

Air Force Institute of Technology

AFIT Scholar

Theses and Dissertations

Student Graduate Works

12-1994

A State-Space Model of a Large, Lightly Damped Space Structure Using the Eigensystem Realization Algorithm

Robert D. Woods

Follow this and additional works at: <https://scholar.afit.edu/etd>

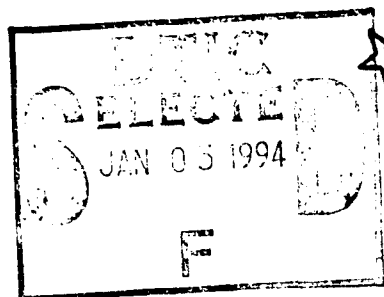


Part of the [Space Vehicles Commons](#)

Recommended Citation

Woods, Robert D., "A State-Space Model of a Large, Lightly Damped Space Structure Using the Eigensystem Realization Algorithm" (1994). *Theses and Dissertations*. 6333.
<https://scholar.afit.edu/etd/6333>

This Thesis is brought to you for free and open access by the Student Graduate Works at AFIT Scholar. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AFIT Scholar. For more information, please contact AFIT.ENWL.Repository@us.af.mil.

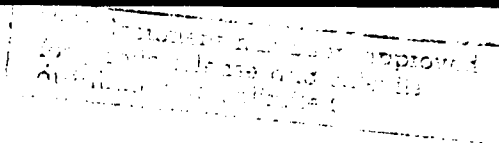


A STATE-SPACE MODEL OF A LARGE, LIGHTLY DAMPED
SPACE STRUCTURE USING THE EIGENSYSTEM
REALIZATION ALGORITHM

THESIS
Robert Daryl Woods
Captain, USAF

AFIT/GA/ENY/94D-2

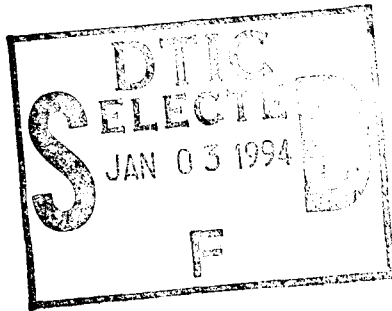
19941228 078



DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GA/ENY/94D-2



A STATE-SPACE MODEL OF A LARGE, LIGHTLY DAMPED
SPACE STRUCTURE USING THE EIGENSYSTEM
REALIZATION ALGORITHM

THESIS

Robert Daryl Woods
Captain, USAF

AFIT/GA/ENY/94D-2

Accession for	<input checked="checked" type="checkbox"/>
DTIC GRAFI	<input checked="checked" type="checkbox"/>
DTIC T/E	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Auto Report on	
By	
DTIC Date	
Air Force	
DTIC	
A-1	

Approved for public release; distribution unlimited

AFIT/GA/ENY/94D-2

A STATE-SPACE MODEL OF A LARGE, LIGHTLY DAMPED
SPACE STRUCTURE USING THE EIGENSYSTEM
REALIZATION ALGORITHM

THESIS

Presented to the Faculty of the Graduate School of Engineering
of the Air Force Institute of Technology
Air Education and Training Command
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Astronautical Engineering

Robert Daryl Woods, B.S.
Captain, USAF

DTIC QUALITY INSPECTED 2

December, 1994

Approved for public release; distribution unlimited

Acknowledgements

This thesis is dedicated to my wife, [REDACTED], and son, [REDACTED]. A special thank you goes out to the both of them for the support they gave and the sacrifices they made during my thesis work as well as my entire AFIT experience. Their love and encouragement were essential to the successful completion of this thesis.

This thesis could not have been completed without the help, advice, and encouragement of others. I would like to thank my thesis advisor, Dr. Bradley Liebst for his invaluable guidance and advice and Capt. Richard Cobb, who coded the ERA algorithm, for his help and advice throughout the research and writing of this thesis. I also thank Capt. Mark Hunter for his help in proof-reading the rough draft of this thesis. I would also like to express my appreciation to Mr. Jay Anderson, Mr. Andy Pitts, Mr. Dan Rioux, and Mr. Mark Derriso who were always available to help with the many equipment problems that were encountered during the experimental portion of this thesis. A special thanks also goes out to Mr. Robert Forbes and Mr. Robert Gordon for their computer expertise.

