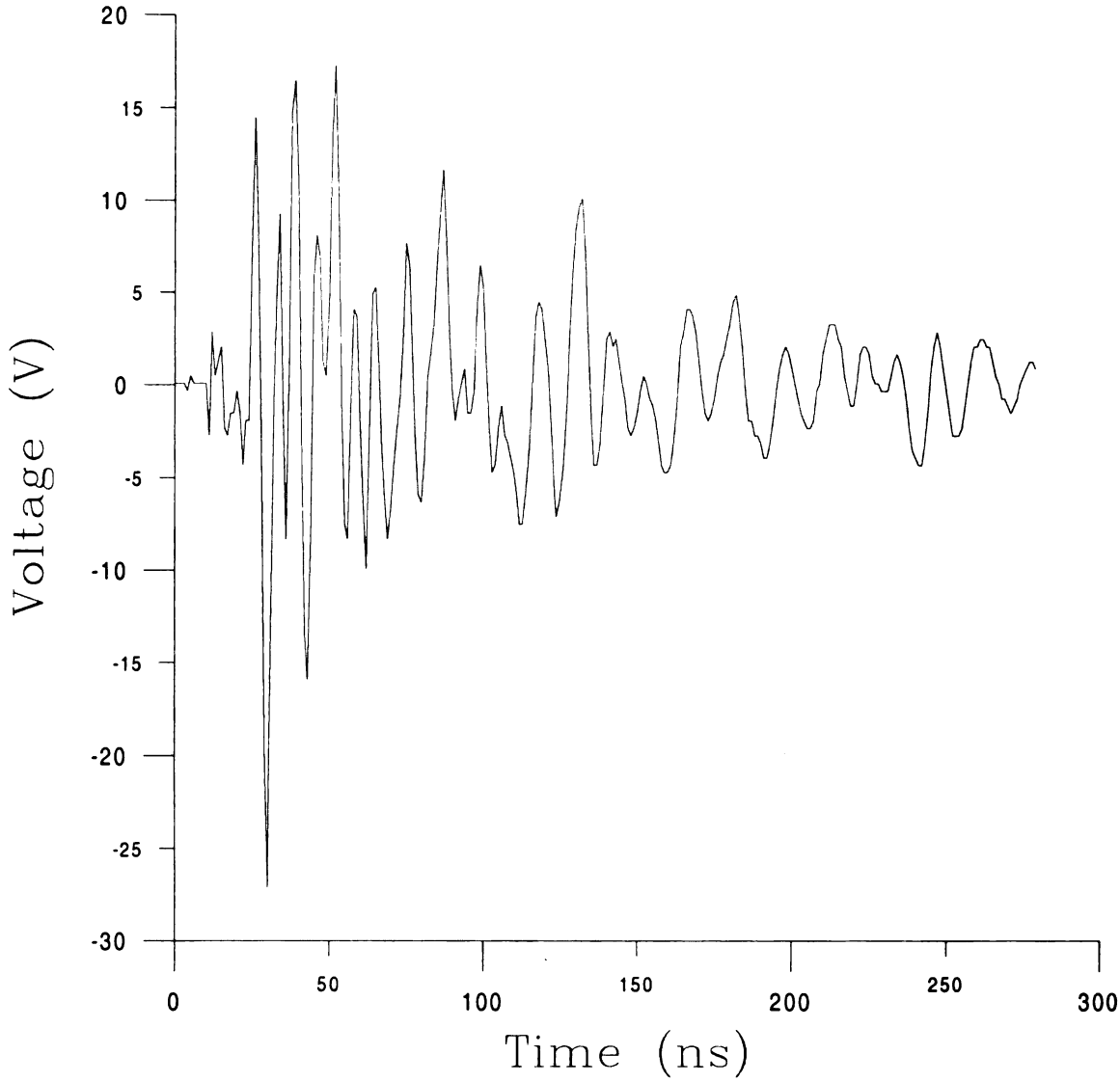

Figure 3-13: Conducted Emission Transient for Forward Biased, Traveling, Blue Line Motor with Negative Triggering (Blue 06)

Figu

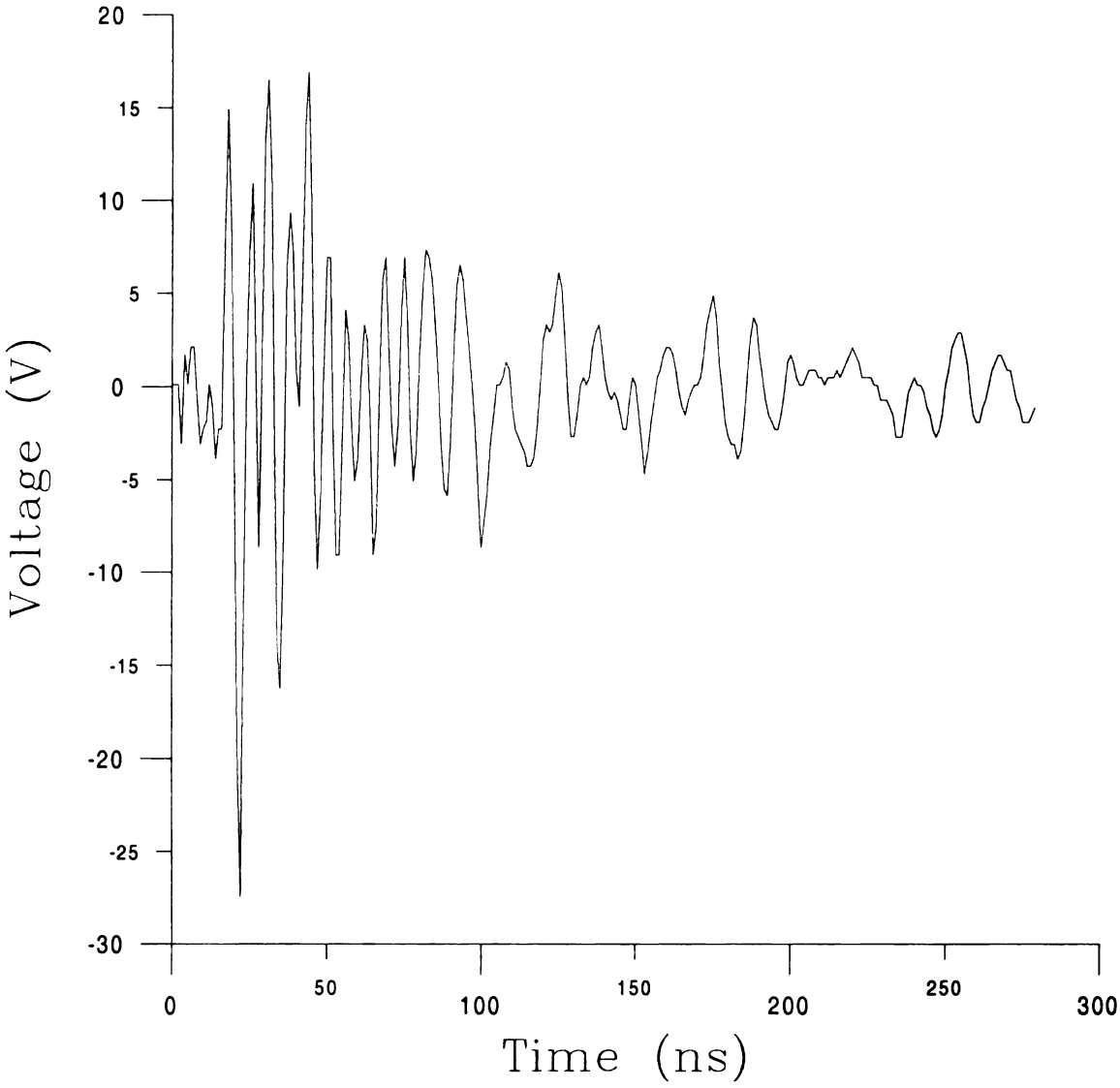

Figure 3-14: Conducted Emission Transient for Reverse Biased, Traveling, Blue Line Motor with Positive Triggering (Blue 03)

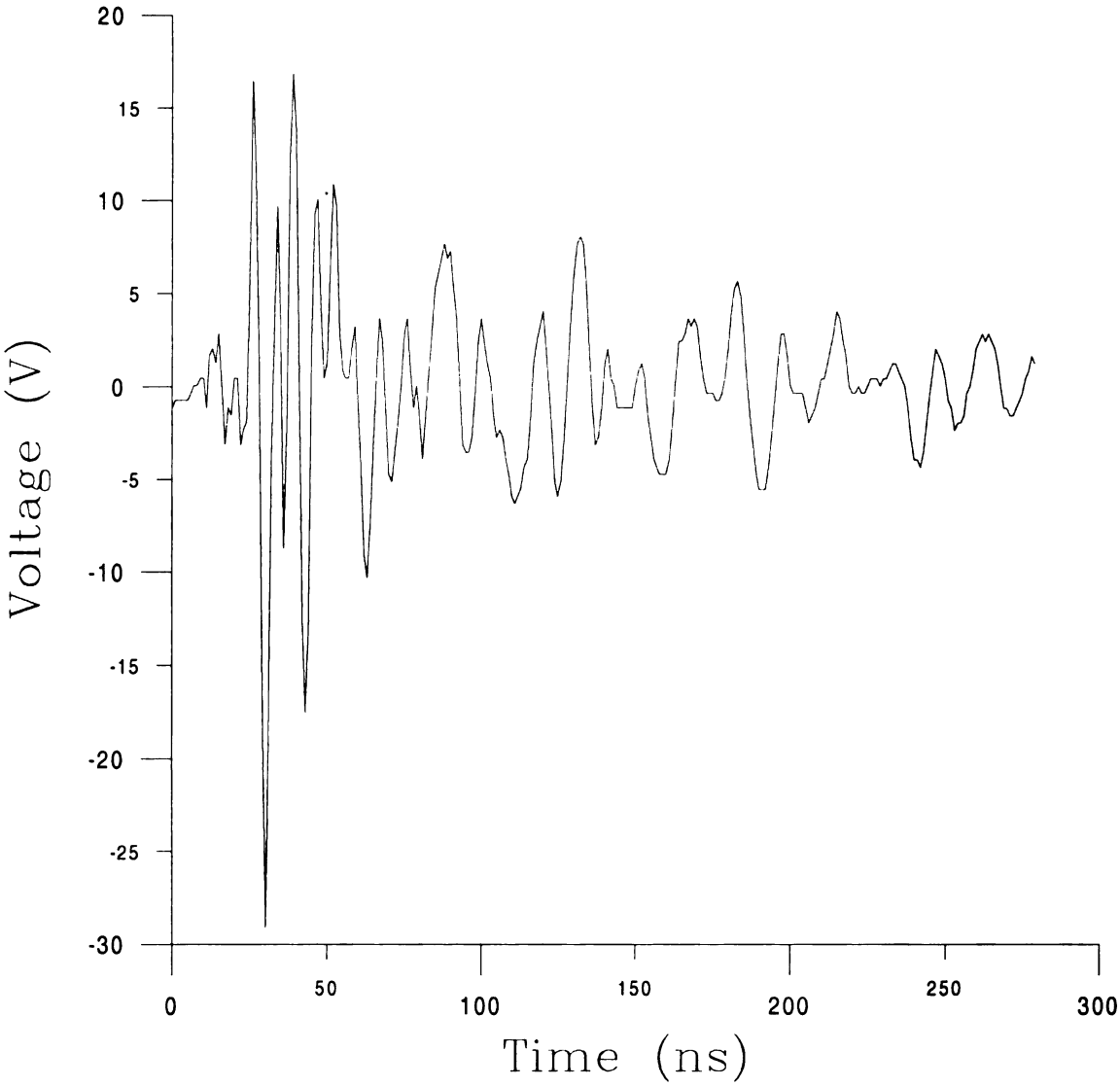

Figure 3-15: Conducted Emission Transient for Reverse Biased, Traveling, Blue Line Motor with Negative Triggering (Blue 02)

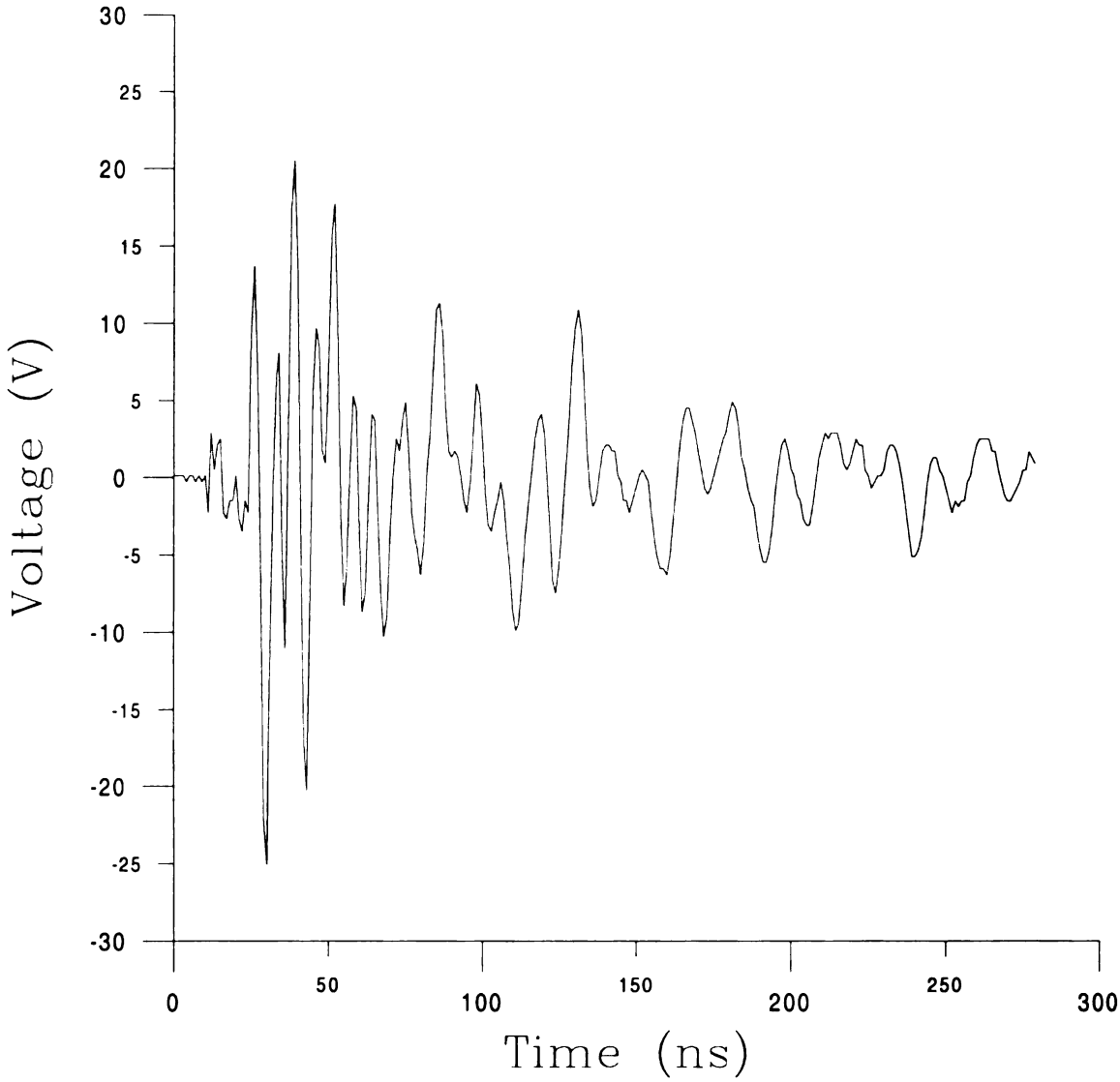

Figure 3-16: Conducted Emission Transient for Forward Biased, Stalled, Green Line Motor with Negative Triggering (Green 16)

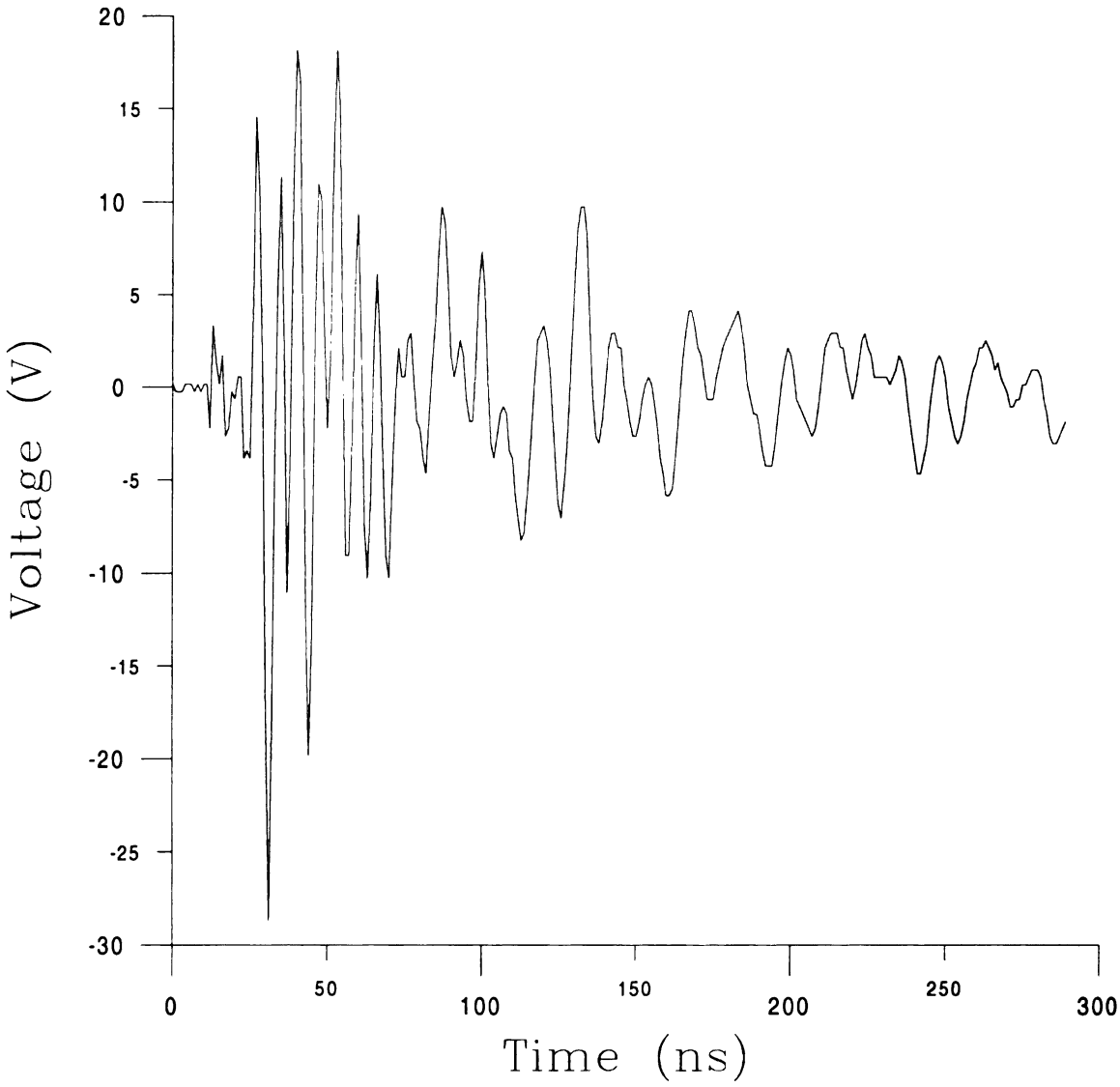

Figure 3-17: Conducted Emission Transient for Reverse Biased, Stalled, Green Line Motor with Positive Triggering (Green 13)

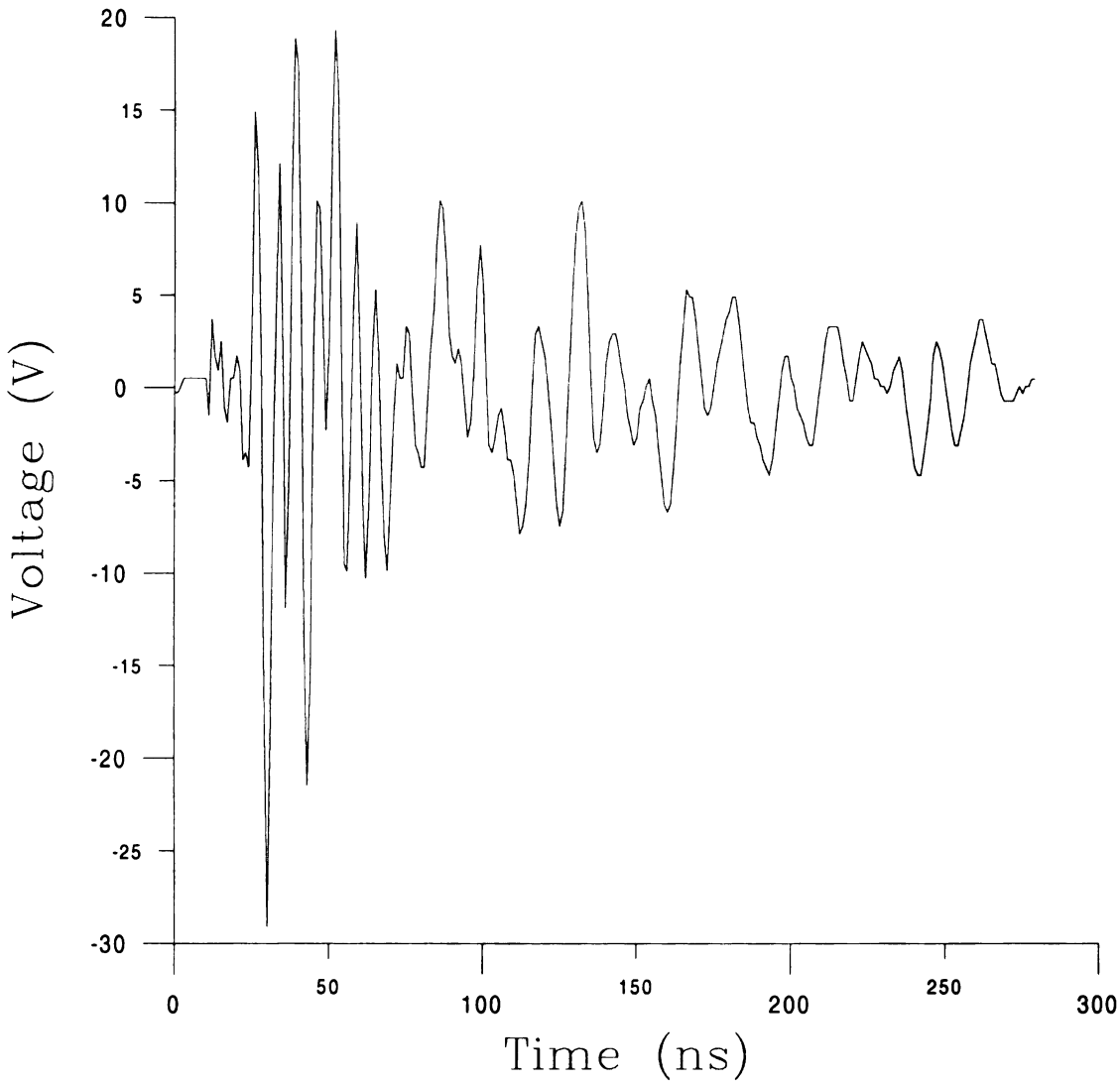

Figure 3-18: Conducted Emission Transient for Reverse Biased, Stalled, Green Line Motor with Negative Triggering (Green 12)

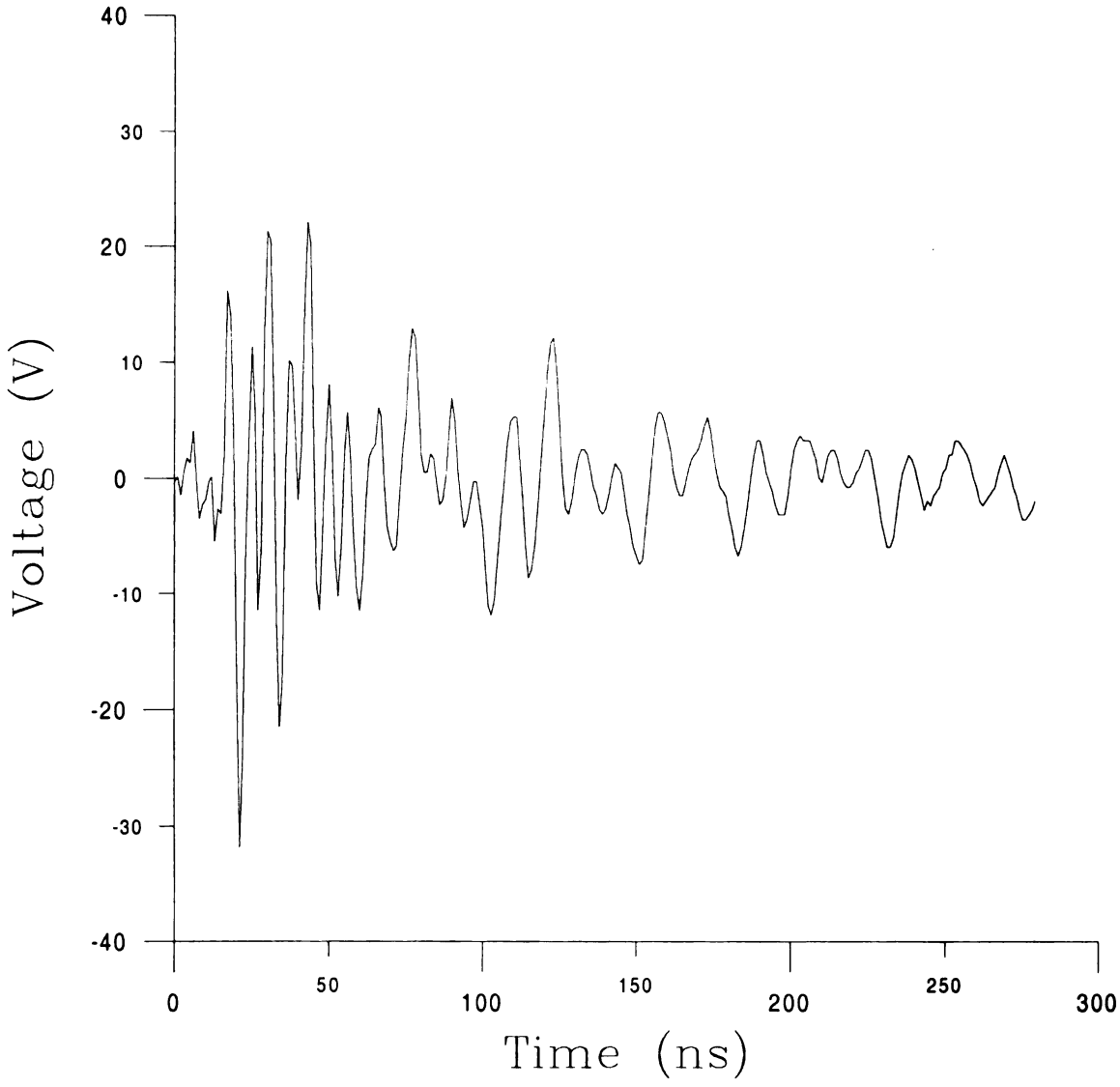

Figure 3-19: Conducted Emission Transient for Forward Biased, Traveling, Green Line Motor with Positive Triggering (Green 15)

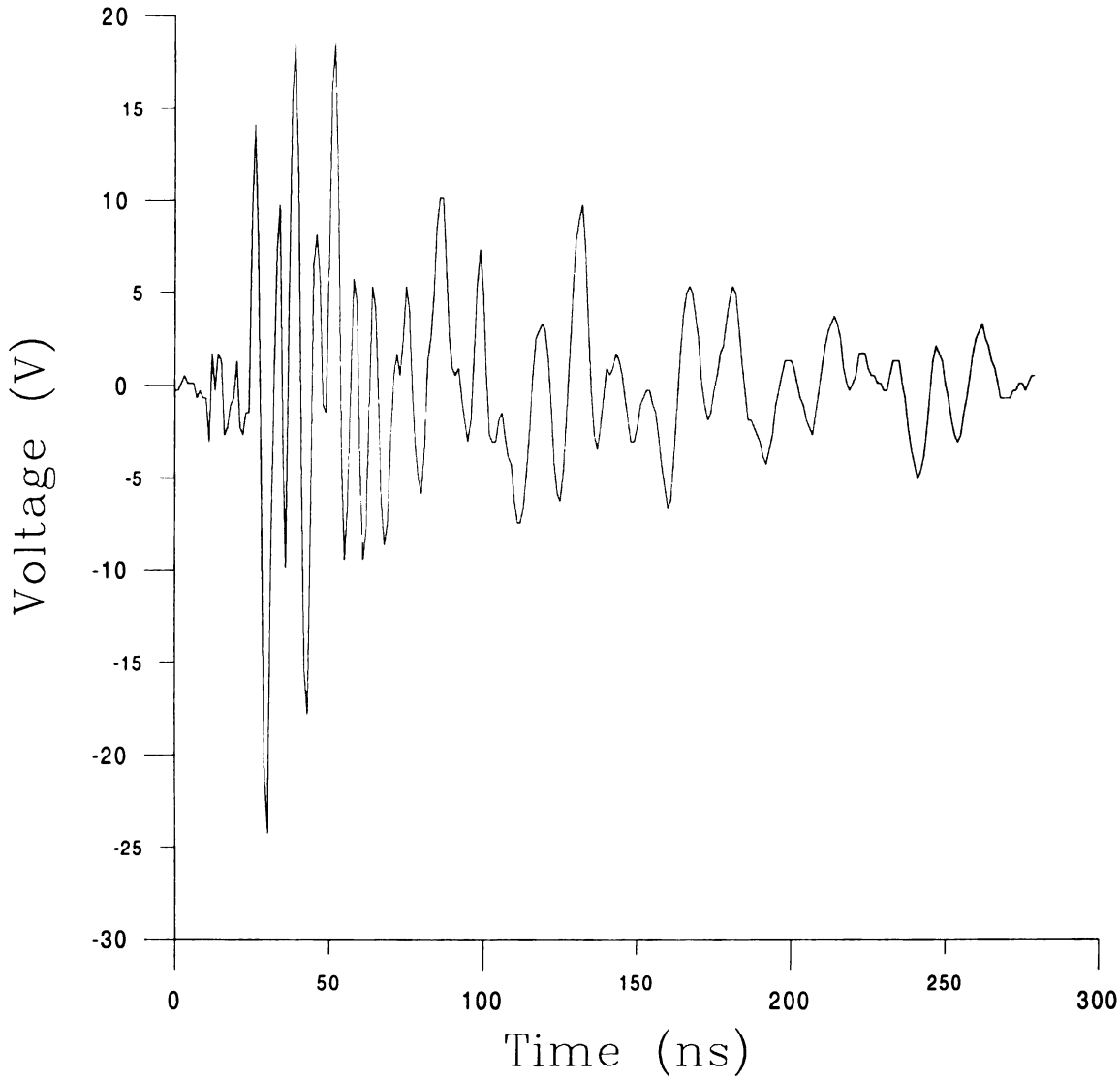

Figure 3-20: Conducted Emission Transient for Forward Biased, Traveling, Green Line Motor with Negative Triggering (Green 14)

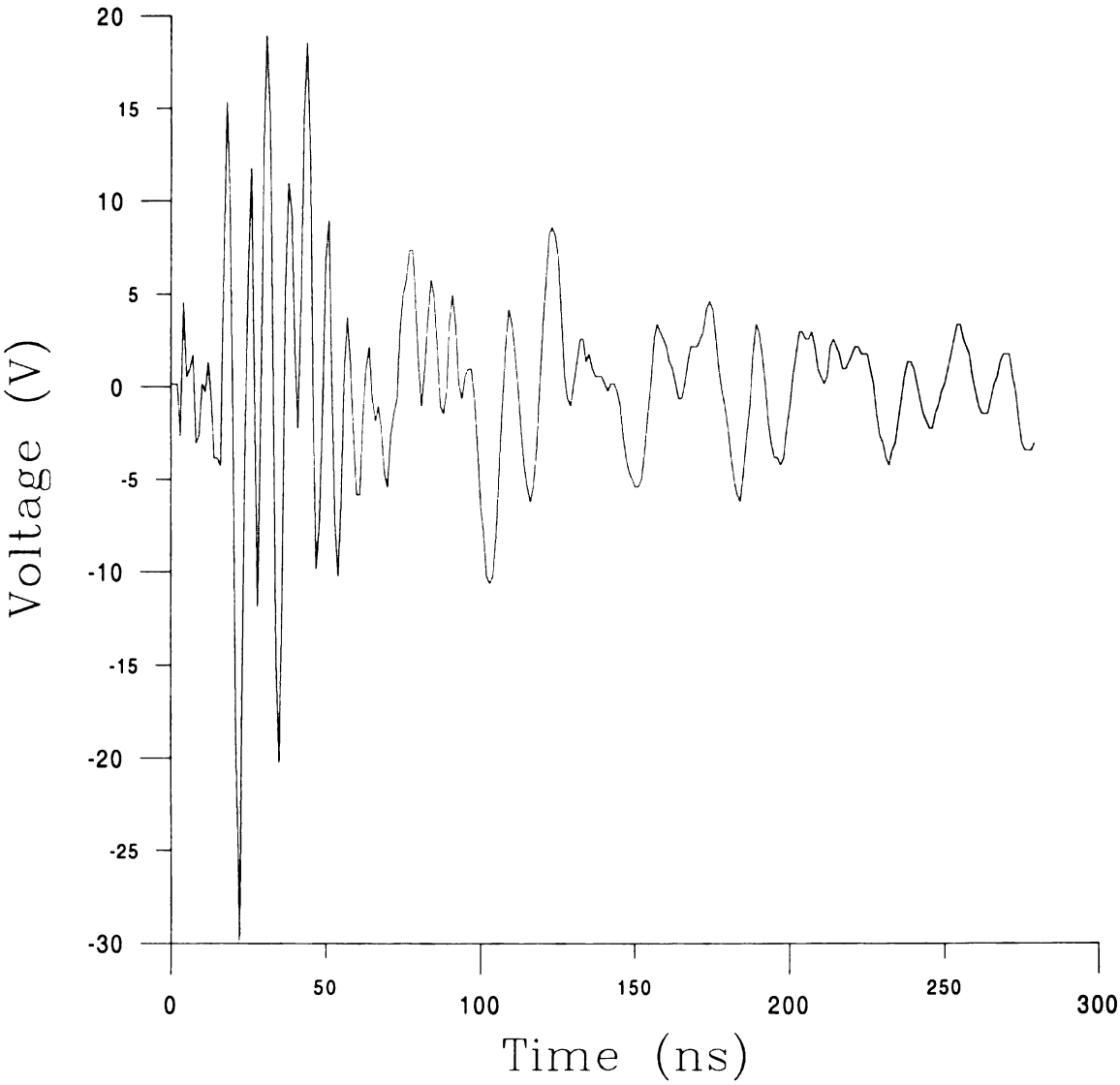

Figure 3-21: Conducted Emission Transient for Reverse Biased, Traveling, Green Line Motor with Positive Triggering (Green 11)

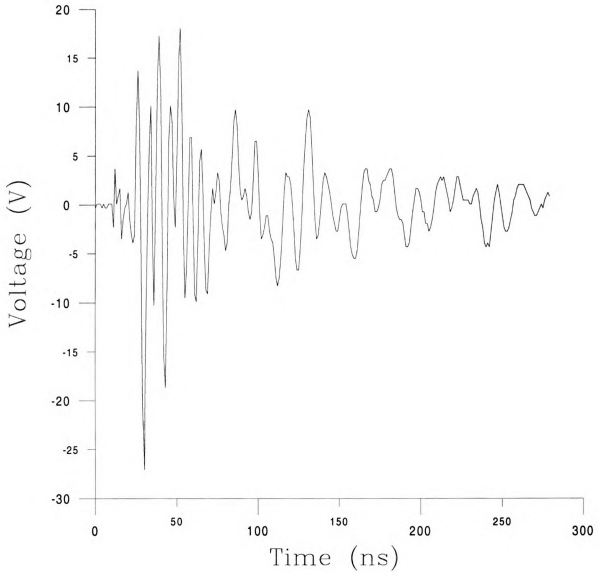

Figure 3-22: Conducted Emission Transient for Reverse Biased, Traveling, Green Line Motor with Negative Triggering (Green 10)

3.4 Interpretation

Chapter 4 discusses the modeling of the conducted emission waveforms that are measured in the previous section. These models accurately reflect the natural modes of each motor. However, before moving on to modeling the motor transients it is important to examine the data. The frequency content of these measurements is of particular interest. By taking the Fourier transform of each measured waveform, the frequency

content is observed. The following figures 3-23 to 3-26 show the frequency content of each motor. For each motor, all of the measured states are plotted on the same figure.

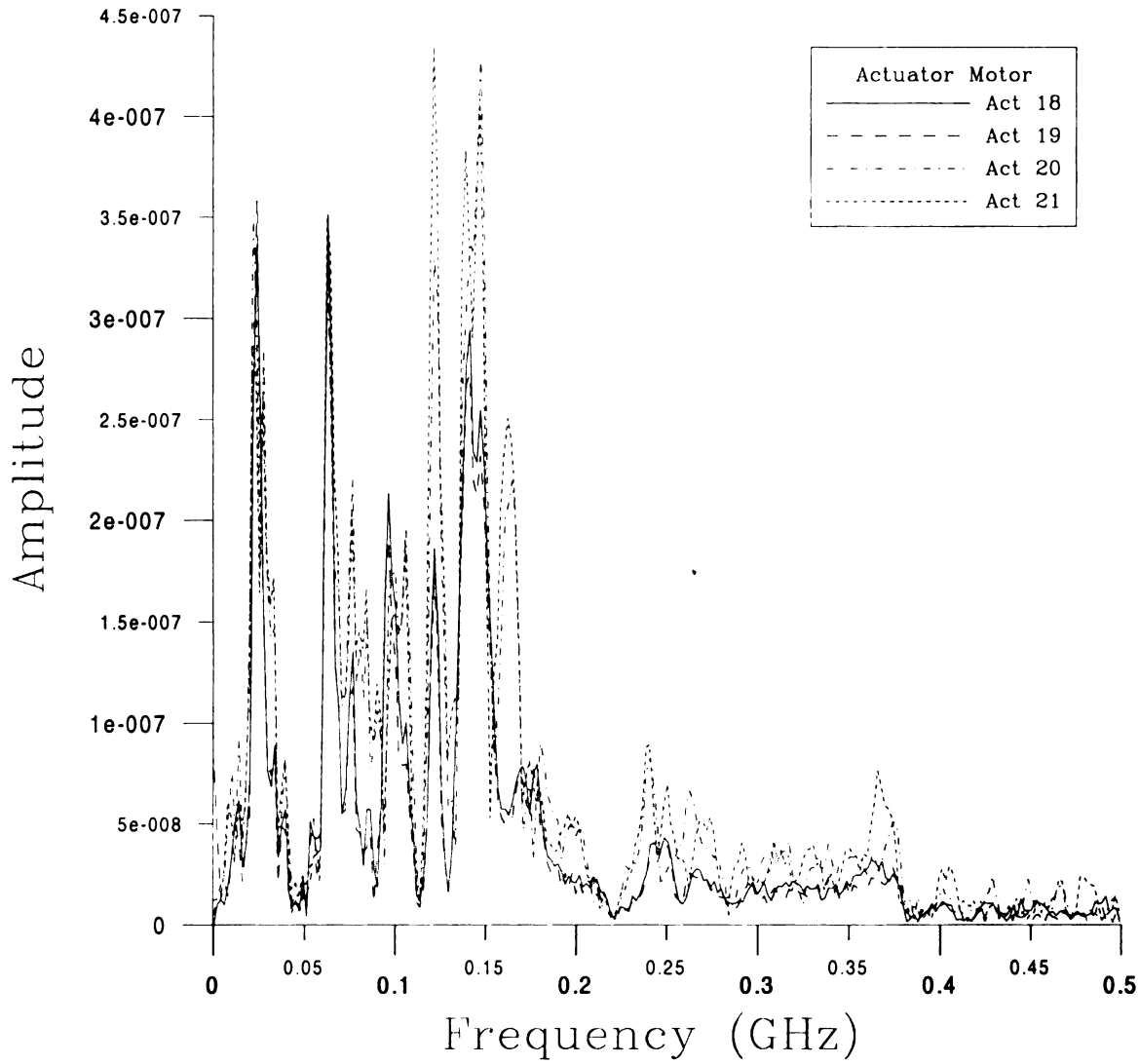

Figure 3-23: Frequency Content of the Actuator Motor

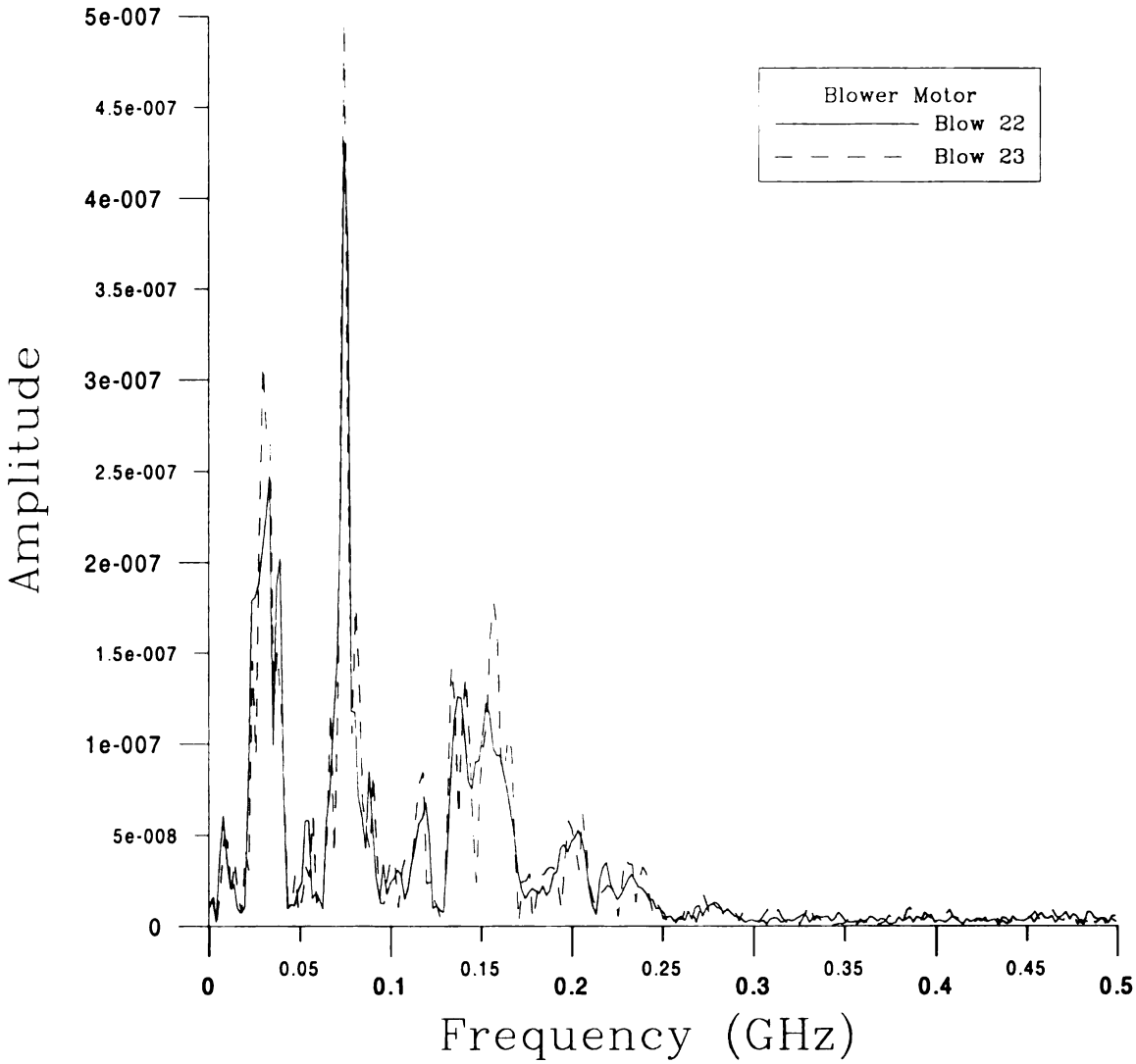

Figure 3-24: Frequency Content of the Blower Motor

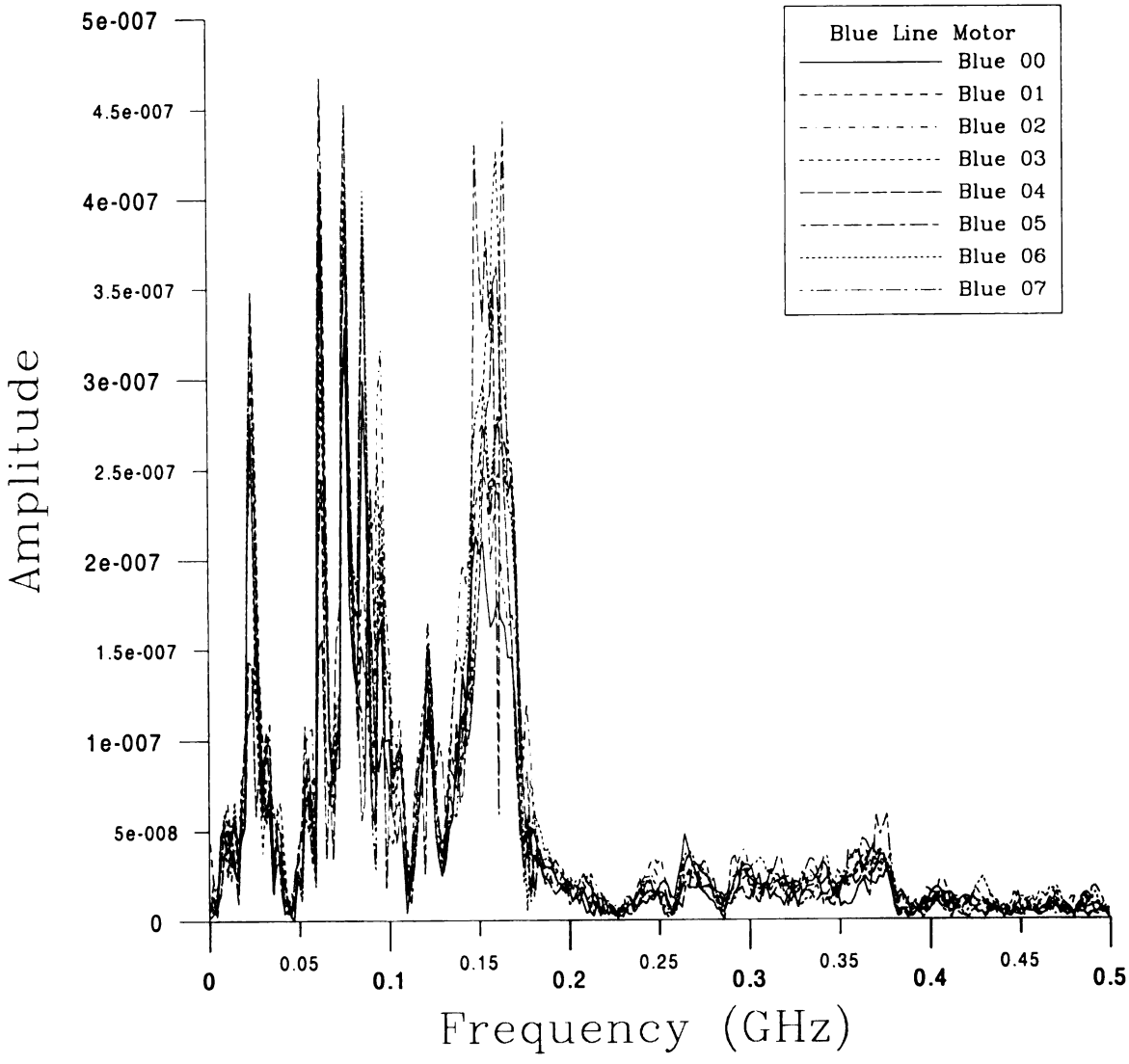

Figure 3-25: Frequency Content of the Blue Line Motor

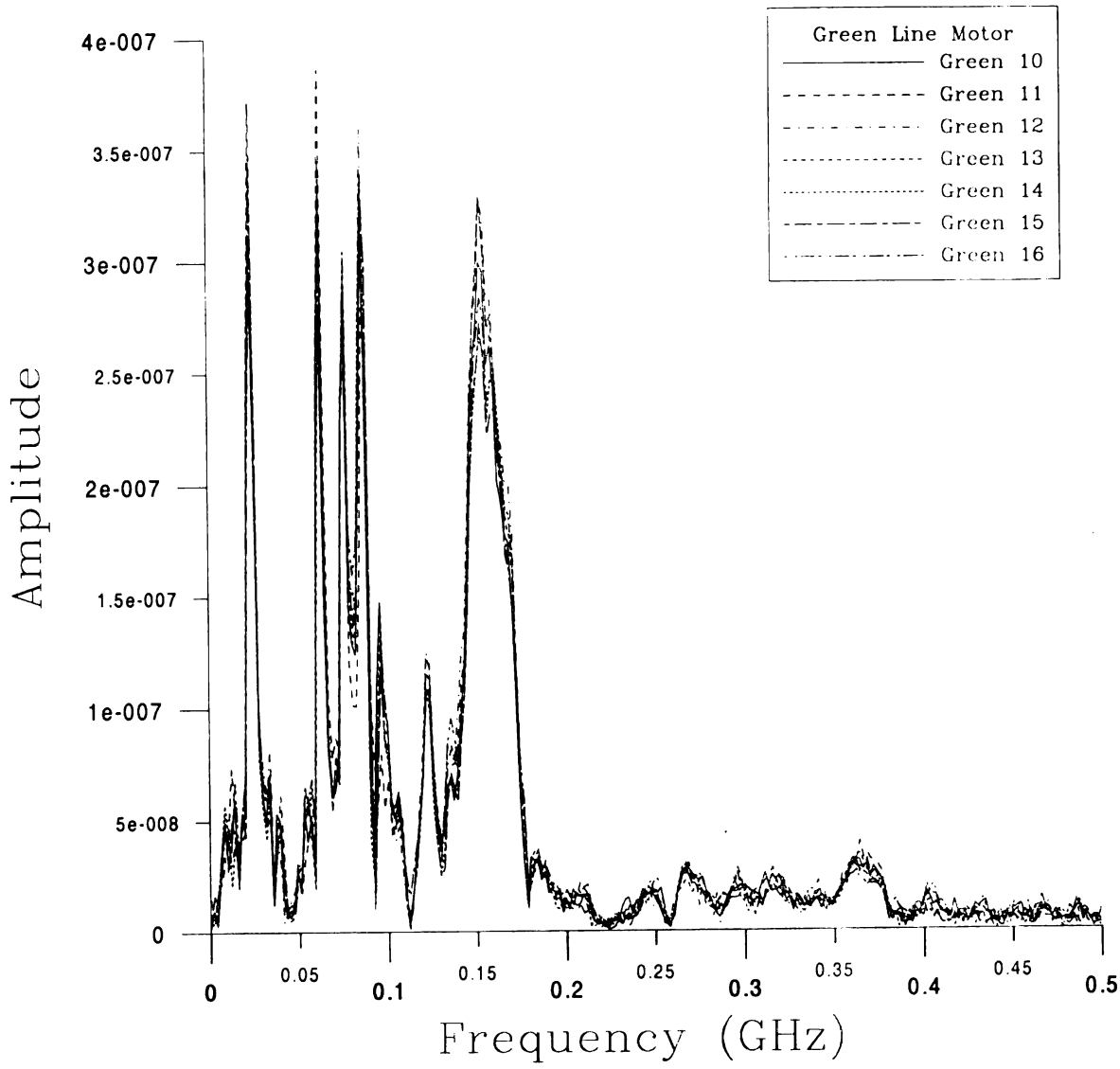

Figure 3-26: Frequency Content of the Green Line Motor

Figure 3-23

frequency

receiver for

display sum

or skilled. A

in mind, ma

motor type

Figure 3-23 shows that the triggering method used for measurements does not affect the frequency content of the motor. However, the actuator motor has different frequency behavior for forward and reverse biasing. Figures 3-24 through 3-26, on the other hand display similar frequency content for all motor states. Whether the motors are traveling or stalled, forward or reverse biased, the frequency content remains the same. With this in mind, modeling in Chapter 4 is done for only one representative motor state for each motor type.

CHAPTER 4

TRANSIENT ELECTRICAL MOTOR MODELS

4.1 Overview

This chapter discusses the modeling of transient electrical motors. It begins with a theoretical discussion of motor modeling [4,5]. A discussion of accomplishing this model using a computer program follows. Data is presented for transient motor models generated using the previous theory and computer program. Finally, the modeled data is discussed and compared to the measured data.

4.2 Derivation of the E-Pulse Method

It is assumed that the motor switching voltage transient can be represented as a finite sum of damped sinusoids.

$$m(t) = \sum_{n=1}^{N} a_n e^{\sigma_n t} \cos(\omega_n t + \phi_n), \qquad T_L < t < T_W \qquad \textbf{(4.1)}$$

where $s_n = \sigma_n + j\omega_n$ is the natural frequency of the nth mode, a_n and ϕ_n are the amplitude and phase of the nth mode, respectively, T_L is the beginning of the late time period, and T_W is the end of the measurement window. N is the number of natural modes and is chosen by inspecting the measured frequency content of each motor. Examining the frequency spectrum of the actuator motor voltage, as seen in Figure 4-1, it can be observed that this motor has five natural frequencies, which are represented by the five major peaks with arrows pointing to them.

Amplitude

3e-1

2e-1

1e-1

0

T

the motor

pulse, or

that satisf

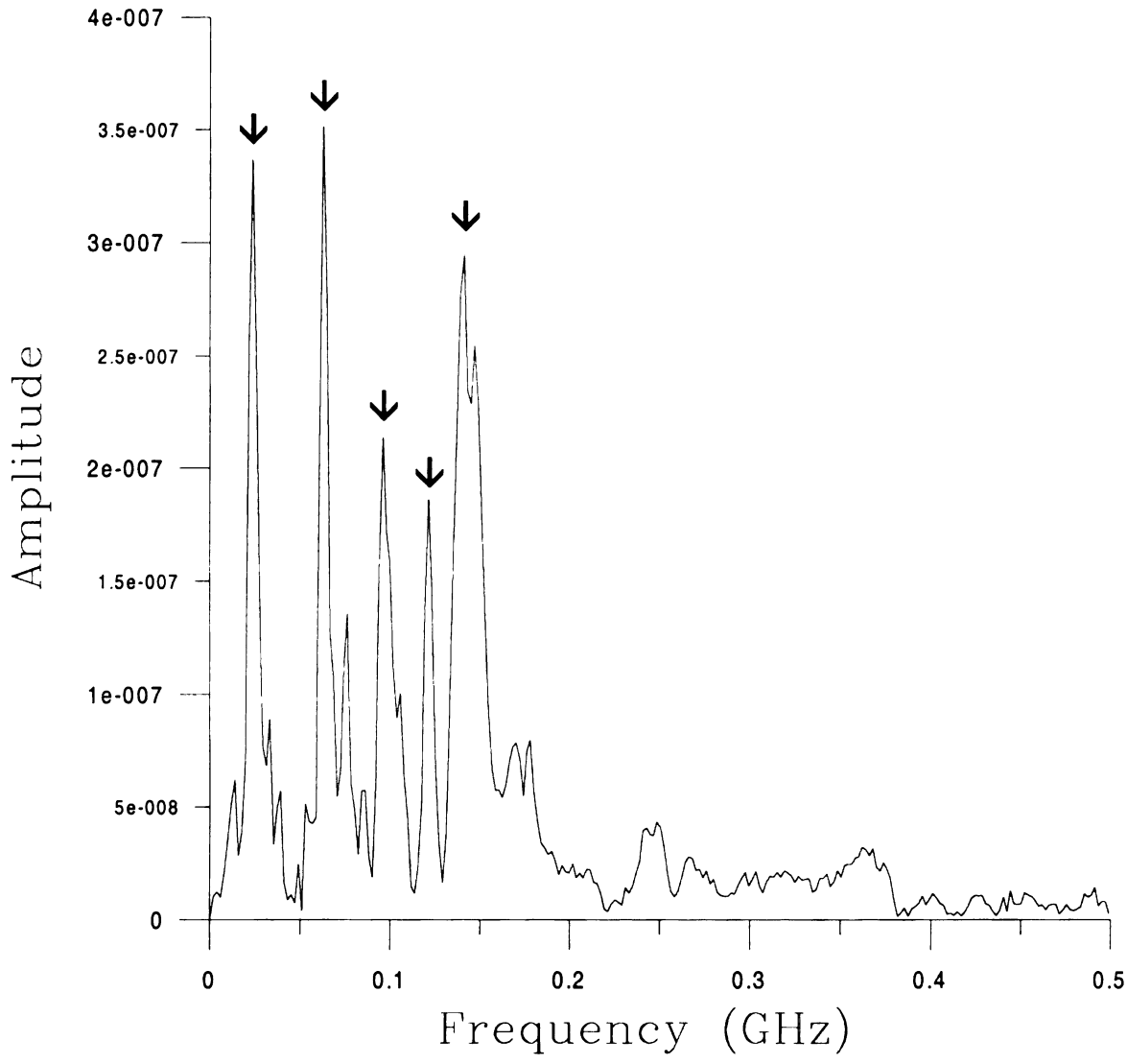

Figure 4-1: Natural Modes for the Actuator Motor

There exists a waveform function that yields a zero result when convolved with the motor switching transient, $m(t)$. This waveform function is known as an extinction pulse, or E-pulse. Thus, an E-pulse, $e(t)$ is defined as a waveform of finite duration, T_e, that satisfies

where

It is

a zero resu

requires

where

is the

$$
\begin{aligned}
c(t) &= e(t) * m(t) \\
&= \int_0^{T_e} e(t')m(t-t')dt' \\
&= \sum_{n=1}^{N} a_n e^{\sigma_n t}\left[A_n \cos(\omega_n t) + B_n \sin(\omega_n t)\right], \qquad t > T_L + T_e
\end{aligned} \tag{4.2}
$$

where

$$
\begin{Bmatrix} A_n \\ B_n \end{Bmatrix} = \int_0^{T_e} e(t') e^{-\sigma_n t'} \begin{Bmatrix} \cos\omega_n t' \\ \sin\omega_n t' \end{Bmatrix} dt'. \tag{4.3}
$$

It is desirable that the E-pulse convolved with the motor switching transient yield a zero result. Thus

$$
c(t) = e(t) * m(t) = 0, \qquad t > T_L + T_e \tag{4.4}
$$

requires

$$
A_n = B_n = 0, \qquad 1 \le n \le N \tag{4.5}
$$

This convolution waveform can also be written in the form

$$
c(t) = \sum_{n=1}^{N} a_n \left|E(s_n)\right| e^{\sigma_n t} \cos(\omega_n t + \psi_n), \qquad t > T_L + T_e \tag{4.6}
$$

where

$$
E(s_n) = L\{e(t)\} = \int_0^{T_e} e(t) e^{-s_n t} dt, \qquad n = 1, 2, ..., N \tag{4.7}
$$

*is th*e Laplace transform of the E-pulse waveform, and

$$
\psi_n = \phi_n + \tan^{-1}\left(-\frac{E_{in}}{E_{rn}}\right) \tag{4.8}
$$

where

$$E_{in} = \mathrm{Im}\{E(s_n)\} \qquad E_{rn} = \mathrm{Re}\{E(s_n)\} \tag{4.9}$$

Since $c(t) = 0$ for $t > T_L + T_e$,

$$E_{in} = E_{rn} = 0 \qquad 1 \le n \le N \tag{4.10}$$

which also requires

$$E(s_n) = E(s_n^*) = 0 \qquad 1 \le n \le N \tag{4.11}$$

Now, $e(t)$ is broken into two components,

$$e(t) = e^f(t) + e^e(t), \tag{4.12}$$

where $e^f(t)$ is a forcing component that excites a waveform, and $e^e(t)$ is an extinction component that extinguishes the response due to $e^f(t)$. The forcing component is user defined, while the extinction component is determined by expanding $e^e(t)$ into a set of basis functions

$$e^e(t) = \sum_{m=1}^{M} \alpha_m f_m(t) \tag{4.13}$$

Thus, equation (4.12) becomes

$$e(t) = e^f(t) + \sum_{m=1}^{M} \alpha_m f_m(t) \tag{4.14}$$

where $M = 2N$ and $\{f_m(t)\}$ are defined as unit pulse basis functions with unknown durations, and $\{\alpha_m\}$ are the unknown amplitudes of the basis functions.

The solution of the basis function pulse widths and amplitudes is solved using the program *EP.FOR*, written by Dr. Ed Rothwell, Dr. Ponniah Ilavarasan, and Dr. John E. Ross III. This program defines the total width of the e-pulse as T_k, and the number of basis functions as $2N + 1$ (the forcing function plus $2N$ extinction components). The program iteratively steps through a series of E-pulse widths and the basis function amplitudes are computed at each iteration. Substituting equation (4.14) into equation (4.4) yields

$$c(t) = \left[e^f(t) + \sum_{m=1}^{M} \alpha_m f_m(t) \right] * m(t) = 0, \qquad t > T_L + T_e. \quad \textbf{(4.15)}$$

This equation is solved in the program, forcing $c(t) = 0$ at chosen time points. It must be noted that $c(t)$ is forced to equal zero only at these matching points and it is very possible that $c(t) \neq 0$ at points in between the matching points. Thus the program generates unique values for α_m for each iteration of T_k.

Now that the extinction pulse is known, it must be used to reconstruct the measured waveform. The first step for this reconstruction is determining the natural frequencies. Taking the Laplace Transform of equation (4.13) yields

$$\begin{aligned} E^e(s_n) &= \sum_{m=1}^{M} \alpha_m L\{f_m(t)\} \\ &= \sum_{m=1}^{M} \alpha_m F_m(s_n) \qquad n = 1, 2, \ldots, N \end{aligned} \quad \textbf{(4.16)}$$

The extinction pulse must be zero at the natural frequencies, thus

$$E(s_n) = \sum_{m=1}^{M} \alpha_m F_m(s_n) = 0, \qquad n = 1, 2, \ldots, N \quad \textbf{(4.17)}$$

Equation (4.17) can be solved for each s_n using the definition of the Laplace transform.

$$F_m(s_n) = \int_{-\infty}^{\infty} f_m(t) e^{-s_n t} dt, \qquad n = 1, 2, ..., N \tag{4.18}$$

The pulse $f_m(t)$ has a duration of $\Delta = \dfrac{T_k}{2N+1}$. Thus equation (4.18) can be rewritten as

$$\begin{aligned} F_m(s_n) &= \int_{(m-1)\Delta}^{m\Delta} 1 e^{-s_n t} dt \\ &= -\frac{1}{s}\left[e^{-s_n t} \right]_{(m-1)\Delta}^{m\Delta} \\ &= \frac{e^{-s_n m\Delta} - e^{-s_n (m-1)\Delta}}{s_n} \\ &= e^{-s_n m\Delta} \frac{1 - e^{-s_n \Delta}}{s_n}, \qquad n = 1, 2, ..., N \end{aligned} \tag{4.19}$$

Now, substituting the result from equation (4.19) into equation (4.17) yields

$$E(s_n) = \sum_{m=1}^{M} \alpha_m \left(\frac{1 - e^{-s_n \Delta}}{s_n} \right) e^{-s_n m\Delta} = 0, \qquad n = 1, 2, ..., N \tag{4.20}$$

Now, moving $\left(\dfrac{1 - e^{-s_n \Delta}}{s_n} \right)$ outside the summation and letting $z_n = e^{-s_n \Delta}$ the problem reduces to a polynomial equation

$$\sum_{m=1}^{M} \alpha_m z_n^m = 0 \qquad n = 1, 2, ..., N \tag{4.21}$$

with known α_m already determined by the program. The polynomial in equation (4.21) is solved for each z_n, which can now be used to determine the natural frequencies.

$$\begin{aligned} z_n &= e^{-s_n \Delta} \\ \ln z_n &= -s_n \Delta \\ s_n &= -\frac{1}{\Delta} \ln z_n \end{aligned} \tag{4.22}$$

With the natural frequencies now known, it is possible to recreate the waveform. Let $r(t)$ be the recreated waveform

$$r(t) = \sum_{n=1}^{N} A_n e^{\sigma_n t} \cos(\omega_n t + \phi_n) \tag{4.23}$$

where σ_n and ω_n are the real and imaginary parts of the natural frequencies respectively

$$s_n = \sigma_n + j\omega_n \tag{4.24}$$

and A_n and ϕ_n are the unknown amplitude and phase of the reconstructed waveform. Equation (4.18) can be rewritten as

$$r(t) = \sum_{n=1}^{N} e^{\sigma_n t} \left[A_n \cos \omega_n t + B_n \sin \omega_n t \right] \tag{4.25}$$

A_n and B_n can be determined by using least squares curve fitting to the measured data. This gives a reconstructed waveform for each value of T_k. The iteration with the least squared error is chosen as optimal and given as output from the program.

4.3 Results and Discussion of Modeling for Each Motor

Modeling for each motor is accomplished by first choosing a representative data set for each motor type. Data set Act 18 is chosen for the actuator motor while data sets Blow 23, Blue 00, and Green 10 are chosen for the blower motor, blue line motor, and green line motor respectively. Next, the frequency content of each motor is examined and a number of expected modes is determined. Often the actual number of expected modes is not readily apparent, so a range of possible natural modes is modeled for each motor.

By examining the frequency content of the actuator motor in Figure 2-10 it is seen that the actuator has four to six significant natural modes. Thus, Figures 4-2 through 4-7 show the modeled actuator motor for six, five and four natural modes. Examining the frequency content of the modeled data for six and five expected natural modes in Figures 4-5 and 4-6, it can be seen that five natural modes are modeled in both cases. This is reflected in the nearly identical modeled time-domain data they produce, which is seen in Figures 4-2 and 4-3. Additionally, it is noted that the frequency data beyond 100 MHz does not match up nearly as well as the data below 100 MHz. The modeled time-domain data seen in Figure 4-4, assuming four expected modes, however, does not match up with the measured data nearly as well. This is reflected in the frequency data as the three natural modes below 100 MHz match up very well, while there is very little agreement between measured and modeled data beyond 100 MHz.

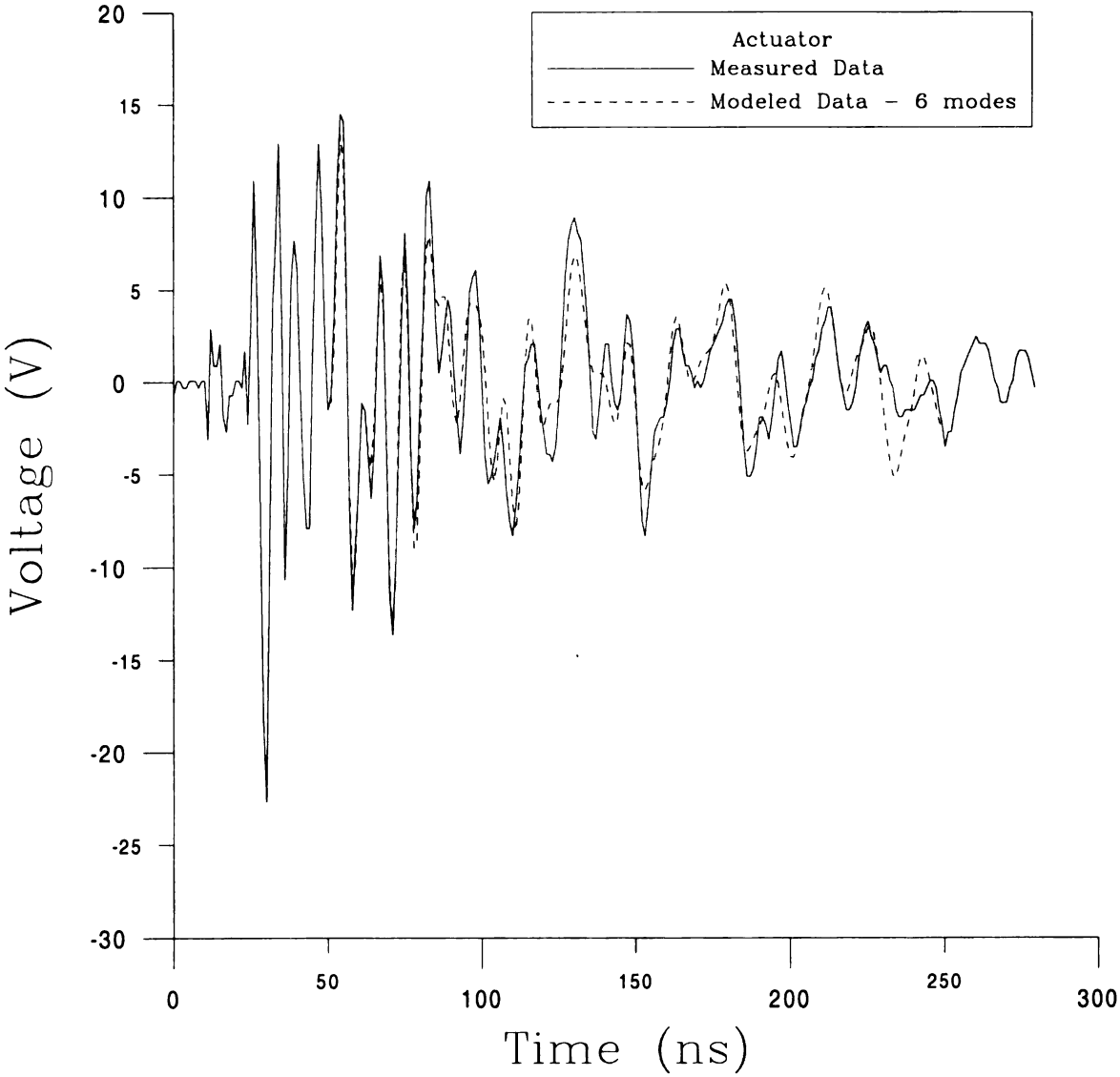

Figure 4-2: Modeled Actuator Motor Data Assuming 6 Natural Modes

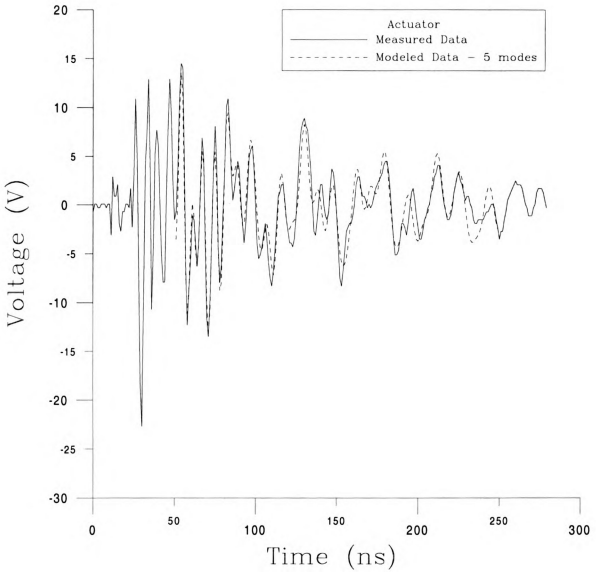

Figure 4-3: Modeled Actuator Motor Data Assuming 5 Natural Modes

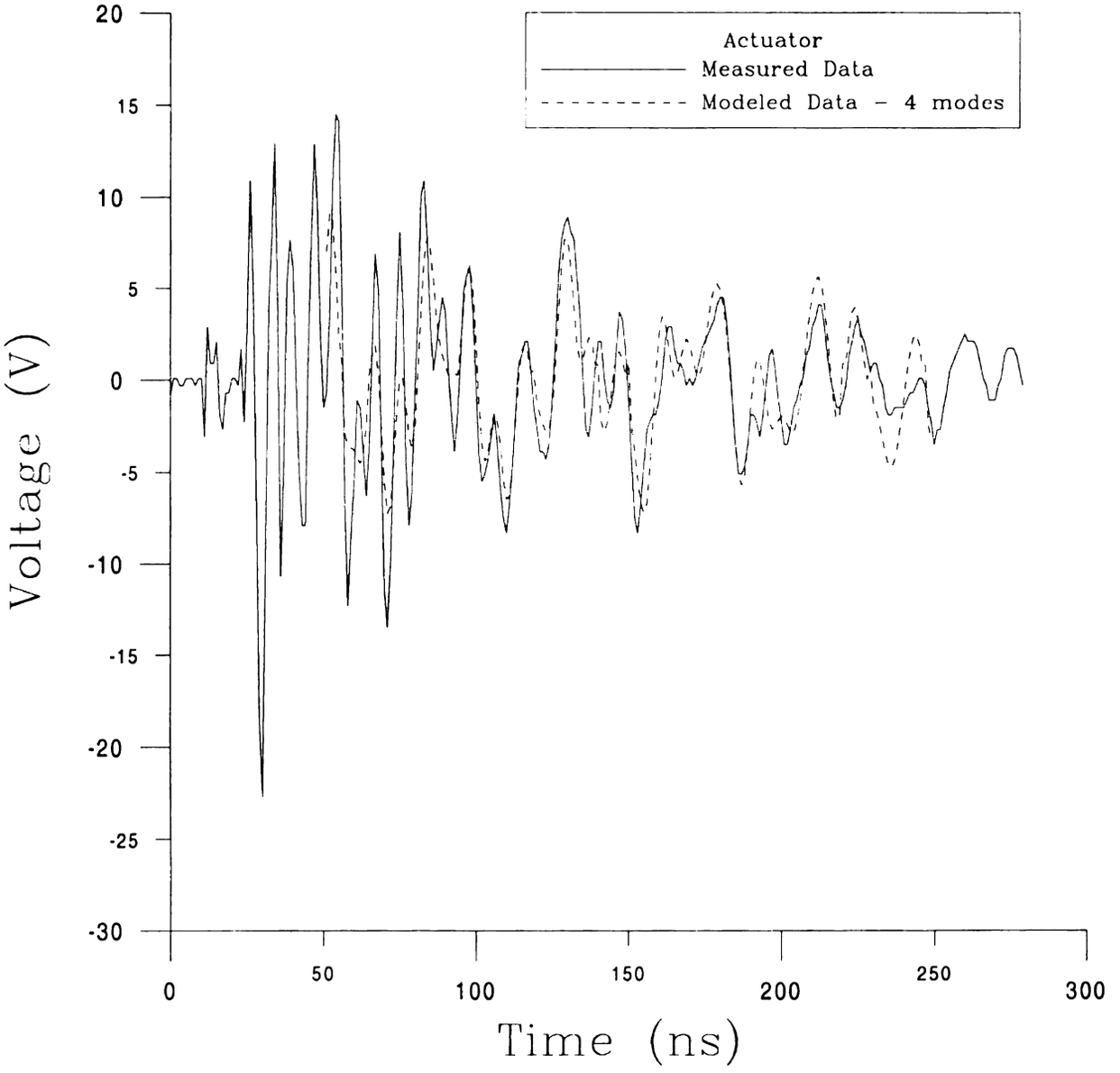

Figure 4-4: Modeled Actuator Motor Data Assuming 4 Natural Modes

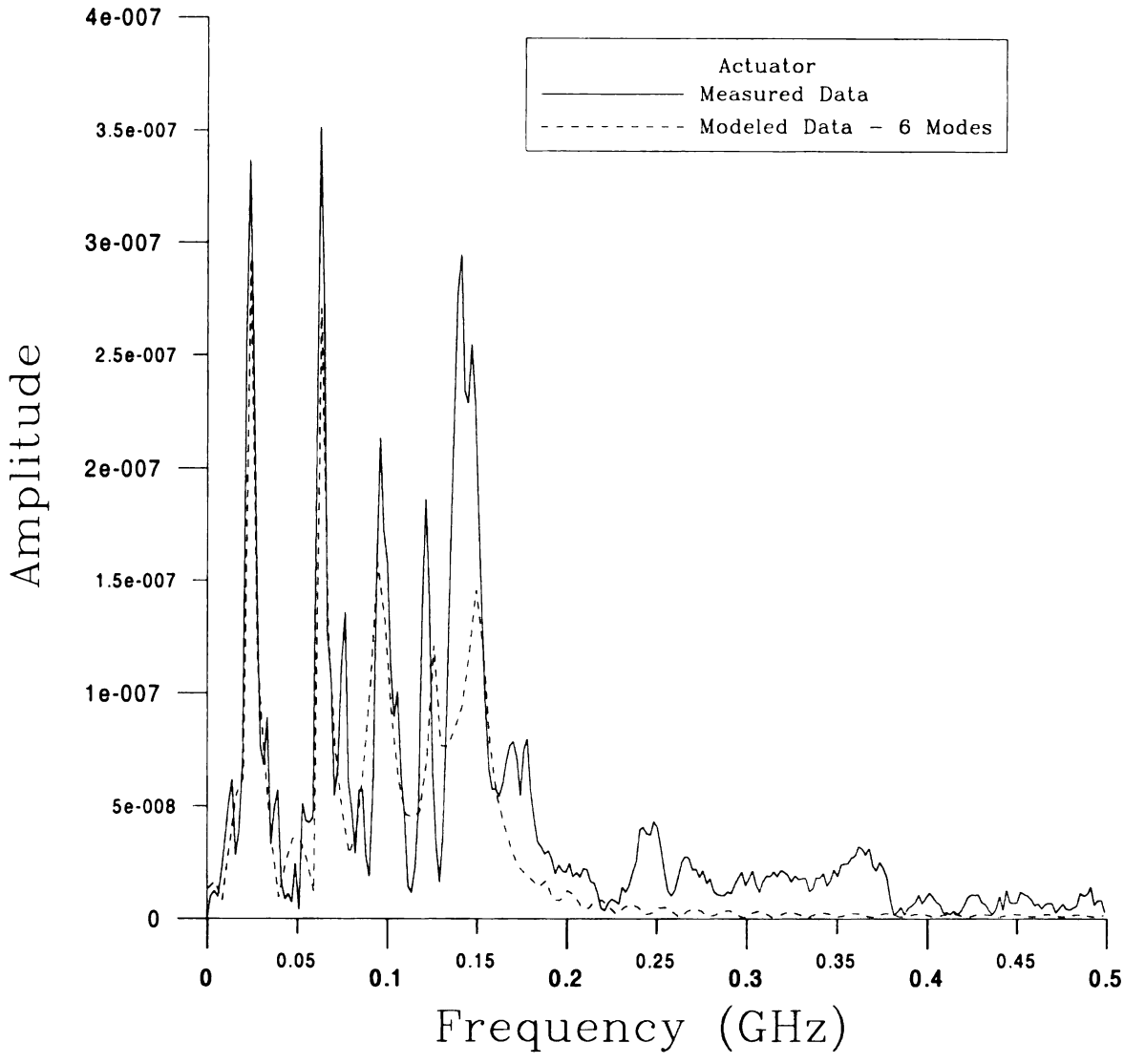

Figure 4-5: Frequency Content of Modeled Actuator Motor Data Assuming 6 Natural Modes

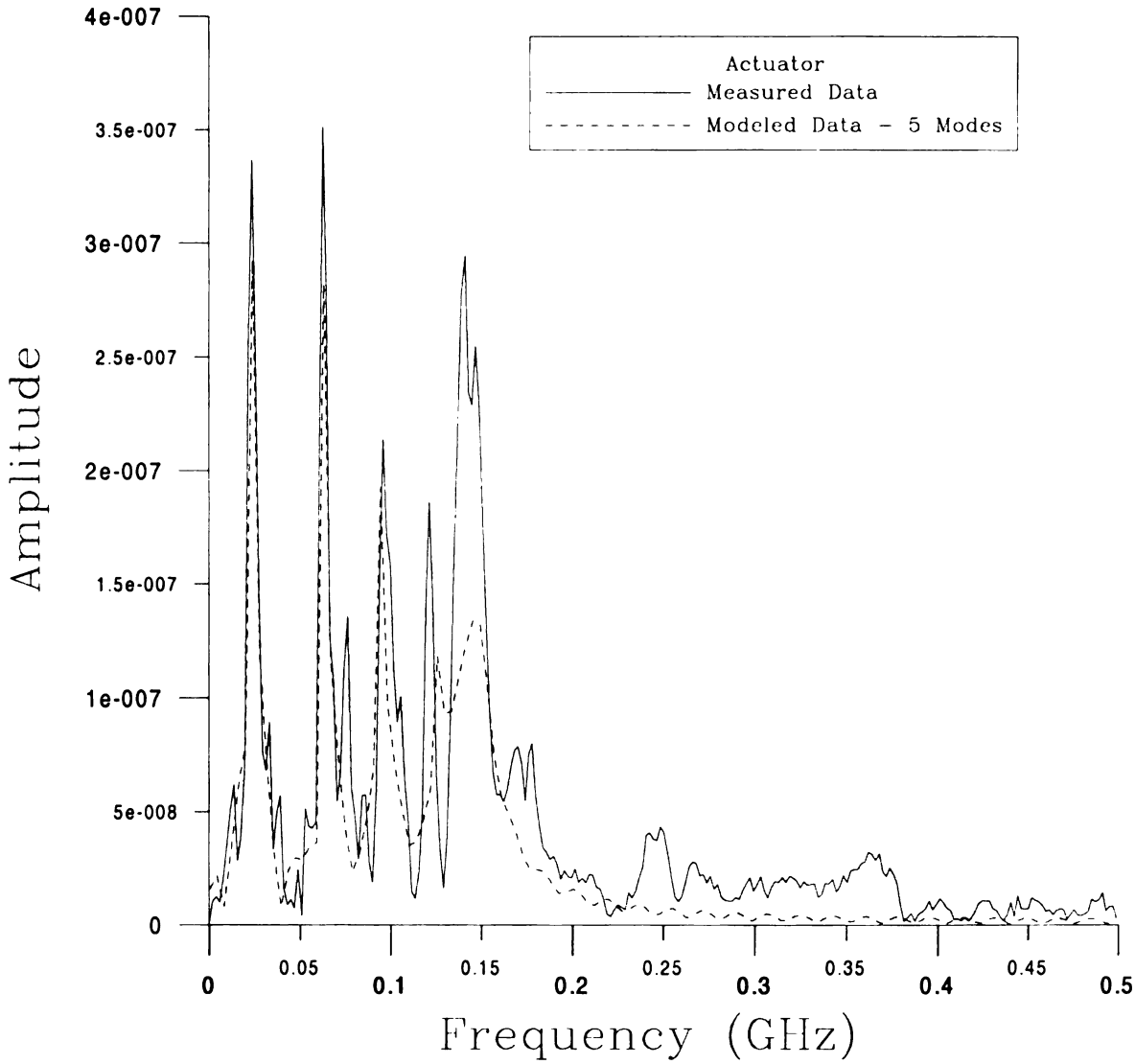

Figure 4-6: Frequency Content of Modeled Actuator Motor Data Assuming 5 Natural Modes

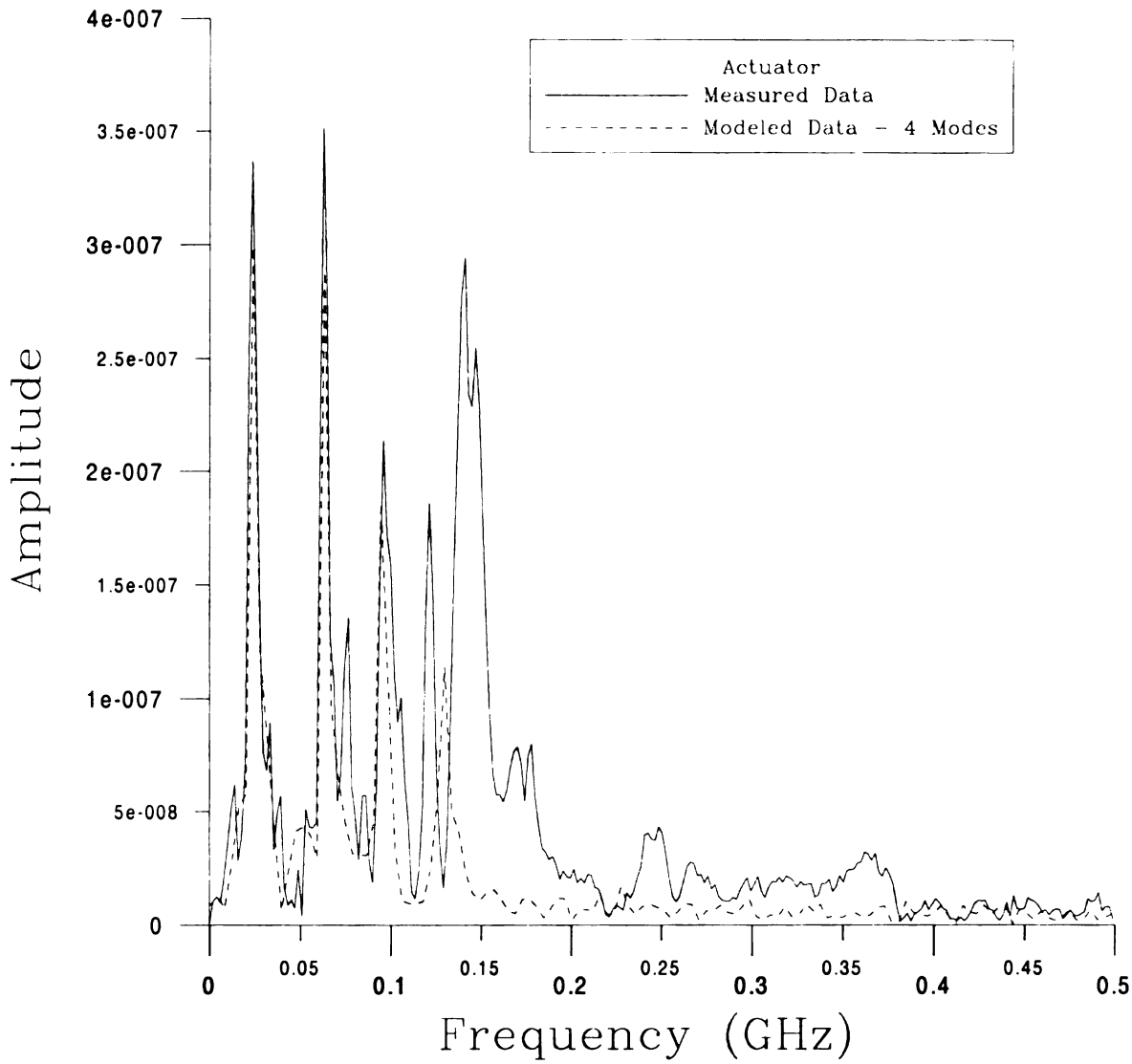

Figure 4-7: Frequency Content of Modeled Actuator Motor Data Assuming 4 Natural Modes

It was noted in Chapter 2 that the linear model for each motor tends to break down beyond a certain frequency. For the actuator motor this occurs beyond 100 MHz. This breakdown in the linear model appears to have affected the actuator motor as discussed above. Modeling has also been done on the motors with their frequency content truncated in order to observe the effect on the models.

By examining the frequency content of the actuator motor with frequency truncated at 100 MHz in Figure 2-10, it can be seen that the actuator motor now has three expected natural modes. Models were computed for two through five expected natural modes in Figures 4-8 through 4-15. As expected, in the cases of three through five expected natural modes, shown in Figures 4-12 through 4-14, only three natural modes were modeled. The accuracy of the model was not improved through the additional modeled modes. This is also reflected in the nearly identical time domain models in Figures 4-5 through 4-10, all of which match the measured data fairly well. In the case where only two expected natural modes were modeled, the first two modes were modeled nicely, as seen in Figure 4-15, yet the third natural mode at 100 MHz was not. The effect of this third natural mode being omitted can be observed in the deviation of the time domain modeled data from the measured data in Figure 4-11.

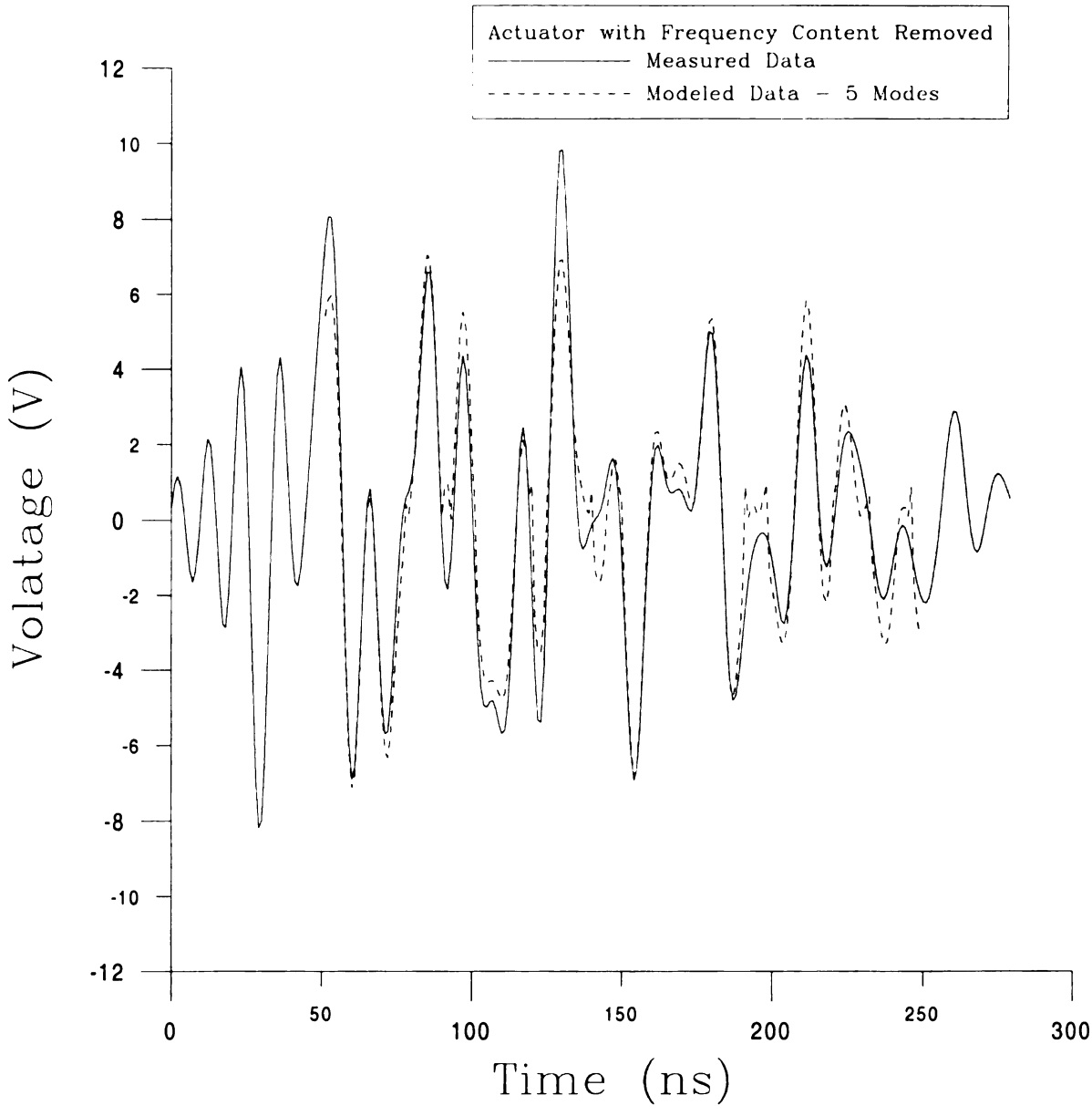

Figure 4-8: Modeled Actuator Motor Data with Frequency Truncated at 100 MHz Assuming 5 Natural Modes

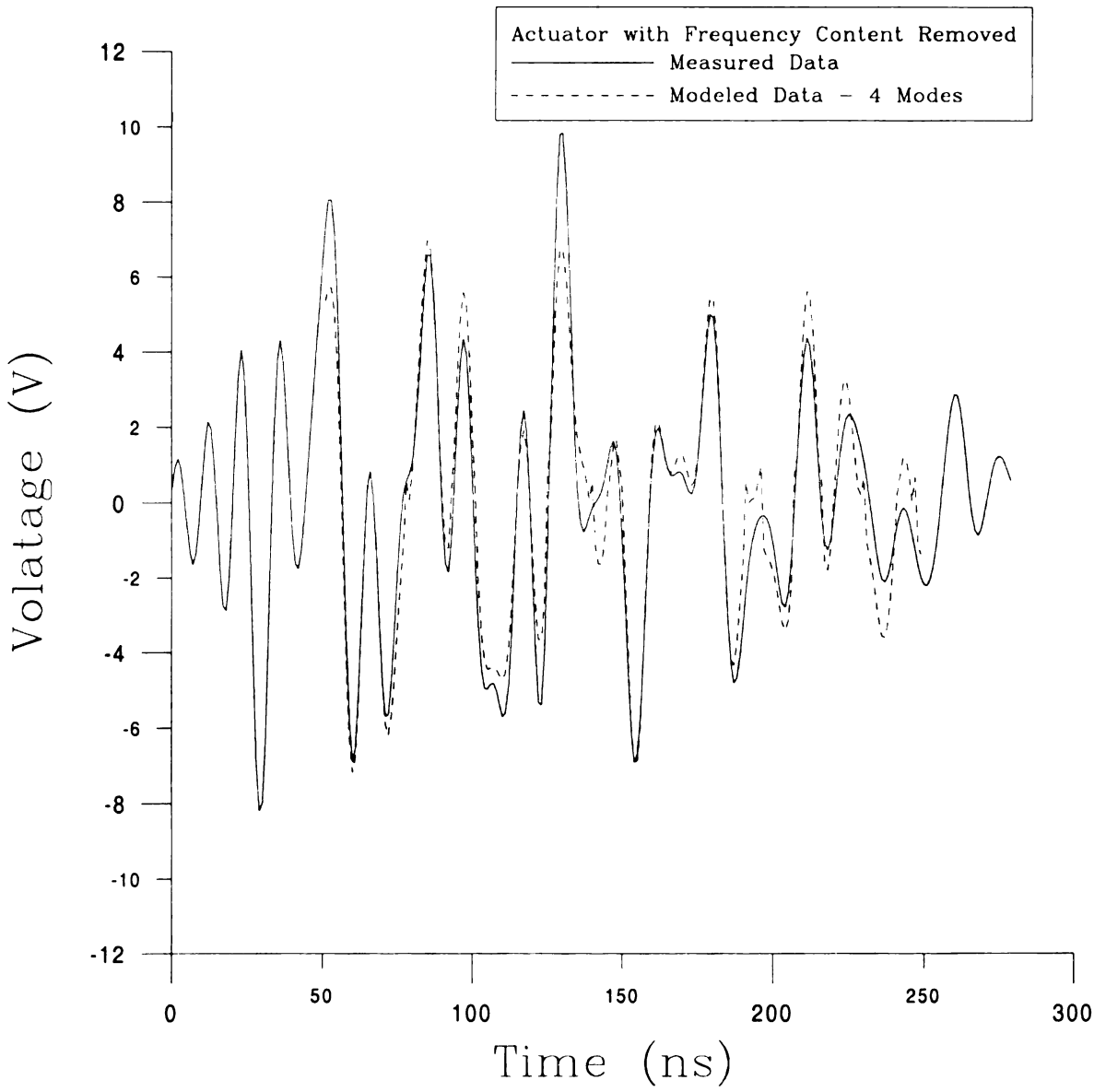

Figure 4-9: Modeled Actuator Motor Data with Frequency Truncated at 100 MHz Assuming 4 Natural Modes

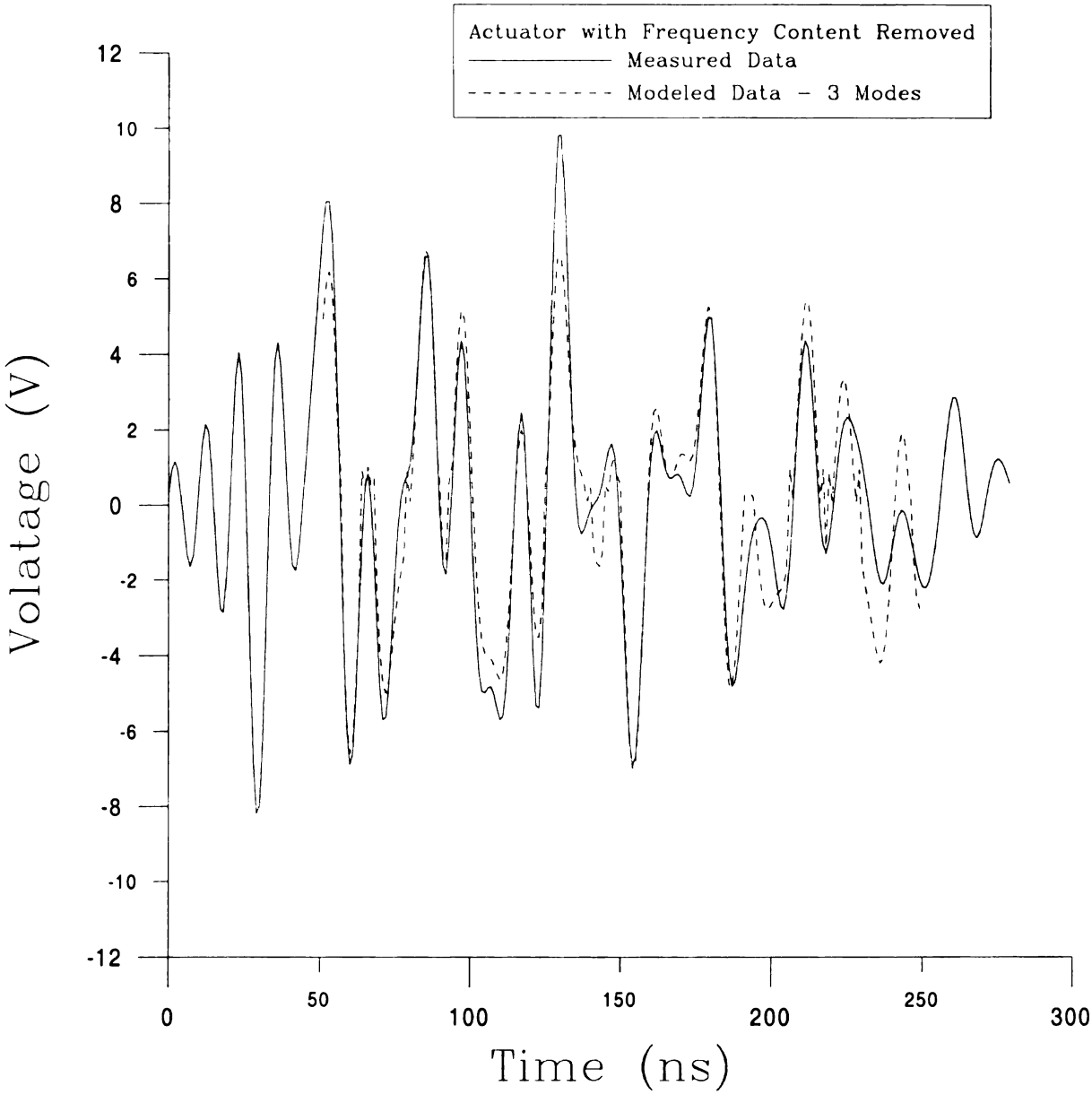

Figure 4-10: Modeled Actuator Motor Data with Frequency Truncated at 100 MHz Assuming 3 Natural Modes

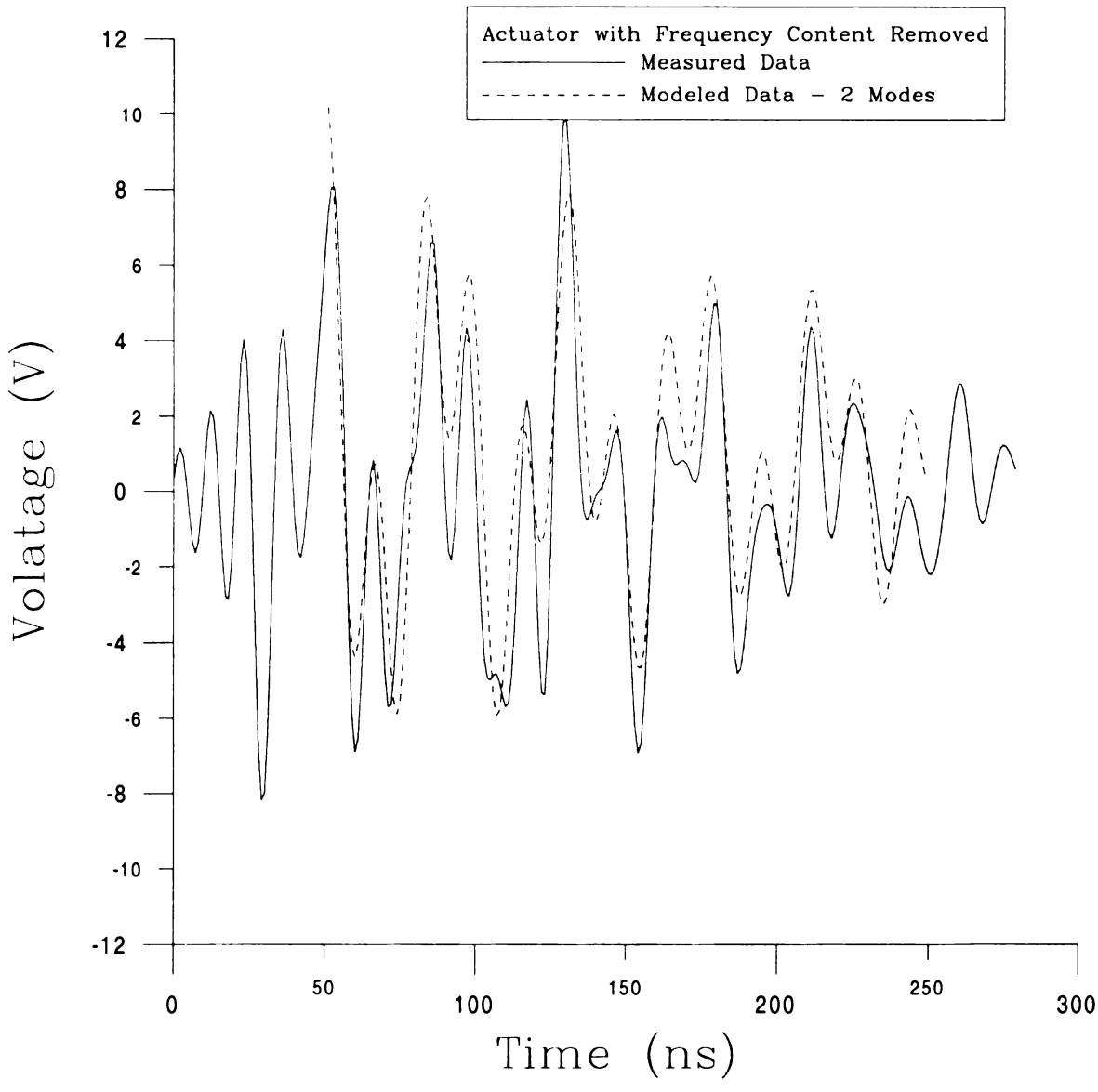

Figure 4-11: Modeled Actuator Motor Data with Frequency Truncated at 100 MHz Assuming 2 Natural Modes

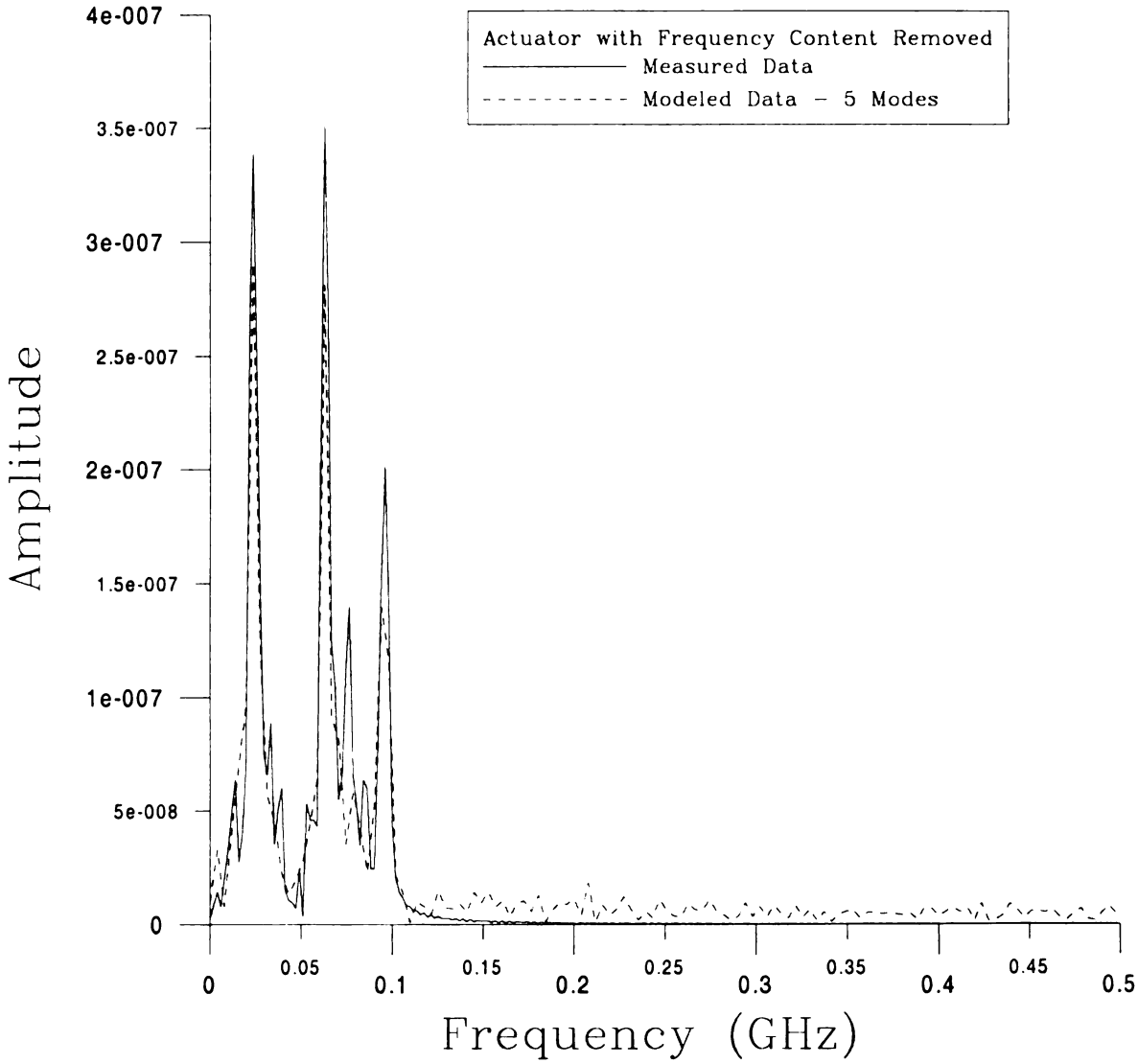

Figure 4-12: Frequency Content of Modeled Actuator Motor Data with Frequency Truncated at 100 MHz Assuming 5 Natural Modes