Robert Daryl Woods

Table of Contents

	Page
Acknowledgements	ii
List of Figures	vii
List of Tables	xii
Abstract	xiii
 I. Introduction	 1-1
1.1 Problem and General Approach	1-1
1.2 The PACOSS Program	1-2
 II. Experimental Equipment	 2-1
2.1 Description of the DTA	2-1
2.1.1 Ring Truss	2-1
2.1.2 Tripod	2-1
2.1.3 Solar Arrays	2-2
2.1.4 Suspension System	2-2
2.2 Active Vibration Control System	2-3
2.2.1 Reaction Mass Actuator	2-4
2.2.2 Accelerometer	2-4
2.2.3 Motor Control Unit	2-5
2.3 Optima/3	2-5
2.4 Additional Hardware and Software	2-6

	Page
III. Theory	3-1
3.1 Coulomb Damping	3-1
3.1.1 Case I: $\dot{x} < 0$	3-3
3.1.2 Case II: $\dot{x} > 0$	3-4
3.1.3 Determining the Coulomb Damping Coefficient	3-5
3.2 System Identification	3-6
IV. Experimental Procedure	4-1
4.1 DTA and Actuator Preparation	4-1
4.2 Validation of ERA Routines	4-2
4.3 Coulomb Damping Identification	4-3
4.4 Coulomb Damping Compensation	4-4
4.5 Data Acquisition	4-5
4.5.1 Frequency Response Functions	4-5
4.5.2 Frequency Response Functions with Friction Compensation	4-6
V. Results	5-1
5.1 Coulomb Damping Coefficients	5-1
5.2 ERA Fit of the Truth Model	5-2
5.2.1 Truth Model without Noise	5-2
5.2.2 Truth Model with Noise	5-5
5.3 ERA Fit of the PACOSS DTA	5-5
5.4 Friction Compensation: Is it worth it?	5-8
5.5 Effects of Instrumentation Wire Removal	5-11
VI. Recommendations and Conclusions	6-1
Appendix A. State-Space Model FRFs of PACOSS DTA	A-1

	Page
Appendix B. State-Space Model FRFs of PACOSS DTA (Cont.)	B-1
Appendix C. State-Space Model FRFs of PACOSS DTA (Cont.)	C-1
Appendix D. State-Space Model FRFs of PACOSS DTA (Cont.)	D-1
Appendix E. State-Space Model FRFs of PACOSS DTA (Cont.)	E-1
Appendix F. State-Space Model FRFs of PACOSS DTA (Cont.)	F-1
Appendix G. State-Space Model FRFs of PACOSS DTA (Cont.)	G-1
Appendix H. State-Space Model FRFs of PACOSS DTA (Cont.)	H-1
Appendix I. State-Space Model FRFs of Truth Model	I-1
I.1 Truth Model without Noise	I-1
I.2 Truth Model with Noise	I-6
Appendix J. Computer Routines	J-1
J.1 ERA Routines	J-1
J.1.1 frfinter.m	J-1
J.1.2 mirror.m	J-1
J.1.3 mimoera.m	J-1
J.1.4 weave.m and mimohank.m	J-2
J.1.5 parm.m	J-2
J.1.6 frffilt.m	J-2
J.2 Parameter Settings	J-2
J.2.1 PACOSS DTA	J-2
J.2.2 Truth Model	J-3

	Page
Appendix K. C Code Used in Friction Compensation	K-1
K.1 C Code for Coulomb Damping Compensator	K-1
K.2 Coulomb Damping "Tweaking" Routine	K-5
Appendix L. MATLAB TM Code	L-1
L.1 MATLAB TM ERA Code	L-1
L.1.1 frffilt.m	L-1
L.1.2 mimohank.m	L-4
L.1.3 mimoera.m	L-5
L.1.4 frfinter.m	L-7
L.1.5 mirror.m	L-8
L.1.6 weave.m	L-9
L.1.7 parm.m	L-10
L.2 MATLAB TM ERA Setup Routine	L-12
L.2.1 erasetup.m	L-12
L.2.2 frfbuild.m	L-14
L.2.3 clearxfers	L-15
L.2.4 plotfrfs.m	L-16
L.3 Friction Coefficient Calculation Routine	L-17
Bibliography	BIB-1
Vita	VITA-1