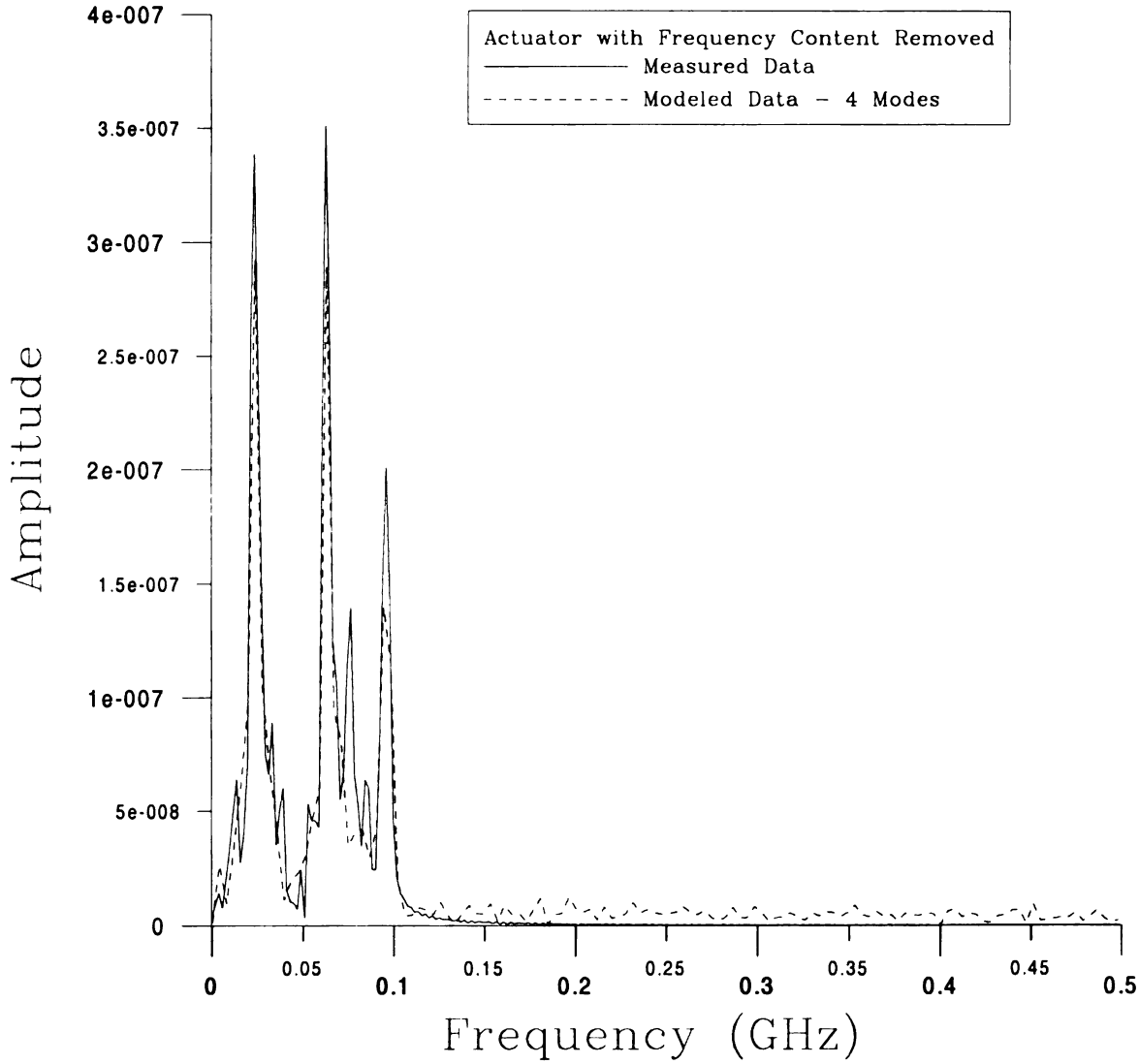

Figure 4-13: Frequency Content of Modeled Actuator Motor Data with Frequency Truncated at 100 MHz Assuming 4 Natural Modes

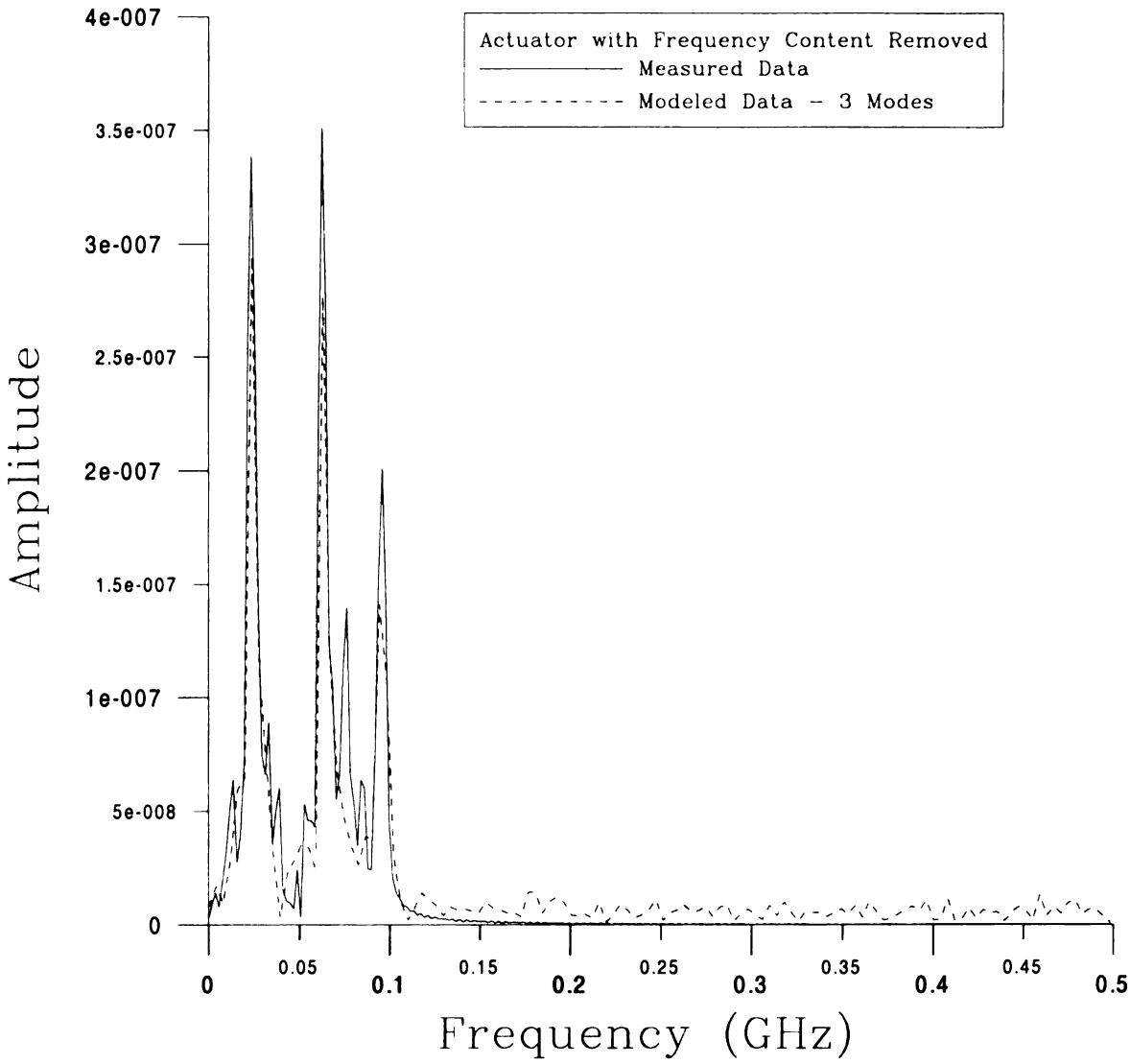

Figure 4-14: Frequency Content of Modeled Actuator Motor Data with Frequency Truncated at 100 MHz Assuming 3 Natural Modes

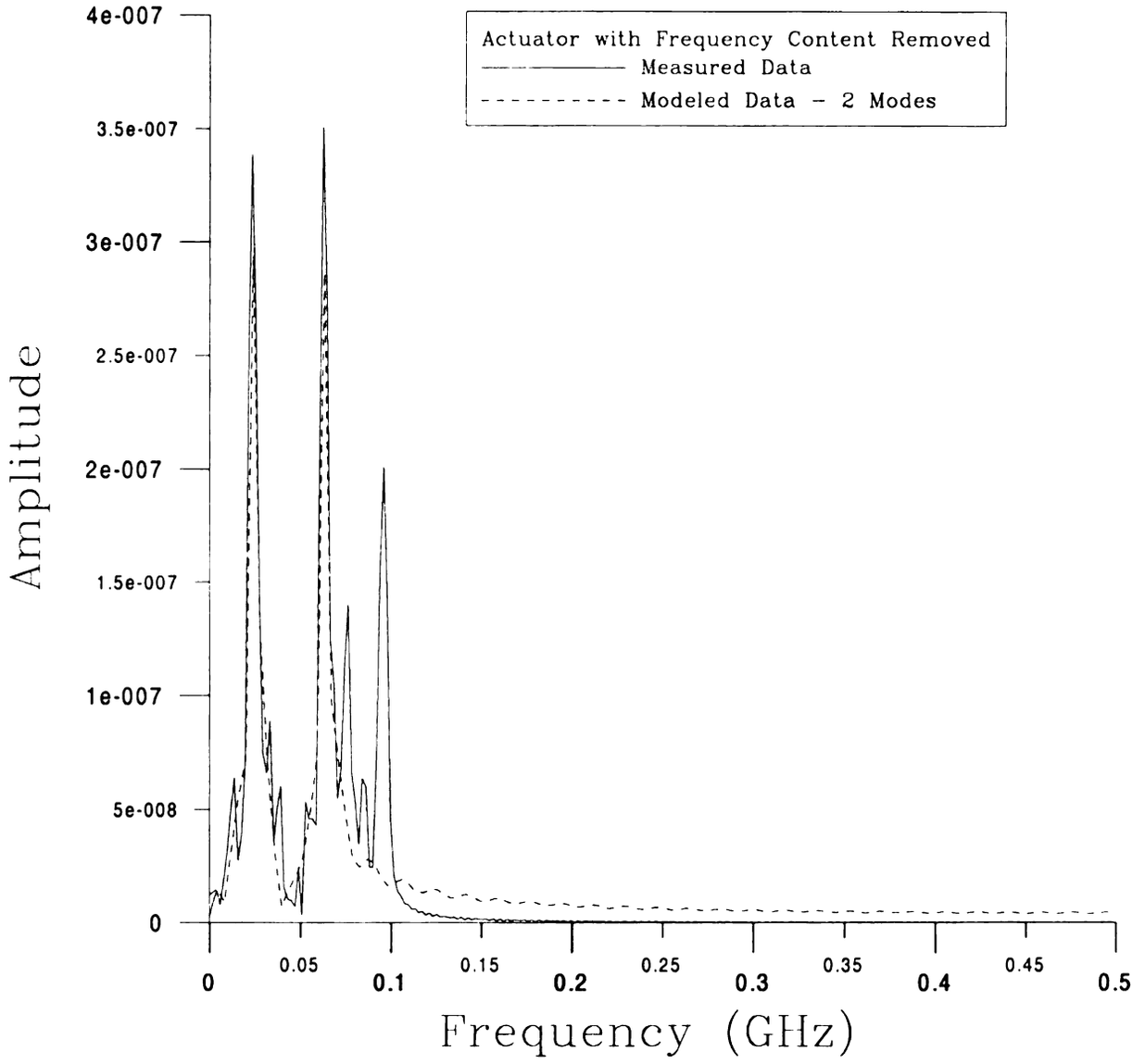

Figure 4-15: Frequency Content of Modeled Actuator Motor Data with Frequency Truncated at 100 MHz Assuming 2 Natural Modes

Examining the frequency content of the blower motor in Figure 2-12, it is seen that this motor is dominated by two natural modes below 100 MHz, with some smaller natural modes occurring after 100 MHz. Modeling is done for two through six expected natural modes. Similar to the findings with the actuator motor, only the frequency content before 100 MHz is accurately modeled, as seen in Figures 4-21 through 4-25. This seems to agree with the theory made in Chapter 2 that the linear model for the blower motor breaks down beyond 100 MHz.

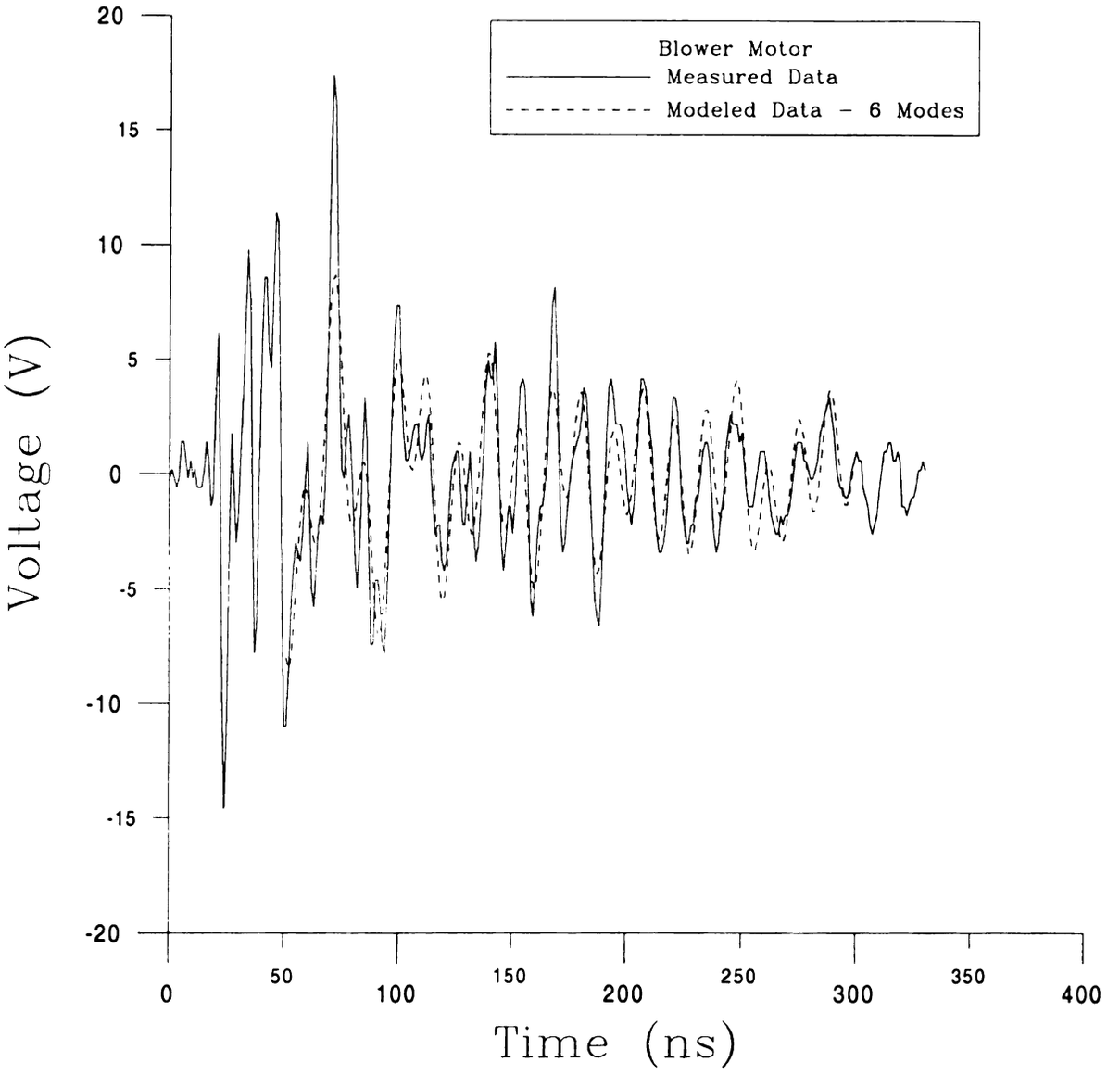

Figure 4-16: Modeled Blower Motor Data Assuming 6 Natural Modes

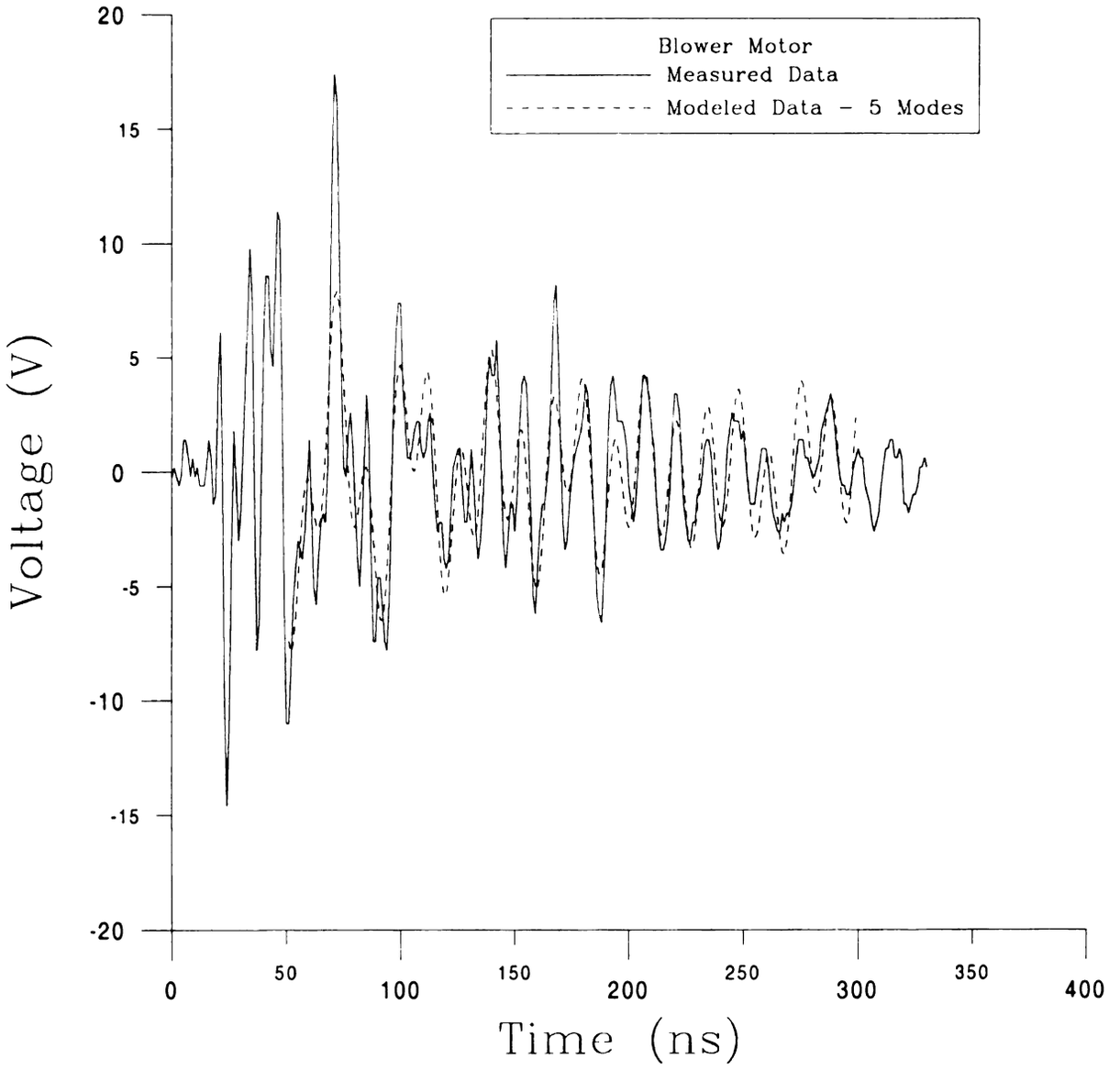

Figure 4-17: Modeled Blower Motor Data Assuming 5 Natural Modes

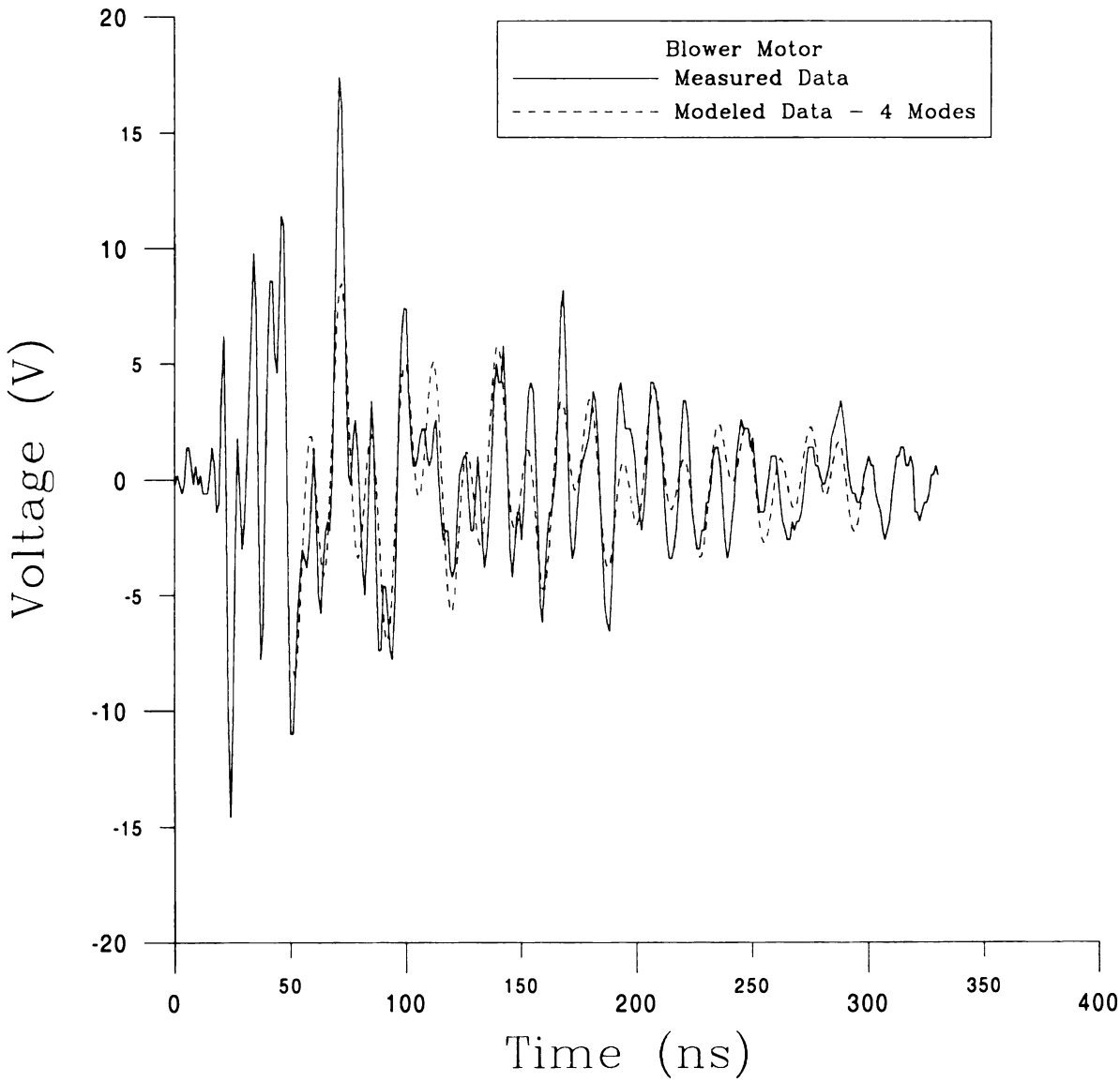

Figure 4-18: Modeled Blower Motor Data Assuming 4 Natural Modes

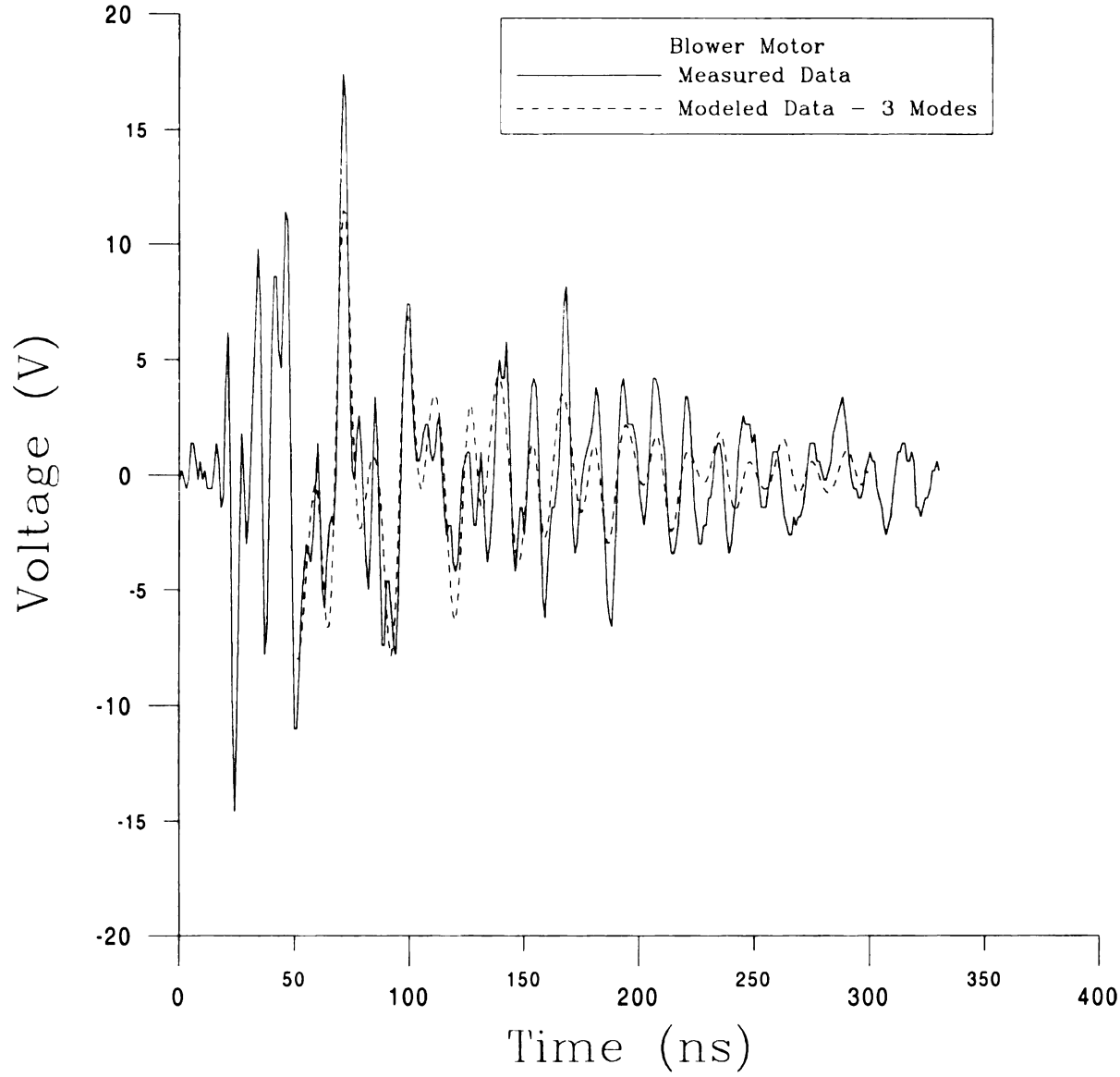

Figure 4-19: Modeled Blower Motor Data Assuming 3 Natural Modes

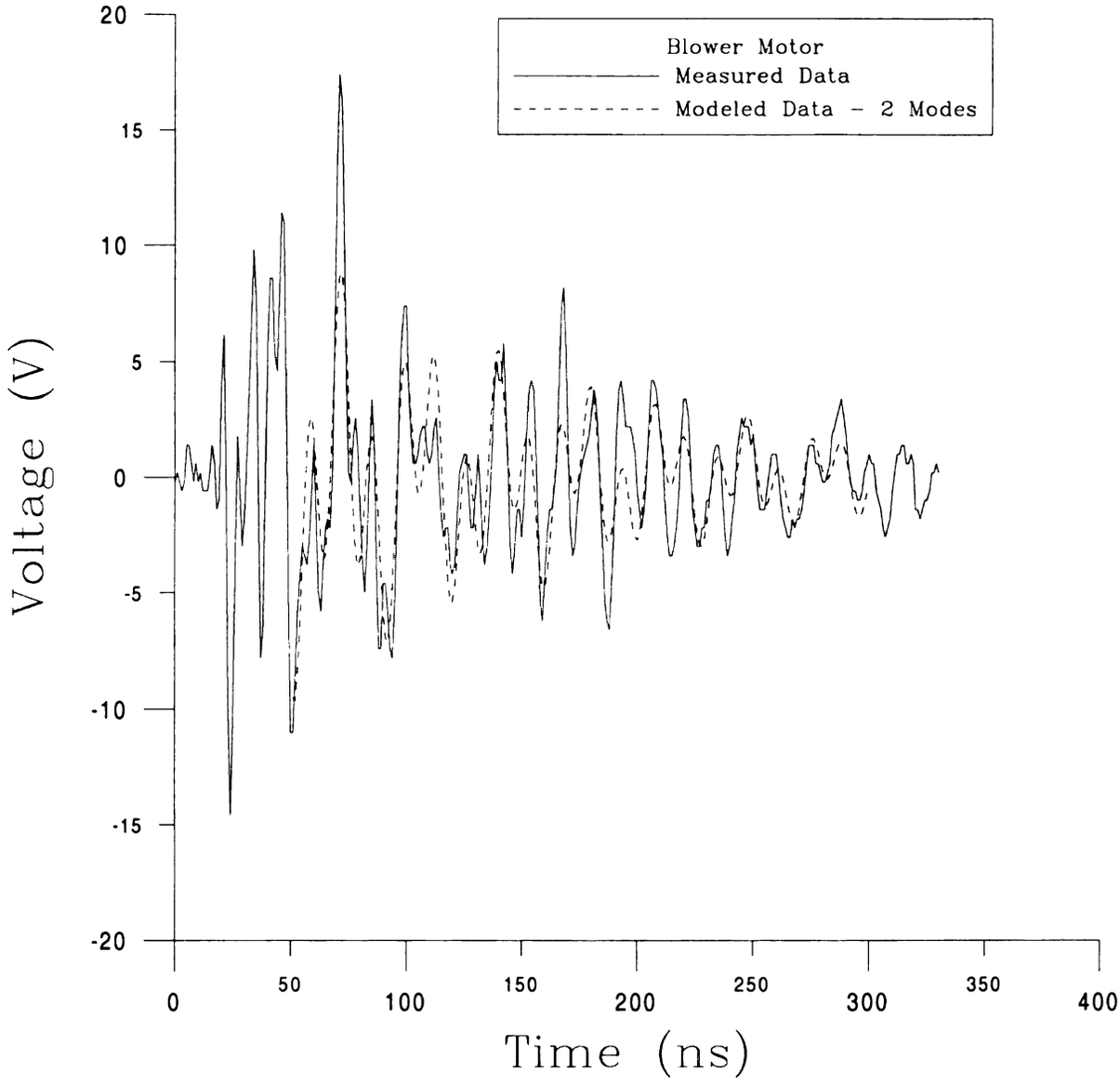

Figure 4-20: Modeled Blower Motor Data Assuming 2 Natural Modes

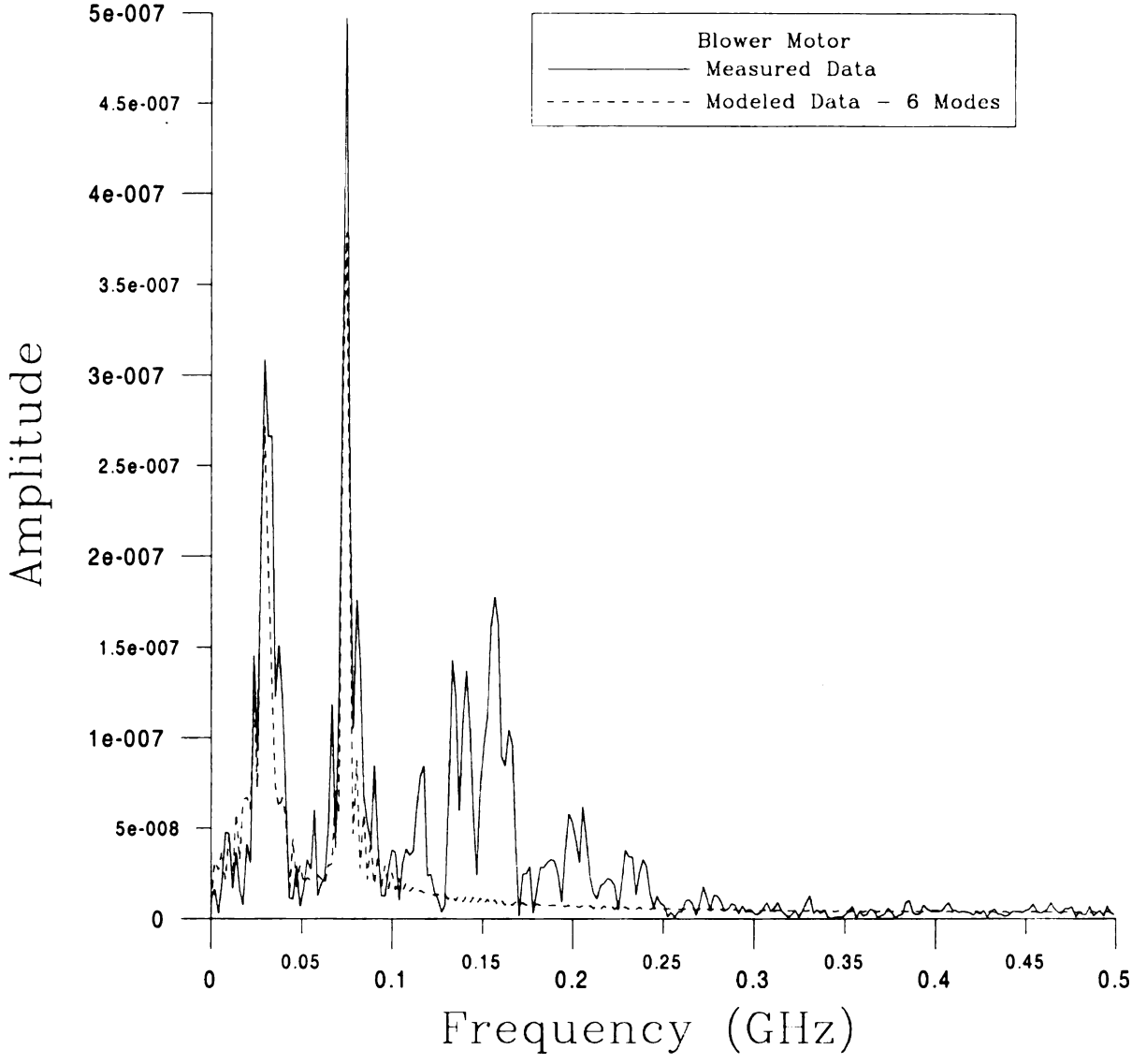

Figure 4-21: Frequency Content of Modeled Blower Motor Data Assuming 6 Natural Modes

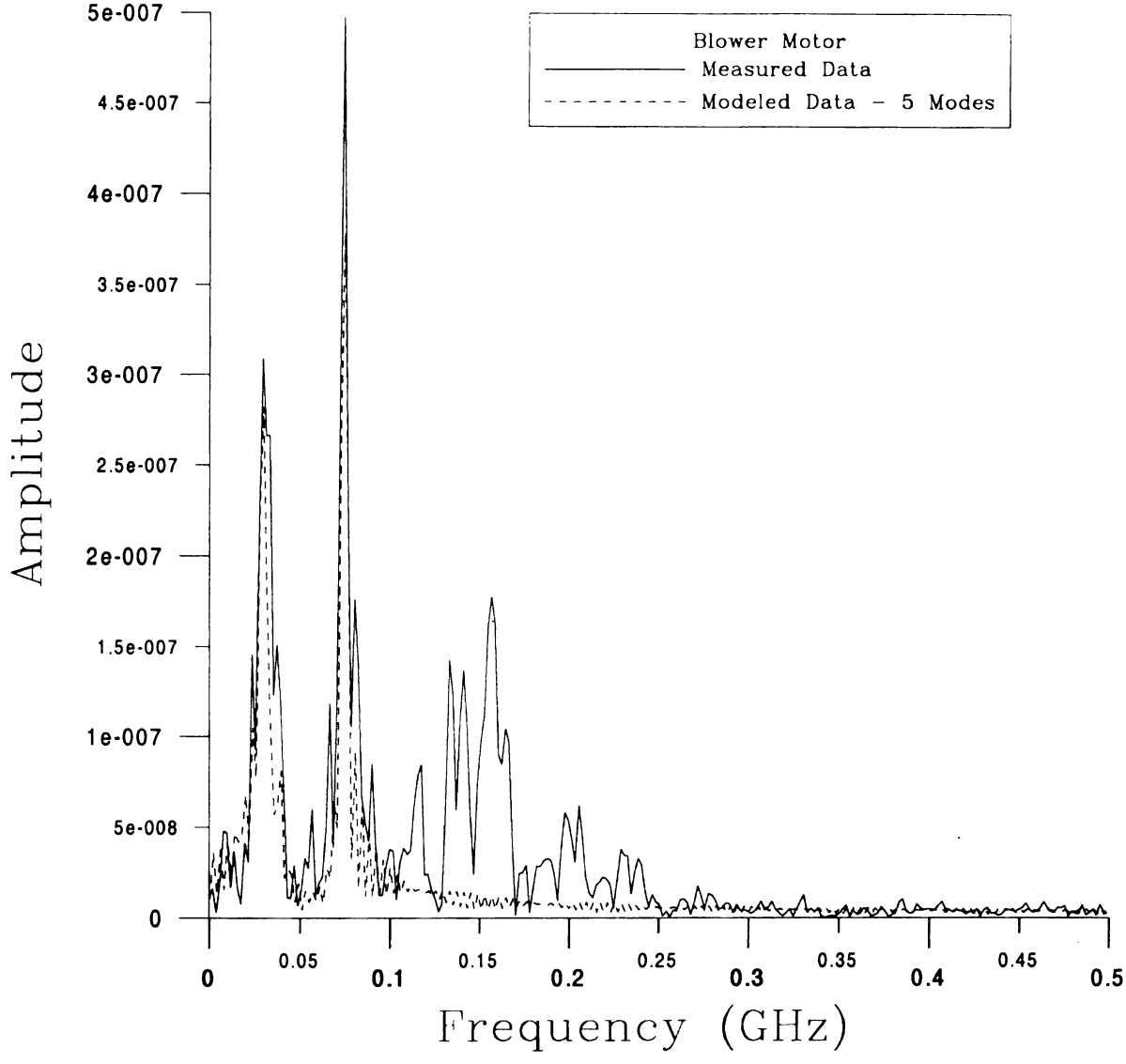

Figure 4-22: Frequency Content of Modeled Blower Motor Data Assuming 5 Natural Modes

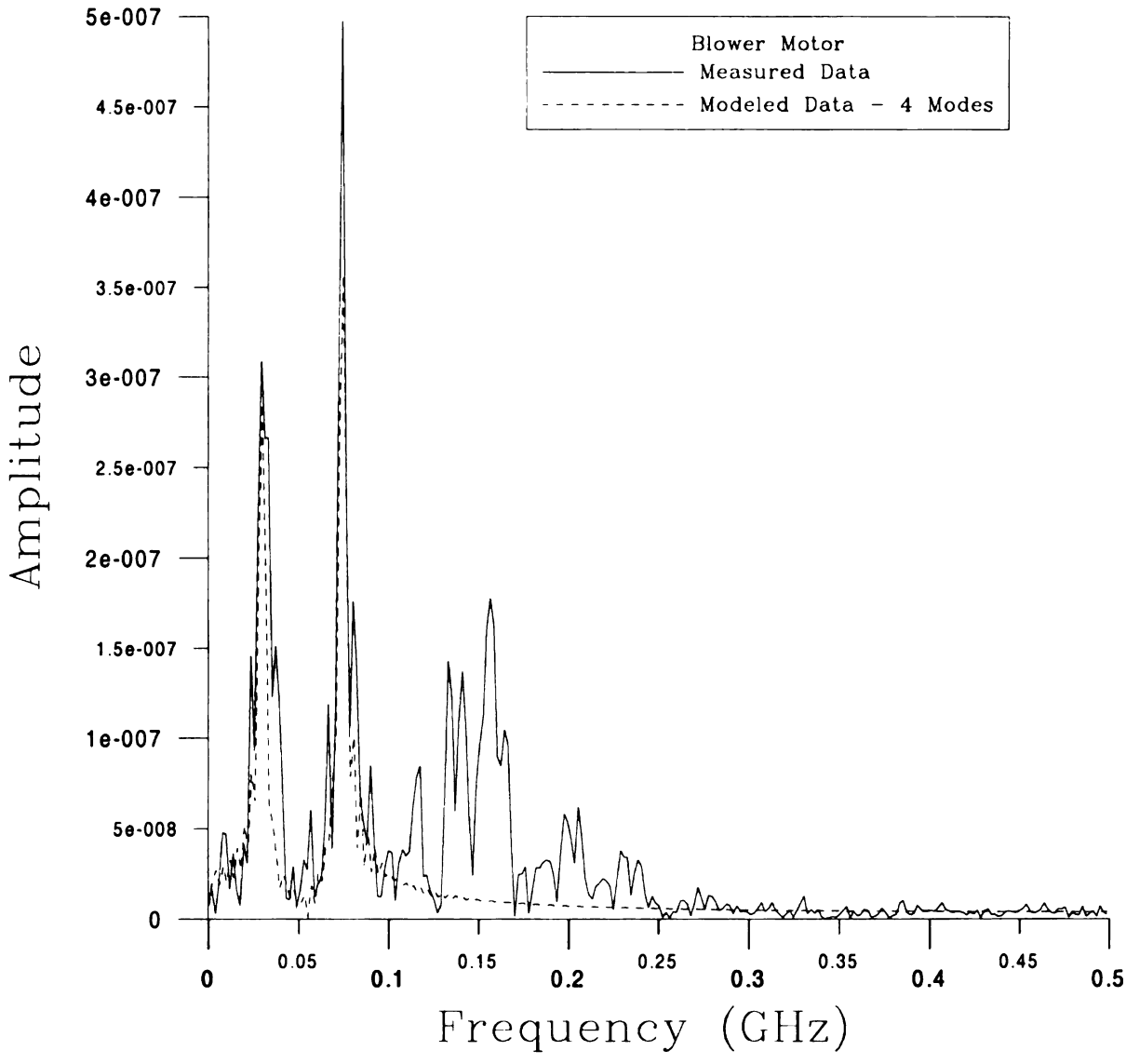

Figure 4-23: Frequency Content of Modeled Blower Motor Data Assuming 4 Natural Modes

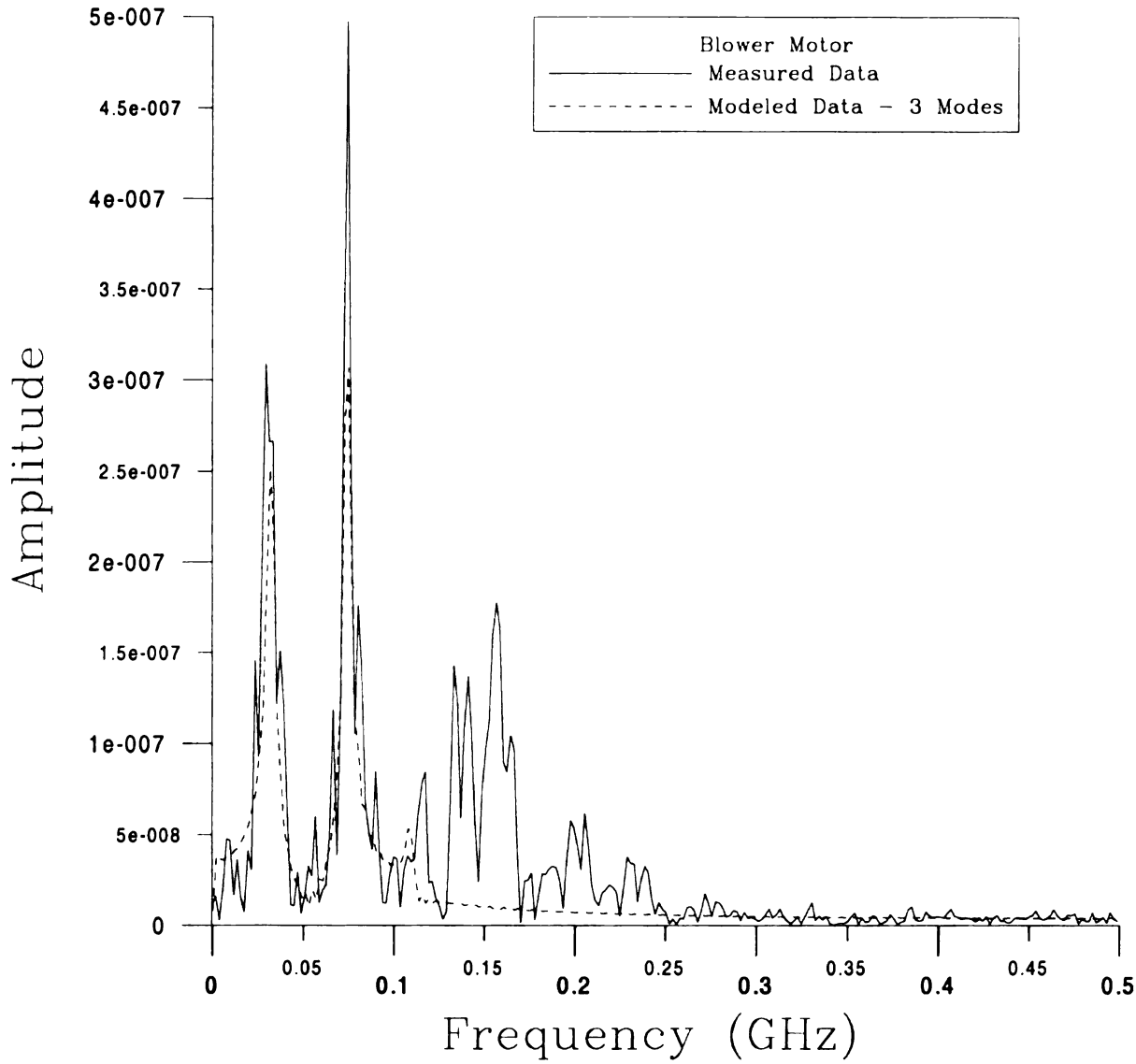

Figure 4-24: Frequency Content of Modeled Blower Motor Data Assuming 3 Natural Modes

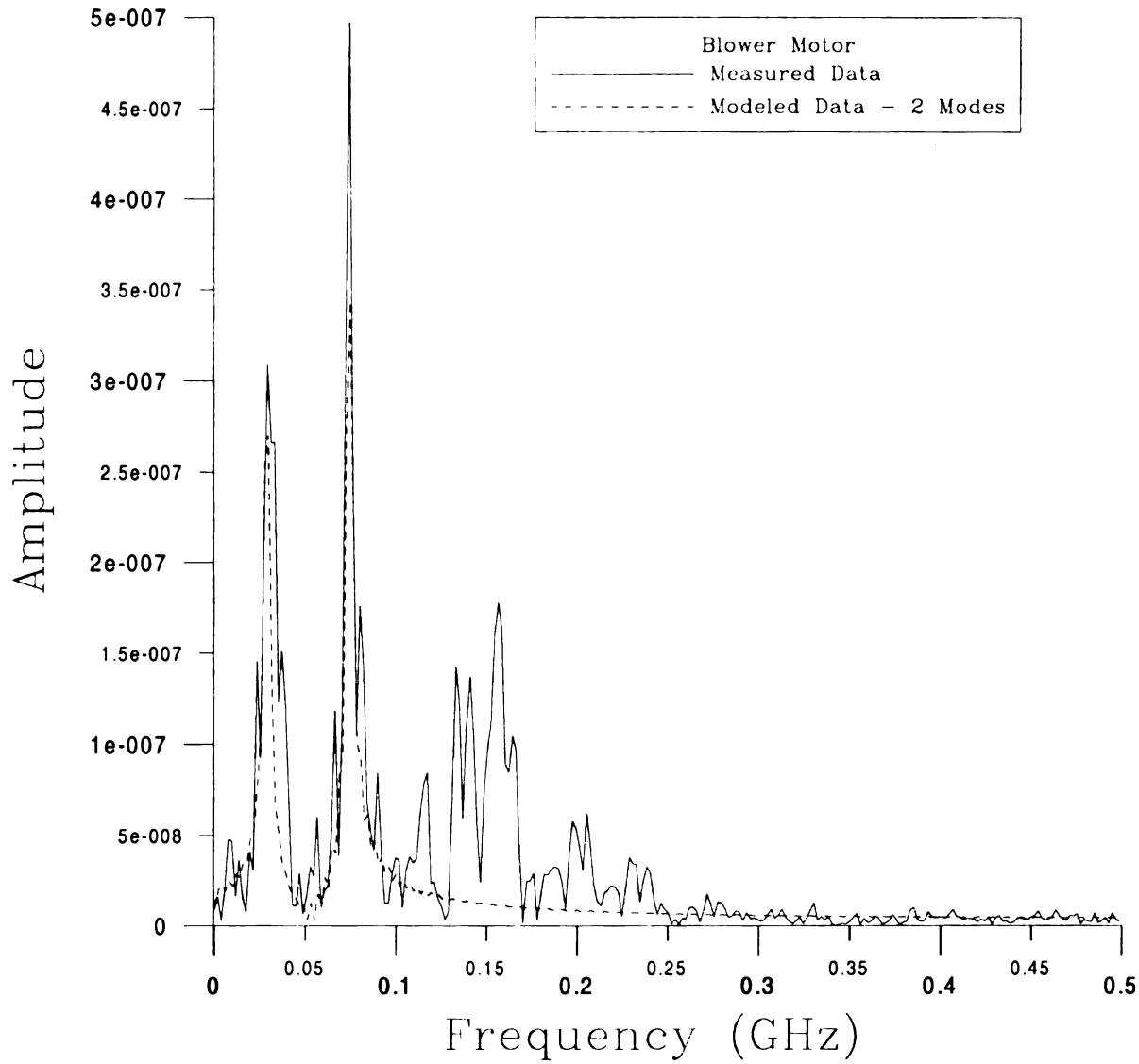

Figure 4-25: Frequency Content of Modeled Blower Motor Data Assuming 2 Natural Modes

The blower motor with frequency data truncated at 100 MHz appears to have two natural modes, as seen in Figure 2-12. Models were computed with one through three expected natural modes. As expected, both the models with two and three expected modes contained two natural modes, as seen in Figures 4-29 and 4-30. However, the match between measured and modeled time domain data seen in Figures 4-26 and 4-27 appears to be no better than for the non-truncated models in Figures 4-16 through 4-20. Looking back at the comparison of the time domain data with and without the frequency content beyond 100 MHz in Figure 2-13, however, it is observed that the missing frequency content beyond 100 MHz actually has very little impact on the shape of the time domain curve. The modeled data for one expected mode is shown in Figures 4-28 and 4-31, and is interesting as it shows a waveform consisting of a single damped sinusoid, as well as another waveform consisting of the same damped sinusoid combined with a second damped sinusoid, which is due to the other natural frequency.