List of Figures

Figure	Page
2.1. DTA Configuration	2-2
2.2. DTA and Support Structure	2-3
2.3. Actuator/Sensor Unit	2-4
2.4. Motor Control Unit: Front Panel	2-5
3.1. Functional Diagram of Reaction Mass Actuator	3-2
3.2. Free Decay Response Function	3-3
4.1. Truth Model Diagram	4-2
4.2. Equipment Configuration for Open Loop System	4-5
4.3. Equipment Configuration with Active Friction Compensation	4-6
5.1. Displacement Y1 Force Input F2 for the Truth Model with No Noise	5-3
5.2. Displacement Y2 Force Input F2 for the Truth Model with No Noise	5-3
5.3. Displacement Y1 Force Input F2 for the Truth Model with Noise	5-4
5.4. Displacement Y2 Force Input F2 for the Truth Model with Noise	5-4
5.5. Non-collocated Pair: Measurement 1 Excitation Actuator 2	5-6
5.6. Collocated Pair: Measurement 6 Excitation Actuator 6	5-6
5.7. Measurement 1 Excitation Actuator 6	5-7
5.8. FRFs for Measurement 1 Actuator 2	5-8
5.9. FRFs for Measurement 4 Actuator 4	5-9
5.10. ERA Fit for Compensated System: Measurement 1 Actuator 1	5-10
5.11. ERA Fit for Non-Compensated System: Measurement 1 Actuator 1	5-10
5.12. Comparison of FRFs	5-11
A.1. Measurement 1 Excitation Actuator 1	A-2

Figure	Page
A.2. Measurement 2 Excitation Actuator 1	A-3
A.3. Measurement 3 Excitation Actuator 1	A-4
A.4. Measurement 4 Excitation Actuator 1	A-5
A.5. Measurement 5 Excitation Actuator 1	A-6
A.6. Measurement 6 Excitation Actuator 1	A-7
A.7. Measurement 7 Excitation Actuator 1	A-8
A.8. Measurement 8 Excitation Actuator 1	A-9
B.1. Measurement 1 Excitation Actuator 2	B-2
B.2. Measurement 2 Excitation Actuator 2	B-3
B.3. Measurement 3 Excitation Actuator 2	B-4
B.4. Measurement 4 Excitation Actuator 2	B-5
B.5. Measurement 5 Excitation Actuator 2	B-6
B.6. Measurement 6 Excitation Actuator 2	B-7
B.7. Measurement 7 Excitation Actuator 2	B-8
B.8. Measurement 8 Excitation Actuator 2	B-9
C.1. Measurement 1 Excitation Actuator 3	C-2
C.2. Measurement 2 Excitation Actuator 3	C-3
C.3. Measurement 3 Excitation Actuator 3	C-4
C.4. Measurement 4 Excitation Actuator 3	C-5
C.5. Measurement 5 Excitation Actuator 3	C-6
C.6. Measurement 6 Excitation Actuator 3	C-7
C.7. Measurement 7 Excitation Actuator 3	C-8
C.8. Measurement 8 Excitation Actuator 3	C-9
D.1. Measurement 1 Excitation Actuator 4	D-2
D.2. Measurement 2 Excitation Actuator 4	D-3
D.3. Measurement 3 Excitation Actuator 4	D-4

Figure	Page
D.4. Measurement 4 Excitation Actuator 4	D-5
D.5. Measurement 5 Excitation Actuator 4	D-6
D.6. Measurement 6 Excitation Actuator 4	D-7
D.7. Measurement 7 Excitation Actuator 4	D-8
D.8. Measurement 8 Excitation Actuator 4	D-9
E.1. Measurement 1 Excitation Actuator 5	E-2
E.2. Measurement 2 Excitation Actuator 5	E-3
E.3. Measurement 3 Excitation Actuator 5	E-4
E.4. Measurement 4 Excitation Actuator 5	E-5
E.5. Measurement 5 Excitation Actuator 5	E-6
E.6. Measurement 6 Excitation Actuator 5	E-7
E.7. Measurement 7 Excitation Actuator 5	E-8
E.8. Measurement 8 Excitation Actuator 5	E-9
F.1. Measurement 1 Excitation Actuator 6	F-2
F.2. Measurement 2 Excitation Actuator 6	F-3
F.3. Measurement 3 Excitation Actuator 6	F-4
F.4. Measurement 4 Excitation Actuator 6	F-5
F.5. Measurement 5 Excitation Actuator 6	F-6
F.6. Measurement 6 Excitation Actuator 6	F-7
F.7. Measurement 7 Excitation Actuator 6	F-8
F.8. Measurement 8 Excitation Actuator 6	F-9
G.1. Measurement 1 Excitation Actuator 7	G-2
G.2. Measurement 2 Excitation Actuator 7	G-3
G.3. Measurement 3 Excitation Actuator 7	G-4
G.4. Measurement 4 Excitation Actuator 7	G-5
G.5. Measurement 5 Excitation Actuator 7	G-6