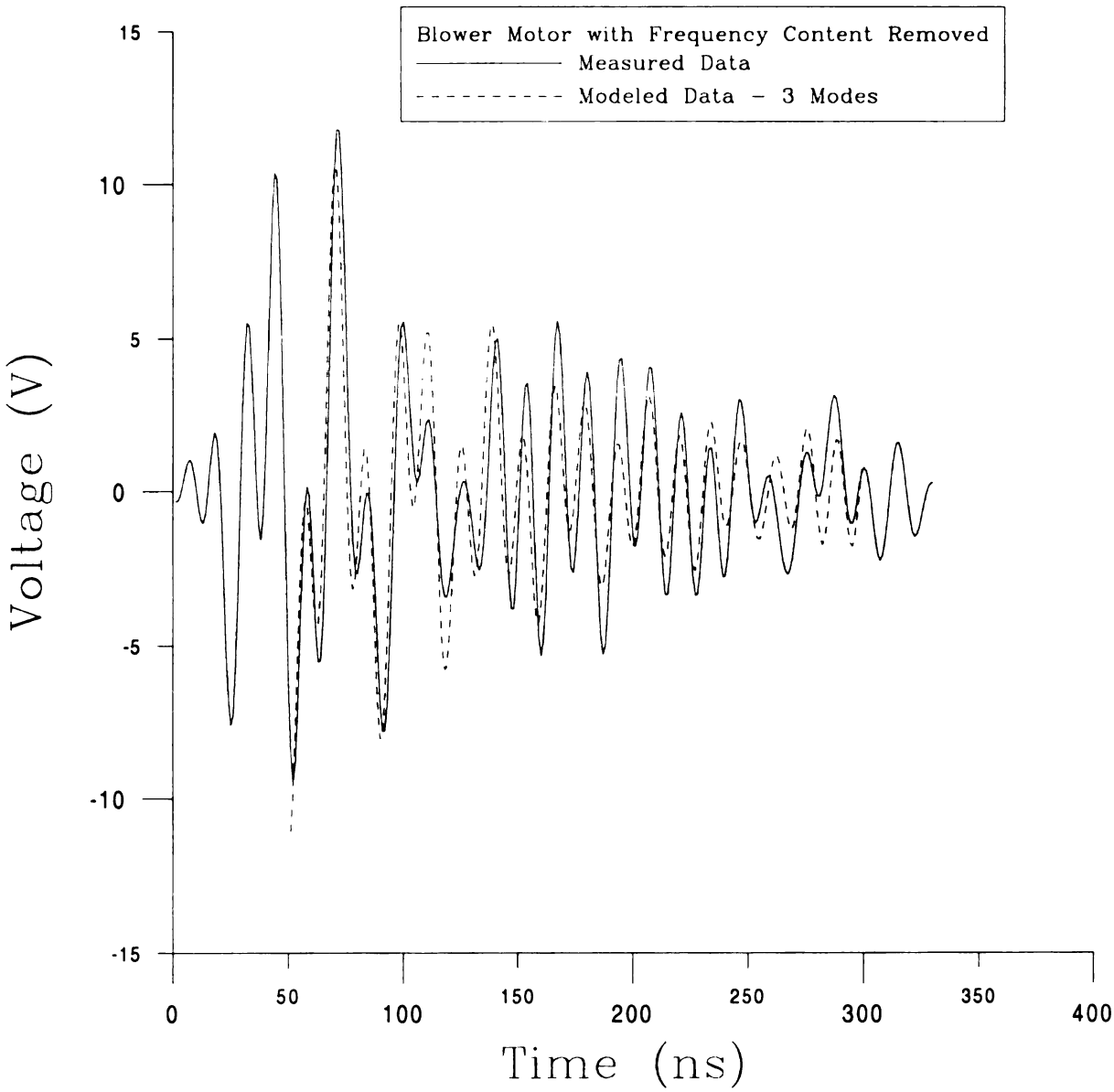

Figure 4-26: Modeled Blower Motor Data with Frequency Truncated at 100 MHz Assuming 3 Natural Modes

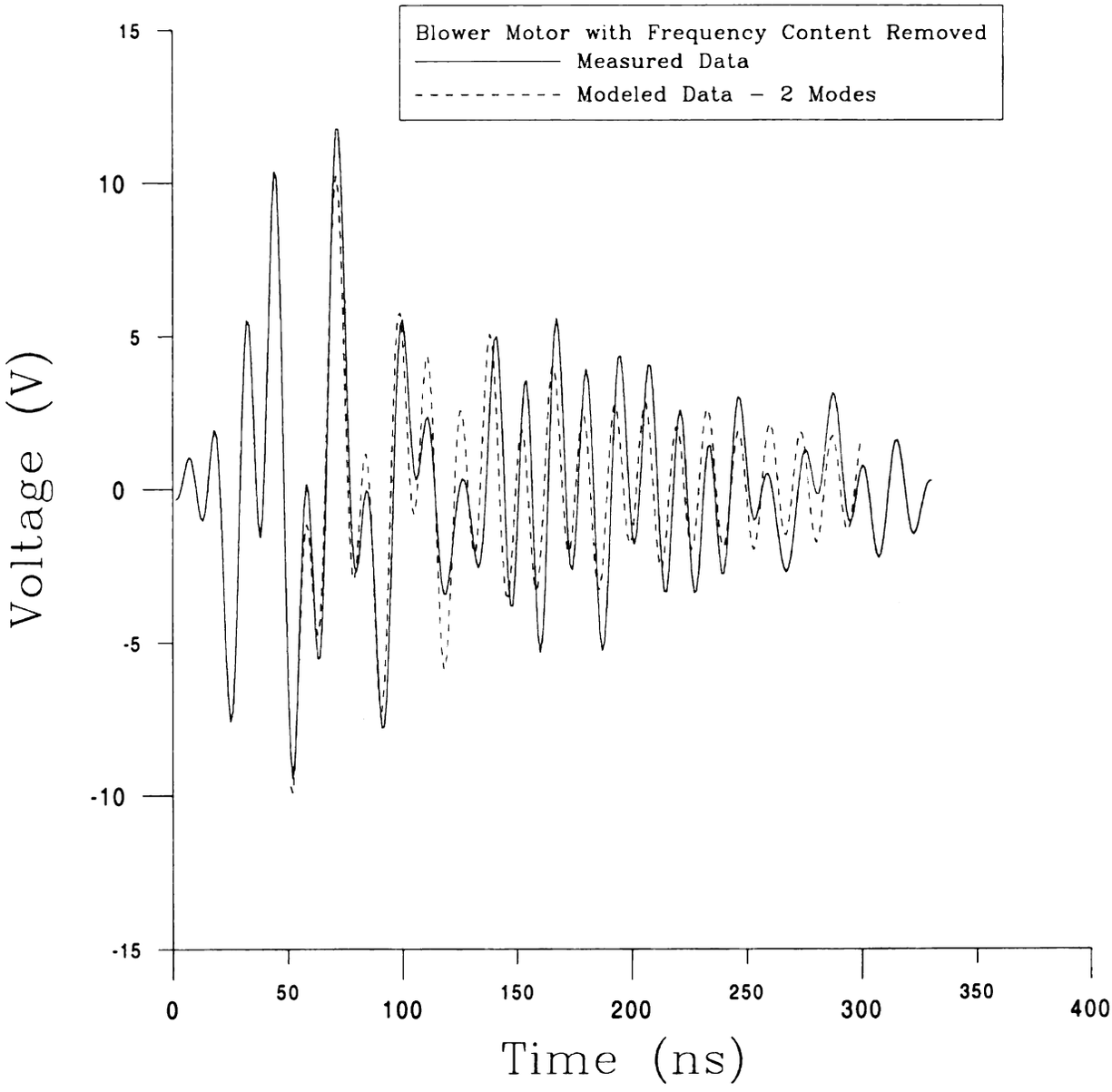

Figure 4-27: Modeled Blower Motor Data with Frequency Truncated at 100 MHz Assuming 2 Natural Modes

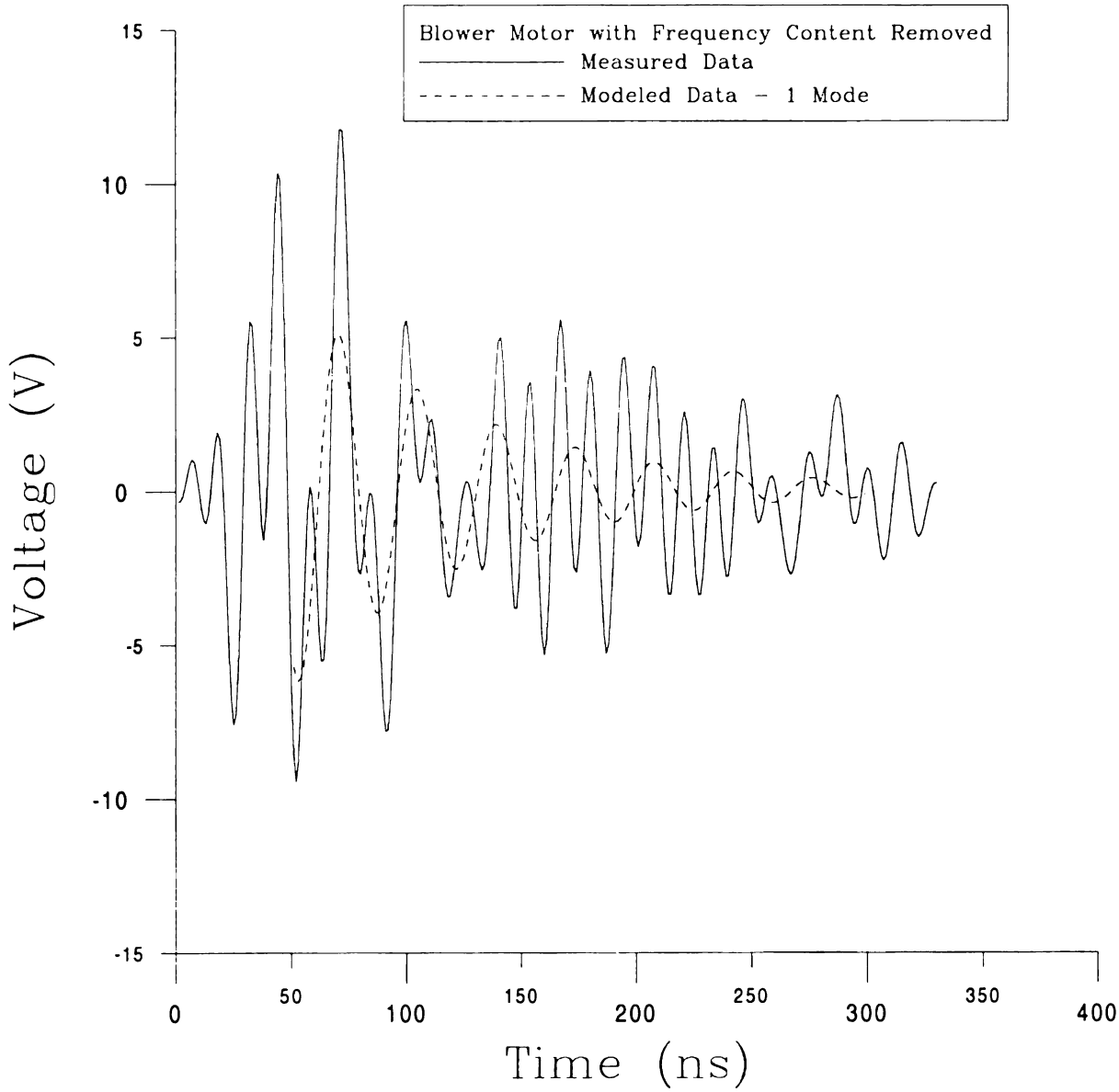

Figure 4-28: Modeled Blower Motor Data with Frequency Truncated at 100 MHz Assuming 1 Natural Mode

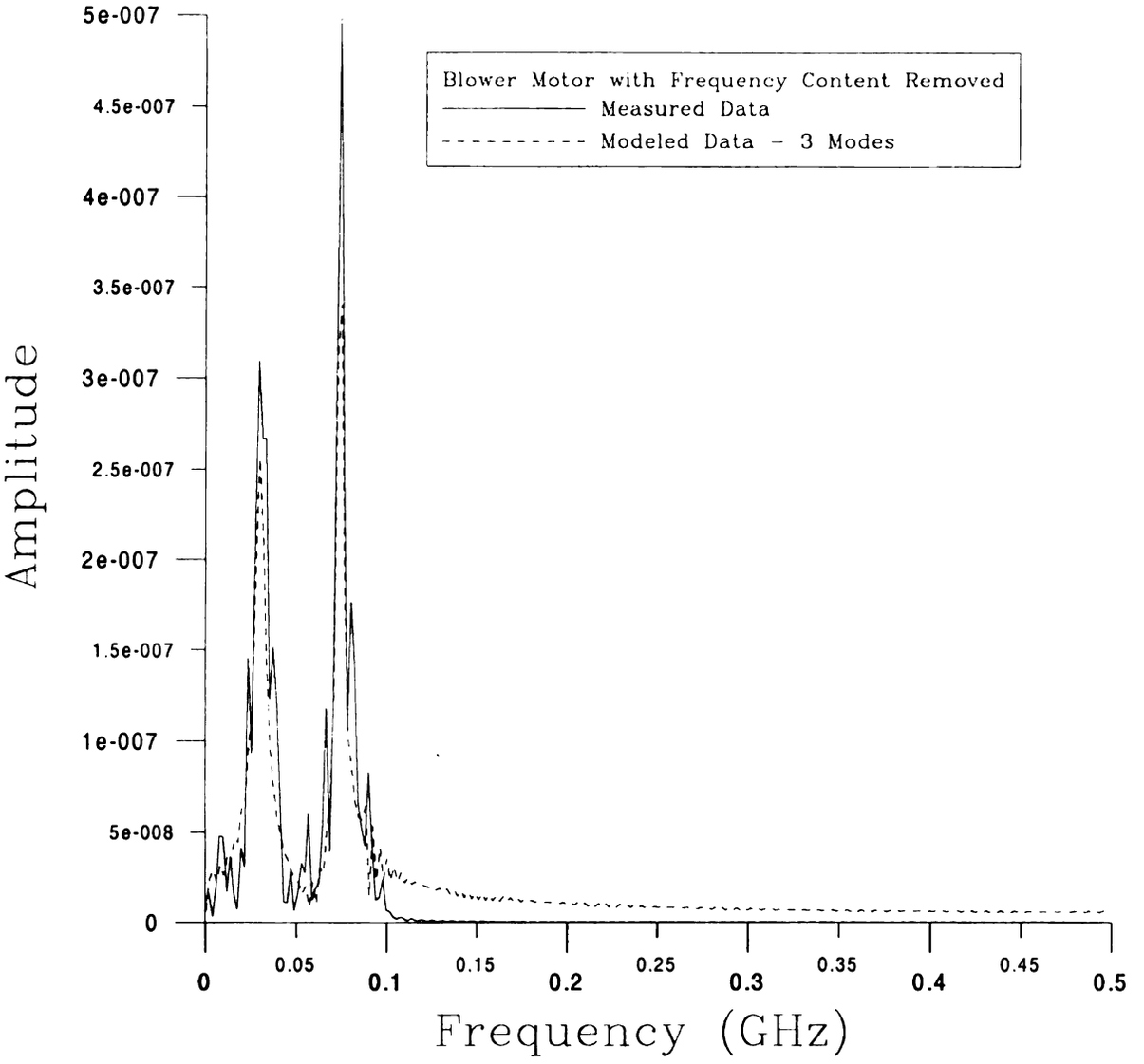

Figure 4-29: Frequency Content of Modeled Blower Motor Data with Frequency Truncated at 100 MHz Assuming 3 Natural Modes

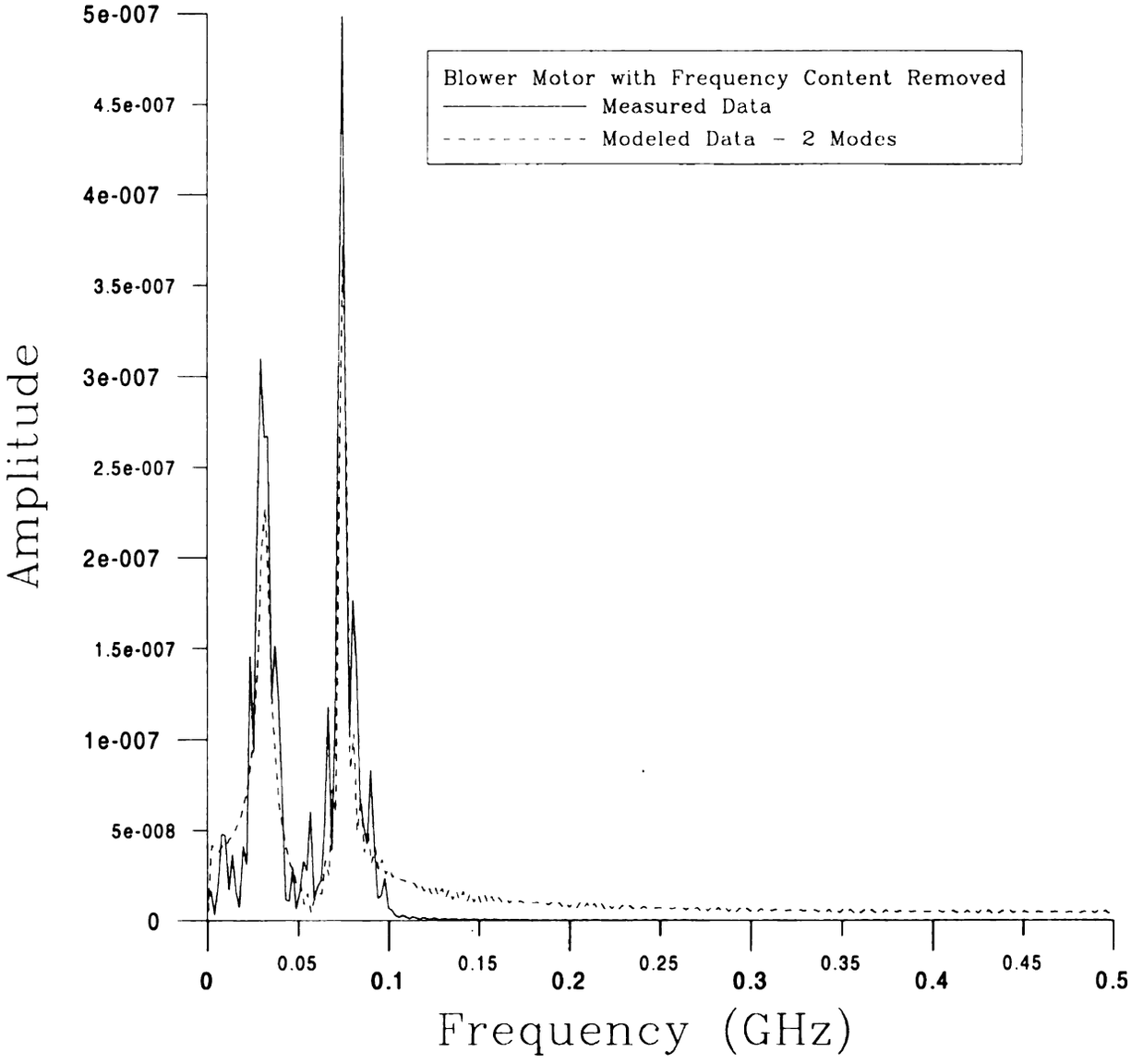

Figure 4-30: Frequency Content of Modeled Blower Motor Data with Frequency Truncated at 100 MHz Assuming 2 Natural Modes

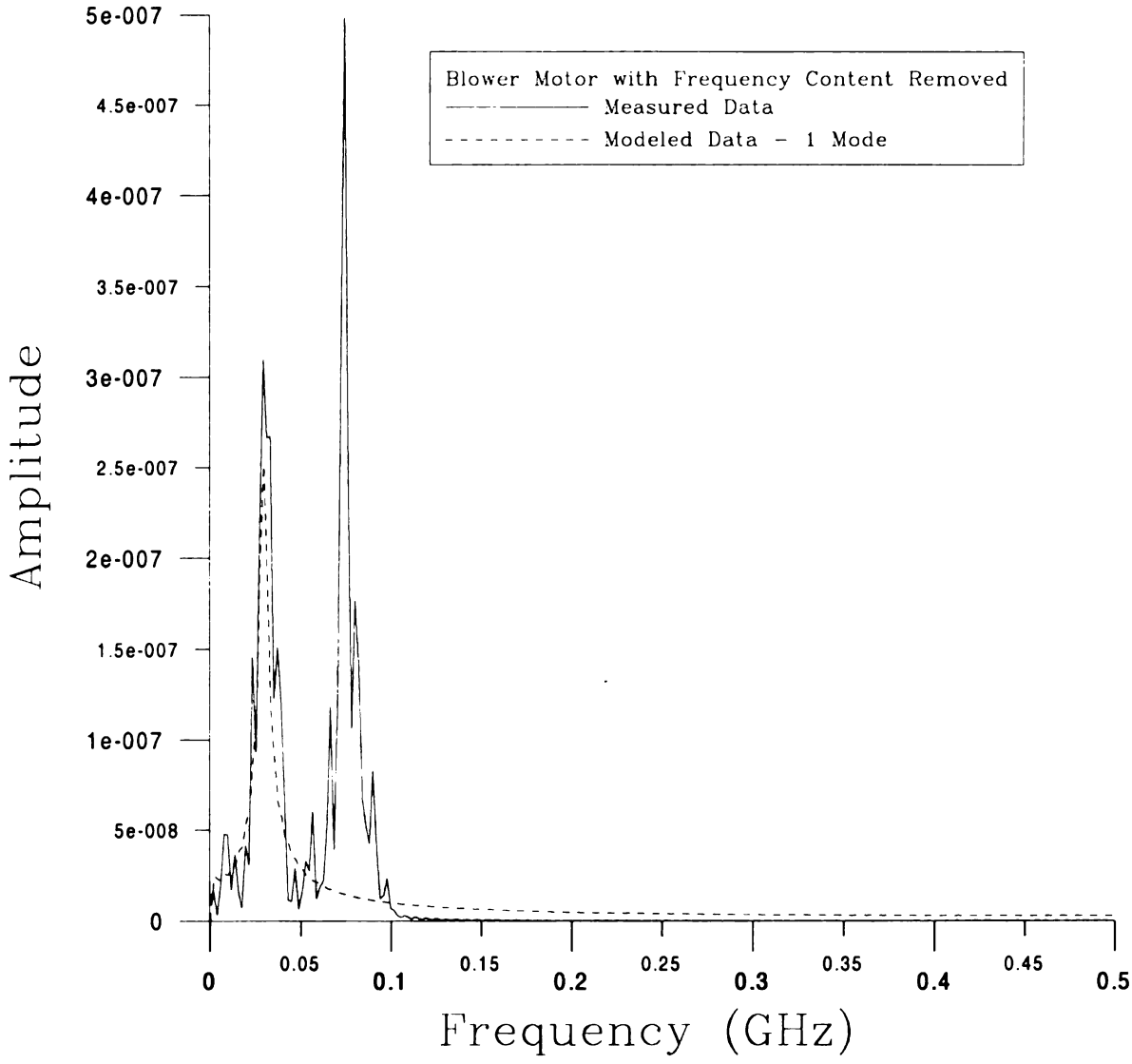

Figure 4-31: Frequency Content of Modeled Blower Motor Data with Frequency Truncated at 100 MHz Assuming 1 Natural Mode

The frequency content for the blue line motor can be observed in Figure 2-14. Truncating the frequency data at 200 MHz does not appear to remove any significant frequency content. This waveform appears to consist of between four and seven significant natural modes. The last major natural mode, at about 150 MHz, has a remarkably wide frequency content. Modeling was done for both the measured and truncated data models with four through eight expected natural modes.

The best time domain fit occurred using eight assumed natural modes as seen in Figures 4-32 and 4-42. The frequency content, seen in Figures 4-37 and 4-47, modeled the first 5 natural modes, but nothing was successfully modeled beyond 100 MHz. Seven expected natural modes, shown in Figures 4-33, 4-38, 4-43, and 4-48, yielded nearly as accurate models, with only the first four natural modes modeled. Perhaps the most interesting blue line model was the one with six expected natural modes. As seen in Figures 4-39 and 4-49, the first two modes were successfully modeled, while the next two modes were partially modeled. Yet this is the only blue line motor model to successfully replicate any natural modes beyond 100 MHz. The frequency data for the model using five expected modes in Figures 4-40 and 4-50 is nearly identical to the six mode model except it lacks the natural mode beyond 100 MHz. Comparing the two models in the time domain, in Figures 4-34, 4-35, 4-44, and 4-45, it is seen that the modeled natural mode beyond 100 MHz contributes to the shape of the earliest part of the modeled time domain. Finally, the model assuming 4 natural modes, seen in Figures 4-36, 4-41, 4-46 and 4-51 successfully models only the first two natural modes, which leads to a very poor match between measured and modeled time-domain data.

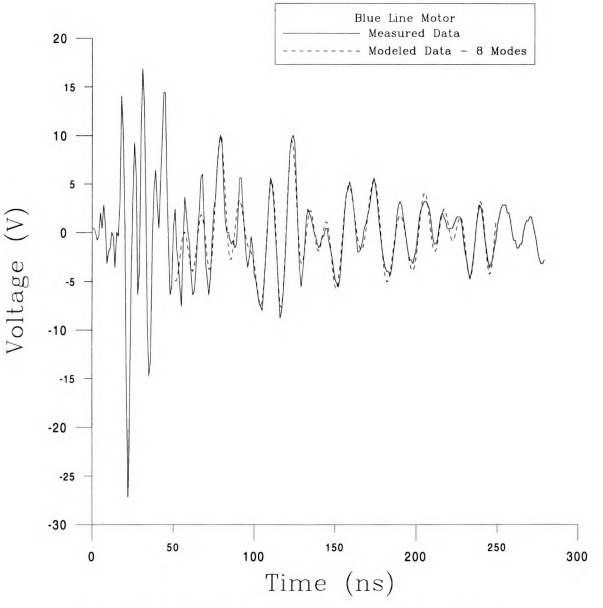

Figure 4-32: Modeled Blue Line Motor Data Assuming 8 Natural Modes

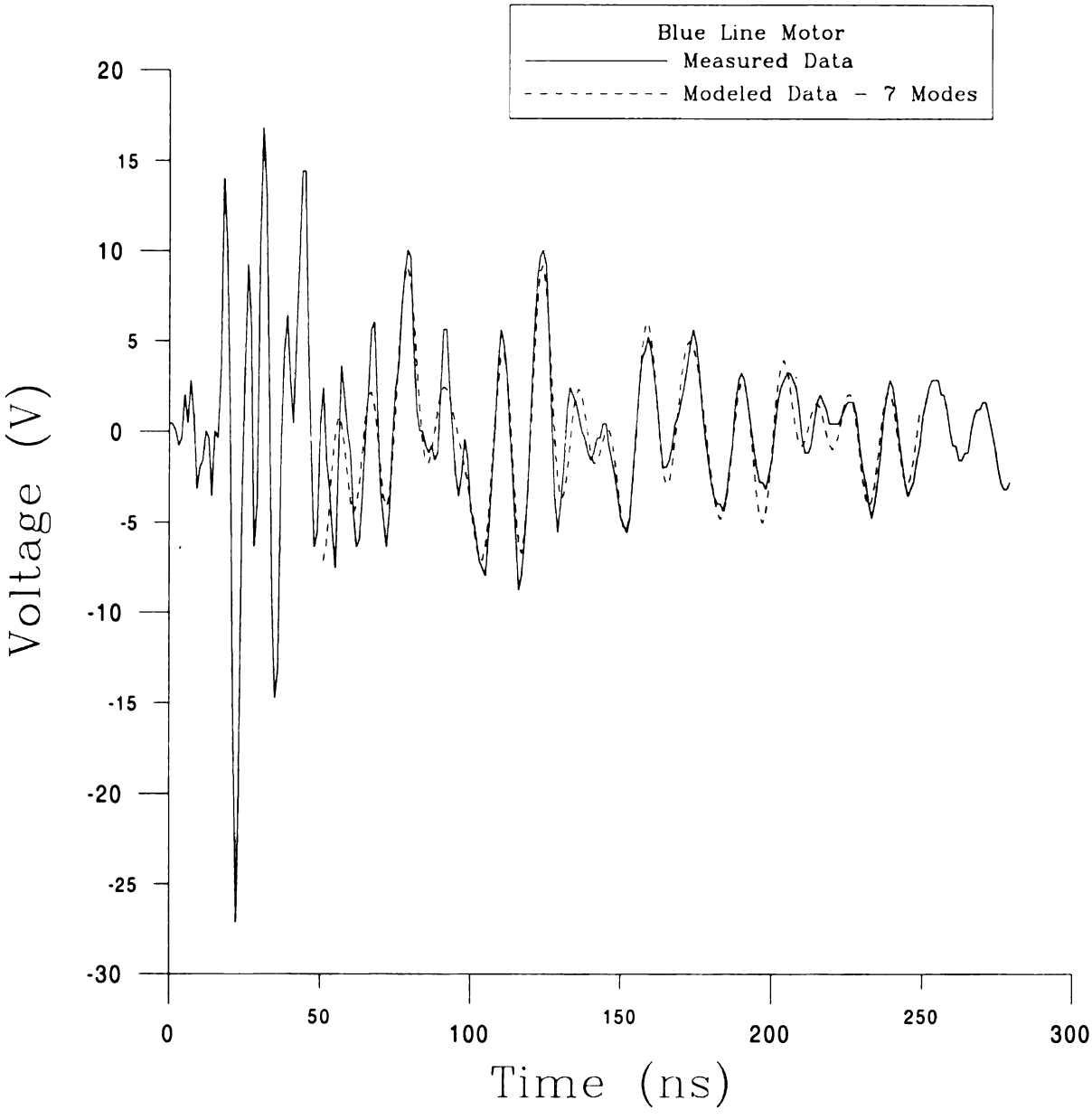

Figure 4-33: Modeled Blue Line Motor Data Assuming 7 Natural Modes

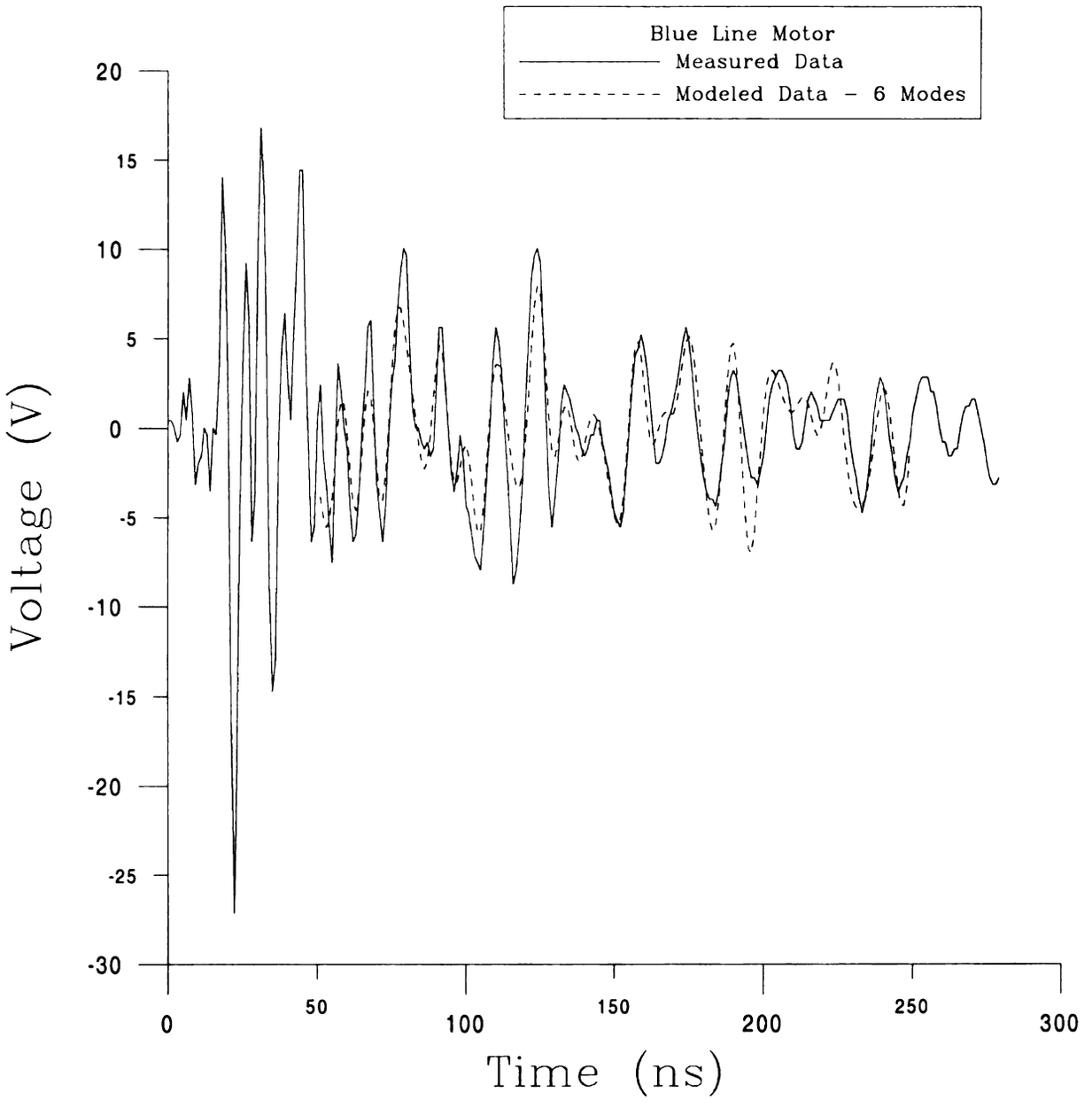

Figure 4-34: Modeled Blue Line Motor Data Assuming 6 Natural Modes

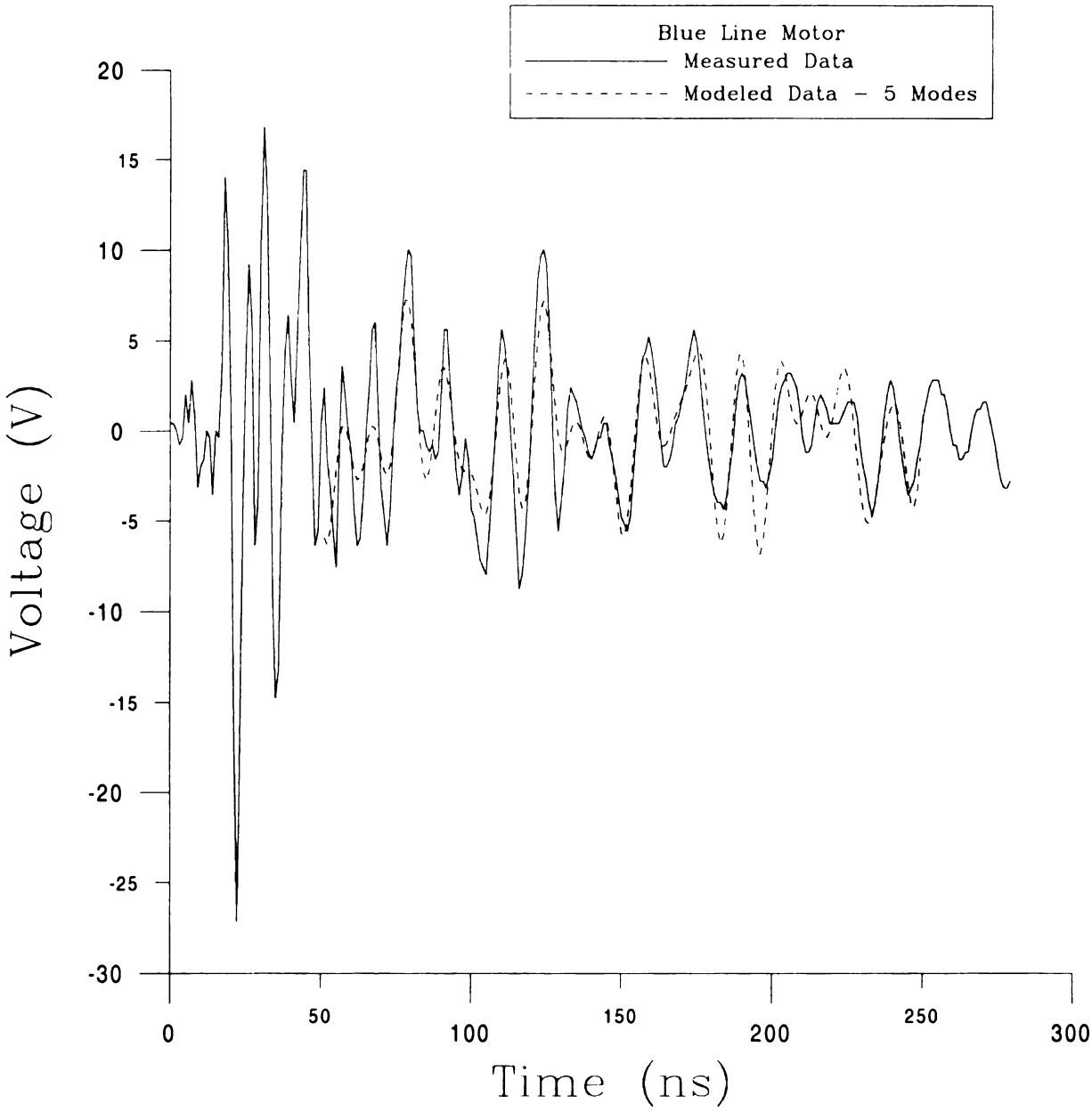

Figure 4-35: Modeled Blue Line Motor Data Assuming 5 Natural Modes

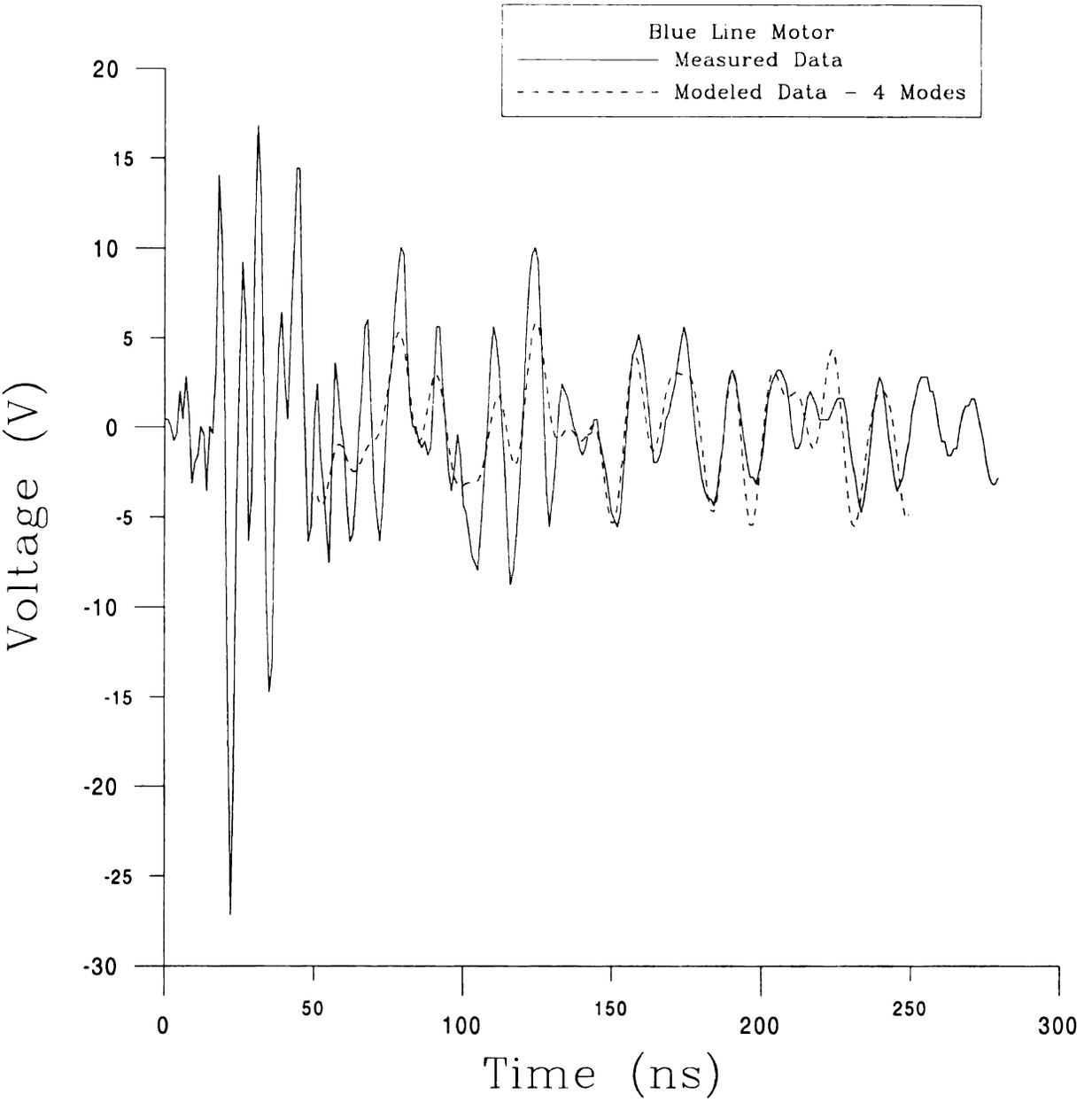

Figure 4-36: Modeled Blue Line Motor Data Assuming 4 Natural Modes

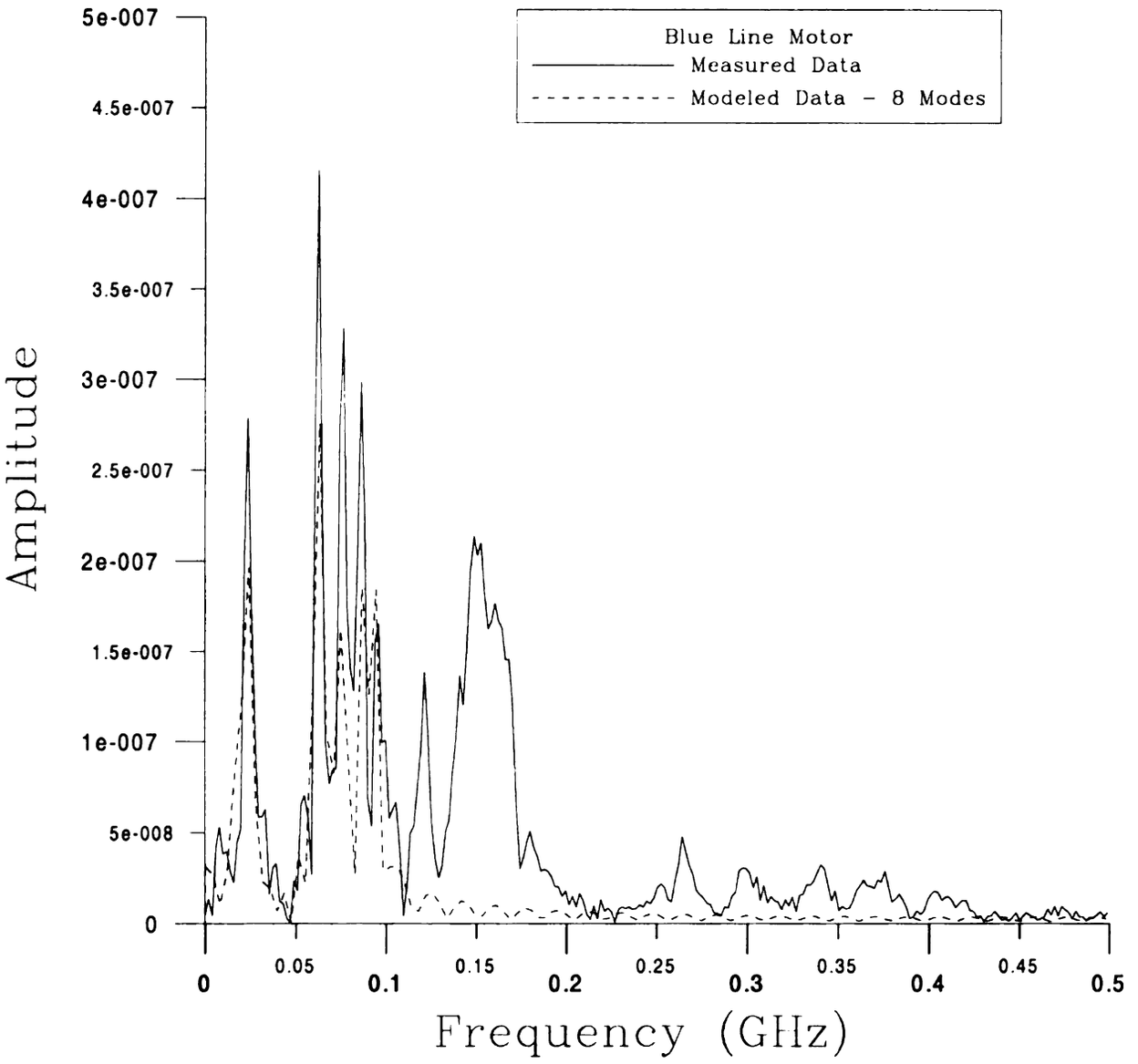

Figure 4-37: Frequency Content of Modeled Blue Line Motor Data Assuming 8 Natural Modes

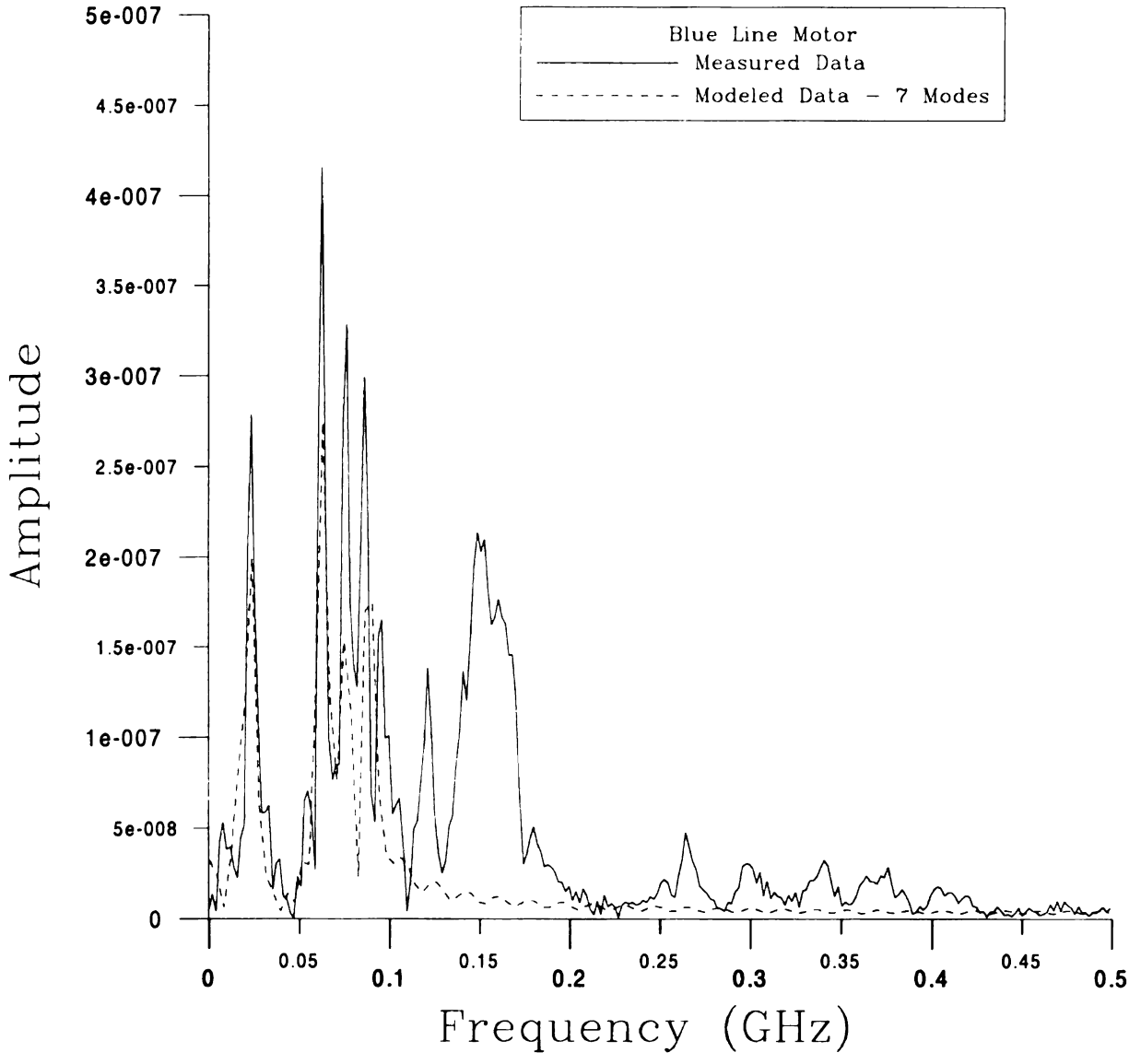

Figure 4-38: Frequency Content of Modeled Blue Line Motor Data Assuming 7 Natural Modes

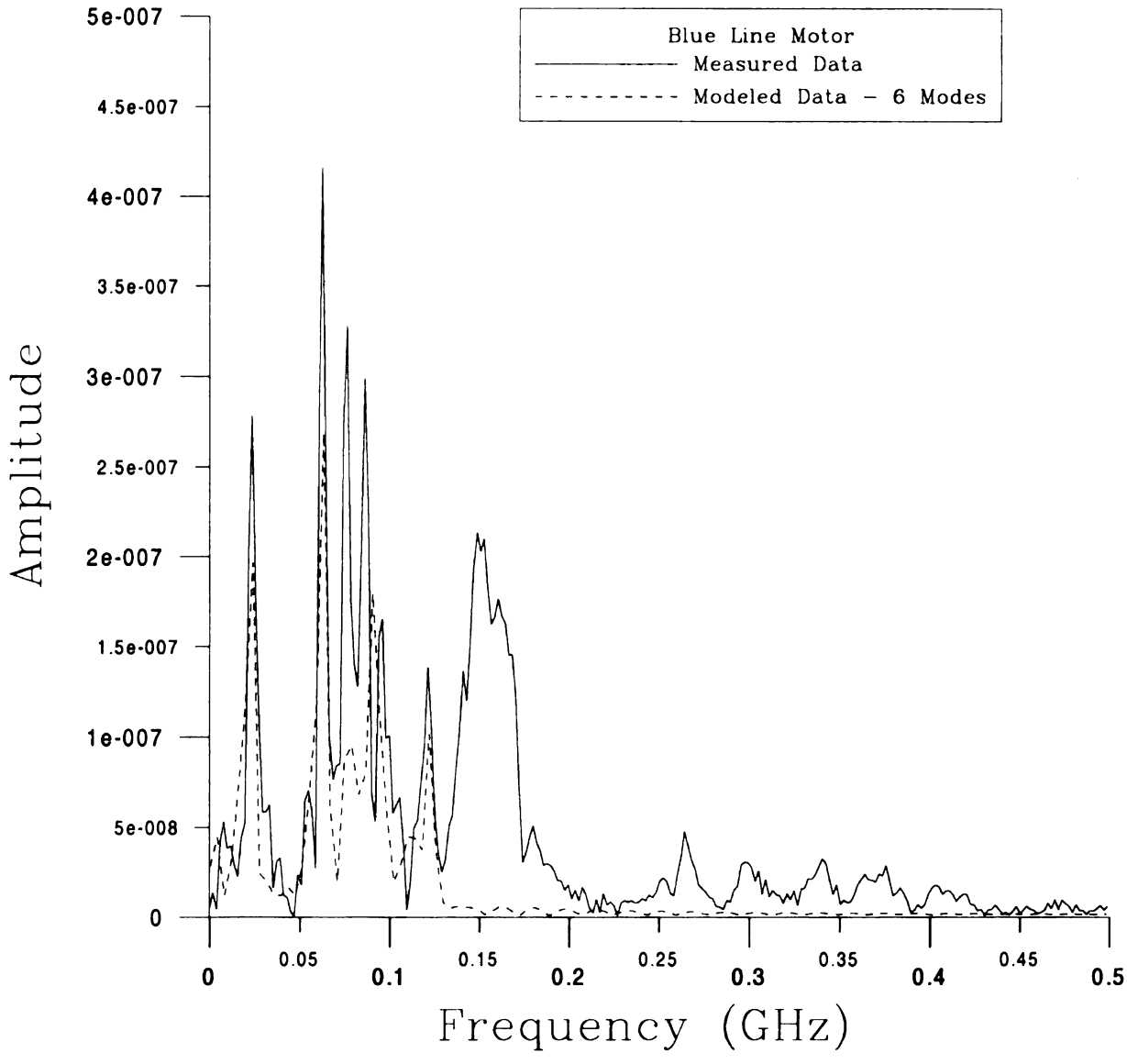

Figure 4-39: Frequency Content of Modeled Blue Line Motor Data Assuming 6 Natural Modes

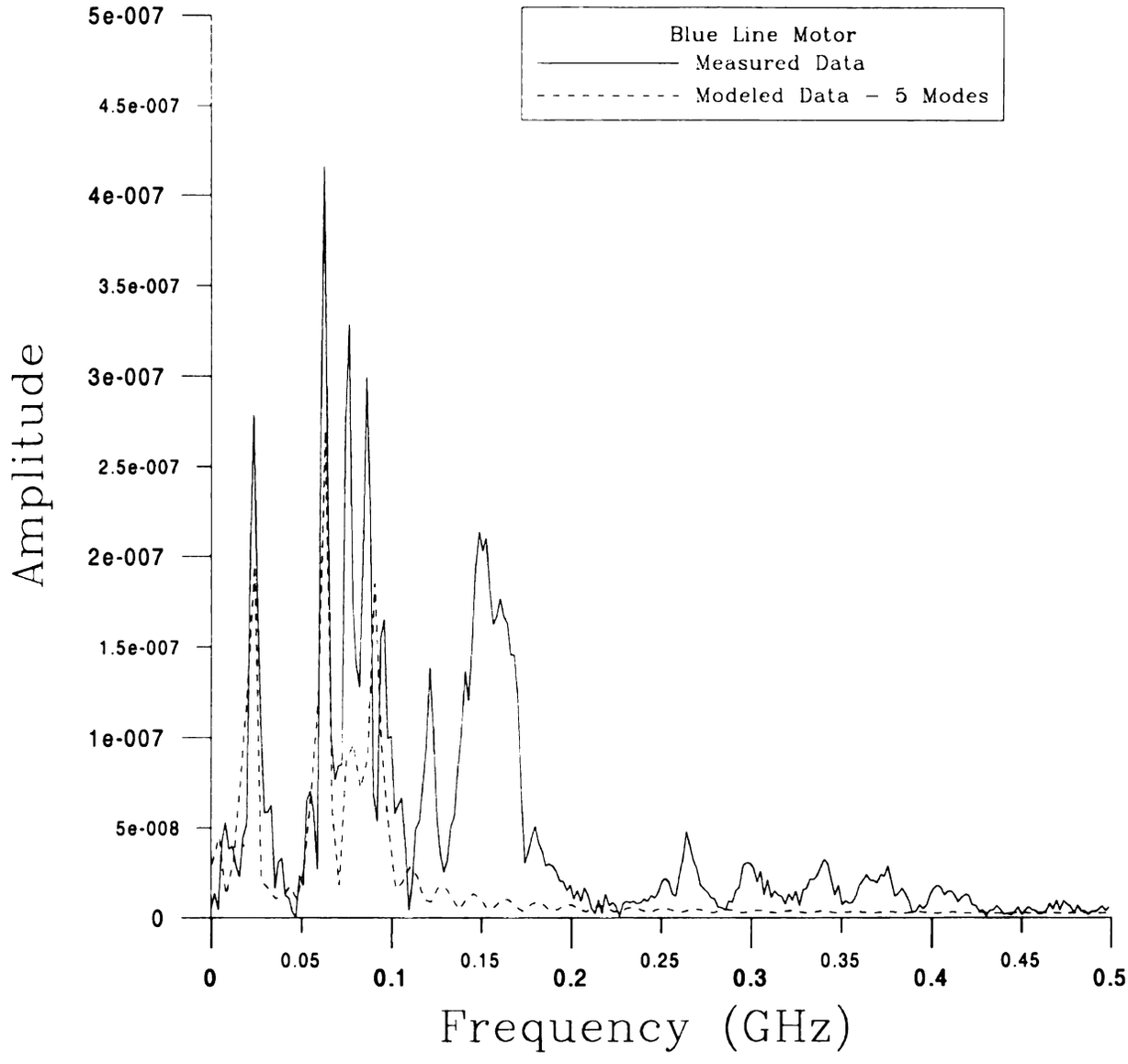

Figure 4-40: Frequency Content of Modeled Blue Line Motor Data Assuming 5 Natural Modes