Figure	Page
G.6. Measurement 6 Excitation Actuator 7	G-7
G.7. Measurement 7 Excitation Actuator 7	G-8
G.8. Measurement 8 Excitation Actuator 7	G-9
H.1. Measurement 1 Excitation Actuator 8	H-2
H.2. Measurement 2 Excitation Actuator 8	H-3
H.3. Measurement 3 Excitation Actuator 8	H-4
H.4. Measurement 4 Excitation Actuator 8	H-5
H.5. Measurement 5 Excitation Actuator 8	H-6
H.6. Measurement 6 Excitation Actuator 8	H-7
H.7. Measurement 7 Excitation Actuator 8	H-8
H.8. Measurement 8 Excitation Actuator 8	H-9
I.1. Displacement Y1 Force Input F1	I-1
I.2. Displacement Y2 Force Input F1	I-2
I.3. Displacement Y3 Force Input F1	I-2
I.4. Displacement Y1 Force Input F2	I-3
I.5. Displacement Y2 Force Input F2	I-3
I.6. Displacement Y3 Force Input F2	I-4
I.7. Displacement Y1 Force Input F3	I-4
I.8. Displacement Y2 Force Input F3	I-5
I.9. Displacement Y3 Force Input F3	I-5
I.10. Displacement Y1 Force Input F1	I-6
I.11. Displacement Y2 Force Input F1	I-7
I.12. Displacement Y3 Force Input F1	I-7
I.13. Displacement Y1 Force Input F2	I-8
I.14. Displacement Y2 Force Input F2	I-8
I.15. Displacement Y3 Force Input F2	I-9

Figure		Page
I.16.	Displacement Y1 Force Input F3	I-9
I.17.	Displacement Y2 Force Input F3	I-10
I.18.	Displacement Y3 Force Input F3	I-10

List of Tables

Table	Page
5.1. Coulomb Damping Coefficients	5-2
5.2. Truth Model Results	5-5

Abstract

Large, lightweight space structures of the future will require state-of-the-art vibration suppression systems. To design such a system, it is necessary to have a mathematical model that adequately describes the motion of the system. Coulomb damping compensation was utilized to remove a known nonlinearity from the frequency response functions obtained from the Passive and Active Control of Space Structures (PACOSS) Dynamic Test Article (DTA). Via the Eigensystem Realization Algorithm (ERA), the frequency response functions were then used to generate a mathematical representation of the structure's dynamics. Two models were obtained. The first model, without friction compensation, acted as a baseline. The second was obtained using friction compensation. Comparing the models, it was determined that active friction compensation was worthwhile and resulted in a more accurate mathematical description of the system.

A STATE-SPACE MODEL OF A LARGE, LIGHTLY DAMPED SPACE STRUCTURE USING THE EIGENSYSTEM REALIZATION ALGORITHM

I. Introduction

Future military and civilian space structures will be large in size due to design requirements; however, in order to reduce launch costs, these structures must also be lightweight. These characteristics, when combined, lead to a structure likely to experience vibrations. These vibrations could hinder the system's performance and, if left uncontrolled, may lead to failure of the mission. This is especially true of those systems requiring precision pointing, precise shape control, and rapid targeting maneuvers (3:11). These systems will require state-of-the-art, optimal control systems in order to remain useful over their entire life span. To perform an optimal control design, an accurate state space model of the system is necessary.

1.1 Problem and General Approach

The goal of this research is to develop an accurate state space model of the Passive and Active Control of Space Structures (PACOSS) Dynamic Test Article (DTA), a testbed for large flexible space structures. The DTA, like actual large space structures, possesses high modal density at low frequencies (3:15,17). This leads to a need for a system identification technique capable of identifying multiple, closely spaced, low frequency modes. In this research effort, the Eigensystem Realization Algorithm (ERA) was used to develop the state-space model (2:1564). This identification algorithm works best with low-noise or perfect data.