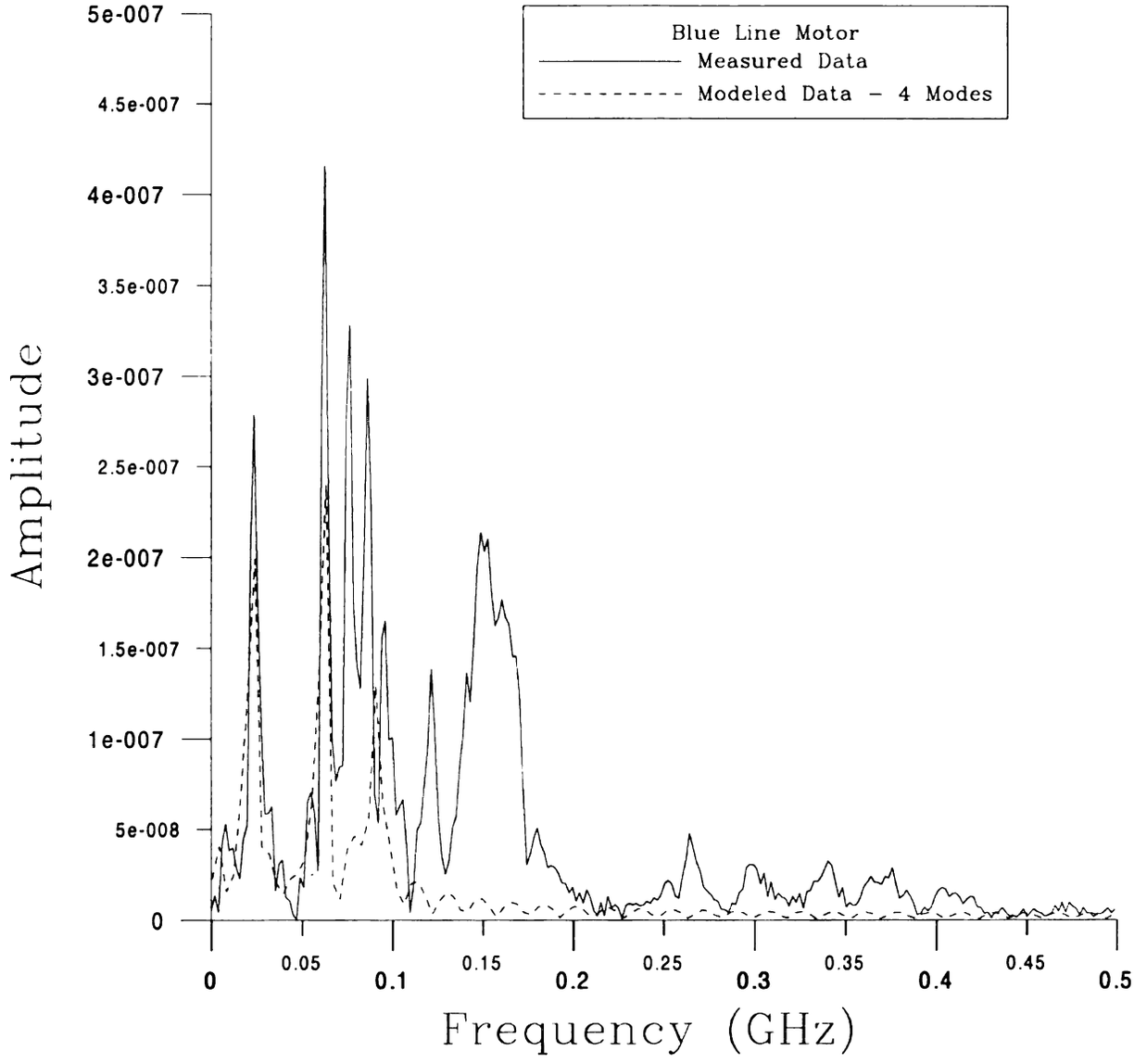

Figure 4-41: Frequency Content of Modeled Blue Line Motor Data Assuming 4 Natural Modes

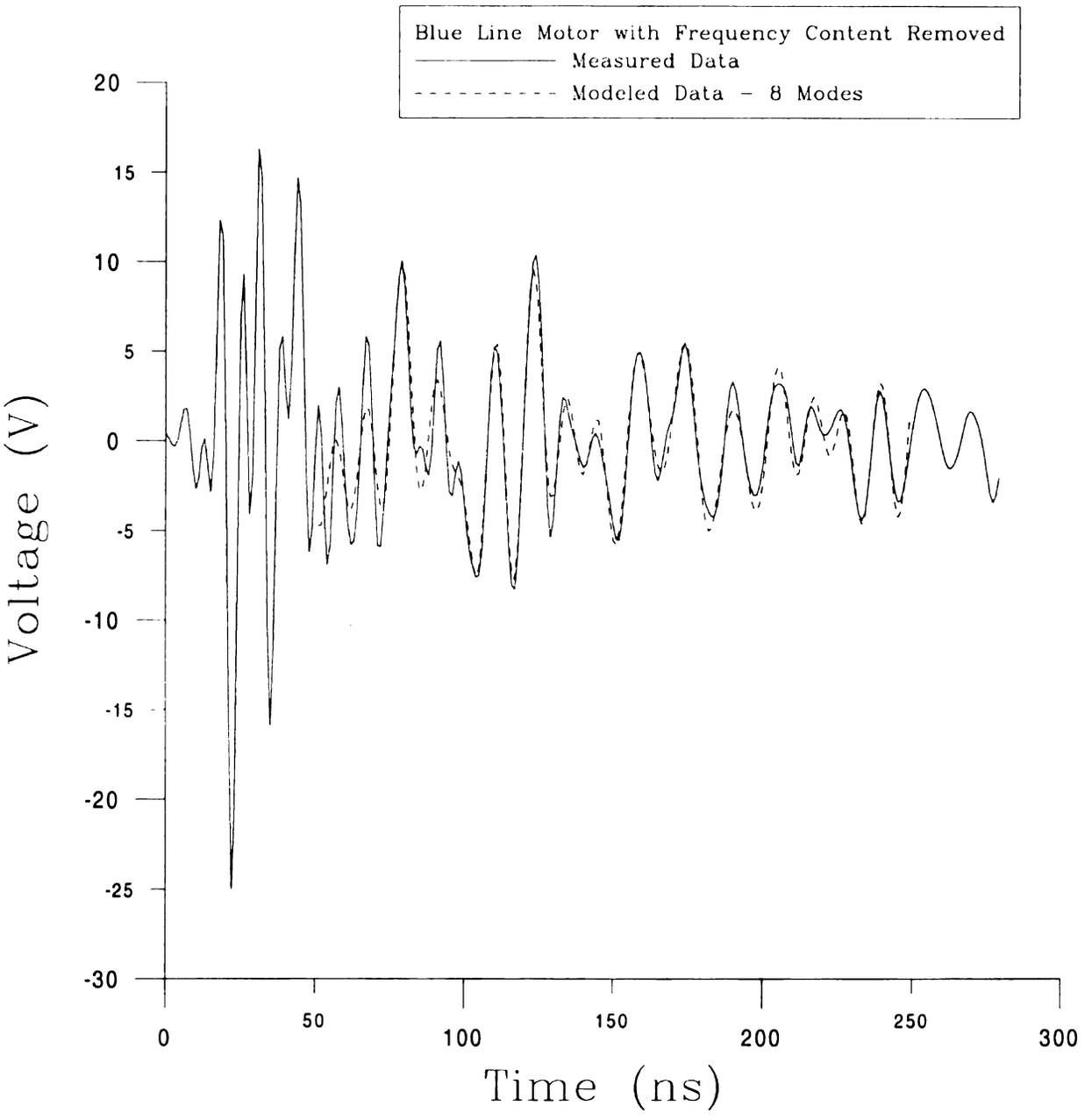

Figure 4-42: Modeled Blue Line Motor Data with Frequency Truncated at 200 MHz Assuming 8 Natural Modes

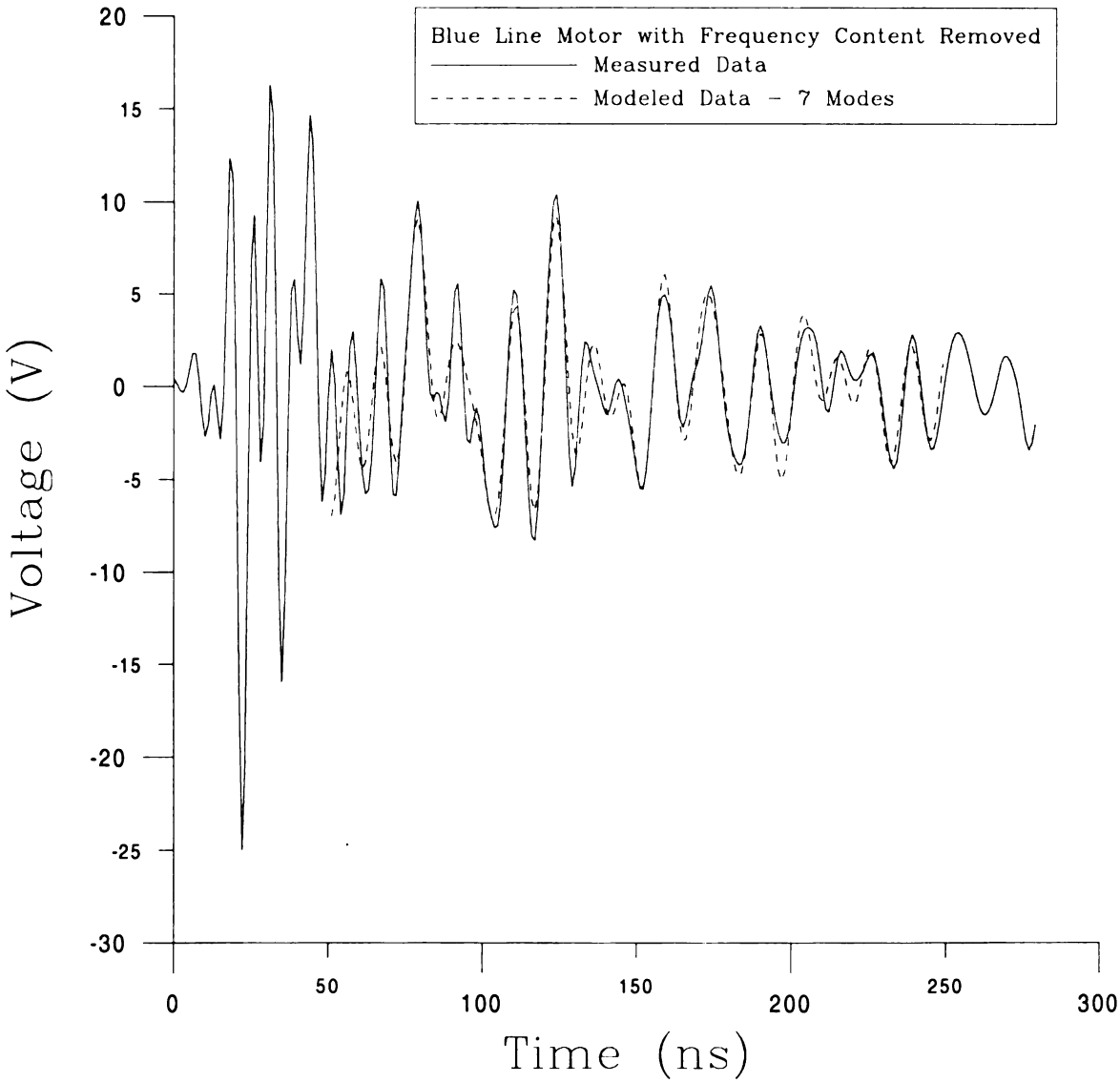

Figure 4-43: Modeled Blue Line Motor Data with Frequency Truncated at 200 MHz Assuming 7 Natural Modes

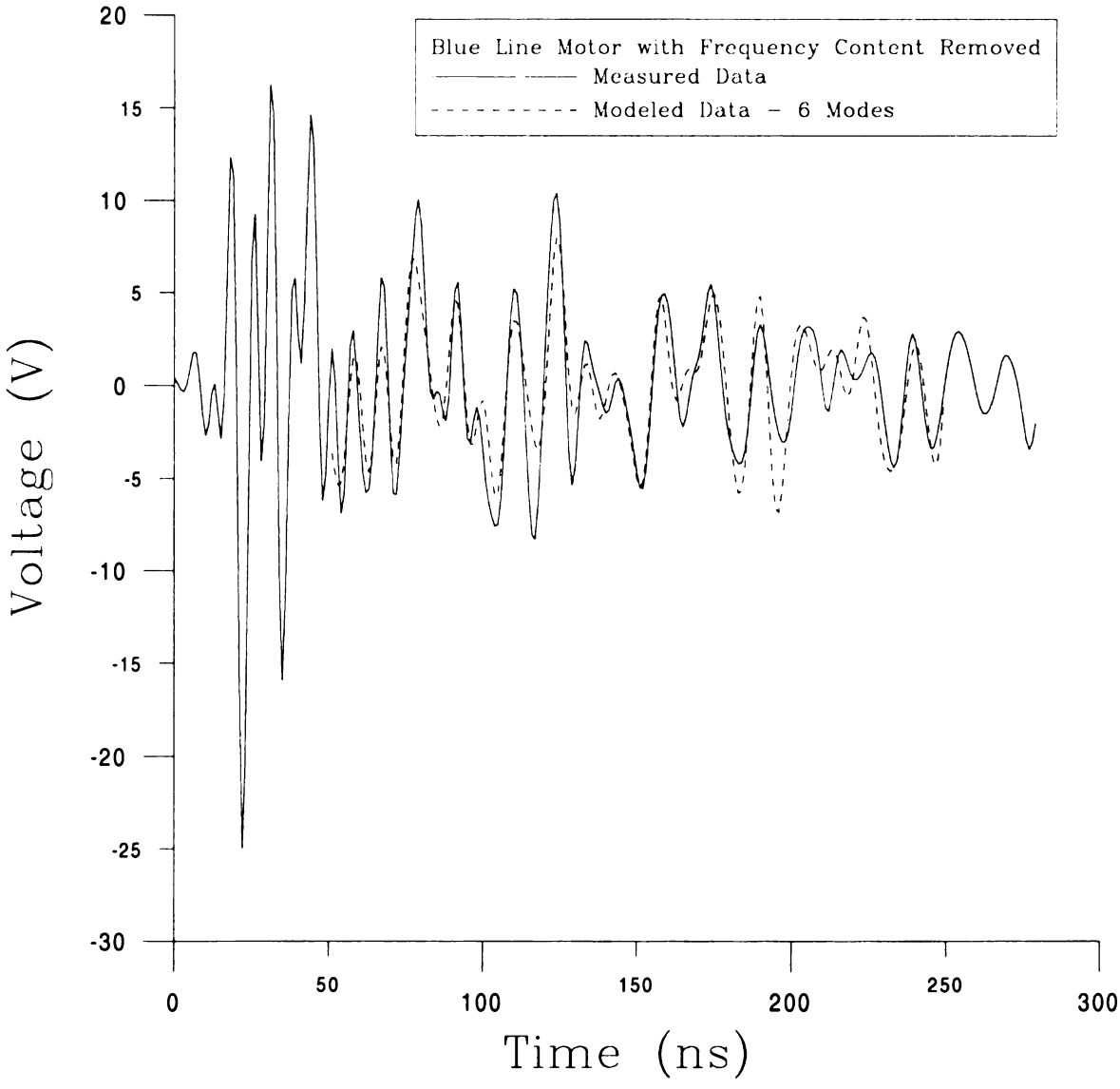

Figure 4-44: Modeled Blue Line Motor Data with Frequency Truncated at 200 MHz Assuming 6 Natural Modes

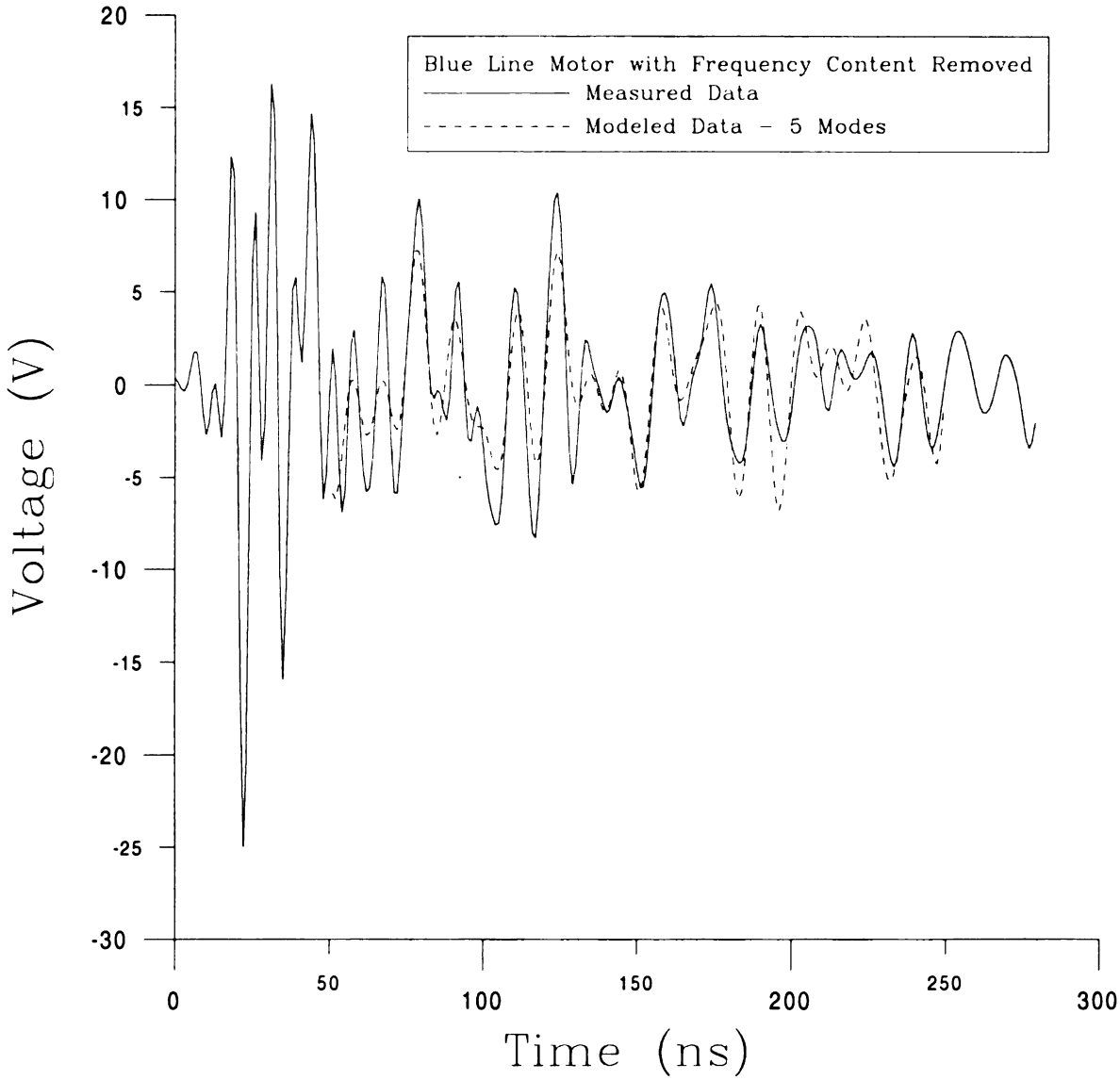

Figure 4-45: Modeled Blue Line Motor Data with Frequency Truncated at 200 MHz Assuming 5 Natural Modes

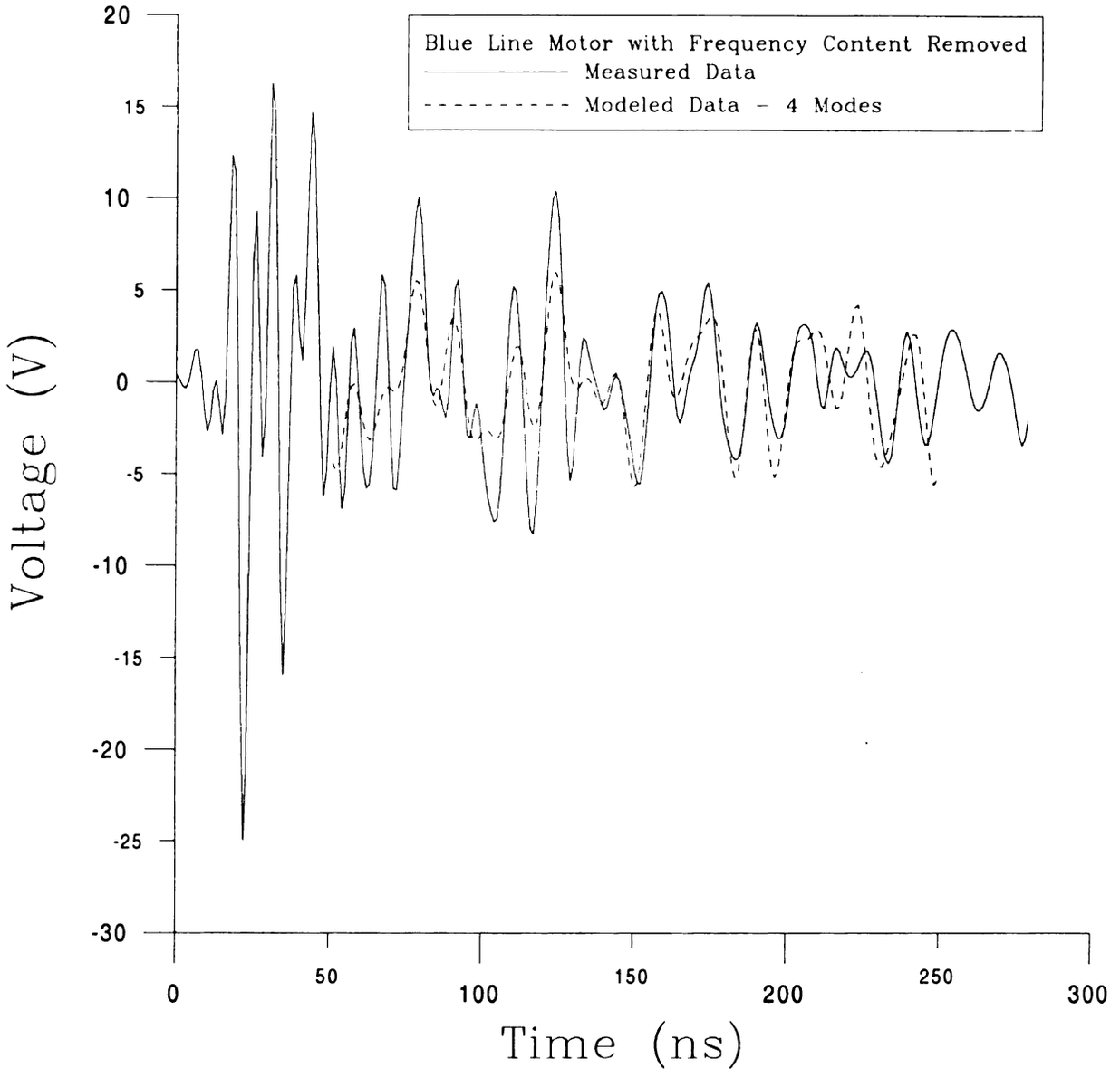

Figure 4-46: Modeled Blue Line Motor Data with Frequency Truncated at 200 MHz Assuming 4 Natural Modes

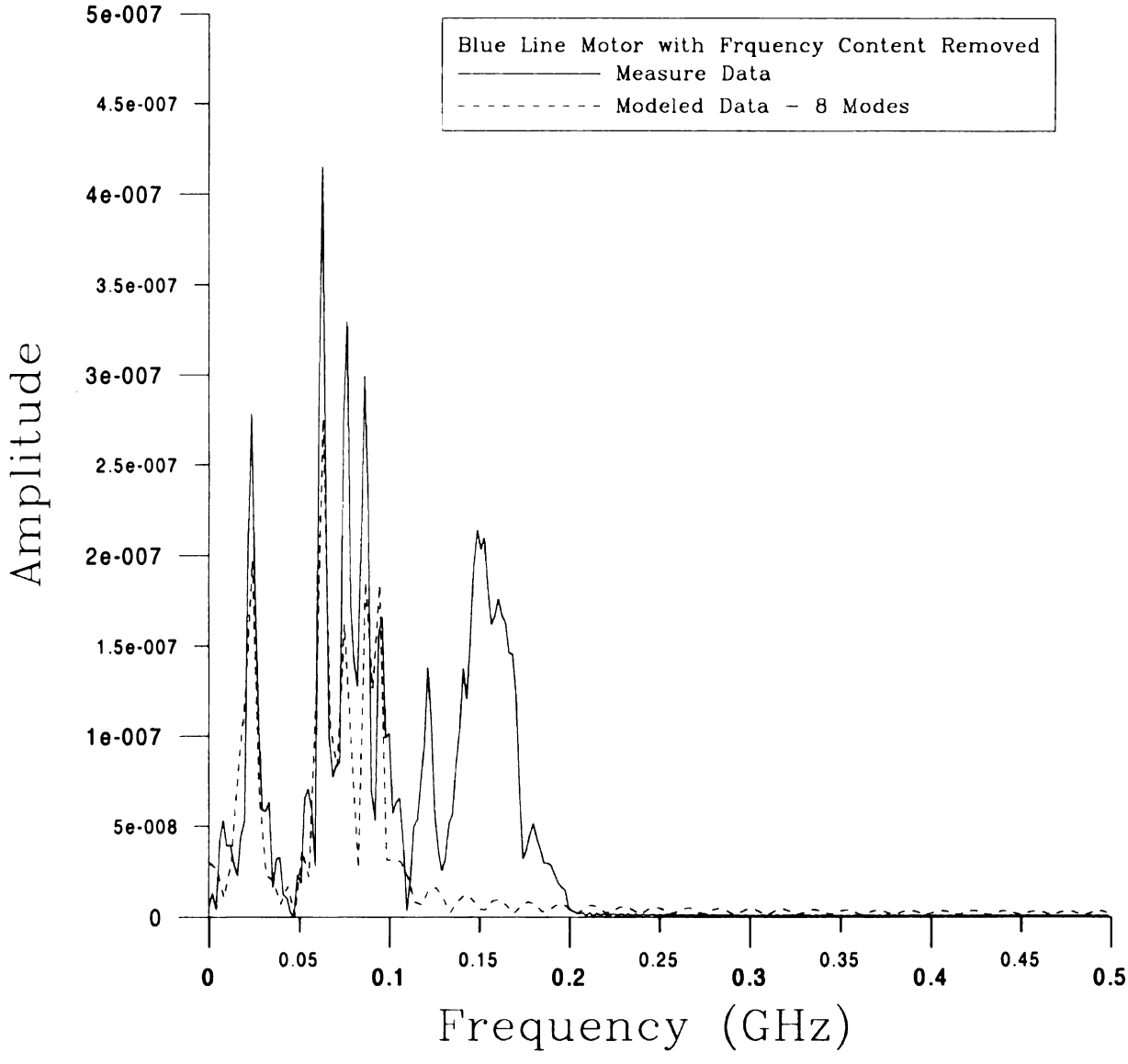

Figure 4-47: Frequency Content of Modeled Blue Line Motor Data with Frequency Truncated at 200 MHz Assuming 8 Natural Modes

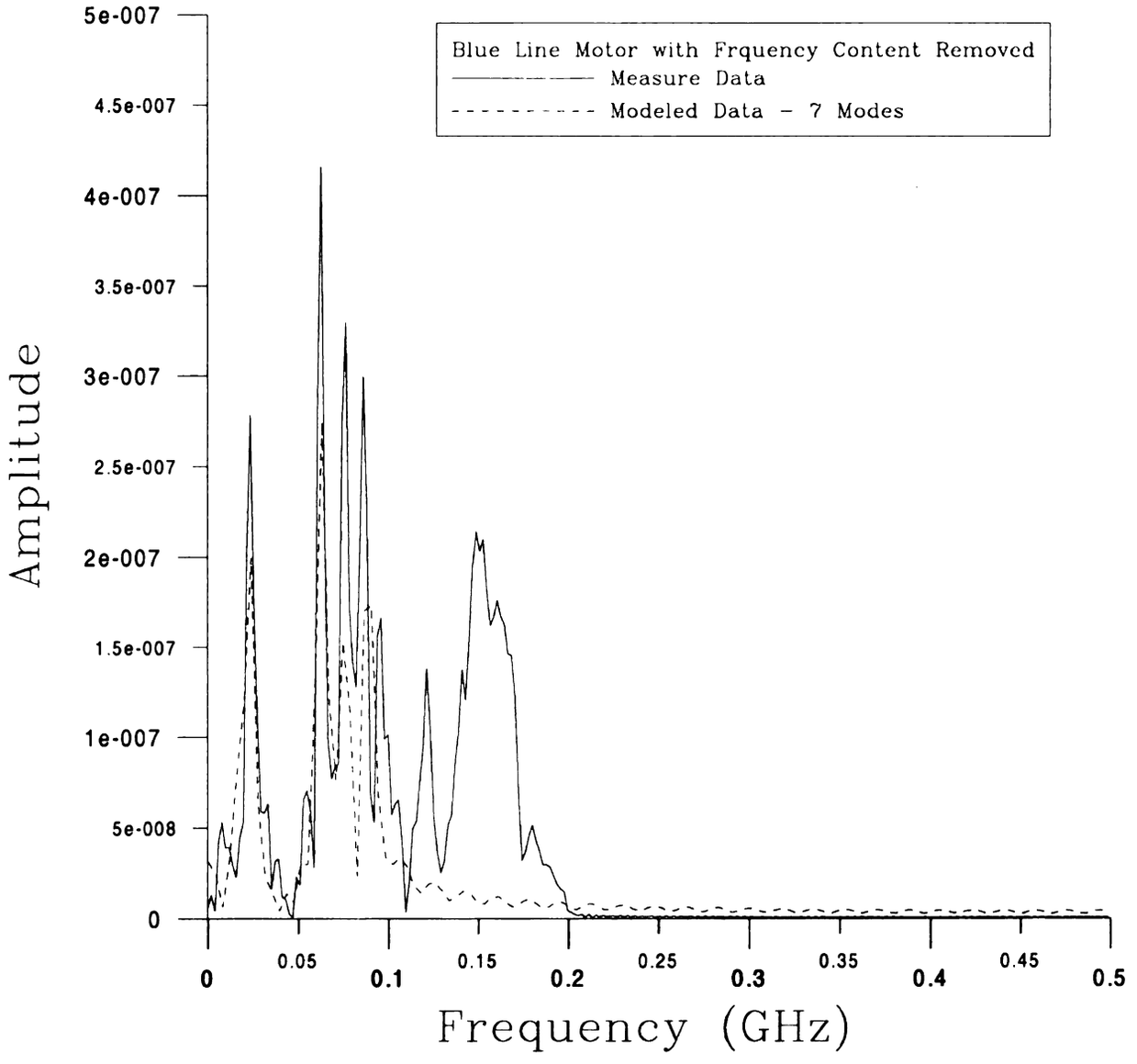

Figure 4-48: Frequency Content of Modeled Blue Line Motor Data with Frequency Truncated at 200 MHz Assuming 7 Natural Modes

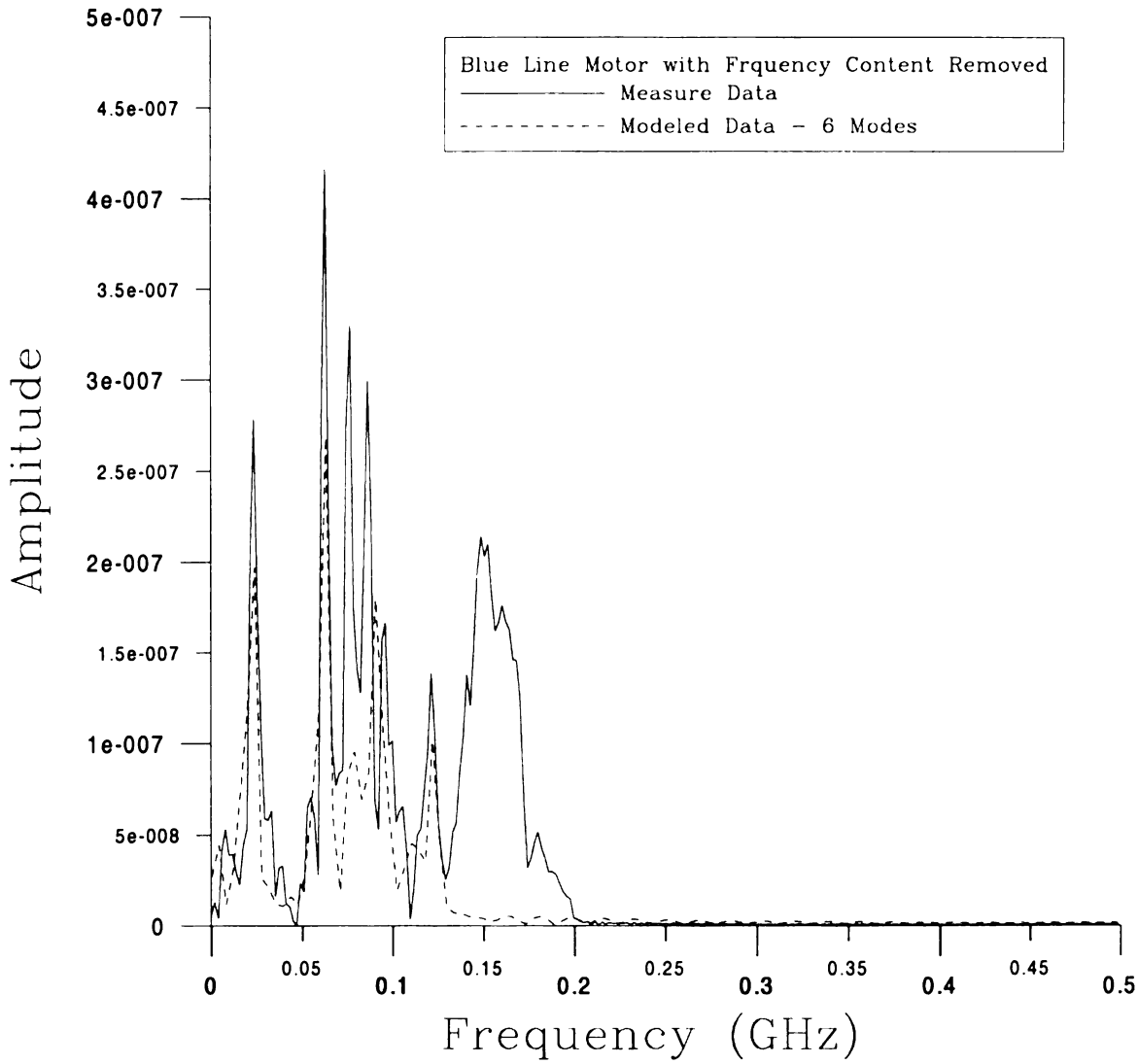

Figure 4-49: Frequency Content of Modeled Blue Line Motor Data with Frequency Truncated at 200 MHz Assuming 6 Natural Modes

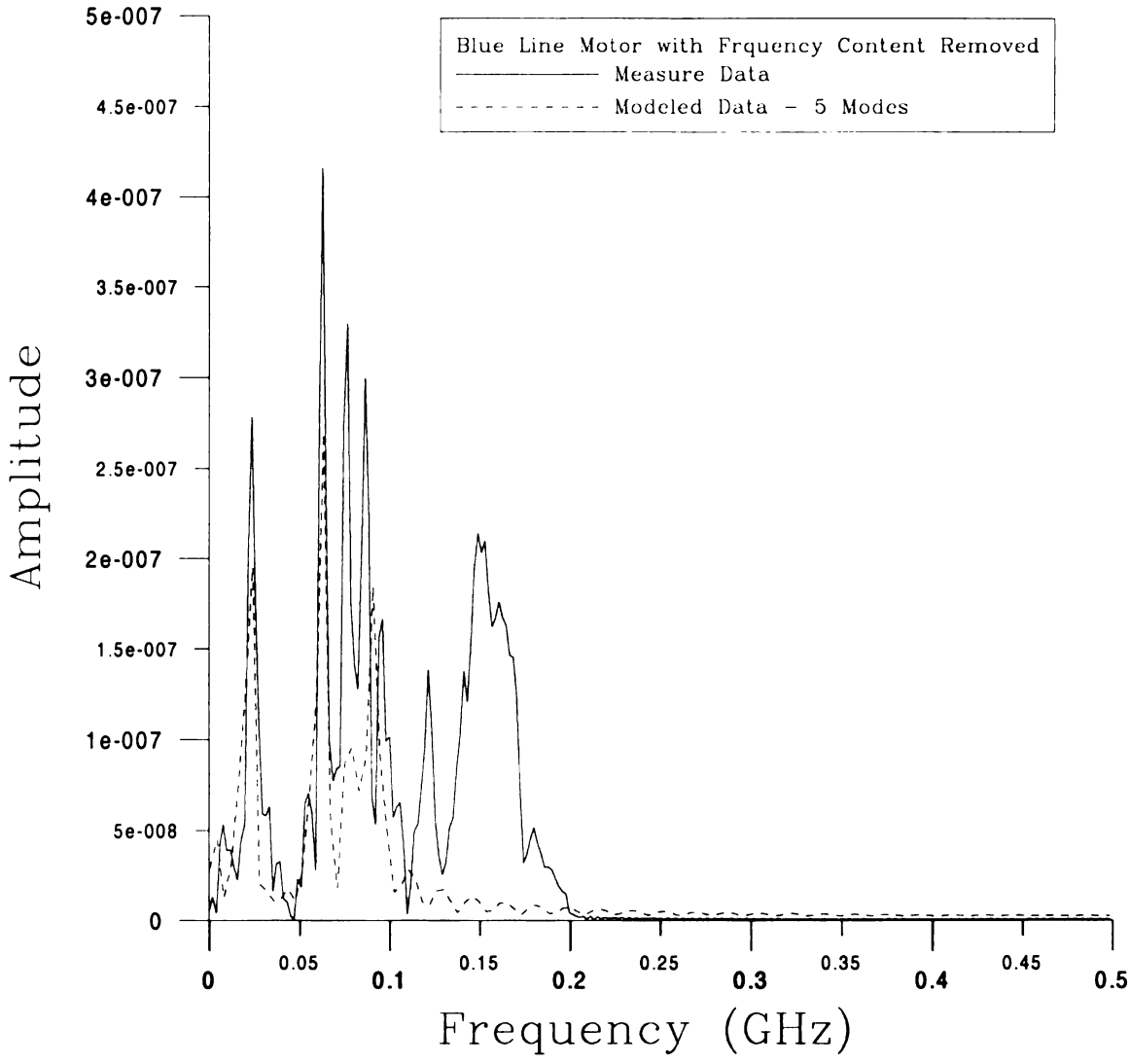

Figure 4-50: Frequency Content of Modeled Blue Line Motor Data with Frequency Truncated at 200 MHz Assuming 5 Natural Modes

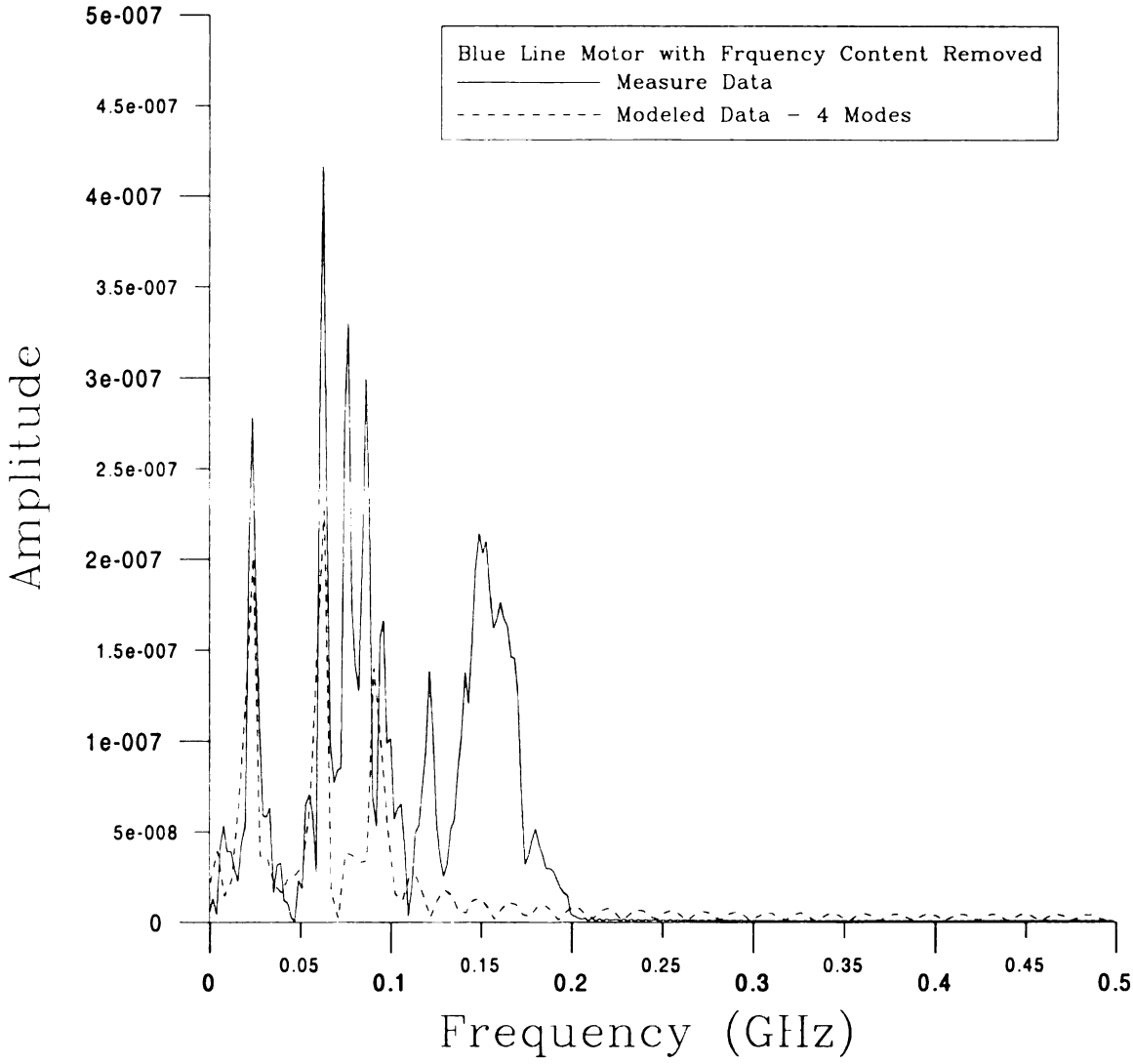

Figure 4-51: Frequency Content of Modeled Blue Line Motor Data with Frequency Truncated at 200 MHz Assuming 4 Natural Modes

Similar to the blue line motor, the green line motor has no significant frequency contributions beyond 200 MHz, as seen in Figure 2-16. Additionally, the green line motor also has a very wide natural mode at around 150 MHz; however, it has much greater amplitude than the one for the blue line motor. Both the blue line and green line motors are positional motors used in a rear view mirror unit, so it is expected that they will have very similar characteristics.

The models for seven and eight expected modes, in Figures 4-52, 4-53, 4-57, and 4-58, are virtually identical. They match the time domain measured data fairly well, and in the frequency domain the first four modes are matched to some extent. Additionally both the modes beyond 100 MHz are also matched, although the amplitude of the modeled mode at around 150 MHz is much less than the measured data. The modeled data with frequency truncated at 200 MHz, in Figures 4-58 and 4-68, is the same as the non-truncated model, except the wide pulse at 150 MHz is actually modeled as two pulses, one at 150 MHz and one at 175 MHz. There is very little difference between the modeled time domain data of the truncated (Figures 4-62 and 4-63) and non-truncated (Figures 4-52 and 4-53) data Models, although the truncated data models appear to be marginally less accurate. For the six expected natural mode models, seen in Figures 5-59 and 5-69, both the truncated and non-truncated models lack the first mode beyond 100 MHz, but also both break the larger pulse at 150 MHz into two smaller pulses. The five expected natural mode models, in Figures 4-60 and 4-70, are very similar to the six expected natural mode models, except the wide pulse at 150 MHz is modeled as a single pulse. The time domain data for five and six expected natural modes, seen in Figures 4-54, 4-55, 4-64, and 4-65, are all very similar and all agree with the measured data about

the same. The model for four expected natural modes does not model an frequency content beyond 100 MHz and only models three of the earlier modes, as seen in Figures 4-61 and 4-71. The time domain data in Figures 4-56 and 4-66, retains the shape of the measured data, but does not match as closely as the models with more expected modes.

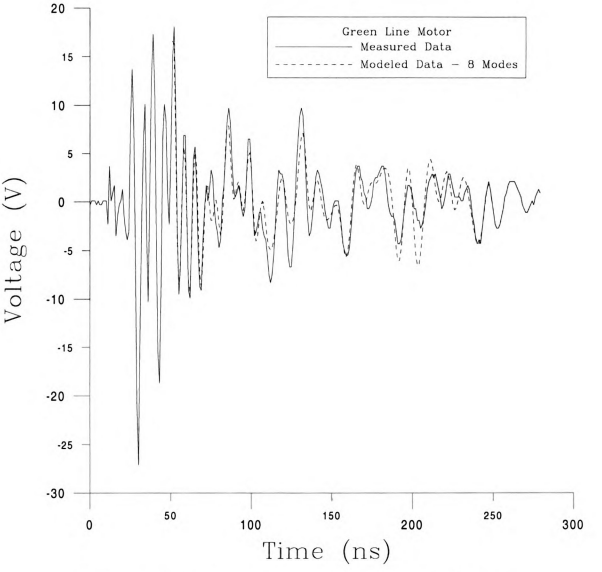

Figure 4-52: Modeled Green Line Motor Data Assuming 8 Natural Modes

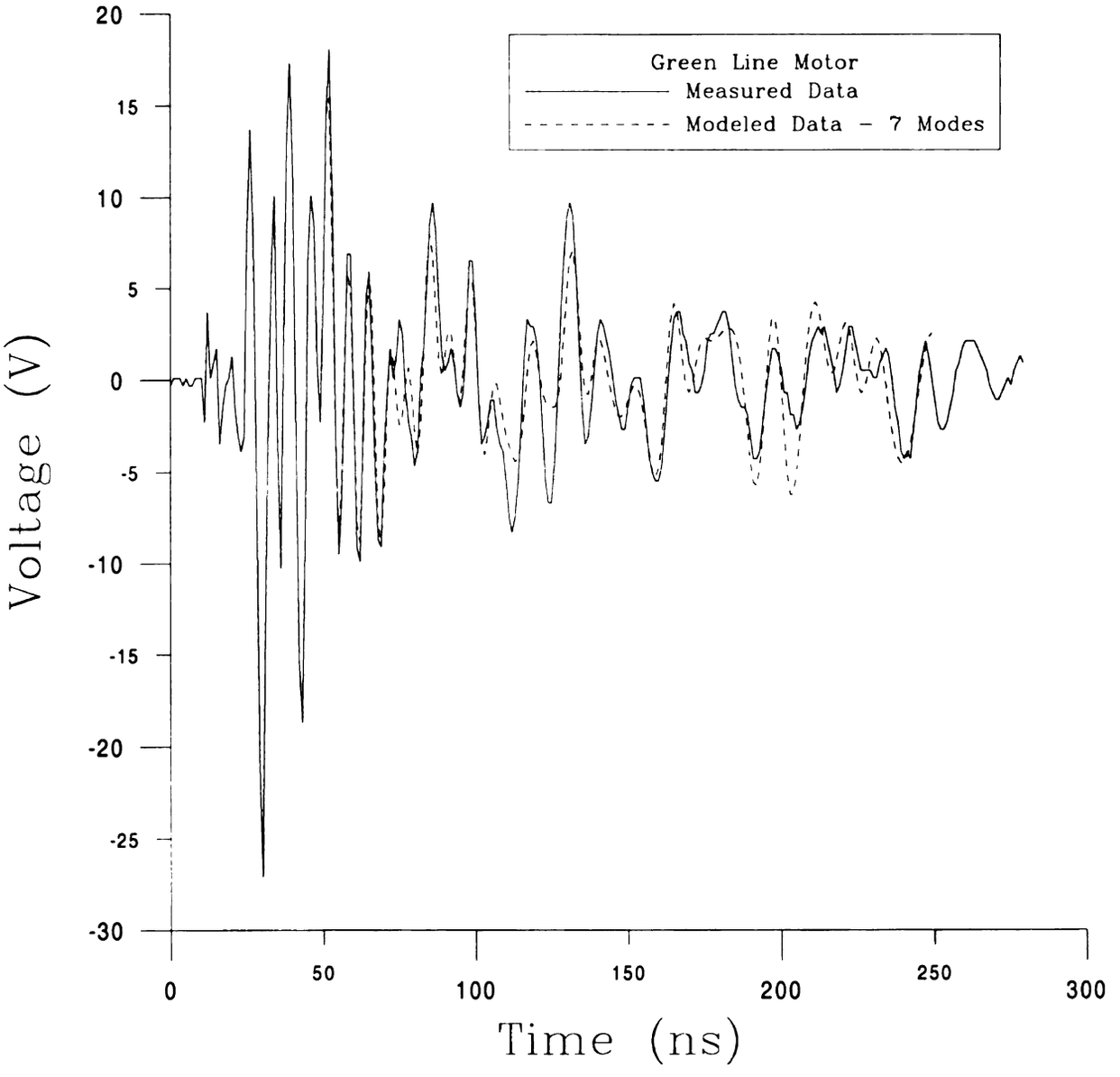

Figure 4-53: Modeled Green Line Motor Data Assuming 7 Natural Modes

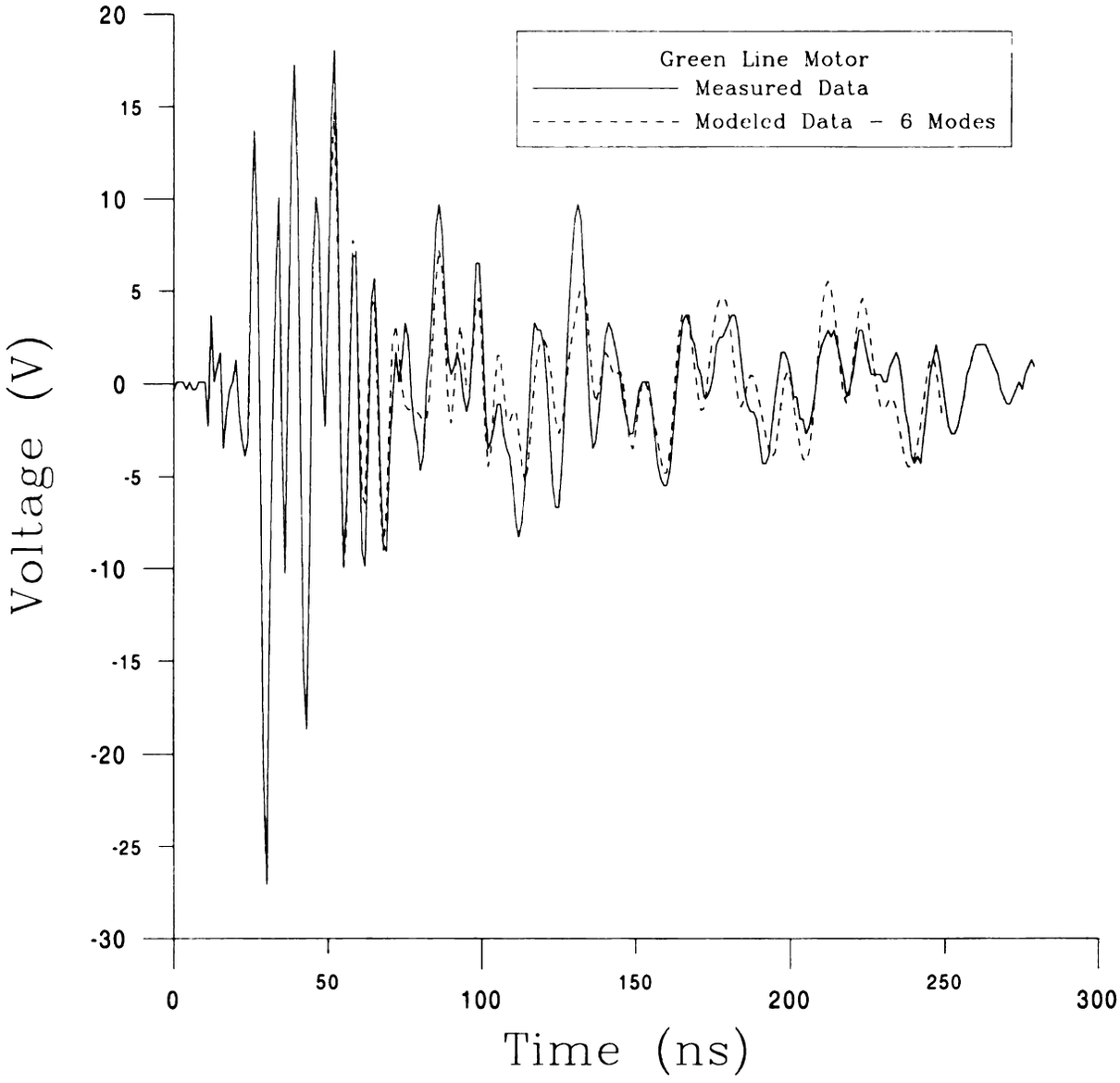

Figure 4-54: Modeled Green Line Motor Data Assuming 6 Natural Modes

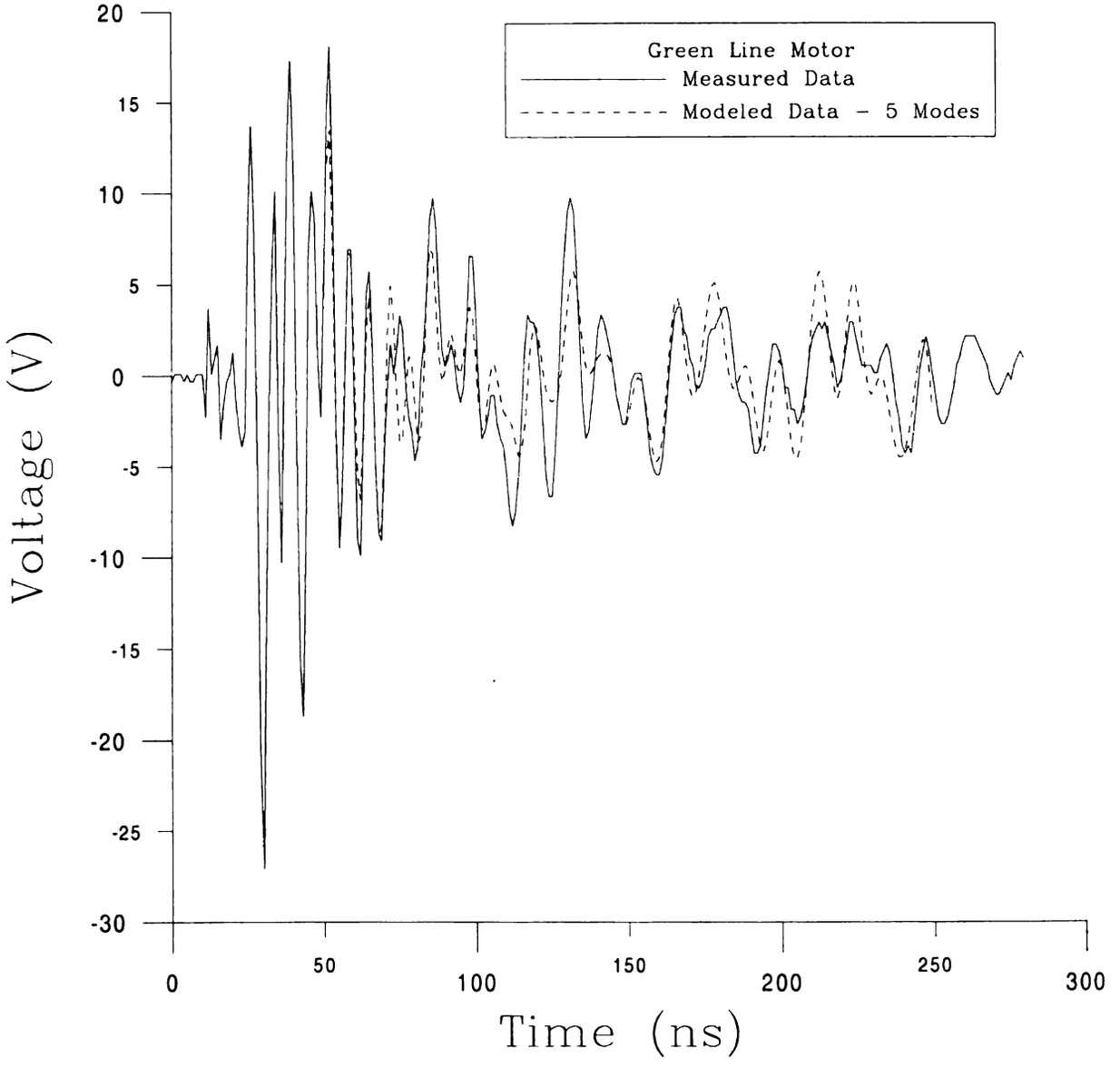

Figure 4-55: Modeled Green Line Motor Data Assuming 5 Natural Modes

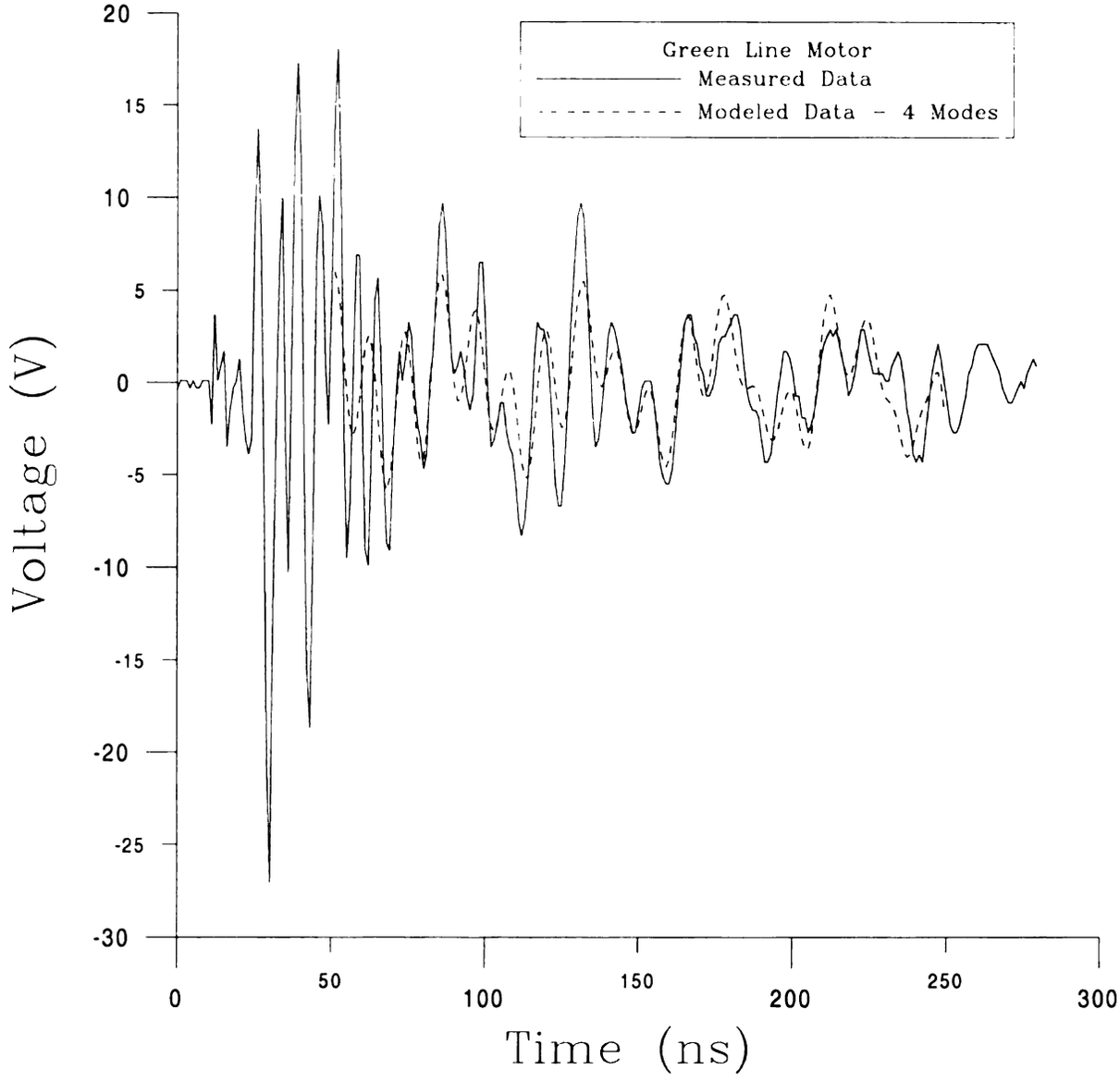

Figure 4-56: Modeled Green Line Motor Data Assuming 4 Natural Modes

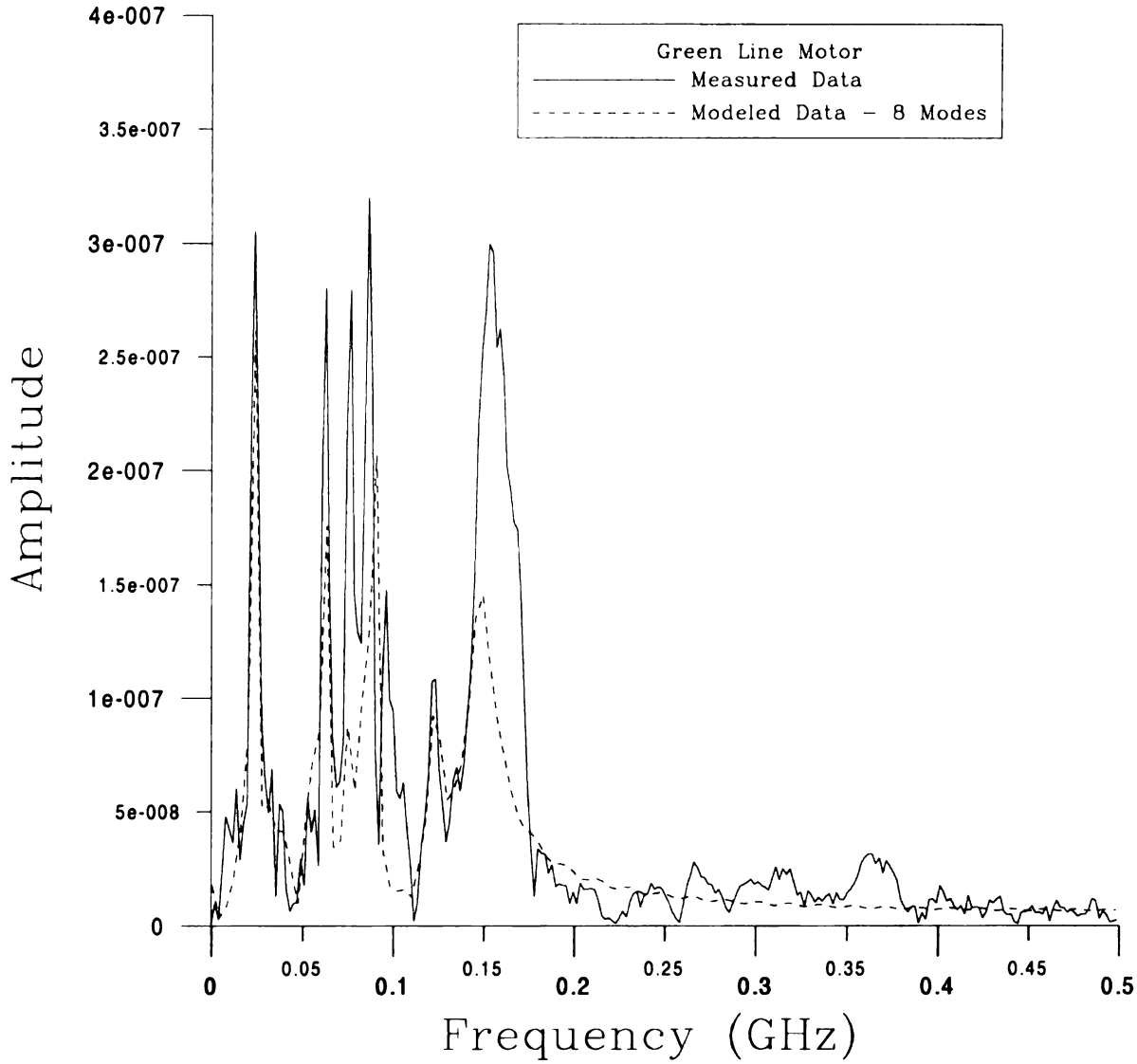

Figure 4-57: Frequency Content of Modeled Green Line Motor Data Assuming 8 Natural Modes

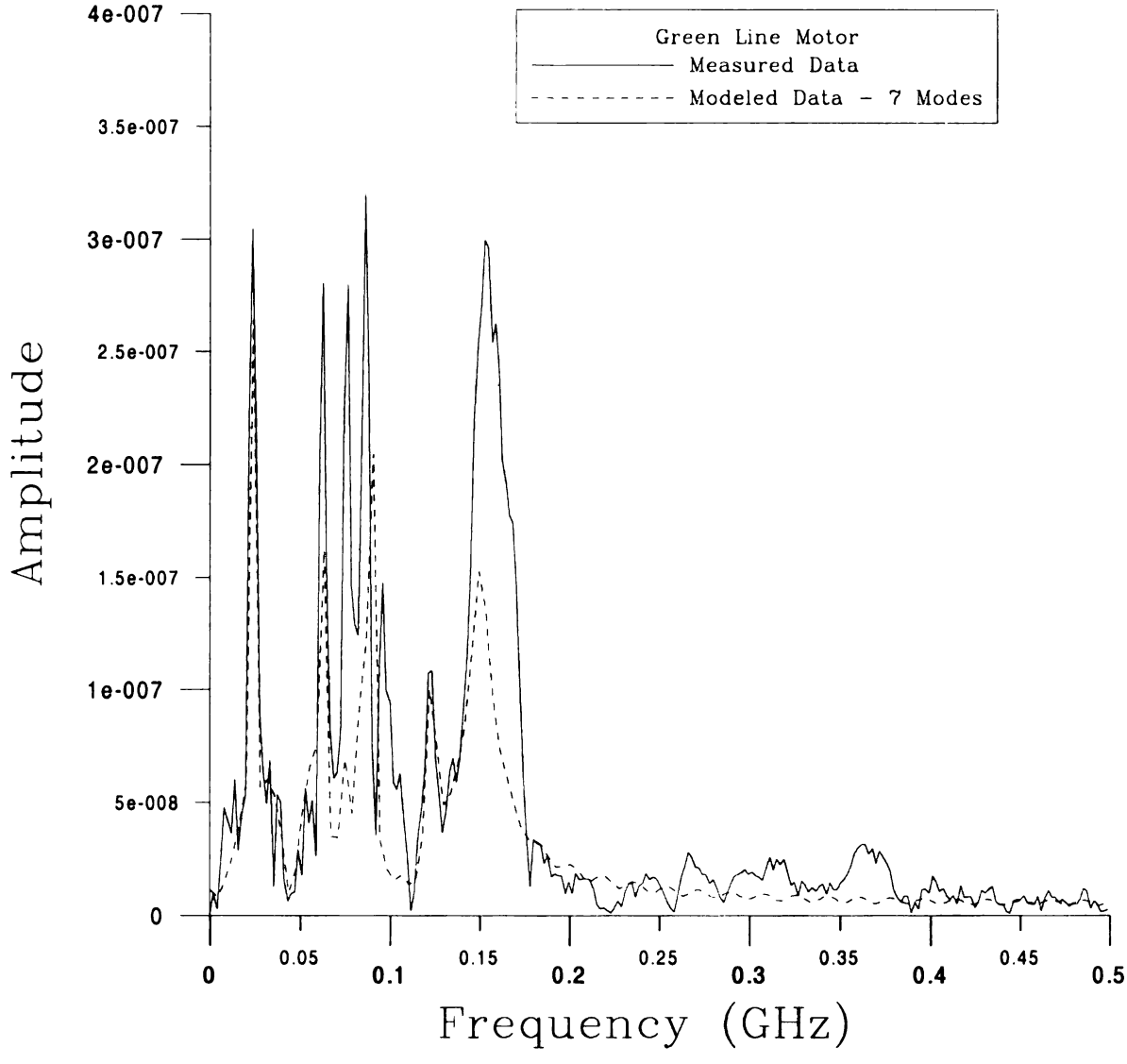

Figure 4-58: Frequency Content of Modeled Green Line Motor Data Assuming 7 Natural Modes

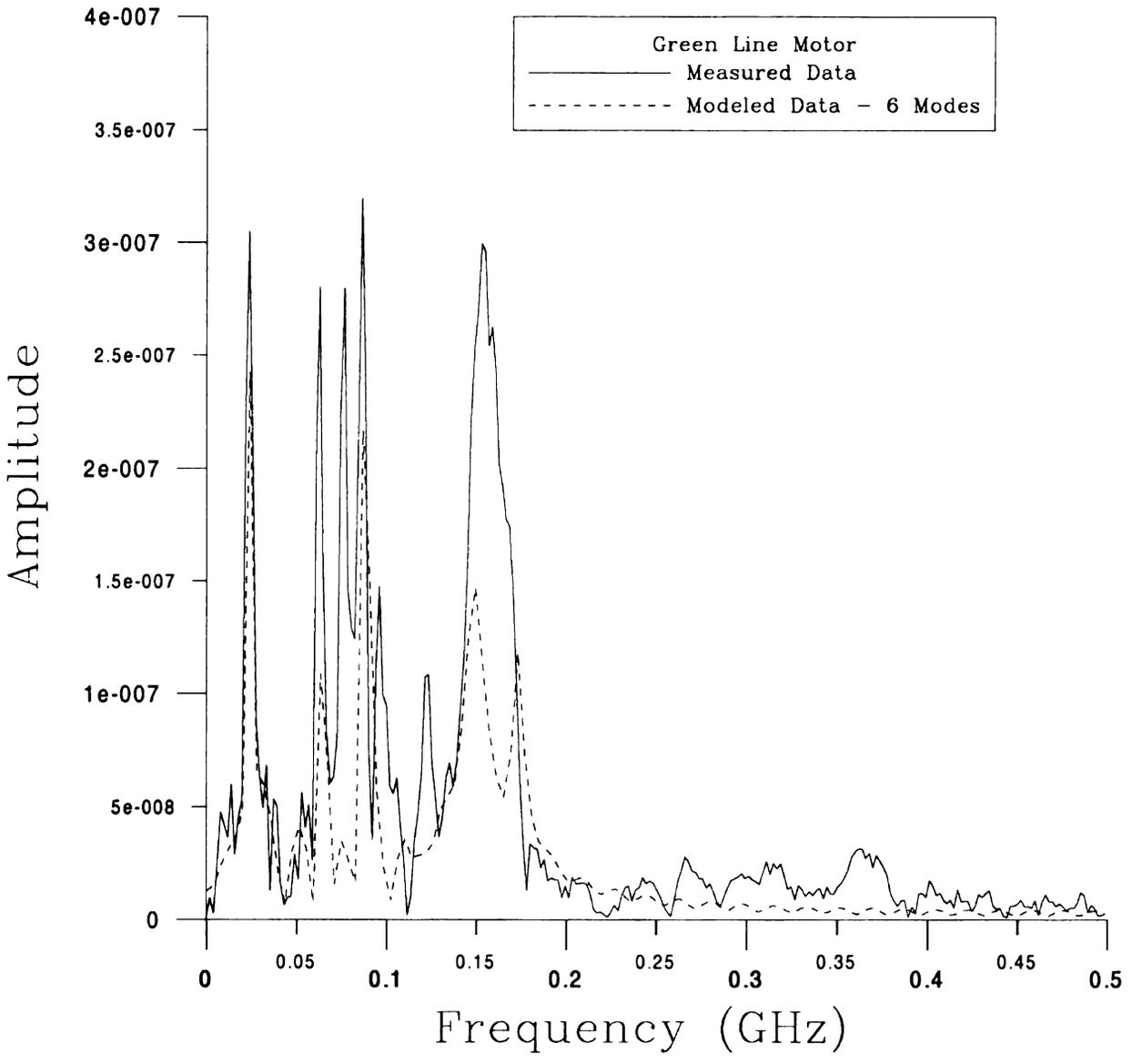

Figure 4-59: Frequency Content of Modeled Green Line Motor Data Assuming 6 Natural Modes

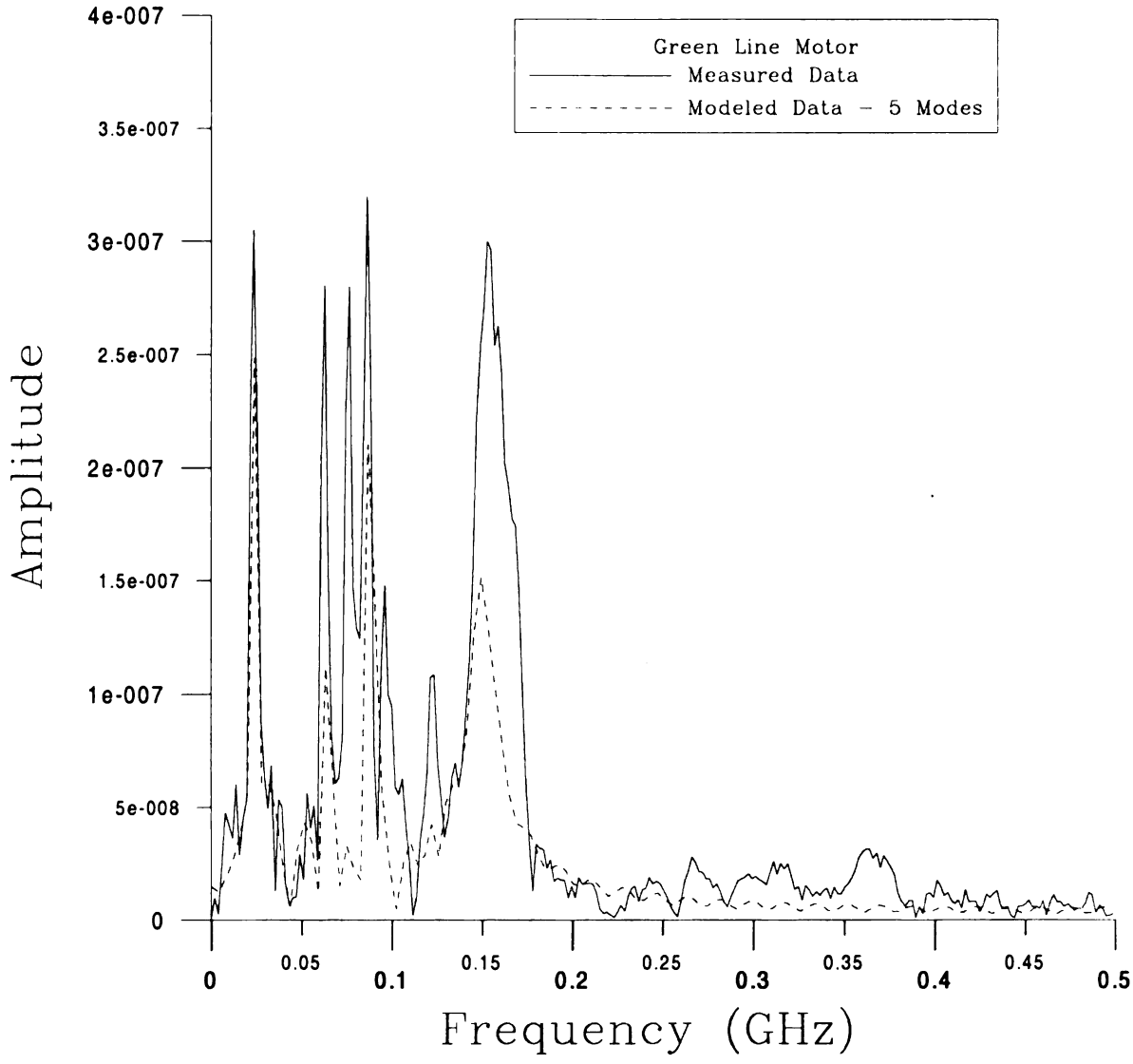

Figure 4-60: Frequency Content of Modeled Green Line Motor Data Assuming 5 Natural Modes

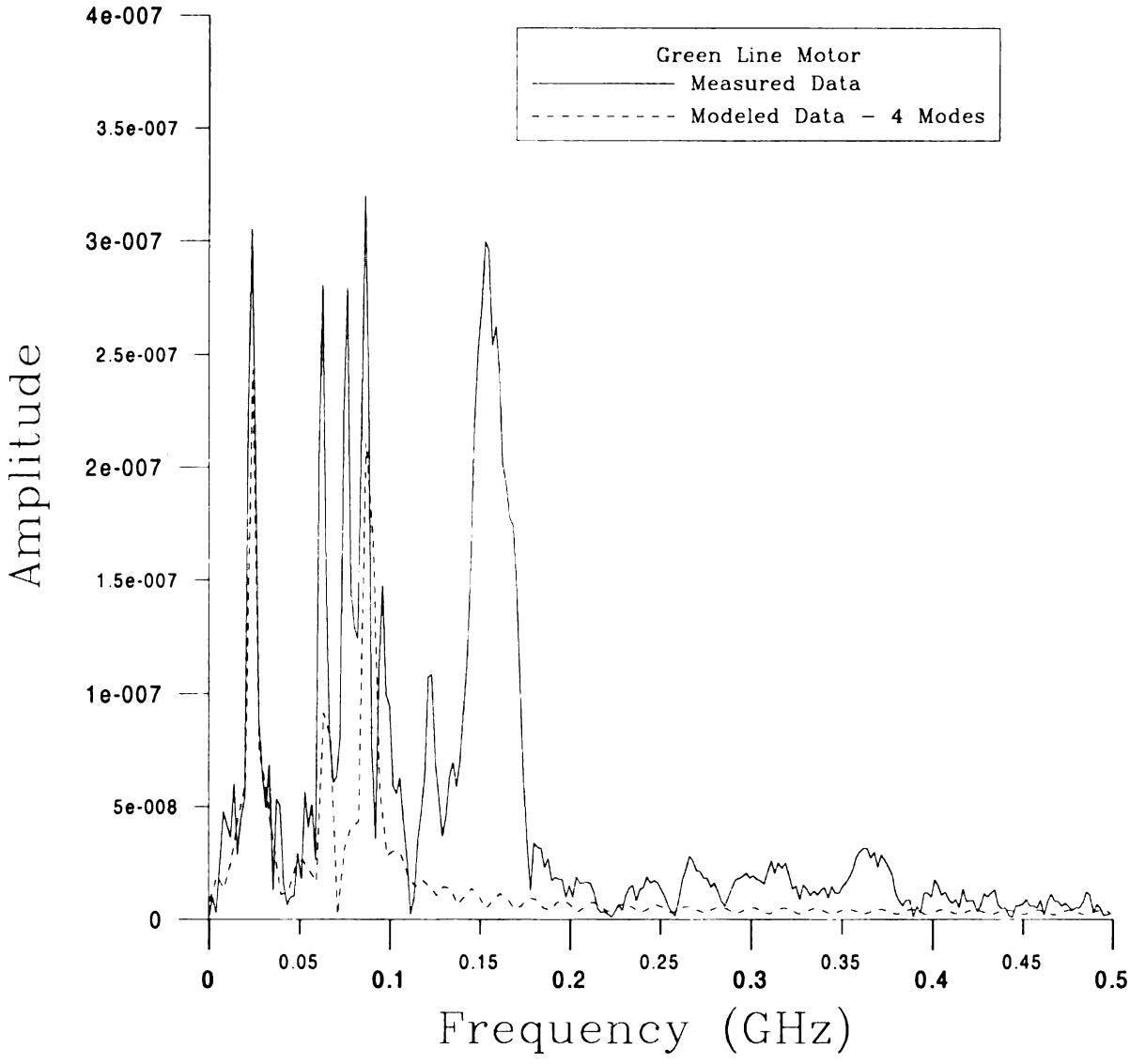

Figure 4-61: Frequency Content of Modeled Green Line Motor Data Assuming 4 Natural Modes

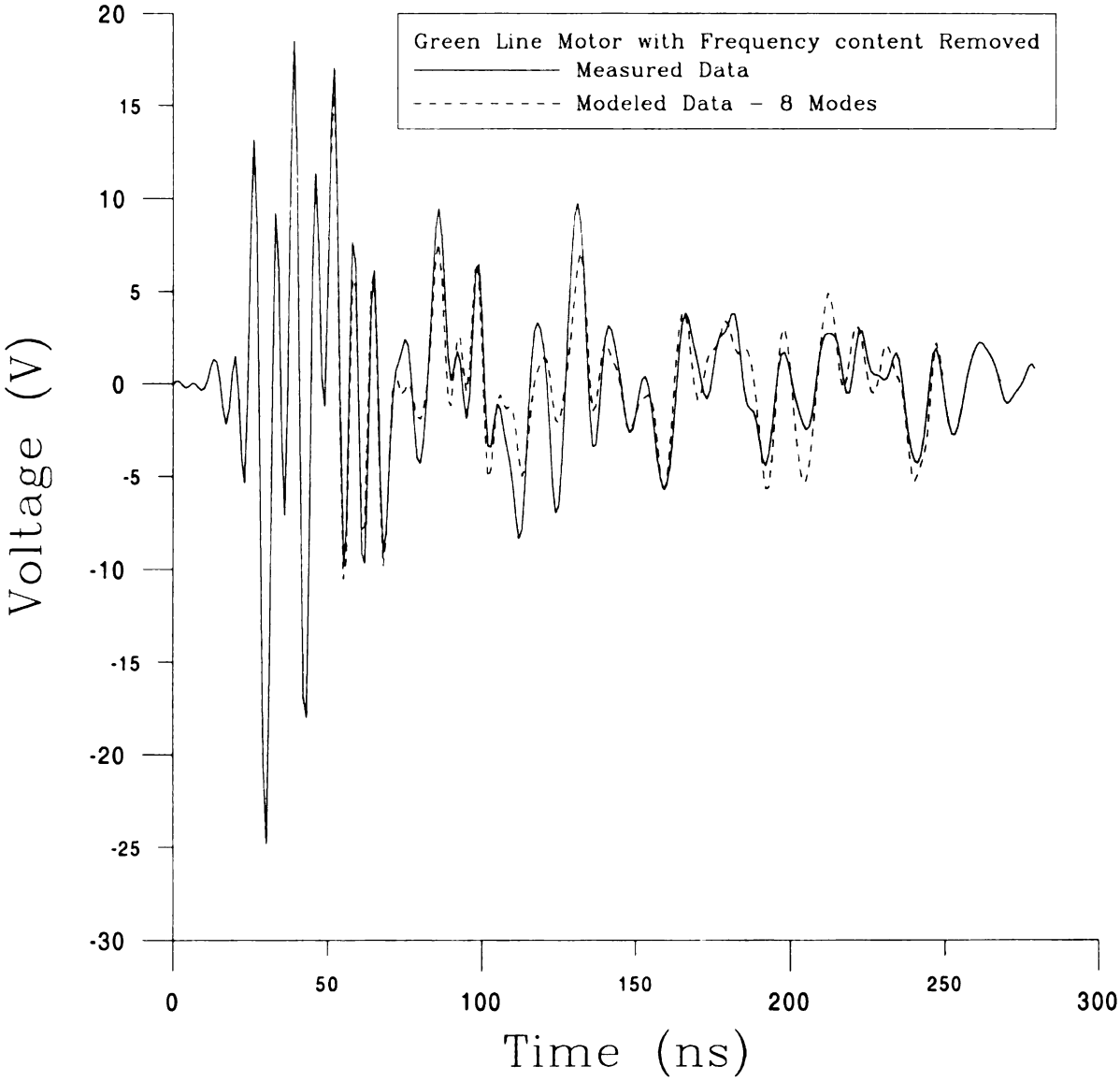

Figure 4-62: Modeled Green Line Motor Data with Frequency Truncated at 200 MHz Assuming 8 Natural Modes

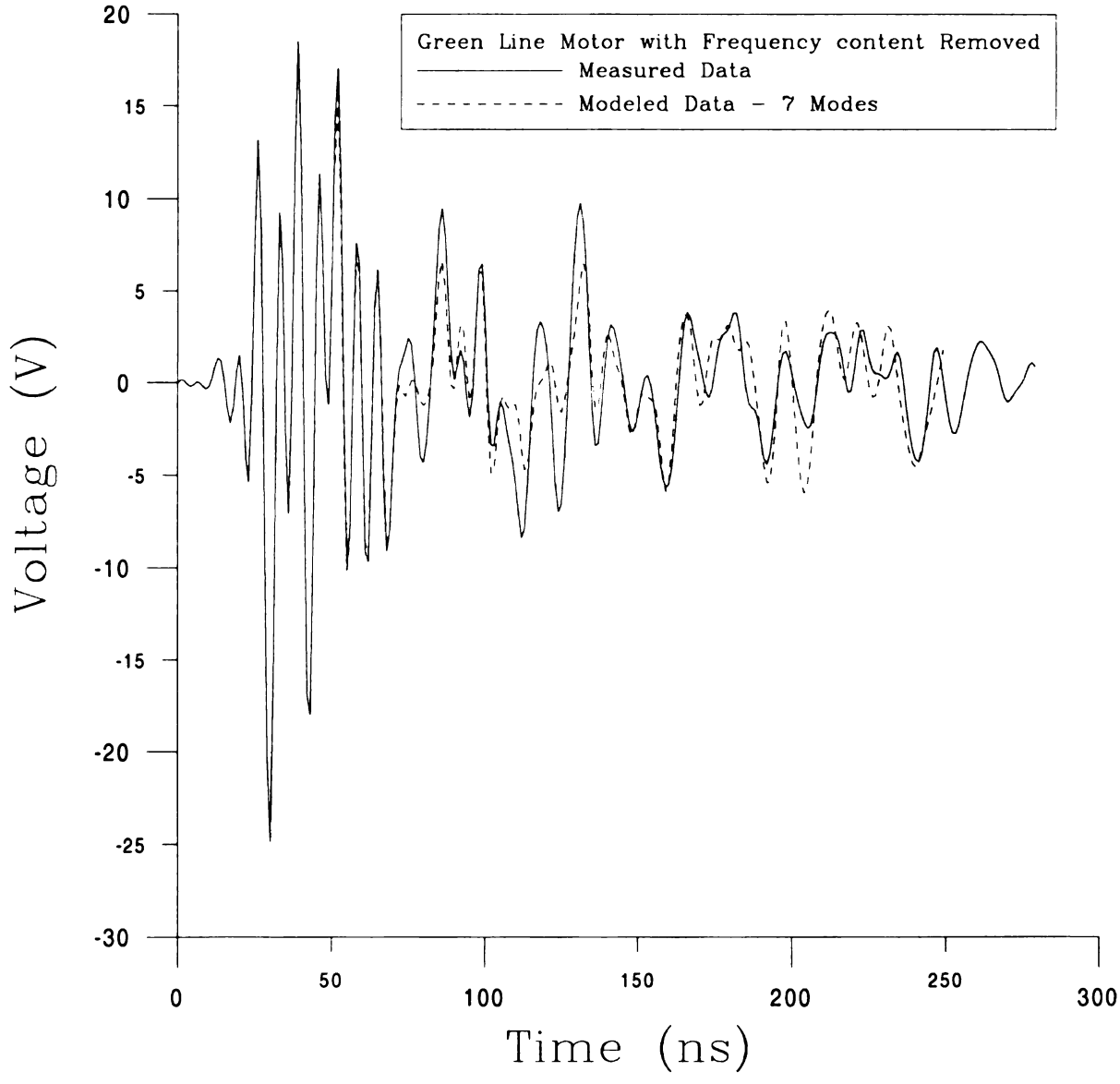

Figure 4-63: Modeled Green Line Motor Data with Frequency Truncated at 200 MHz Assuming 7 Natural Modes

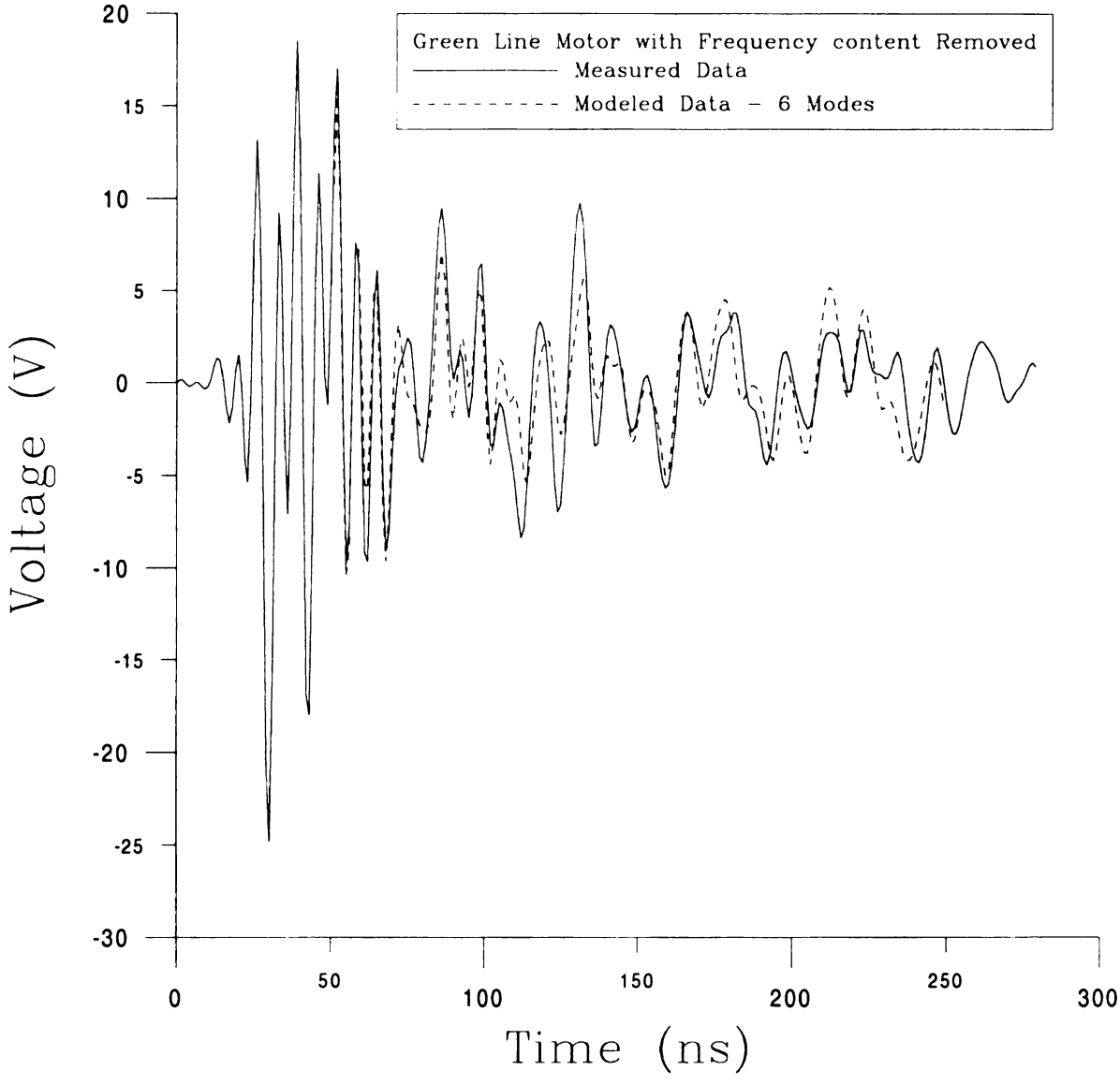

Figure 4-64: Modeled Green Line Motor Data with Frequency Truncated at 200 MHz Assuming 6 Natural Modes

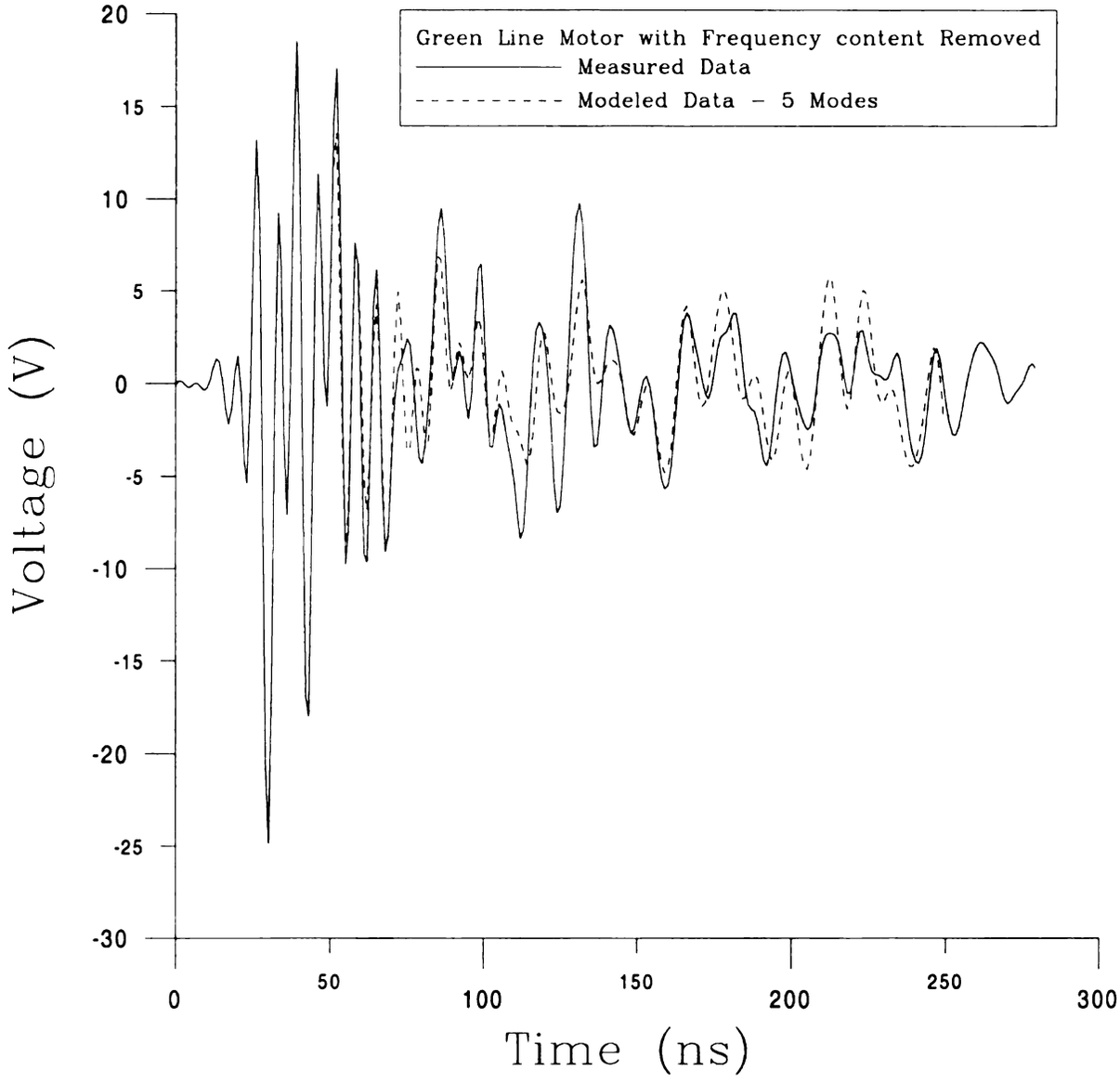

Figure 4-65: Modeled Green Line Motor Data with Frequency Truncated at 200 MHz Assuming 5 Natural Modes

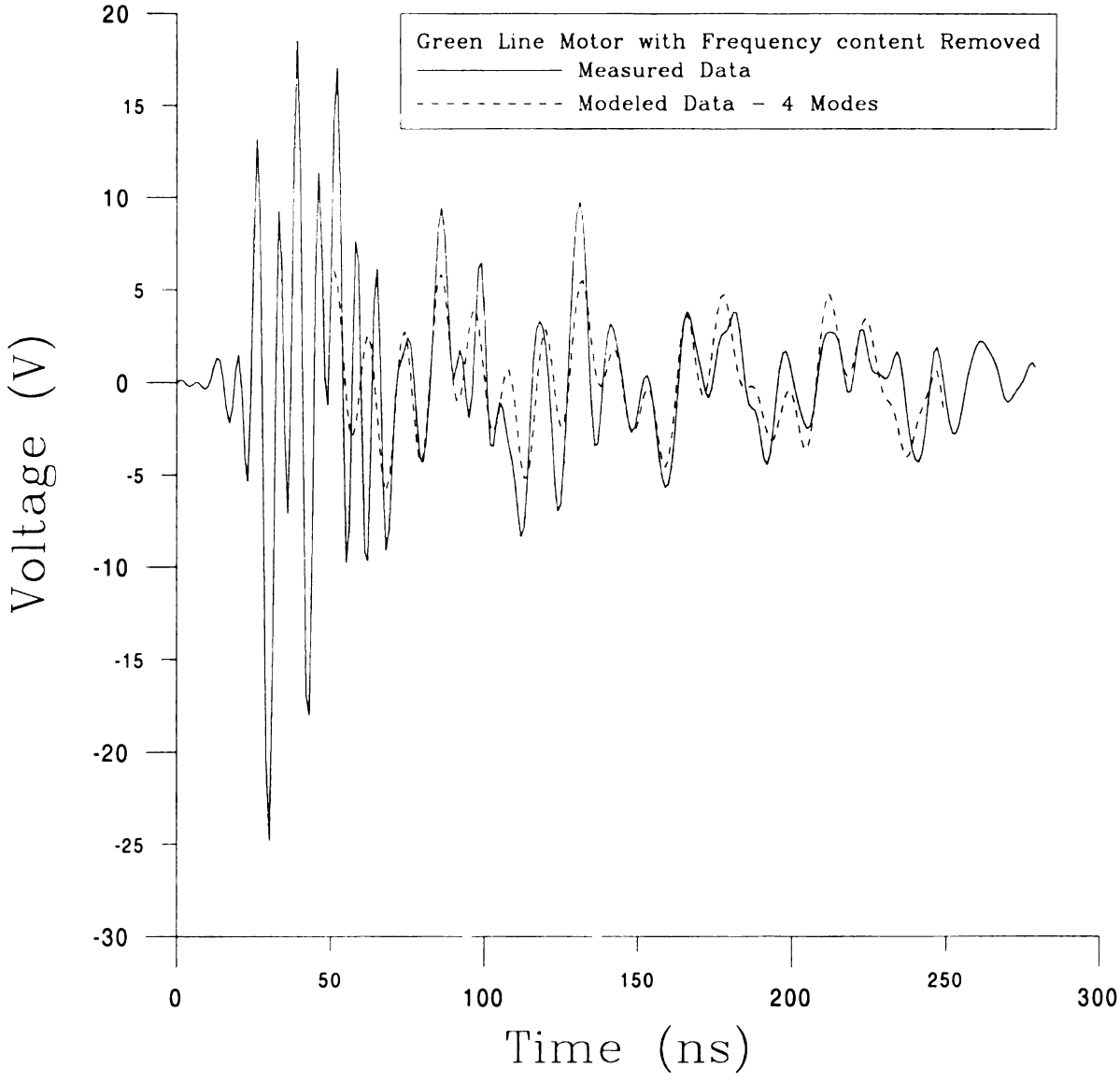

Figure 4-66: Modeled Green Line Motor Data with Frequency Truncated at 200 MHz Assuming 4 Natural Modes

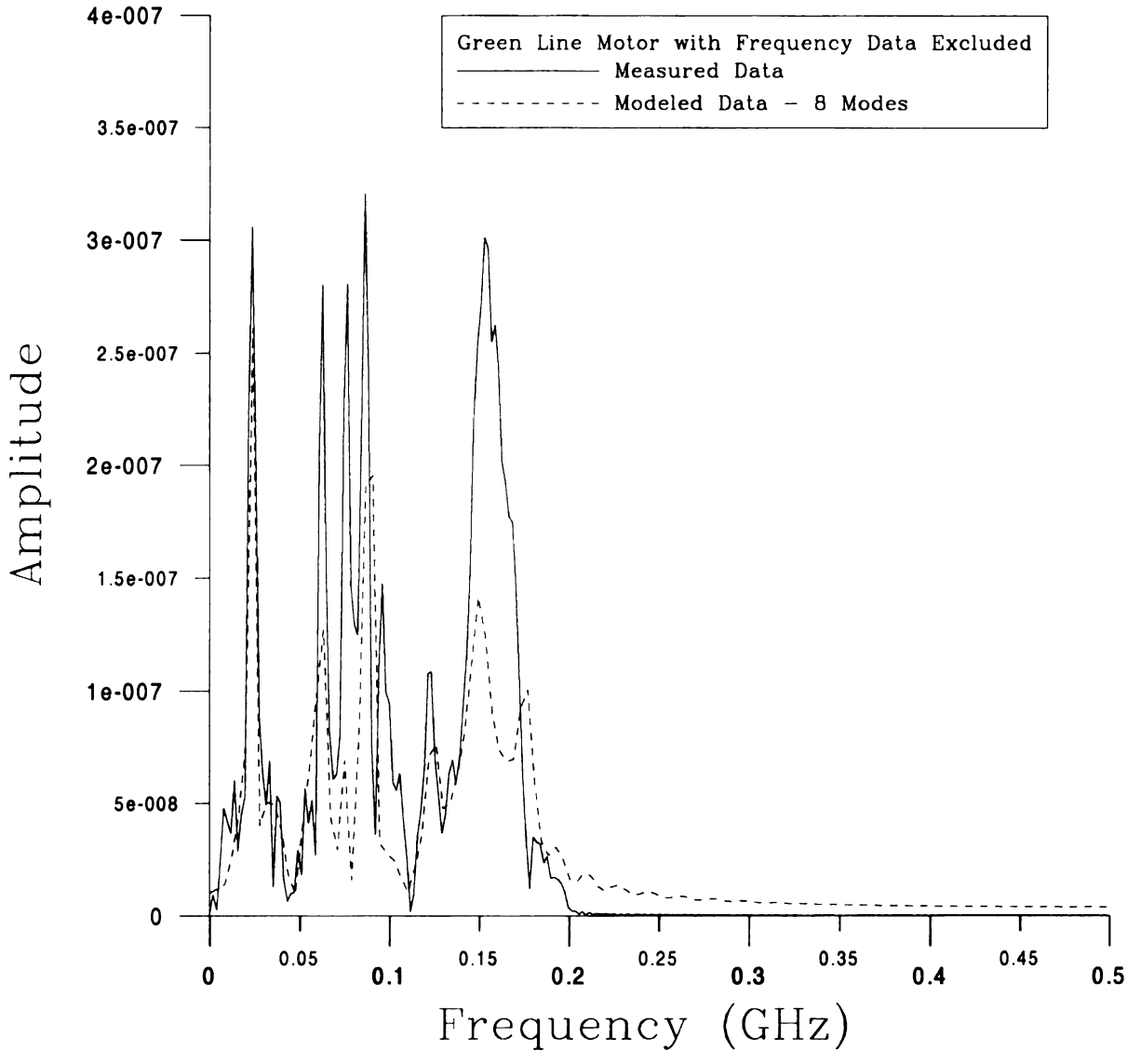

Figure 4-67: Frequency Content of Modeled Green Line Motor Data with Frequency Truncated at 200 MHz Assuming 8 Natural Modes