In order to obtain the most accurate state-space (linear) model, known nonlinearities should be eliminated from the system to the maximum extent possible. Coulomb damping, a nonlinear phenomenon, is known to exist in the reaction mass actuators. A secondary goal of this research is to calculate the Coulomb damping force in each reaction mass actuator and to

remove the effects of Coulomb damping from the resulting frequency response functions. The Coulomb damping coefficients will be calculated using free decay (14:125). After determining the damping coefficients, a nonlinear compensator will be implemented to remove the effects of Coulomb damping from the frequency response functions, which will be used to generate the state-space representation of the PACOSS DTA.

1.2 The PACOSS Program

Martin Marietta Astronautics Group, during the Passive and Active Control of Space Structures program, studied active control and passive damping techniques for use on large space structures. The ultimate goal of the PACOSS program was to demonstrate the benefits of passive damping working in concert with active vibration control on large space systems.

Based upon planned and conceptual civilian and military space systems, a Representative System Article (RSA) was designed and configured such that vibration control design methodology could be better understood. The RSA was not designed to be mission specific; instead, it was a representation of several missions and requirements in a single system. The RSA was composed of seven substructures: a ring truss; a box truss; two solar panels; an equipment tripod; and antenna dish; and an equipment platform. To help ensure dynamic similarity to future large space systems, the size of each substructure was modeled after planned or actual hardware for future spacecraft. This guaranteed that the damping mechanisms developed under the PACOSS program were applicable to many of the future large space structures (7:33,38).

A scaled version of the Representative System Article was constructed. The Dynamic Test Article was designed to have characteristics similar to the RSA, making it representative of future space systems. Thus, damping techniques developed and tested on the DTA could be used in space systems with confidence (7:1). The test article included the same substructures found in the RSA as well as large amounts of passive damping.

After extensive study of the structure, Martin Marietta concluded that a combination of passive damping and active control mechanisms would benefit future military and civilian

space systems. The Astronautics Group did report, however, that improvements were needed in the area of system identification of damped structures with dense modal spectra (7:138,141).

Upon completion of the PACOSS contract, the DTA and its supporting hardware were moved to the Air Force Institute of Technology (AFIT) for further study. After delivery, the test article was modified by removing the box truss, the antenna, the equipment platform, and all of the passive damping mechanisms. Due to logistical constraints, the box truss was replaced by three mass simulators. Having the same dimensions as the originals, two new solar arrays were constructed without any passive damping treatments. These modifications were made to the testbed to approximate a lightly damped, large space structure where purely active control techniques could be developed and investigated for vibration suppression (4:2). Active control techniques are being investigated because of the added launch costs that are incurred when passive damping mechanisms are utilized. This, of course, is due to the added weight of the passive mechanisms.

Since its arrival at the Air Force Institute of Technology, the PACOSS DTA has been the object of three research efforts. The first effort at AFIT was conducted by Capt. Scott George (4). Capt. George conducted an extensive modal survey of the test article using finite element analysis. Lt. Chad Matheson's research involved applying purely active feedback control to attenuate vibrations in the test article (6). Lt. Matheson's research also involved the creation of the first mathematical model of the flexible body's dynamics since its modification. Capt. Tony Nash's research effort was to develop a state space model for the test article using time domain techniques (8).

II. Experimental Equipment

The PACOSS Dynamic Test Article and its supporting hardware now reside at the Air Force Institute of Technology. The support equipment delivered by Martin Marietta includes the following: the air-bearing suspension system, the motor control units, the Optima/3 array processor and a Sun3/50 computer. Additional equipment needed to continue research on the lightly damped structure was obtained by the Air Force. This chapter describes the DTA and its equipment as well as their use in this research effort.

2.1 Description of the DTA

In its present state, the Dynamic Test Article (Fig. 2.1) is composed of four substructures: the ring truss, two solar arrays, and an equipment tripod. In order to control the structure using active measures, eight mass actuators are also mounted on the structure. In addition, three mass simulators are attached to the structure beneath the ring truss. These mass simulators replaced the original DTA box truss, which could not be relocated to AFIT (3:15).

2.1.1 Ring Truss. The first substructure to be designed, the ring truss is the structural backbone of the DTA. The circular truss, comprised of aluminum tubes bonded between aluminum joint blocks, is designed to withstand the gravitational load of the testbed (7:41). Six reaction mass actuators and three mass simulators are mounted on the ring truss, simulating the presence of on-board systems. The actuators and mass simulators were mounted to preserve the axis of symmetry of the structure (6:7).