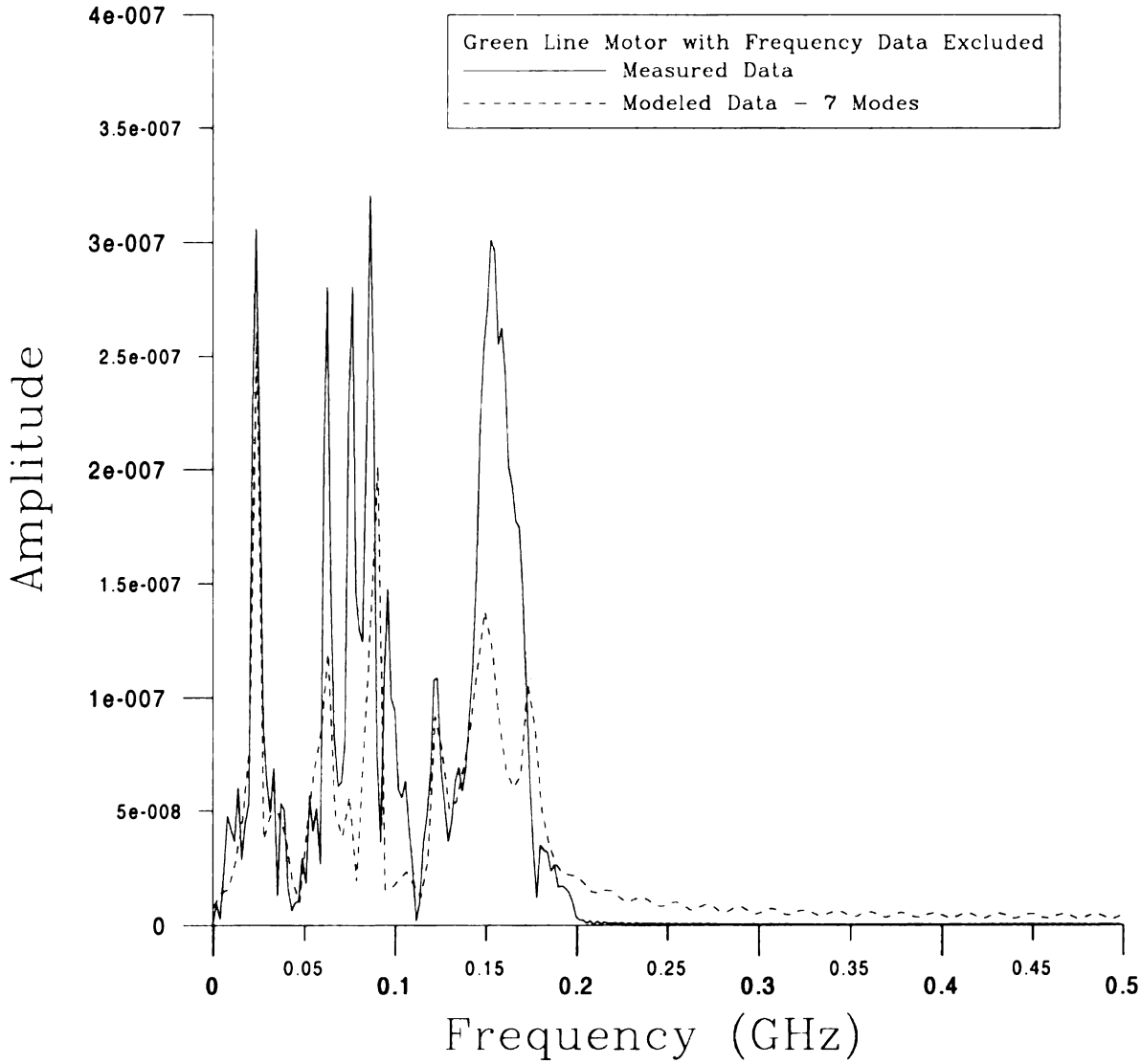

Figure 4-68: Frequency Content of Modeled Green Line Motor Data with Frequency Truncated at 200 MHz Assuming 7 Natural Modes

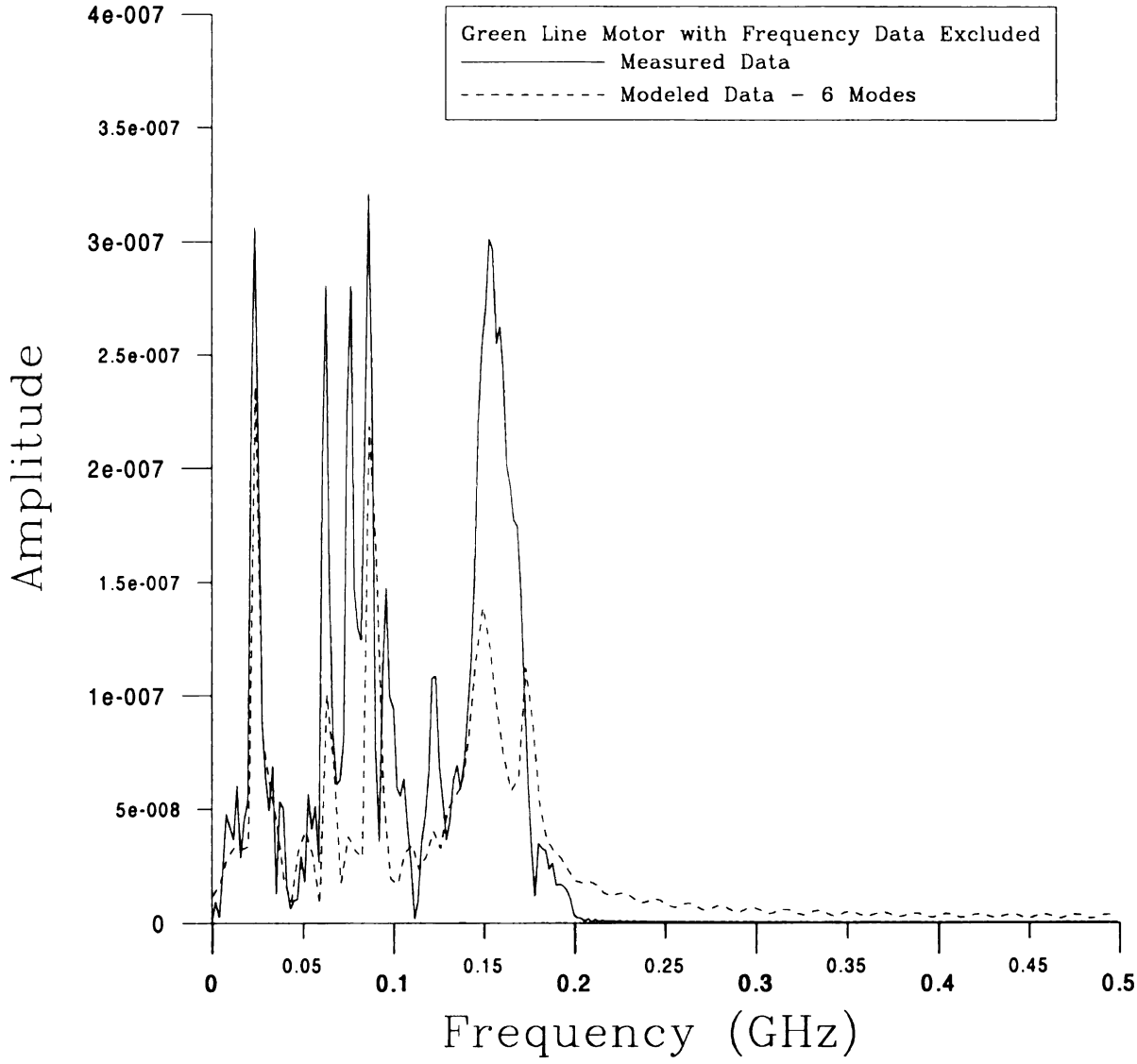

Figure 4-69: Frequency Content of Modeled Green Line Motor Data with Frequency Truncated at 200 MHz Assuming 6 Natural Modes

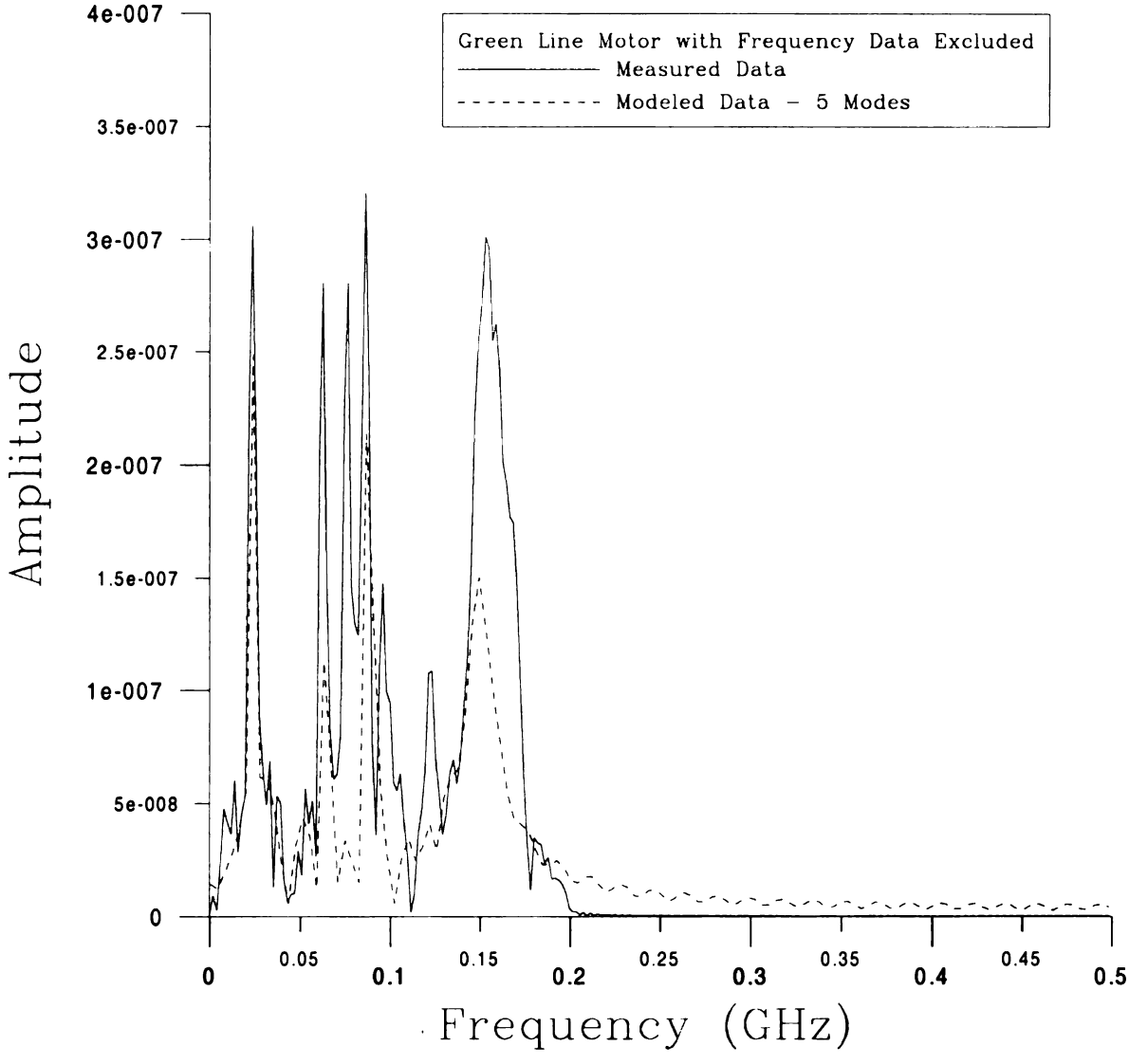

Figure 4-70: Frequency Content of Modeled Green Line Motor Data with Frequency Truncated at 200 MHz Assuming 5 Natural Modes

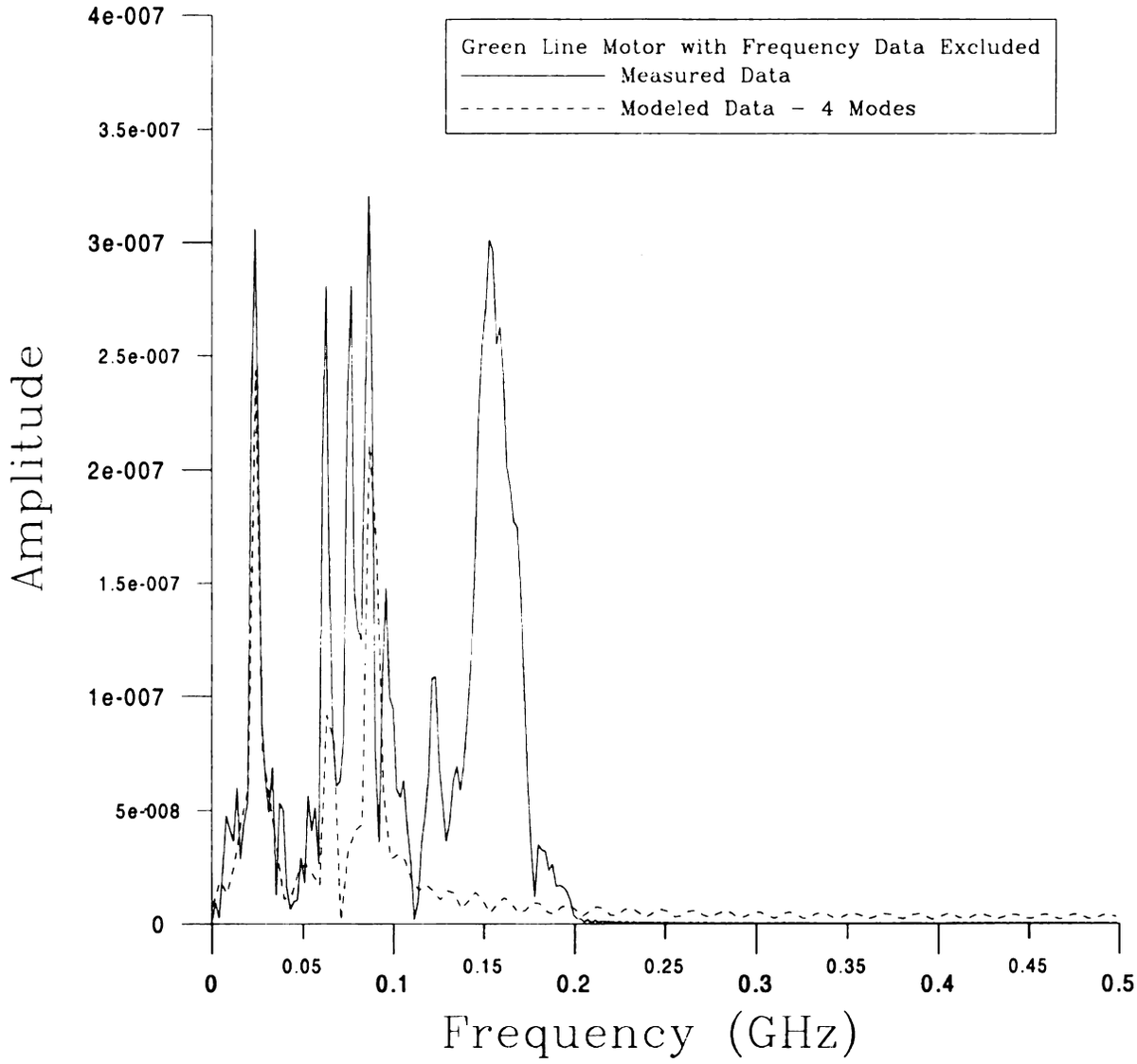

Figure 4-71: Frequency Content of Modeled Green Line Motor Data with Frequency Truncated at 200 MHz Assuming 4 Natural Modes

CHAPTER 5

ELECTROMAGNETIC COMPATIBLITY IMPLICATIONS OF MOTOR SWITCHING TRANSIENTS

5.1 Overview

Up until this chapter this thesis has dealt with the topic of conducted emissions. However, in the field of electromagnetic compatibility radiated emissions are also very important. This chapter discusses radiated emissions and why they are undesirable. A model for radiated emissions is developed and applied to the four motors studied in this thesis. Results from these models are compared to the CISPR 25 radiated emission limits. Conducted emission currents are also computed for each motor model and discussed.

5.2 Radiated Emissions

Radiated emissions are the electric and magnetic fields radiated by one device that may be received by other devices and cause interference in those devices. Although radiated emissions include both electric and magnetic fields, CISPR and other regulatory agencies only require that electric fields be measured for certification. The magnitudes of these fields are measured in dBμV/m and the frequency range for radiated emissions extends from 30 MHz to 40 GHz. CISPR has different sets of regulations for different types of devices. Devices that are marketed for use in commercial, industrial, or business environments are classified as Class A devices. Devices that are marketed for use in residential environments, notwithstanding their use in commercial, industrial, or business

environments are classified as Class B devices. In general the regulations for class B devices are more stringent than those for Class A devices.

Although automobile companies police themselves and often set their standards for radiated emission limits to much more stringent levels than those dictated by CISPR, it is not known exactly what radiated emission limits each automobile manufacturer sets for itself. Thus, for comparison purposes in this thesis, the Class B CISPR limits will be used as a baseline.

5.3 Radiated Emissions Modeling

A wire harness connected to the DC motor can be modeled as a pair of parallel wires of identical length l and separation s. The currents on both wires are directed to the right and are denoted as I_1 and I_2. As seen in Figure 5-1, these currents can be decomposed into common and differential-mode currents

$$\begin{aligned} I_1 &= I_C + I_D \\ I_2 &= I_C - I_D. \end{aligned} \tag{5.1}$$

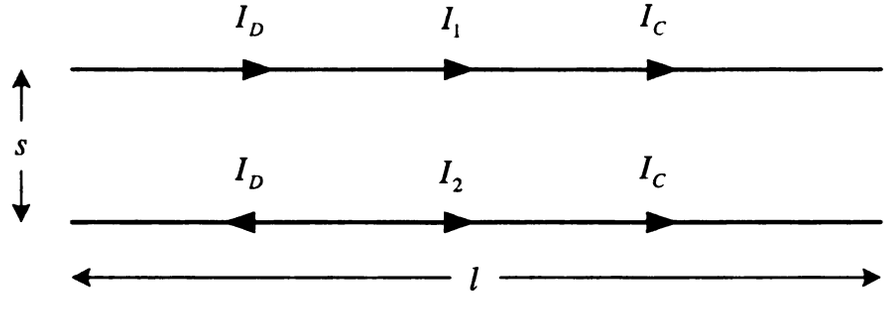

Figure 5-1: Illustration of Common and Differential Mode Currents on the Two-Wire Model

At any given cross section along the harness the differential-mode currents are equal in magnitude but opposite in direction. These differential-mode currents are the ideal currents expected to flow in the harness. However, currents in the harness are not always perfectly balanced in practice. Common-mode currents are the non-ideal, unbalanced currents found in the harness. The common-mode currents are undesirable because they are a source for radiated emissions [1]. For the purposes of this model, a worst-case scenario is assumed where all of the conducted emission currents exiting the motor are common-mode currents. Thus,

$$I_1 = I_2 = I_C \tag{5.2}$$

The total electric field of the harness is determined by superimposing the radiated fields of both conductors in the harness model. Thus, the total radiated field, E_θ, is the sum of the fields generated from the currents I_1 and I_2,

$$E_\theta = E_{\theta,1} + E_{\theta,2} \tag{5.3}$$

In order to determine these electric fields both wires are approximated as dipole antennas. Current in the dipoles are maximized at the source (assuming $l \leq \frac{\lambda}{2}$), which is situated at the center of the dipole, and zero at the ends of the dipole. Thus, the current $I(z)$ is proportional to $\sin(\beta_0 z)$ such that

$$I(z) = \begin{cases} I_C \sin\left[\beta_0\left(\frac{l}{2} - z\right)\right], & 0 > z > \frac{l}{2} \\ I_C \sin\left[\beta_0\left(\frac{l}{2} + z\right)\right], & -\frac{l}{2} > z > 0 \end{cases} \tag{5.4}$$

The field of the total dipole is computed as the superposition of many small Hertzian dipoles of length dz, having a constant current equal to $I(z)$ at that point along the dipole. This is illustrated in Figure 5-2

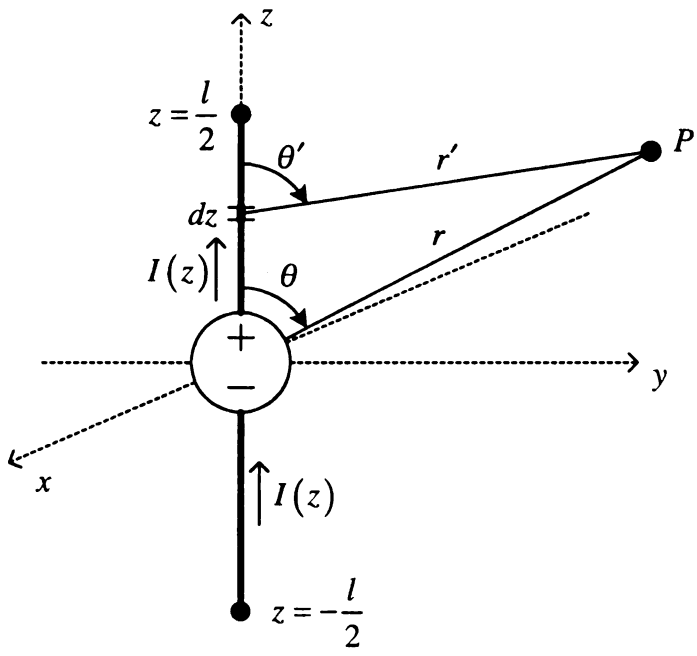

Figure 5-2: Dipole Antenna Model of a Single Conductor in the Two-Wire Model

where P is the observation point where $E_{\theta,1}$ or $E_{\theta,2}$ is computed.

The field at point P due to each tiny dipole segment is

$$dE_{\theta,i} = j\eta_0\beta_0 \frac{I(z)\sin\theta'}{4\pi r_i'} e^{-j\beta_0 r_i'} dz \tag{5.5}$$

However, if P is in the far field, the system can be approximated as seen in Figure 5-3.

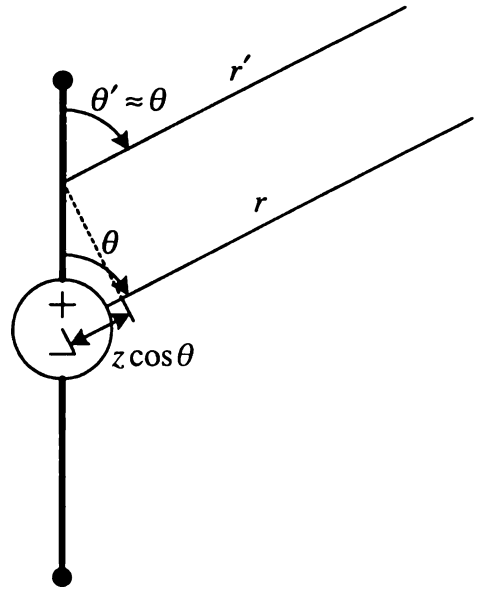

Figure 5-3: Far Zone Approximations for the Dipole Antenna Model of a Single Conductor in the Two-Wire Model

Thus, for a far-field system, $\theta \cong \theta'$ and $r_i \cong r_i'$. This approximation for r_i', however, cannot be used in the phase term because it relies on the electrical distance r_i'/λ_0 rather than the physical distance r_i'. A more reliable approximation for r_i' in the phase term is $r' \cong r - z\cos\theta$. Making these substitutions into Equation (5.5) yields

$$dE_{\theta,i} = j\eta_0\beta_0 \frac{I(z)\sin\theta}{4\pi r_i} e^{-j\beta_0(r_i - z\cos\theta)} dz \tag{5.6}$$

The total radiated electric field due to a single wire is the sum of these tiny dipole contributions

$$E_{\theta,i} = \int_{-\frac{l}{2}}^{\frac{l}{2}} j\eta_0\beta_0 \frac{I(z)\sin\theta}{4\pi r_i} e^{-j\beta_0 r_i} e^{j\beta_0 z\cos\theta} dz \tag{5.7}$$

Substituting Equation (5.4) into Equation (5.7) yields

$$E_{\theta,i} = \int_{-\frac{l}{2}}^{0} j\eta_0\beta_0 I_C \sin\theta \frac{\sin\left[\beta_0\left(\frac{l}{2}+z\right)\right]}{4\pi r_i} e^{-j\beta_0 r_i} e^{j\beta_0 z\cos\theta} dz$$

$$+\int_{0}^{\frac{l}{2}} j\eta_0\beta_0 \sin\theta \frac{I_C \sin\left[\beta_0\left(\frac{l}{2}-z\right)\right]}{4\pi r_i} e^{-j\beta_0 r_i} e^{j\beta_0 z\cos\theta} dz \qquad (5.8)$$

$$= \frac{j\eta_0\beta_0\sin\theta}{4\pi r_i} I_C e^{-j\beta_0 r_i} \left\{ \int_{-\frac{l}{2}}^{0} \sin\left[\beta_0\left(\frac{l}{2}+z\right)\right] e^{j\beta_0 z\cos\theta} dz + \int_{0}^{\frac{l}{2}} \sin\left[\beta_0\left(\frac{l}{2}-z\right)\right] e^{j\beta_0 z\cos\theta} dz \right\}$$

Allowing $\theta = 90^\circ$ to give fields in the broadside of the line and $\phi = 90^\circ$ to give fields in the planes of the wire will maximize the modeled radiated electric field (assuming $l \le \frac{\lambda}{2}$). Equation (5.8) is reduced to

$$E_{\theta,i} = \frac{j\eta_0\beta_0}{4\pi r_i} I_C e^{-j\beta_0 r_i} \left\{ \int_{-\frac{l}{2}}^{0} \sin\left[\beta_0\left(\frac{l}{2}+z\right)\right] dz + \int_{0}^{\frac{l}{2}} \sin\left[\beta_0\left(\frac{l}{2}-z\right)\right] dz \right\} \qquad (5.9)$$

This equation may be simplified to

$$E_{\theta,i} = \frac{j\eta_0\beta_0}{4\pi r_i} I_C e^{-j\beta_0 r_i} \left\{ \frac{2}{\beta_0}\left[1-\cos\left(\beta_0\frac{l}{2}\right)\right] \right\} \qquad (5.10)$$

Thus,

$$E_{\theta,1} = \frac{j\eta_0}{2\pi} I_C \frac{e^{-j\beta_0 r_1}}{r_1}\left[1-\cos\left(\beta_0\frac{l}{2}\right)\right]$$
$$E_{\theta,2} = \frac{j\eta_0}{2\pi} I_C \frac{e^{-j\beta_0 r_2}}{r_2}\left[1-\cos\left(\beta_0\frac{l}{2}\right)\right] \qquad (5.11)$$

where r_1 and r_2 are shown in Figure 5-4.

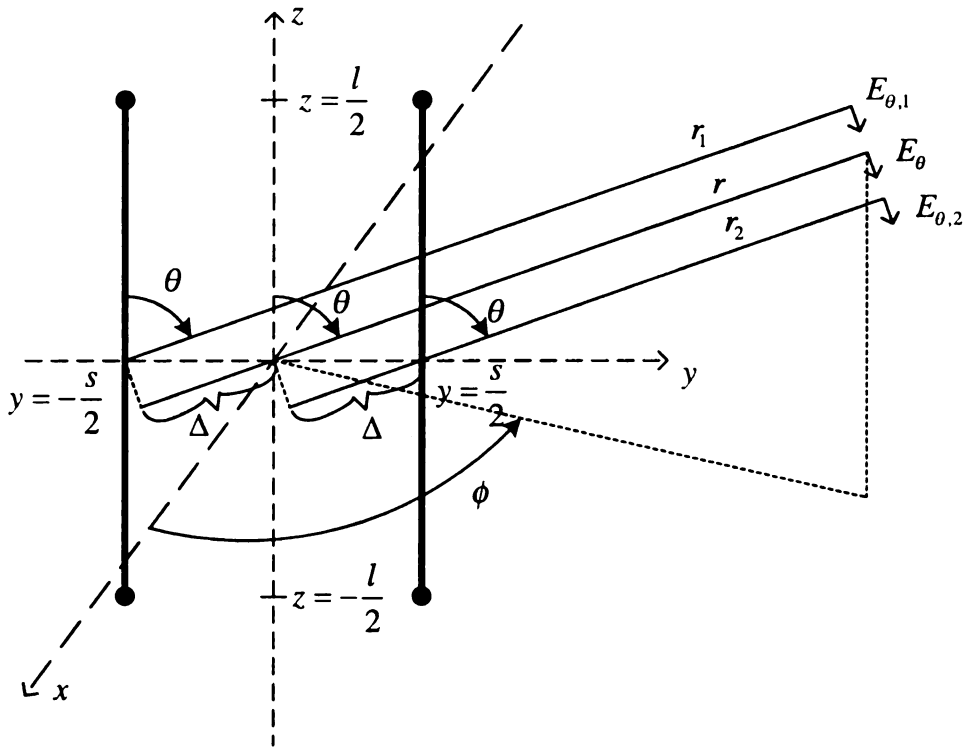

Figure 5-4: Geometry for Total Electric Field Using the Two-Wire Model

The distances r_1 and r_2 can be written in terms of r as follows

$$\begin{aligned} r_1 &= r+\Delta \\ r_2 &= r-\Delta \end{aligned} \tag{5.12}$$

where $\Delta = \frac{1}{2} s \sin\theta \sin\phi$. However, since $\theta = 90^\circ$ and $\phi = 90^\circ$ this reduces to

$\Delta = \frac{1}{2} s \sin\left(90^\circ\right)\sin\left(90^\circ\right) = \frac{s}{2}$. Thus

$$\begin{aligned} E_\theta &= E_{\theta,1} + E_{\theta,2} \\ &= \frac{j\eta_0}{2\pi} I_C \left(\frac{e^{-j\beta_0 r_1}}{r_1} + \frac{e^{-j\beta_0 r_2}}{r_2} \right)\left[1-\cos\left(\beta_0 \frac{l}{2}\right)\right] \\ &= \frac{j\eta_0}{2\pi} I_C \sin\theta \left(\frac{e^{-j\beta_0 (r+\Delta)}}{r+\frac{s}{2}} + \frac{e^{-j\beta_0 (r-\Delta)}}{r-\frac{s}{2}} \right)\left[1-\cos\left(\beta_0 \frac{l}{2}\right)\right] \end{aligned} \tag{5.13}$$

Again, since the observation point is in the far field, the physical distance, $\frac{s}{2}$ is negligible, allowing it to be removed from the denominators. The phase terms cannot be approximated in this way, however, as they depend on electrical distances as explained earlier. Thus,

$$E_\theta = \frac{j\eta_0}{2\pi r} I_C e^{-j\beta_0 r}\left(e^{-j\beta_0\frac{s}{2}} + e^{j\beta_0\frac{s}{2}}\right)\left[1-\cos\left(\beta_0\frac{l}{2}\right)\right] \tag{5.14}$$

Furthermore, $\eta_0 \cong 120\pi$, and $\left(e^{-j\beta_0\frac{s}{2}} + e^{j\beta_0\frac{s}{2}}\right) = 2\cos\left(\frac{1}{2}\beta_0 s\right)$, which leads to

$$E_\theta = j\frac{120 I_C}{r} e^{-j\beta_0 r}\cos\left(\frac{1}{2}\beta_0 s\right)\left[1-\cos\left(\beta_0\frac{l}{2}\right)\right]. \tag{5.15}$$

But the wire spacing s is electrically small, so $\cos\left(\frac{\pi s}{\lambda_0}\right) \approx 1$. Also, $\beta_0 = \frac{2\pi f}{c}$, where c is the speed of light. Thus, the magnitude of the maximum radiated electric field due to common mode currents is

$$|E_\theta|_{\max} = 120\frac{|I_C|}{r}\left[1-\cos\left(\beta_0\frac{l}{2}\right)\right] \tag{5.16}$$

The current I_C should be the worst-case current on the wire harness. Since current is voltage divided by impedance, these worst-case currents correspond to the maximum modeled transient voltages. Since

$$v(t) = r(t) = \sum_{n=1}^{N} A_n e^{\sigma_n t}\cos(\omega_n t + \phi_n) \tag{5.17}$$

and the maximum value of the cosine term is 1, the maximum transient voltage of each term in the series is

$$\left|v(t)\right|_{\max,n} = A_n e^{\sigma_n t}. \tag{5.18}$$

The exponential term decays since σ_n is negative. Thus, the maximum value of equation (5.18) occurs at the beginning of the late time period, $t = T_L$.

$$v_{\max,n} = A_n e^{\sigma_n T_L} \tag{5.19}$$

The maximum voltages occur at the natural frequencies,

$$v_{\max,n} = v(\omega_n) = A_n e^{\sigma_n T_L}. \tag{5.20}$$

Thus,

$$I_{C,n} = I_{\max,n} = I(\omega_n) = \frac{v(\omega_n)}{z(\omega_n)} = \frac{A_n e^{\sigma_n T_L}}{z(\omega_n)}. \tag{5.21}$$

The frequency-domain impedance, $z(\omega)$, was measured earlier and is presented in Chapter 2. The values of $z(\omega_n)$ are obtained through linear interpolation of the value of $z(\omega)$ at the imaginary part of the natural frequencies.

5.4 Radiated and Conducted Emission Model Results

The author wrote a program called *VOME.FOR* that reads in the modeled time-varying voltage data and approximates the maximum voltages due to each natural mode. The program also reads the measured impedance as a function of frequency and computes maximum conducted emission currents due to each natural frequency. The input data for this program corresponds to Figures 4-2, 4-16, 4-32, and 4-52. Predicted natural frequencies that produce conducted emission currents less than 1 mA are considered negligible and are not included. This data is presented in Table 5-1.

Motor Type	Natural Frequency	Voltage at Natural Frequency	Motor Impedance at Natural Frequency	Modeled Conducted Emission Current
Actuator Motor	23.72 MHz	3.75 V	35.05 Ω	110.13 mA
	63.25 MHz	2.54 V	97.58 Ω	26.01 mA
	94.50 MHz	7.06 V	220.34 Ω	32.03 mA
	123.92 MHz	4.58 V	194.43 Ω	23.54 mA
	149.41 MHz	10.55 V	1029.79 Ω	10.24 mA
Blower Motor	29.56 MHz	6.16 V	40.08 Ω	153.79 mA
	41.11 MHz	0.19 V	53.89 Ω	3.54 mA
	73.80 MHz	3.87 V	118.85 Ω	32.54 mA
Blue Line Motor	10.72 MHz	1.01 V	419.47 Ω	2.42 mA
	23.16 MHz	3.98 V	94.80 Ω	42.02 mA
	54.51 MHz	0.54 V	59.23 Ω	9.21 mA
	62.27 MHz	3.89 V	42.03 Ω	92.55 mA
	75.26 MHz	3.21 V	72.55 Ω	44.22 mA
	86.42 MHz	2.77 V	112.93 Ω	24.55 mA
	93.83 MHz	2.59 V	126.79 Ω	20.44 mA
Green Line Motor	23.22 MHz	2.83 V	117.24 Ω	24.16 mA
	62.21 MHz	0.50 V	77.22 Ω	6.50 mA
	89.59 MHz	2.08 V	275.01 Ω	7.57 mA
	122.79 MHz	1.72 V	200.06 Ω	8.62 mA
	148.80 MHz	9.41 V	94.72 Ω	99.37 mA
	174.01 MHz	2.98 V	43.54 Ω	68.54 mA

Table 5-1: Modeled Conducted Emission Currents at the Predicted Natural Frequencies

The voltages and currents on this table reflect the maximum values due to each natural frequency. This worst-case current is used to approximate I_C in equation (5.16) and a worst-case radiated electric field is computed. This computation is carried out by the program *vome.for* for harness lengths ranging from 5 cm to 1m. These harness lengths were chosen since they would be representative harness lengths found in an automobile. The fields are computed at a distance of ten meters from the wire harness in each case in accordance with CISPR Class B regulations. The results are plotted in Figure 5-5.

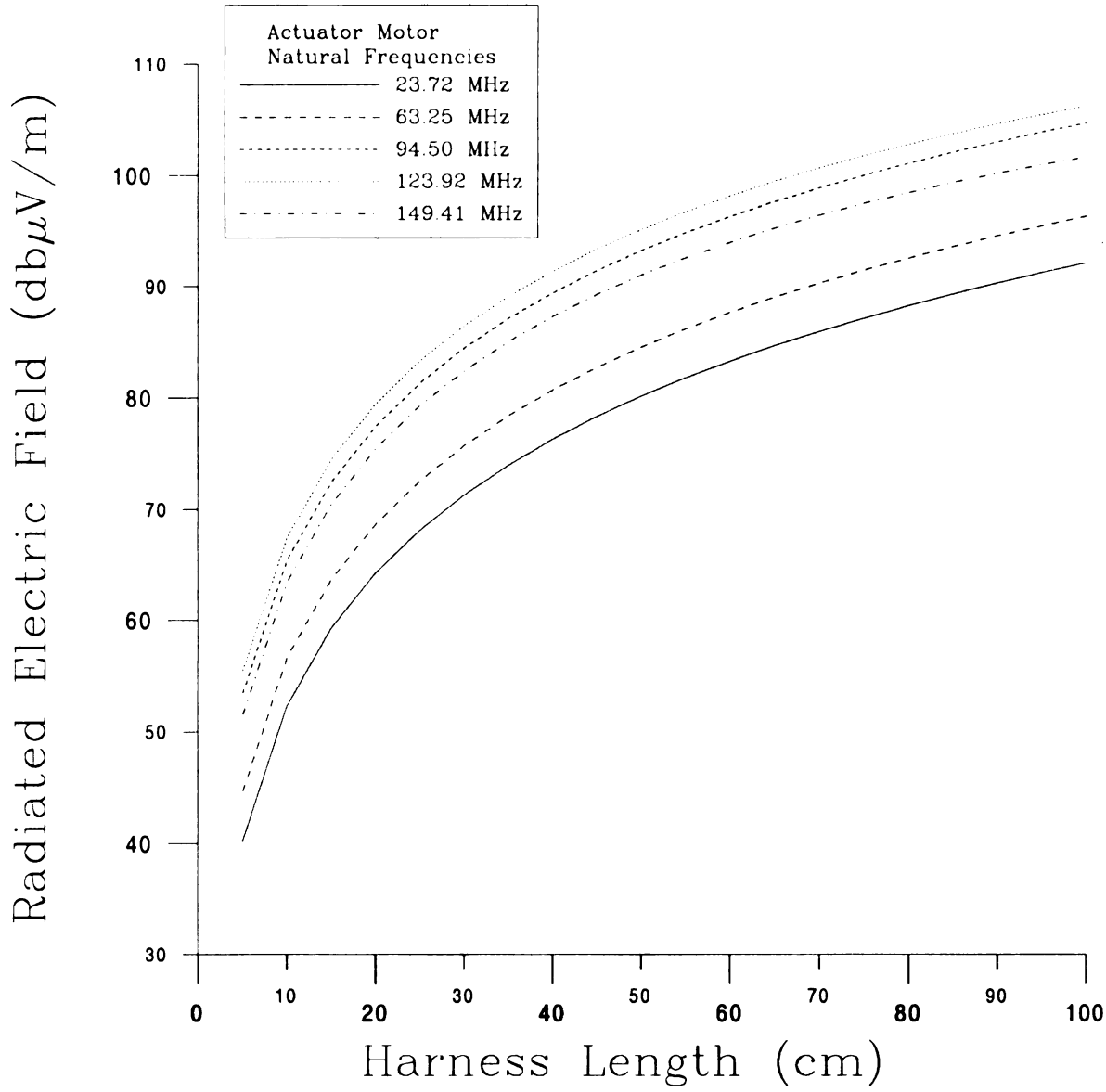

Figure 5-5: Modeled Radiated Emissions vs. Harness Length for the Actuator Motor

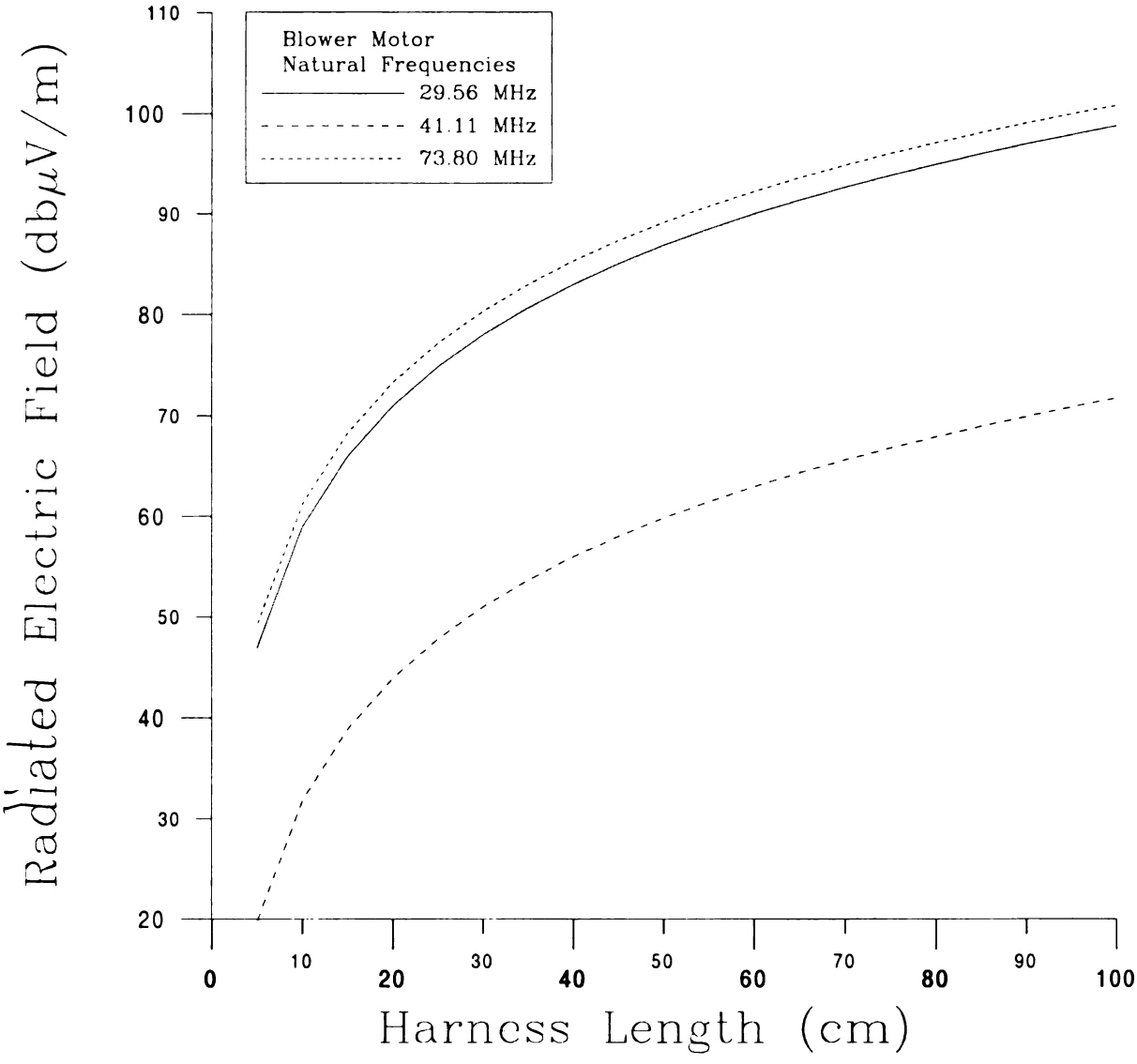

Figure 5-6: Modeled Radiated Emissions vs. Harness Length for the Blower Motor

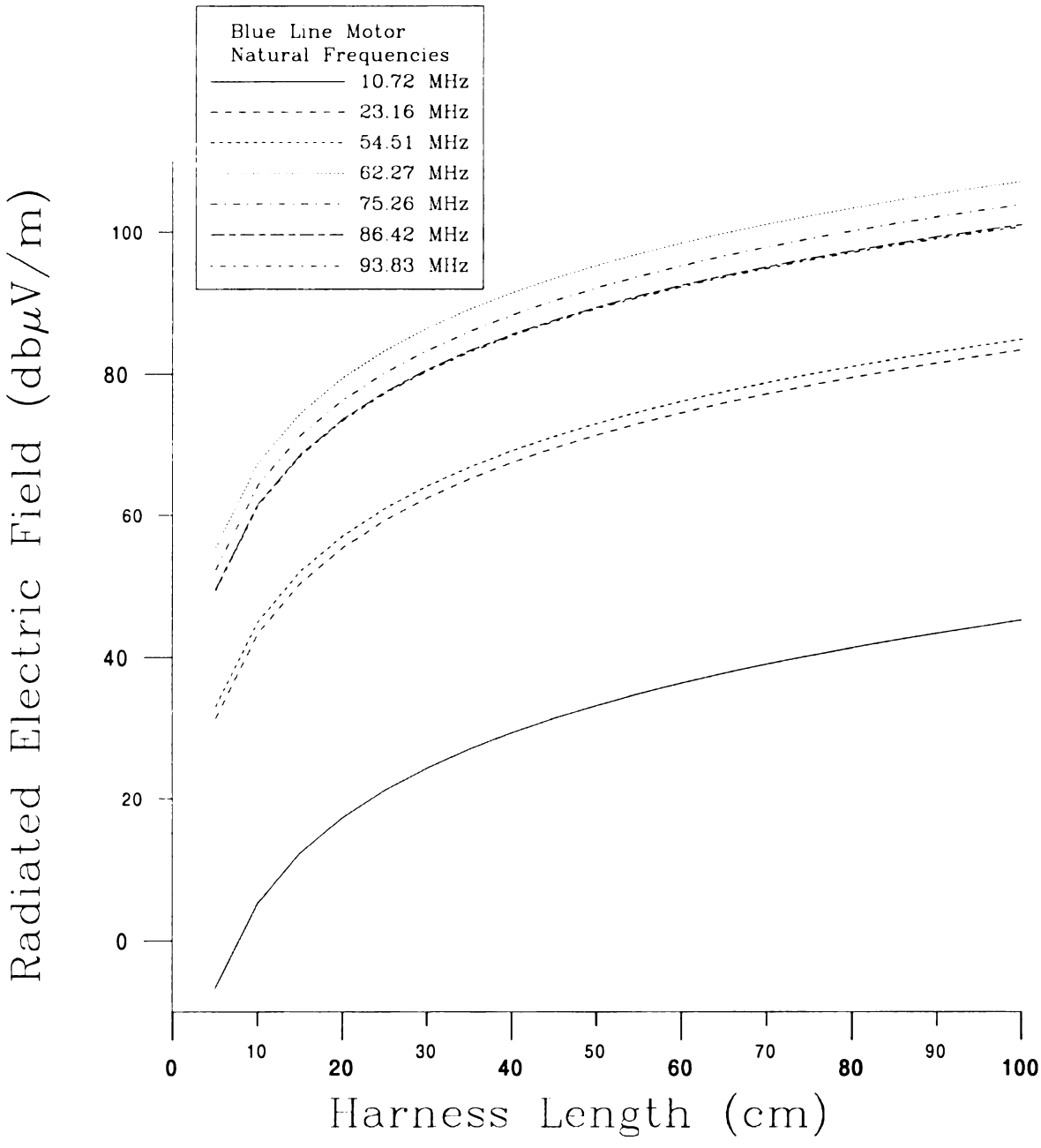

Figure 5-7: Modeled Radiated Emissions vs. Harness Length for the Blue Line Motor

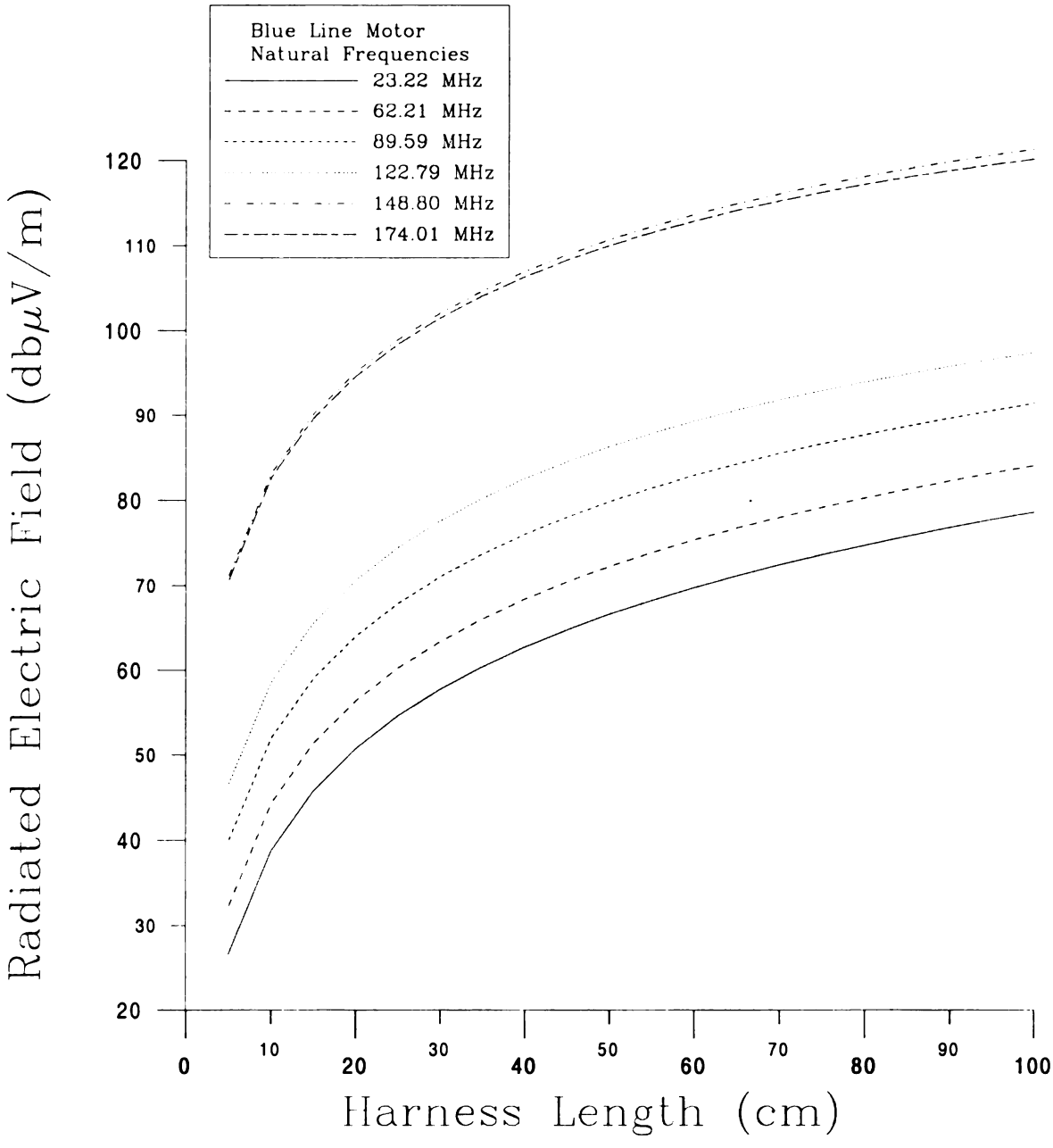

Figure 5-8: Modeled Radiated Emissions vs. Harness Length for the Green Line Motor

In each case the worst-case modeled fields are above the CISPR radiated emission limits of 30 dBμV/m for frequencies between 30-230 MHz. However, these modeled field strengths are worst-case scenarios, computed with currents that are likely much greater than the actual currents present on the wire harness.

CHAPTER 6

CONCLUSIONS

This thesis has presented a comparison between a numerical model and measured data for conducted emission transients. The goal of this thesis was to create a numerical model for conducted emission transients generated due to the switching of DC motors used in automobiles. As discussed in Chapter 4, the numerical model developed does adequately model the measured conducted emission transients. Furthermore, this model made it possible to predict the conducted emission currents and radiated emission electric fields due to the switching transients.

A discussion of conducted emissions as well as the theory behind the procedure for measuring conducted emission transients was presented in Chapter 2. This theory was put to use in Chapter 3 as the measurement procedures and data were presented. The numerical model for the measured conducted emission transients was derived in Chapter 4, utilizing the extinction pulse technique. The modeled data generated through this method was presented and compared to the measured conducted emission transients in Chapter 4 as well.

Finally, Chapter 5 used the numerical model to predict the practical electromagnetic compatibility issues of conducted emission currents and radiated emission electric fields. Conducted emission currents due to each modeled natural frequency were calculated. Also, a dipole model was applied to approximate a wire harness connected to each DC motor. Using this model, the expected radiated electric field was predicted for some representative harness lengths found in automobiles.

Although the goal of this research was achieved, further research could produce better results and possibly improve the utility of the conducted emission transient model. Investigation into a wider variety of motors could lead to the identification of expected natural frequency content for similar motor types. This identification of expected natural frequency spectrums for different motors could lead to a simple method of identifying expected conducted and radiated emissions due to the switching of those motors. Rather than needing to test each motor over a wide frequency range for both conducted and radiated emissions, this natural frequency information would allow the conducted and radiated emission characteristics to be modeled, necessitating only the measurement of the impedance spectrum of the motor.

The radiated emission model developed in Chapter 5 used worst-case currents to model the radiated electric field. These worst-case currents led to modeled radiated emissions that exceed the CISPR 25 limits. However, in practice, the worst-case currents are likely not present on the harness. The current contributions due to each natural frequency will be attenuated somewhat, although this attenuation was difficult to accurately approximate. Future research could attempt to develop a more accurate current model to be used for radiated emission modeling.

The Broadband Artificial Network is a useful tool for measuring conducted emission transients. However, in this thesis, the BAN was found to have a few shortcomings. For the motors tested in this thesis, the BAN did not have sufficiently large impedance at higher test frequencies to support the linear model assumed in the measurement process. Future research could take two approaches to improving the performance of the BAN. The first approach would be to alter the components of the

BAN such that the inductor and capacitor component values would force the BAN to have greater impedance at higher frequencies. This would ensure the impedance of the BAN will be large with respect to the impedance of the motors at higher frequencies and the assumed linear model will remain valid at those frequencies. The second approach to improving the performance of the BAN would require that the assumed linear model be discarded and treat the measurement circuit as a voltage divider. With a known impedance of the motor under test and the BAN over the entire test frequency range a voltage divider would be able to accurately measure the transient voltage.

The area of electromagnetic compatibility is an interesting and fast-growing field of electrical engineering. EMC is becoming prominent in both industry and academia as technology continues to push the limits of high frequency devices, necessitating new testing standards and regulations. This thesis illustrates the need for more research into numerical simulations and measurement techniques in the field of electromagnetic compatibility.

APPENDIX

APPENDIX

FORTRAN PROGRAMS

A.1 Overview

The modeling done in this thesis could not have been accomplished without the aid of computers. The computational speed and efficiency provide by computers is invaluable. This appendix briefly discusses the functions of the two Fortran programs used in this thesis. The source code for these two programs is also included.

A.2 Extinction Pulse Program

The program *ep.for* generates an extinction pulse based on user input. This program was originally used for target identification, but the extinction pulse method can be adopted to model conducted emission transients. The program requires an input file with a measured conducted emission transient data set. The user identifies the time period over which an extinction pulse should be generated as well as an expected number of natural modes. The program then generates an extinction pulse based on the input file and user information. From this extinction pulse, the amplitude and phase of the measured data is approximated, as well as the complex natural frequencies of that data. A modeled waveform is reconstructed using this information.