2.1.2 Tripod. The DTA tripod consists of a secondary steel plate, three rectangular aluminum legs, and mounting hardware to connect the tripod to the ring truss. Designed to have low frequency flexible modes, the tripod's design was based on a Cassegrain system with the tripod supporting the secondary reflecting surface (9:10). Mounted on the secondary steel plate are two reaction mass actuators. The actuators' axes are perpendicular to each other and

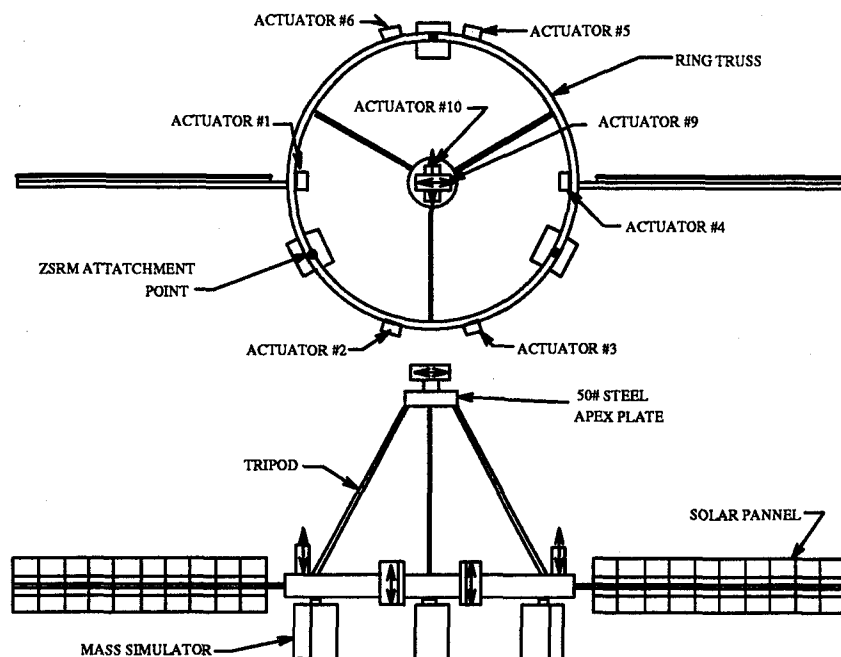


Figure 2.1 DTA Configuration (6)

parallel to the ground. The legs of the tripod are connected to the ring truss by aluminum mounting plates.

2.1.3 Solar Arrays. Attached to the ring truss, the solar arrays, designed to simulate the behavior of actual solar arrays, are sized to have the ability to provide the necessary on-orbit power required by the Cassegrain optical system. Each solar array is comprised of a gridwork of thin aluminum strips attached to a 15 ft. aluminum mast. The solar arrays were designed to simulate the low frequency behavior of actual solar arrays as those modeled for the RSA. The solar arrays are mounted on the ring truss using an aluminum block assembly (9:61).

2.1.4 Suspension System. The DTA is supported by three CSA Engineering Zero Spring Rate Mechanism (ZSRM) supports. Figure 2.2 shows the relationship between the suspension system and the DTA testbed. The ZSRMs, which are attached to the DTA support frame, are able to support the testbed by supplying air pressure beneath a piston to which

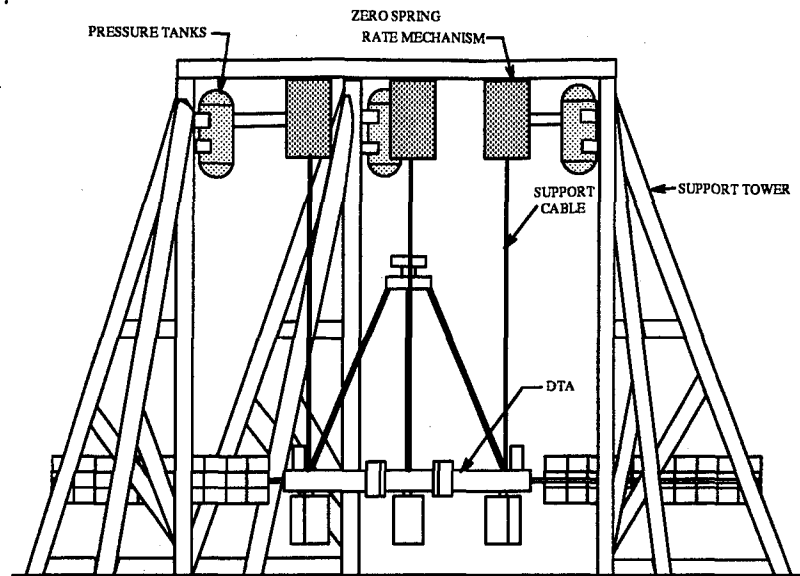


Figure 2.2 DTA and Support Structure (6)