A.3 Source Code for *ep.for*

The following is the source code for *ep.for.*

```
c
c ****************************************
c Double precision version of 27 March 1992
c  Modified by Ed Rothwell
c ****************************************
c
c
c
c
c-----------------------------------------------------------------------
c
c  EP - This program creates a forced E-pulse. The complex natural
c       frequencies contained in the data are obtained as a by
c       product of making the E-pulse.
c
c  EP requires the following files during execution.
c
c  1.  EP.INP - contains information on which data files to use,
c      how many modes to look for, etc. The format for this file
c      is described below.
c
c  2.  Response Files - contain the measured or synthetic transient
c      data in NDF format.
c
c  The EP.INP file has the following format.
c
c  Line #          Datatype              Explanation
c
c     1            L,L,L,L      Logical switches for program control
c                               dpsw, szsw, dcsw, irsw
c
c           dpsw = .true. >>>> discard positive sigma terms
c           szsw = .true. >>>> set positive sigma mode amplitudes to
c                              zero
c           dcsw = .true. >>>> generate E-pulse to kill DC
c           irsw = .true. >>>> print intermediate results
c
c     2             A30          filename for results summary
c     3             A30          filename for E-pulse output
c     4             A30          filename for best fit data
c     5             A30          filename for complex natural frequencies
c     6             I            # data sets used k
c     7             A30          filename of data set #1
c     8             R,R          start and end of late-time of data set #1
c     9             A30          filename of data set #2
c     10            R,R          start and end of late-time of data set #2
c     |             |            |
c     |             |            |
c     |             |            |
c     6+2k-1        A30          filename of data set #k
c     6+2k          R,R          start and end of late-time of data set #k
c     7+2k          I            # of modes to extract, # of match points
c                                in late-time
c     8+2k          I            # points to skip
c     9+k         R,R,R          lower limit, upper limit, increment
c                                in search for optimum E-pulse duration
c     10+k          I            # significant digits squared error per
c                                point
c
```

```
c-------------------------------------------------------------------------
c
c  Sample EP.INP
c
c     The natural mode frequencies of Boieng 747 (B747) are extracted
c     using three different aspect angle measurements.
c
c    .false., .false., .true., .false.    Program control switches
c    B747.res                             Filename for results summary
c    B747.epu                             Filename for E-pulse output
c    B747.dat                             Filename for best fit data
c    B747.frq                             Filename for natural frq
c    3                                    # of data sets
c    B7471.ndf                            B747 data at aspect #1
c    7.0,20.0                             start, end of late-time #1
c    B7472.ndf                            B747 data at aspect #2
c    6.5,20.0                             start, end of late-time #2
c    B7473.ndf                            B747 data at aspect #3
c    6.0, 20.0                            start, end of late time #3
c    5, 40                                # modes, # match points
c    1                                    skipping every other point
c    2.0, 7.0, 0.1                        lower, upper, increment
c    3                                    # significant digits
c
c-------------------------------------------------------------------------
c
c
c Overview of program operation:
c
c  1. EP reads the EP.INP file
c
c  Note #1 :  The late-time period of each data set does not have
c             to start or end at the same time.  Each data set
c             must have the same sampling interval.
c
c  Note #2 :  To provide good estimates of the complex natural
c             frequencies it is best to use responses that have
c             large late-time energy. Often, it is not necessary
c             to use all measured responses in order to obtain good
c             estimation of natural frequencies.  Unused responses
c             can de used as an independent check of the quality of
c             the resulting E-pulses with the program DISCRIMS.FOR.
c
c  2. EP searches for the optimum E-pulse duration. The optimum duration
c     is the one that yields the minimum squared error per point between
c     the original data and a waveform constructed using the extracted
c     modal frequencies, amplitues, and phases.
c
c  3. The program computes the amplitudes for an E-pulse of each value
c     of duration in the search. Based on the pulse amplitudes of the E-
c     pulse the complex natural frequencies are obtained. The amplitudes
c     and phases that result in a best fit to the data are then
c     computed. The squarred error per point for that value is then
c     reported via SQRTOT.
c
c  4. After the optimum value of duration within the specified search
c     range is obtained the pulse amplitudes of the E-pulse is computed
c     and written to the specified disk file.
c
c  5. The natural frequencies are written to a specified disk file.
c
c  6. The amplitudes and phases that result in the best fit to each
c     data set is computed. The summary of mode frequencies, amplitudes
c     and phases is written to a specified disk file.
```

```
c
c  7. The squarred error per point for each value on duration is written
c     to the file TKDATA.DAT. This data is useful in estimating the
c     limits of the duration search and the number of evaluation points
c     in the search.
c
c  Note #4 : Natural mode frequencies with a positive damping
c            coefficient (sigma) may be encountered with measured data.
c            These values are obviously non-physical. All modes of
c            passive targets must have a negative sigma. The cause of
c            the non-physical values is usually due to excessive noise
c            present in the measurement or by imperfect modeling of the
c            transmitted pulse. It is also conceivable that choosing the
c            time window for mode extraction too small can also result
c            in poor estimates natural frequencies. Incorrect estimates
c            of the number of modes in the data could also cause the
c            same effect.
c
c            P. Ilavarasan has attempted make a version of EP.FOR that
c            would constrain the sigma to be negative. This problem
c            seems to be very difficult and has not yet been
c            successfully implemented.
c
c  Note #5 : The duration of the natural E-pulse width is given as
c
c                                          number of modes
c    Natural E-pulse duration  = ---------------------------------
c                                highest natural frequency excited
c
c    From this relationship, the duration of the forced E-pulse
c    can be estimated as follows.
c
c    0.25 natural duration < forced duration < 4 natural duration
c
c    EP searches for the optimum  forced E-pulse duration within the
c    specified range.
c
c  References:
c
c    1.  E. J. Rothwell and K. M. Chen, "A Hybrid E-pulse/
c        Least Squares Technique for Natural resonance Extraction,"
c        Proceedings of the IEEE, pp. 296-298, March 1988.
c
c    2.  C. E. Baum, E. J. Rothwell, K. M. Chen and D. P. Nyquist,
c        "The Singualrity Expansion Method and its Application to
c        Target Identification," to appear in Special Proceedings of
c        IEEE on Electromagnetics, October 1991.
c
c-----------------------------------------------------------------------
c
      program ep
c
      implicit real*8 (a-h,o-z)
c
      parameter (luna=10, luntk=11, mxpts=2048, mxdts=10,
     &           mxnms=20, mxnps=42, mxtkpts=1024)
c
      character*30 respnm(mxdts), filers, fileep, filefr
      logical dpsw, szsw, dcsw, irsw
c
      common /switch/ dpsw, szsw, dcsw, irsw
      common /files/ respnm, filers, fileep, filefr
      common /system/ nresp, tstart(mxdts), tend(mxdts), nskip
     &                , modes, nptslt
```

```
      common /sample/ npulse, dt1, dt2(mxdts), te(mxdts), nmodes
c
      write(*,*) '***********************************************'
      write(*,*) '**   Program : EP.FOR                        **'
      write(*,*) '**             Version 1.1                   **'
      write(*,*) '**                                           **'
      write(*,*) '**   Purpose : Natural Mode Extraction via   **'
      write(*,*) '**             Hybrid E-Pulse/Least Squares  **'
      write(*,*) '**             Technique.                    **'
      write(*,*) '**                                           **'
      write(*,*) '**   Update  : November 12, 1991             **'
      write(*,*) '**                                           **'
      write(*,*) '**   Authors : Ponniah Ilavarasan            **'
      write(*,*) '**             John E. Ross III              **'
      write(*,*) '**             Edward J. Rothwell            **'
      write(*,*) '**                                           **'
      write(*,*) '**   Department of Electrical Engineering    **'
      write(*,*) '**   Michigan State University                **'
      write(*,*) '**   East Lansing, MI 48824                  **'
      write(*,*) '**                                           **'
      write(*,*) '**   Copyright (c) 1991                      **'
      write(*,*) '***********************************************'
      write(*,*)
c
c  read ep.inp file to get start up information
c
      call rdepinp
      write (*,*) 'passed rdepinp'
c
c  read in specified response data sets and put in array h(,)
c
      call rdresp
      write (*,*) 'passed rdresp'
c
c  process response data sets, round off start & stop times, generate
c  arrays with slopes and time values
c
      call procres
      write (*,*) 'passed procres'
c
c  skip data points for best fit and put in array hbf(,)
c
      call skip
      write (*,*) 'passed skip'
c
c  calculate energy in late time of responses
c
      call energy
      write (*,*) 'passed energy'
c
c  open file for output of search for optimum E-pulse duration (tk)
c
      open(luntk,file='tkdata.dat')
c
c  Number of pulses used here is twice the # of modes plus one
c  because forcd E-pulse. In the case dc forced E-pulse, it is
c  plus one to the above number
c
      npulse = 2*modes+1
      if (dcsw) npulse = npulse + 1
c
c  Minimize the function sqrtot(tk) to find optimum tk and best E-pulse
c
      call minfunc
```

```
c
c  Write summary, best fit, and E-pulses to disk files
c
      call wrres
c
      close(luntk)
c
      end
c
c-----------------------------------------------------------------------
c
c  RDEPINP - Reads EP.INP file information.
c
      subroutine rdepinp
c
      parameter (luna=10, mxdts=10)
c
      implicit real*8 (a-h,o-z)
c
      character*30 respnm(mxdts), filers, fileep, filefr
      logical dpsw, szsw, dcsw, irsw
c
      common /switch/ dpsw, szsw, dcsw, irsw
      common /files/ respnm, filers, fileep, filefr
      common /system/ nresp, tstart(mxdts), tend(mxdts), nskip
     &                , modes, nptslt
      common /search/ tkstrt, tkend, tkstep, nsigtk
c
 1    format(a)
c
c  open file for input
c
      open(luna,file='ep.inp')
c
c  read in program control
c    dpsw = .true. >>>> discard positive sigma terms
c    szsw = .true. >>>> set positive sigma mode amplitudes to zero
c    dcsw = .true. >>>> generate E-pulse to kill DC
c    irsw = .true. >>>> print intermediate results
c
      read(luna,*) dpsw, szsw, dcsw, irsw
c
c  read filename for results summary output
c
      read (luna,1) filers
c
c  read filename for E-pulse output
c
      read (luna,1) fileep
c
c  read filename for complex natural frequency output
c
      read(luna,1) filefr
c
c  read in # of data sets
c
      read  (luna,*) nresp
      write (*,*) 'number of data sets is ',nresp
c
c  read the response filenames and it's corresponing
c  start and end of late-time
c
      do 10 i = 1, nresp
        read(luna,1) respnm(i)
```

```
        write (*,*) 'i=',i
        write (*,*) 'respnm=',respnm(i)
        read(luna,*) tstart(i),tend(i)
        write (*,*) 'tstart,tend=',tstart(i),tend(i)
10      continue
c
c  read # modes, # of match points in late time
c
      read(luna,*) modes, nptslt
      write (*,*) 'modes=',modes
      write (*,*) 'nptslt=',nptslt
c
c  read in number of points to skip
c
      read(luna,*) nskip
      write (*,*) 'nskip=',nskip
c
c  read in lower, upper limit, increment in duration search
c  of optimum E-pulse
c
      read(luna,*) tkstrt, tkend, tkstep
      write (*,*) 'tkstrt,tkend,tkstep=',tkstrt,tkend,tkstep
c
c  read in accuracy in squared error per point in duration search
c  of optimum E-pulse
c
      read(luna,*) nsigtk
      write (*,*) 'nsigtk=',nsigtk
c
      write (*,*) 'All info read OK'
      close(luna)
c
c  write out a message, when upper limit used in search of optimum
c  exceeds the minimum late-time period
c
      temp = 1.0d10
c
      do 20 i = 1, nresp
20      if(temp .ge. (tend(i)-tstart(i))) temp = tend(i)-tstart(i)
c
      if (tkend .ge. temp) then
        write(*,*)' Upper limit used in search of optimum E-pulse'
        write(*,*)' has exceeded the minimum late-time period'
        stop
      end if
c
      return
c
      end
c
c-----------------------------------------------------------------------
c
c  RDRESP - Reads response files and loads data into two dimensional
c           array H( , ).
c
      subroutine rdresp
c
      implicit real*8 (a-h,o-z)
c
      parameter (mxpts=2048, mxdts=10)
c
      character*30 respnm(mxdts), filers, fileep, filefr
      character*30 fname, ftype
c
```

```
      real*8 xr(mxpts), xi(mxpts)
c
      common /files/ respnm, filers, fileep, filefr
      common /system/ nresp, tstart(mxdts), tend(mxdts), nskip
     &                , modes, nptslt
      common /rtime/ dt, nstart(mxdts), nend(mxdts), npts
      common /btime/ dtbf, nptsbf(mxdts), nbftot
      common /harray/ h(mxdts,mxpts), hbf(mxdts,mxpts)
      common /tarray/ t(mxpts), tbf(mxpts)
c
c  read in all responses one at a time
c
      do 20 i = 1, nresp
c
        fname = respnm(i)
        call rddata(fname,ftype,s0,ds,npts,sigma,omega,xr,xi)
c
c  put response data in H(,) array
c
        do 10 j = 1, npts
 10       h(i,j) = xr(j)
c
 20   continue
c
c  construct an array containing time values
c
      do 30 j = 1, npts
 30     t(j) = (j-1)*ds
c
c  save sampling interval in common block variable
c
      dt = ds
c
      return
c
      end
c
c-----------------------------------------------------------------------
c
c   PROCRES - Round off start and end times. Calculate slopes of
c             response data.
c
      subroutine procres
c
      implicit real*8 (a-h,o-z)
c
      parameter (mxpts=2048, mxdts=10)
c
      common /system/ nresp, tstart(mxdts), tend(mxdts), nskip
     &                , modes, nptslt
      common /rtime/ dt, nstart(mxdts), nend(mxdts), npts
      common /btime/ dtbf, nptsbf(mxdts), nbftot
      common /harray/ h(mxdts,mxpts), hbf(mxdts,mxpts)
      common /sarray/ slope(mxdts,mxpts)
c
c   Round off start and end time of each data set
c
      do 10 i = 1, nresp
        nstart(i) = 1 + int(tstart(i) / dt)
        nend(i) = int(tend(i) / dt)
10    continue
c
      do 15 i = 1, nresp
        tstart(i) = dt * nstart(i)
```

```
        tend(i) = dt * nend(i)
15    continue
c
c  Calculate slopes of responses
c
      iend = npts - 1
      do 30 i = 1, nresp
        do 20 j = 1, iend
          slope(i,j) = (h(i,j+1) - h(i,j)) / dt
 20     continue
 30   continue
c
      return
c
      end
c
c-----------------------------------------------------------------------
c
c  SKIP - Skips the specified number of points in the late time data.
c    Skipping allows use data of sets with dense sampling without
c    excessive run time.
c
      subroutine skip
c
      implicit real*8 (a-h,o-z)
c
      parameter (mxpts=2048, mxdts=10)
c
      logical dpsw, szsw, dcsw, irsw
c
      common /switch/ dpsw, szsw, dcsw, irsw
      common /system/ nresp, tstart(mxdts), tend(mxdts), nskip
     &                , modes, nptslt
      common /rtime/ dt, nstart(mxdts), nend(mxdts), npts
      common /btime/ dtbf, nptsbf(mxdts), nbftot
      common /harray/ h(mxdts,mxpts), hbf(mxdts,mxpts)
      common /tarray/ t(mxpts), tbf(mxpts)
c
c  Calculating sampling interval in best fit
c
      dtbf = dt * (nskip+1)
c
c  Take every nskip th point and put it into hbf(,) array and tbf()
c     array
c
      i = 0
      n = -nskip
c
10    i = i + 1
      n = n + nskip + 1
      if (n .gt. npts) goto 30
      tbf(i) = t(n)
c
      do 20 k = 1, nresp
        hbf(k,i) = h(k,n)
20    continue
      goto 10

30    nbftot = i-1
c
c  recalculate nstart and nend to match the skipped data
c
      do 40 i = 1, nresp
        nstart(i) = 1 + int(tstart(i) / dtbf)
```

```
        nend(i) = int(tend(i) / dtbf)
        nptsbf(i) = nend(i) - nstart(i)
40    continue
c
      if (irsw) then
        do 50 i = 1, nresp
50        write (*,*) '# points in best fit in data #1= ',nptsbf(i)
      endif
c
      return
c
      end
c
c-----------------------------------------------------------------------
c
c  ENERGY - Calculates energy.
c
      subroutine energy
c
      implicit real*8 (a-h,o-z)
c
      parameter (mxpts=2048, mxdts=10)
c
      common /system/ nresp, tstart(mxdts), tend(mxdts), nskip
     &                , modes, nptslt
      common /rtime/ dt, nstart(mxdts), nend(mxdts), npts
      common /btime/ dtbf, nptsbf(mxdts), nbftot
      common /harray/ h(mxdts,mxpts), hbf(mxdts,mxpts)
      common /enerlt/ enerl(mxdts)
c
      do 10 i = 1, nresp
10      enerl(i) = 0.d0
c
c  Calculate the late-time energy of each response
c
      do 25 k = 1, nresp
        do 20 i = nstart(k), nend(k)
            enerl(k) = enerl(k) + hbf(k,i)*hbf(k,i)
20    continue
25    continue
c
      write (*,*) 'Energy calculated'
      esav = enerl(1)
      write (*,*) 'esav=',esav
      do 40 k = 1, nresp
      write (*,*) 'Relative late-time energy of data set ',k,'  =',
     &                enerl(k) / esav
40    continue
c
      return
c
      end
c
c-----------------------------------------------------------------------
c
c  MINFUNC - Searches for the minimum point of a function.
c
      subroutine minfunc
c
      implicit real*8 (a-h,o-z)
c
      parameter (mxpts=2048, mxdts=10, mxtkpts=1024)
c
      real*8 serr(mxtkpts)
```

```
c
      common /search/ tkstrt, tkend, tkstep, nsigtk
      common /final/ tkmin, smin
c
      external sqrtot
c
c  Step through tk and locate minimum squared error pt
c
      tk = tkstrt
      itk = 0
10    itk = itk + 1
      serr(itk) = sqrtot(tk)
      tk = itk * tkstep + tkstrt
      if (tk .le. tkend) go to 10
c
      imin = 1
      smin = serr(1)
      do 20 i = 1, itk
        if (serr(i) .lt. smin) then
          smin = serr(i)
          imin = i
        endif
20    continue
c
c  Locate minimum point
c
      if ((imin .eq. 1) .or. (imin .eq. itk)) then
        tkmin = tkstrt + (imin-1)*tkstep
        smin = sqrtot(tkmin)
      else
c
        asq = tkstrt + (imin-2)*tkstep
        bsq = asq + tkstep
        csq = bsq + tkstep
c
c  Calculate accuracy from number of significant figures
c
        acc = 10.d0**(-nsigtk)
c
        smin  = golden (asq,bsq,csq,sqrtot,acc,tktmp)
c
c  Reassign temporary variable used in call list to common block
c  variable
c
        tkmin = tktmp
c
      endif
c
      return
c
      end
c
c-----------------------------------------------------------------------
c
c  WRRES - Writes results to output files.
c
      subroutine wrres
c
      implicit real*8 (a-h,o-z)
c
      parameter (luna=10, luna1=12, mxpts=2048, mxdts=10, mxnms=20,
     &           mxnps=42, mxtkpts=1024)
c
      character*30 respnm(mxdts), filers, fileep, filefr
```

```
      character*30 fname, ftype
      character*20 filnam(20)
c
      real*8 b(mxnps), hbfs(mxpts), xi(mxpts)
c
      common /files/ respnm, filers, fileep, filefr
      common /system/ nresp, tstart(mxdts), tend(mxdts), nskip
     &               , modes, nptslt
      common /rtime/ dt, nstart(mxdts), nend(mxdts), npts
      common /btime/ dtbf, nptsbf(mxdts), nbftot
      common /harray/ h(mxdts,mxpts), hbf(mxdts,mxpts)
      common /tarray/ t(mxpts), tbf(mxpts)
      common /sample/ npulse, dt1, dt2(mxdts), te(mxdts), nmodes
      common /final/ tkmin, smin
      common /freq/ x(mxnps)
c
c  create file names for best fit waveforms
c
        filnam(1) = 'bf1.dat'
        filnam(2) = 'bf2.dat'
        filnam(3) = 'bf3.dat'
        filnam(4) = 'bf4.dat'
        filnam(5) = 'bf5.dat'
        filnam(6) = 'bf6.dat'
        filnam(7) = 'bf7.dat'
        filnam(8) = 'bf8.dat'
        filnam(9) = 'bf9.dat'
        filnam(10) = 'bf10.dat'
        filnam(11) = 'bf11.dat'
        filnam(12) = 'bf12.dat'
        filnam(13) = 'bf13.dat'
        filnam(14) = 'bf14.dat'
        filnam(15) = 'bf15.dat'
        filnam(16) = 'bf16.dat'
        filnam(17) = 'bf17.dat'
        filnam(18) = 'bf18.dat'
        filnam(19) = 'bf19.dat'
        filnam(20) = 'bf20.dat'
c
c  Write E-pulse waveform data to disk
c
      call ampltd(b)
c
      fname = fileep
      ftype = 'Time'
      s0 = 0
      ds = tkmin / npulse
      np = npulse
c
      call wrdata(fname,ftype,s0,ds,np,b,xi)
c
c  Open file for summary output
c
      open(luna,file=filers)
      open (luna1, file='mode.dat',status='unknown')
      write (luna1,*) nresp,nmodes
c
      write (luna,*) 'results --------------------------'
      write (luna,*) 'tk = ',tkmin
      write (luna,*) 'squared error per pt = ',smin
c
      do 30 k = 1, nresp
        write (luna,*) 'data set ',k
c
```

```
        do 10 j = 1, nbftot
10        hbfs(j) = hbf(k,j)
c
        jdc = 1
        tstemp = tstart(k)
        tetemp = tend(k)
        ntmp = nmodes
        dttemp = dtbf
        call linear (x,ntmp,jdc,tstemp,tetemp,hbfs,tbf,dttemp,b)
c
        do 20 j = 1, nmodes
          eng = ener(b(2*j-1),x(2*j-1),x(2*j),b(2*j),tstemp,tetemp)
          write (luna,*) 'mode no. ',j
          write (luna,*) 'sigma     = ',x(2*j-1)
          write (luna,*) 'omega     = ',x(2*j)
          write (luna,*) 'amp       = ',b(2*j-1)
          write (luna,*) 'phase     = ',b(2*j)
          write (luna,*) 'energy    = ',eng
20        continue
        write (luna,*) 'dc        = ',b(2*nmodes+1)
c
c  write out modal info
c
        do 25 j=1, nmodes
          ar = b(2*j-1)*cos(b(2*j))
          ai = b(2*j-1)*sin(b(2*j))
          write (luna1,*) ar,ai,x(2*j-1),x(2*j)
25        continue
        write (luna1,*) b(2*nmodes+1)
c
30      continue
c
      close(luna)
      close (luna1)
c
c  open file for best fit data
c
      do 60 k = 1, nresp
      open (luna,file=filnam(k),status='unknown')
c
      do 35 j = 1, nbftot
35        hbfs(j) = hbf(k,j)
c
        tstemp = tstart(k)
        tetemp = tend(k)
        ntmp = nmodes
        dttemp = dtbf
        call linear (x,ntmp,jdc,tstemp,tetemp,hbfs,tbf,dttemp,b)
c
        do 50 i = nstart(k), nend(k)
          ti = tbf(i)
          f = b(2*nmodes+1)
          do 40 j = 1, nmodes
40          f = f + b(2*j-1)*exp(x(2*j-1)*ti)*cos(x(2*j)*ti+b(2*j))
          write (luna,*) ti,hbf(k,i),f
50        continue
        close (luna)
60      continue
c
c  open file for complex natural frequency output
c
      open(luna,file=filefr)
c
      do 70 i = 1, nmodes
```

```
70      write(luna,*) x(2*i-1), x(2*i)
c
      close(luna)
c
      return
c
      end
c
c-----------------------------------------------------------------------
c
c  ENER - Calculates energy in a single mode waveform between
c         t = t1 and t2
c
      real*8 function ener (amp,sigma,omega,phi,t1,t2)
c
      implicit real*8 (a-h,o-z)
      complex*16 s, jphi
c
      jphi = dcmplx(0.,phi)
      s = dcmplx(sigma,omega)
c
      if (sigma .eq. 0.d0) then
        term = 2.d0*(t2-t1)
      else
        term = (exp(2.d0*sigma*t2) - exp(2.d0*sigma*t1)) / sigma
      endif
c
      ener = (amp*amp/4.d0)*(term + dreal(exp(2.d0*(s*t2+jphi))/s) -
     &          dreal(exp(2.d0*(s*t1+jphi))/s))
c
      return
c
      end
c
c-----------------------------------------------------------------------
c
c  SQRTOT - Calculates the squared error per point between the actual
c    data and the data constructed from extracted natural frequencies
c    for a given E-pulse duration. The E-pulse method is used to extract
c    the natural frequencies.  This function is minimized to find the
c    optimum fit between actual and reconstructed data.  This should
c    yield the optimum E-pulse.
c
      real*8 function sqrtot (tk)
c
      implicit real*8 (a-h,o-z)
c
      parameter (luntk=11, mxpts=2048, mxdts=10, mxnms=20,
     &           mxnps=42)
c
      logical dpsw, szsw, dcsw, irsw
c
      real*8 sg(mxnps-1), om(mxnps-1)
      complex*16 zc(mxnps), zr(mxnps), zz
      real*8 hbfs(mxpts), b(mxnps)
c
      common /switch/ dpsw, szsw, dcsw, irsw
      common /system/ nresp, tstart(mxdts), tend(mxdts), nskip
     &                , modes, nptslt
      common /search/ tkstrt, tkend, tkstep, nsigtk
      common /rtime/ dt, nstart(mxdts), nend(mxdts), npts
      common /btime/ dtbf, nptsbf(mxdts), nbftot
      common /harray/ h(mxdts,mxpts), hbf(mxdts,mxpts)
      common /tarray/ t(mxpts), tbf(mxpts)
```

```
      common /enerlt/ enerl(mxdts)
      common /sample/ npulse, dt1, dt2(mxdts), te(mxdts), nmodes
      common /freq/ x(mxnps)
c
c  Calculate start of late time, and fining out the
c  biggest end time
c
      tendb = tend(1)
c
      do 5 i = 1, nresp
        if ( tendb .le. tend(i) ) tendb = tend(i)
        te(i) = tstart(i) + tk
        dt2(i) = (tend(i)-te(i)) / (nptslt-1)
5     continue
c
      dt1 = tk /npulse
c
c  Calculate pulse function amplitudes for E-pulse
c
      call ampltd(b)
c
c  Calculate complex zero's of E-pulse. These correspond to the
c  natural frequencies of the response.
c
      do 10 i = 1, npulse
10      zc(i) = b(i)
c
      ndeg = npulse - 1
      call zroots (zc,ndeg,zr,.true.)
c
c  Calculate complex frequencies
c
      do 30 i = 1, ndeg
        zz = (-1.d0/dt1) * log(zr(i))
        sg(i) = dreal(zz)
30      om(i) = abs(dimag(zz))
c
      if (irsw) then
        do 40 i = 1, ndeg
40        write (*,*) 'sg = ',sg(i),'   om = ',om(i)
      endif
c
c  Identify complex conjugate pairs
c
      nmodes = 0
      ndegm1 = ndeg - 1
      do 60 j = 1, ndegm1
        iflag = 0
        jp1 = j + 1
c
        do 50 i = jp1, ndeg
c
          if (om(j) .eq. 0.d0) om(j) = 1.d-5
          test = abs(om(i) - om(j)) / om(j)
          if (test .lt. 1.d-4) iflag=1
50      continue
c
c  test for multiple frequency
c
        if (nmodes .ge. 1) then
          do 45 i = 1, nmodes
            test = abs(om(i) - om(j)) / om(j)
            if (test .lt. 1.d-4) iflag = 0
45        continue
```

```
        endif
c
        if (iflag.eq.1) then
c
          if ((sg(j) .gt. 0.d0) .and. (szsw)) then
            nmodes = nmodes + 1
            om(nmodes) = om(j)
            sg(nmodes) = 0.d0
            if (irsw) write (*,*) 'set sigma = 0.0'
          else if ((sg(j) .gt. 0.d0) .and. (dpsw)) then
            if (irsw) write (*,*) 'discarding +sigma term'
          else if (sg(j)*tendb .gt. npts*dt) then
            if (irsw) write (*,*) 'discard large +sigma term'
          else
            nmodes = nmodes + 1
            om(nmodes) = om(j)
            sg(nmodes) = sg(j)
          endif
        endif
60    continue
c
      do 70 i = 1, nmodes
        x(2*i-1) = sg(i)
        x(2*i) = om(i)
70    continue
c
      sqrtot = 0.d0
c
      do 120 k = 1, nresp
c
        do 110 j = 1, nbftot
110        hbfs(j) = hbf(k,j)
c
        jdc = 1
c
c  Assign temporary variables for call list instead of using
c  common block variables
c
        tstmp = tstart(k)
        tetmp = tend(k)
        ntmp = nmodes
c
        call linear (x,ntmp,jdc,tstmp,tetmp,hbfs,tbf,dtbf,b)
c
        if (irsw) then
          write (*,*) 'data set # ',k
          do 100 i = 1 ,nmodes
            write (*,*) 'sg = ',x(2*i-1),'   om = ',x(2*i)
100         write (*,*) 'am = ',b(2*i-1),'   ph = ',b(2*i)
          write (*,*) 'dc = ',b(2*nmodes+1)
        endif
c
c  calculate squared error for data set k
c
        serror = 0.d0
c
        do 90 i = nstart(k), nend(k)
          ti = tbf(i)
          f = b(2*nmodes+1)
c
          do 80 j = 1, nmodes
80         f = f + b(2*j-1)*exp(x(2*j-1)*ti)*cos(x(2*j)*ti+b(2*j))
c
          serror = serror + (f-hbf(k,i))*(f-hbf(k,i))
```

```
90     continue
c
        sqrerr = serror/nptsbf(k)
c
        sqrtot = sqrtot + sqrerr/enerl(k)
c
120    continue
c
      write (*,*) 'tk = ',tk,'     sqrtot = ', sqrtot
c
c  Write squared error data to disk
c
      write(luntk,*) tk, sqrtot
c
      return
c
      end
c
c-----------------------------------------------------------------------
c
      subroutine ampltd (b)
c
      implicit real*8 (a-h,o-z)
c
      parameter (mxpts=2048, mxdts=10, mxnps=42)
c
      logical dpsw,szsw,dcsw,irsw
c
c  Solves for E-pulse pulse function amplitudes
c
      real*8 b(mxnps)
      real*8 a(mxnps,mxnps)
      real*8 diag(mxnps)
      real*8 save(mxnps)
      integer ipvt(mxnps)
c
      common /switch/ dpsw, szsw, dcsw, irsw
      common /system/ nresp, tstart(mxdts), tend(mxdts), nskip
     &                , modes, nptslt
      common /search/ tkstrt, tkend, tkstep, nsigtk
      common /rtime/ dt, nstart(mxdts), nend(mxdts), npts
      common /btime/ dtbf, nptsbf(mxdts), nbftot
      common /harray/ h(mxdts,mxpts), hbf(mxdts,mxpts)
      common /sarray/ slope(mxdts,mxpts)
      common /tarray/ t(mxpts), tbf(mxpts)
      common /enerlt/ enerl(mxdts)
      common /sample / npulse, dt1, dt2(mxdts), te(mxdts), nmodes
c
c  fill matrix
c
      np1 = npulse - 1
c
      do 10 n = 1, np1
        do 5 m = 1, np1
          a(m,n) = 0.d0
5       continue
      b(n) = 0.d0
10    continue
c
      do 50 k = 1, nresp
        energ  = enerl(k)
c
        do 40 m = 1, nptslt
c
```

```
          tm = te(k) + dt2(k)*(m-1)
c
          do 30 n = 1, npulse
c
            t1n = (n-1)*dt1
            t2n = t1n + dt1
c
            tl = tm - t2n
            tu = tm - t1n
            ml = 2 + int((tl - t(1)) / dt)
            mu = 1 + int((tu - t(1)) / dt)
            deltal = tl - t(ml - 1)
            deltau = tu - t(mu)
c
            sum1 = 0.d0
            sum2 = 0.d0
            iend = mu - 1
c
            do 20 i = ml, iend
              sum1 = sum1 + slope(k,i)
              sum2 = sum2 + h(k,i)
 20         continue
c
            temp = (dt-deltal)*(0.5d0*slope(k,ml-1)*
     &             (dt+deltal)+h(k,ml-1)) +
     &             deltau*(0.5d0*slope(k,mu)*dt+h(k,mu)) +
     &             dt*(0.5d0*dt*sum1+sum2)
            save(n) = temp/energ
c
 30       continue
c
          do 12 kk = 1, np1
            do 11 jj = 1, np1
              a(kk,jj) = a(kk,jj) + save(jj)*save(kk)
 11         continue
            b(kk) = b(kk) + save(kk)*(-save(npulse))
 12       continue
c
 40     continue
c
 50   continue
c
c  fill in for dc e-pulse
c
      if (dcsw) then
        do 60 kk = 1, npulse
          a(kk,npulse) = 1.d0
 60     continue
c
        do 70 jj = 1, npulse
          a(npulse,jj) = 1.d0
 70     continue
c
        a(npulse,npulse) = 0.d0
        b(npulse) = -1.d0
      endif
c
c  solve matrix problem
c
      if (dcsw) then
        nsolve = npulse
      else
        nsolve = np1
      endif
```

```
c
      call decomp (nsolve,a,ipvt,diag)
      call solve  (nsolve,a,ipvt,b)
c
      b(npulse) = 1.d0
c
      return
      end
c
c-----------------------------------------------------------------------
c
c  LINEAR - Peforms linear regression (least squares) to obtain
c           amplitudes and phases + dc
c
      subroutine linear (x,nmodes,jdc,tstart,tend,h,t,dt,b)
c
      implicit real*8 (a-h,o-z)
c
      parameter (mxpts=2048, mxnps=42)
c
      real*8 a(mxnps,mxnps), diag(mxnps)
      real*8 b(mxnps), x(mxnps)
      real*8 h(mxpts), t(mxpts)
      integer ipvt(mxnps)
c
      pi = 4.d0 * atan(1.d0)
c
      nstart = 2 + int(tstart / dt)
      nend = 1 + int(tend / dt)
      nmodt2 = 2*nmodes
      nmodp = nmodt2 + 1
c
c  fill B vector
c
      do 20 i = 1, nmodp
      b(i) = 0.d0
20    continue
c
      do 100 i = nstart, nend
        hi = h(i)
        ti = t(i)
        n = 1
        do 50 j = 1, nmodes
          ex = exp(x(n)*ti)
          b(n) = b(n) + hi*ex*cos(x(n+1)*ti)
          b(n+1) = b(n+1) + hi*ex*sin(x(n+1)*ti)
          n = n + 2
50      continue
        b(nmodp) = b(nmodp) + hi
100   continue
c
c  fill a matrix
c
      do 120 i = 1, nmodp
        do 110 j = 1, nmodp
          a(i,j) = 0.d0
110     continue
120   continue
c
      do 200 i = nstart, nend
        ti = t(i)
        n = 1
        do 150 j = 1, nmodes
          exn = exp(x(n)*ti)
```

```
            cen = exn*cos(x(n+1)*ti)
            sen = exn*sin(x(n+1)*ti)
            m = 1
            do 140 k = 1, nmodes
              exm = exp(x(m)*ti)
              cem = exm*cos(x(m+1)*ti)
              sem = exm*sin(x(m+1)*ti)
              a(n,m) = a(n,m) + cen*cem
              a(n,m+1) = a(n,m+1) + cen*sem
              a(n+1,m) = a(n+1,m) + sen*cem
              a(n+1,m+1) = a(n+1,m+1) + sen*sem
              m = m + 2
140         continue
            a(n,nmodp) = a(n,nmodp) + cen
            a(n+1,nmodp) = a(n+1,nmodp) + sen
            n = n + 2
150       continue
          m = 1
          do 160 k = 1, nmodes
            exm = exp(x(m)*ti)
            cem = exm*cos(x(m+1)*ti)
            sem = exm*sin(x(m+1)*ti)
            a(nmodp,m) = a(nmodp,m) + cem
            a(nmodp,m+1) = a(nmodp,m+1) + sem
            m = m + 2
160       continue
200     continue
c
        a(nmodp,nmodp) = a(nmodp,nmodp) + dfloat(nend-nstart+1)
c
        nmat = nmodt2
        if (jdc .eq. 1) nmat = nmodp
c
        call decomp (nmat,a,ipvt,diag)
        call solve  (nmat,a,ipvt,b)
c
c  put into phase-amplitude form
c
        n = 1
        do 250 i = 1, nmodes
          an = sqrt(b(n)*b(n) + b(n+1)*b(n+1))
          phin = atan(-b(n+1)/b(n))
          if (b(n) .lt. 0.d0) phin = phin + pi
          b(n) = an
          b(n+1) = phin
          n = n + 2
250     continue
c
        return
        end
c
c-----------------------------------------------------------------------
c
c  DECOMP - Decomposes matrix A to triangular form.
c           From: Linear Algebra by Gilbert Strang
c
        subroutine decomp (n,a,ipvt,diag)
c
        implicit real*8 (a-h,o-z)
c
        parameter (mxnps=42)
c
        integer n,ipvt(mxnps)
        integer nm1,i,j,k,kp1,m
```

```
      real*8 a(mxnps,mxnps)
      real*8 diag(mxnps)
c
      ipvt(n) = 1
      if (n .eq. 1) go to 70
      nm1     = n-1
c
      do 60 k=1,nm1
      kp1     = k+1
c
c  find pivot p
c
      m       = k
      do 10 i=kp1,n
10    if (abs(a(i,k)) .gt. abs(a(m,k))) m=i
      ipvt(k) = m
      if (m .ne. k) ipvt(n) = -ipvt(n)
      p       = a(m,k)
      a(m,k) = a(k,k)
      a(k,k) = p
      diag(k) = p
      if (p  .eq. 0.0d0) go to 60
c
c  compute multipliers
c
20    do 30 i=kp1,n
30    a(i,k) = -a(i,k)/p
c
c  interchange rows and columns
c
      do 50 j=kp1,n
      t       = a(m,j)
      a(m,j) = a(k,j)
      a(k,j) =  t
      if (t .eq. 0.0d0) go to 50
      do 40 i=kp1,n
      a(i,j) = a(i,j) + a(i,k)*t
40    continue
50    continue
60    continue
c
70    diag(n) = a(n,n)*dfloat(ipvt(n))
c
      return
      end
c
c-----------------------------------------------------------------------
c
c  SOLVE - Solves matrix eqn Ax = B with a decomposed dy DECOMP
c
      subroutine solve (n,a,ipvt,b)
c
      implicit real*8 (a-h,o-z)
c
      parameter (mxnps=42)
c
      integer n,ipvt(mxnps)
      integer nm1,k,kb,kp1,km1,m,i
      real*8 a(mxnps,mxnps)
      real*8 b(mxnps)
c
c  forward elimination
c
      if (n .eq. 1) go to 30
```

```
      nm1     = n-1
      do 10 k=1,nm1
      kp1     = k+1
      m       = ipvt(k)
      s       = b(m)
      b(m)    = b(k)
      b(k)    = s
      do 10 i=kp1,n
10    b(i)    = b(i) + a(i,k)*s
c
c  back substitution
c
      do 20 kb=1,nm1
      km1     = n-kb
      k       = km1+1
      b(k)    = b(k)/a(k,k)
      s       = -b(k)
      do 20 i=1,km1
20    b(i)    = b(i) + a(i,k)*s
c
30    b(1)    = b(1)/a(1,1)
c
      return
      end
c
c-----------------------------------------------------------------------
c
c  ZROOTS - Calculates complex roots. From Numerical Recipies.
c
      subroutine zroots(a,m,roots,polish)
      implicit real*8 (a-h,o-z)
      parameter (eps=1.d-6,maxm=101)
      complex*16 a(m+1),roots(m),ad(maxm),x,b,c
      logical polish
      do 11 j=1,m+1
      ad(j)=a(j)
11    continue
      do 13 j=m,1,-1
      x=dcmplx(0.d0,0.d0)
      call laguer(ad,j,x,eps,.false.)
      if(abs(dimag(x)).le.2.d0*eps**2*abs(dreal(x)))
     2   x=dcmplx(dreal(x),0.d0)
      roots(j)=x
      b=ad(j+1)
      do 12 jj=j,1,-1
        c=ad(jj)
        ad(jj)=b
        b=x*b+c
12      continue
13    continue
      if (polish) then
      do 14 j=1,m
        call laguer(a,m,roots(j),eps,.true.)
14      continue
      endif
      do 16 j=2,m
      x=roots(j)
      do 15 i=j-1,1,-1
        if(dreal(roots(i)).le.dreal(x))go to 10
        roots(i+1)=roots(i)
15      continue
      i=0
10      roots(i+1)=x
16    continue
```

```
      return
      end
c
c-------------------------------------------------------------------------
c
c LAGUER - From Numerical Recipies
c
      subroutine laguer(a,m,x,eps,polish)
      implicit real*8 (a-h,o-z)
      complex*16 a(m+1),x,dx,x1,b,d,f,g,h,sq,gp,gm,g2,zero
      logical polish
      parameter (zero=(0.d0,0.d0),epss=6.d-8,maxit=100)
      dxold=abs(x)
      do 12 iter=1,maxit
      b=a(m+1)
      err=abs(b)
      d=zero
      f=zero
      abx=abs(x)
      do 11 j=m,1,-1
        f=x*f+d
        d=x*d+b
        b=x*b+a(j)
        err=abs(b)+abx*err
11      continue
      err=epss*err
      if(abs(b).le.err) then
        dx=zero
        return
      else
        g=d/b
        g2=g*g
        h=g2-2.d0*f/b
        sq=sqrt((m-1)*(m*h-g2))
        gp=g+sq
        gm=g-sq
        if(abs(gp).lt.abs(gm)) gp=gm
        dx=m/gp
      endif
      x1=x-dx
      if(x.eq.x1)return
      x=x1
      cdx=abs(dx)
      if(iter.gt.6.and.cdx.ge.dxold)return
      dxold=cdx
      if(.not.polish)then
        if(abs(dx).le.eps*abs(x))return
      endif
12    continue
      pause 'too many iterations'
      return
      end
c
c-------------------------------------------------------------------------
c
c GOLDEN - From Numerical Recipies
c
      real*8 function golden (ax,bx,cx,f,tol,xmin)
      implicit real*8 (a-h,o-z)
      r=.61803399d0
      c=1.d0-r
      x0=ax
      x3=cx
      if(abs(cx-bx).gt.abs(bx-ax))then
```

```
        x1=bx
        x2=bx+c*(cx-bx)
      else
        x2=bx
        x1=bx-c*(bx-ax)
      endif
      f1=f(x1)
      f2=f(x2)
1      if(abs(x3-x0).gt.tol*(abs(x1)+abs(x2)))then
        if(f2.lt.f1)then
            x0=x1
            x1=x2
            x2=r*x1+c*x3
            f0=f1
            f1=f2
            f2=f(x2)
        else
            x3=x2
            x2=x1
            x1=r*x2+c*x0
            f3=f2
            f2=f1
            f1=f(x1)
        endif
      goto 1
      endif
      if(f1.lt.f2)then
        golden=f1
        xmin=x1
      else
        golden=f2
        xmin=x2
      endif
      return
      end
c
c-----------------------------------------------------------------------
c
c  RDDATA - This subroutine reads in a data set that has uniform
c  sampling.
c    The first line of the data set contains the filetype
c        Time
c        Frequency
c        S-Pulse
c    If file contains S-Pulse data then the next line has the
c    sigma and omega of the mode associated with that S-Pulse.
c    The following lines contain:
c      start time.        nS or Ghz
c      sampling interval  nS or Ghz
c      sampled data points (# for time, # # for frequency and S-Pulse)
c                          (S-Pulse = Cosine, Sine pulses)
c
      subroutine rddata(fnm,ftype,s0,ds,np,sigma,omega,x,y)
c
      implicit real*8 (a-h,o-z)
c
      parameter (luna=10, mxpts=2048)
c
      character*30 fnm, ftype
      real*8 x(mxpts), y(mxpts)
c
      open(unit=luna, file=fnm)
c
c  Read in filetype
```

```
c
      read(luna,1) ftype
1     format(a)
c
      if (ftype.eq.'Spulse') then
      read(luna,*) sigma, omega
      endif
      read(luna,*) s0
      read(luna,*) ds
c
c  Read in the sampled values.
c
      i = 1
      if (ftype.eq.'Time') then
 10     read(luna,*,end=99,err=999) x(i)
      y(i) = 0
      i = i + 1
      goto 10
      else if (ftype.eq.'Spulse') then
 20     read(luna,*,end=99,err=999) x(i), y(i)
      i = i + 1
      goto 20
      endif
c
 99   np = i-1
c
      close(luna)
c
      return
c
 999  write(*,*) '   ***  Error in subroutine rddata ***'
      close(luna)
      stop
c
      end
c
c-----------------------------------------------------------------------
c
c  WRDATA - This subroutine writes uniformly sampled data to a file.
c    The first line of the data set contains the filetype
c        Time
c        Frequency
c        S-Pulse
c    If file contains S-Pulse data then the next line has the
c    sigma and omega of the mode associated with that S-Pulse.
c    The following lines contain:
c      start time.        nS or Ghz
c      sampling interval  nS or Ghz
c      sampled data points (# for time, # # for frequency and S-Pulse)
c                          (S-Pulse = Cosine, Sine pulses)
c
      subroutine wrdata(fnm,ftype,s0,ds,npts,v,x)
      implicit real*8 (a-h,o-z)
      parameter(luna=10, mxpts=2048)
c
      real*8 v(mxpts), x(mxpts)
      character*30 fnm, ftype
c
      open(luna, file=fnm)
c
c  Write out sampling interval.
c
      write(luna,5) ftype
  5   format (a)
```

```
      write(luna,*) s0
      write(luna,*) ds
c
c  Write out sampled values.
c
      if (ftype.eq.'Time') then
      do 20 i = 1, npts
 20     write(luna,*) v(i)
      endif
c
      if (ftype.eq.'Frequency') then
      do 30 i = 1, npts
      write(luna,*) v(i), x(i)
 30     continue
      endif
c
      close(luna)
c
      return
c
      end
c
c-------------------------------------------------------------------------
```

A.4 Radiated Emissions Program

The program *Vome.for* was written to take the input file res.dat generated from the program *ep.for* and extracts the natural frequency information. Impedance data is also inputted from an impedance measurement data file. Worst-case voltages are computed for each natural frequency. From these voltages and impedances, worst-case conducted emission currents are calculated at the natural frequencies and outputted to a data file. Worst-case radiated electric fields are computed for a variety of typical harness lengths found in automobiles and outputted to a data file.

A.5 Source Code for *Vome.for*

The following is the source code for *Vome.for*.

```
      Program Vome
c
c
c     ****************************************************************
c     *                                                              *
c     *     Program Vome                                             *
c     *                                                              *
c     *     Written by Matt Feusse                                   *
c     *     Last updated: March 21, 2001                             *
c     *                                                              *
c     *     This program reads in data from res.dat and the          *
c     *     impedance files for each motor, and computes the         *
c     *     radiated E-fields and conducted emission currents at     *
c     *     the natural frequencies.                                 *
c     *                                                              *
c     ****************************************************************
c
c
      Implicit none
      Integer nummodes,ierror,I,mode(10),II,K,KK,LL,N
      Real*8 pi,d,L,latetime
      Real*8 sigma(10),omega(10),amp(10),phase(10),energy(10)
      Real*8 natfreq(10),freq(10000),imp(10000),ome(10000)
      Real*8 ratio(10),natimp(10),v(10),current(10)
      Real*8 totcurrent(10),beta(10),f(10),efield(10),dbuvefield(10)
      Character*80 motor,res,imped,newfile,newfile2
      pi=3.14159265359
      d=10.0
c
c**************************************
c     Acquiring User defined info
c**************************************
c
      Write(*,*) "Which Motor are you testing?"
      Read(*,*) motor
      Write(*,*) "Enter the name of the results data file:"
      Read(*,*) res
      Write(*,*) "Enter the name of the impedance data file:"
      Read(*,*) imped
      Write(*,*) "Enter the name of the current output file:"
      Read(*,*) newfile
      Write(*,*) "Enter the name of the e-field output file:"
      Read(*,*) newfile2
      Write(*,*) "How many modes?"
      Read(*,*) nummodes
      Write(*,*) "What is the beginning of the late time (ns)?"
      Read(*,*) latetime
      latetime=latetime*1.0e-9
c
c************************
c     Open the files
c************************
c
      Open(Unit=1,File=res,Status='Unknown',Action='read',
     &      Iostat=Ierror)
      Open(Unit=2,File=imped,Status='Unknown',Action='read',
     &      Iostat=Ierror)
      Open(Unit=3,File=newfile,Status='Unknown',Action='Write',
     &      Iostat=Ierror)
```

```
      Open(Unit=4,File=newfile2,Status='Unknown',Action='Write',
     &     Iostat=Ierror)
      Write(3,*) "Output Data File for the",motor,"Motor"
      Write(3,*) "-------------------------------------------------------"
      Write(3,*) "-------------------------------------------------------"
c
c*****************************************************************
c     This section reads the res.dat file and identifies
c     the natural frequencies
c*****************************************************************
c
      Read(1,*)
      Read(1,*)
      Read(1,*)
      Read(1,*)
      I=1
      Do I=1,nummodes
            Read(1,10) mode(I)
 10         Format(20X,I10)
            Read(1,20) sigma(I)
 20         Format(12X,E25.16)
            Read(1,30) omega(I)
 30         Format(12X,E25.16)
            Read(1,40) amp(I)
 40         Format(12X,E25.16)
            Read(1,50) phase(I)
 50         Format(12X,E25.16)
            Read(1,60) energy(I)
 60         Format(12X,E25.16)
            sigma(I)=sigma(I)*1.0e9
            omega(I)=omega(I)*1.0e9
            natfreq(I)=omega(I)/(2*pi)
      Enddo
c
c*****************************************************************
c     This section grabs the impedance and frequency data.
c*****************************************************************
c
      Read(2,*)
      Read(2,*)
      II=1
      Do While (.not. EOF(2))
            Read(2,70) freq(II),imp(II)
 70         Format(F15.8,F20.8)
            freq(II)=1.0e6*freq(II)
            ome(II)=2*pi*freq(II)
            II=II+1
      Enddo
c
c*************************************************************************
c     This section computes the impedances at the natural frequencies
c     using linear interpolation.
c*************************************************************************
c
      Do K=1,I-1
            KK=1
            Do while ((KK.LT.II-2).and.(natfreq(K).GT.freq(KK)))
                  LL=KK
                  KK=KK+1
            enddo
            ratio(K)=natfreq(K)*2/(freq(KK)+freq(KK-1))
            natimp(K)=ratio(K)*(imp(KK)+imp(KK-1))/2
      enddo
c
```

```
c*********************************************************************
c     This section takes the voltages due to each natural frequency
c     and calculates the currents due to those natural frequencies.
c     A radiated electric field is computed for a range of wire
c     harness lengths.  This information is written to an output file.
c*********************************************************************
c
      Do K=1,I-1
            Do N=1,20
                  L=N*0.05
                  v(K)=amp(K)*exp(sigma(K)*latetime)
                  current(K)=v(K)/natimp(K)
                  beta(K)=2*pi*natfreq(K)/(3.0e8)
                  f(K)=1.0-cos(beta(K)*L/2)
                  efield(K)=(abs(f(K))*120*abs(current(K)))*1.0e6/(d)
                  dbuvefield(K)=20*log10(efield(K))
                  Write(4,100) natfreq(K)/1.0e6,L*100,dbuvefield(K)
            Enddo
100         Format(F6.2,5X,F6.2,5X,F8.3)
      Enddo
c
c*********************************************************************
c     This section writes the frequency, impedance,voltage and
c     current data associated with each natural mode to a file.
c*********************************************************************
c
      Do K=1,I-1
            Write(3,*) "Natural Mode #",K
            Write(3,*) "Frequency (MHz):  ",natfreq(K)/1.0e6
            Write(3,*) "Impedance at Natural Frequency:  ",natimp(K)
            Write(3,*) "Maximum Voltage:  ",v(K)
            Write(3,*) "Maximum Current:  ",current(K)
            Write(3,*) "-----------------------------------------------"
      Enddo
      End
```

BIBLIOGRAPHY

[1] C. R. Paul, *Introduction to Electromagnetic Compatibility.* New York: John Wiley & Sons, Inc., 1992.

[2] International Electrotechnical Commission, Genève, Suisse, *CISPR 25: Limits and Methods of Measurement of Radio Disturbance Characteristics for the Protection of Receivers Used on Board Vehicles*, first ed., 1995.

[3] Daimler Chrysler, *DCC EMC Test Specification and Laboratory Procedures*, November 19, 1999.

[4] Edward, J. Rothwell, Kun-Mu Chen, Dennis P. Nyquist, Weimin Sun, "Frequency Domain E-Pulse Synthesis and Target Discrimination," *IEEE Transactions on Antennas and Propagation,* vol. AP-35, pp. 426-434, April 1987.

[5] E. J. Rothwell, K. M. Chen, "A Hybrid E-Pulse / Least Squares Technique for Natural Resonance Extraction," *Proceedings of the IEEE*, vol. 76, pp. 296-298, March 1988.