the support cables are attached. The pistons float on air bearings resulting in a soft, nearly frictionless suspension (4:30). Air tanks attached to the support frame provide air for each piston (3:21). The support cables are attached to the ring truss in three locations positioned in order to maintain the axis of symmetry of the testbed. Thus, the testbed is isolated from any external disturbances.

2.2 Active Vibration Control System

An active vibration suppression system is incorporated into the design of the DTA. The suppression system consists of eight actuator/sensor units which are used to control flexible body modes. Each actuator/sensor unit, shown in Fig. 2.3, is comprised of a reaction mass actuator (RMA) and a Sundstrand QA-1400 accelerometer. Each unit produces two sensor signals: (1) inertial acceleration of the actuator frame at the mounting location and (2) the relative velocity between the reaction mass and the actuator frame. The positioning of the actuator/sensor units is shown in Fig. 2.1.

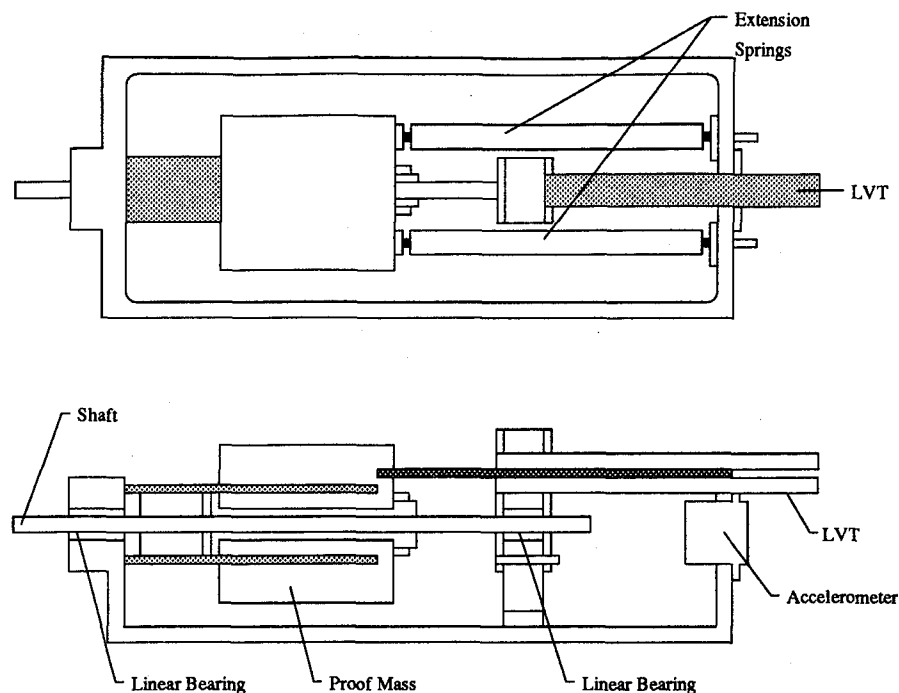


Figure 2.3 Actuator/Sensor Unit

2.2.1 Reaction Mass Actuator. The RMA is comprised of a motor coil, a linear velocity transducer (LVT), a reaction (or proof) mass, extension springs, and a housing. For the six actuators mounted on the ring truss, two extension springs are mounted to both the reaction mass and the assembly housing to provide for gravity off-load for the mass. The actuator has a natural frequency of 1.5-Hz and a stroke length of approximately ± 1 in. (9:90). The two remaining actuators, mounted horizontally on the secondary steel plate, require four extension springs. The RMA, which is attached to the DTA via the housing, measures the relative velocity between the motor mass and actuator housing. Power is supplied to the actuators via the PACOSS Motor Control Units (3:34-38).

2.2.2 Accelerometer. A Sundstrand QA-1400 accelerometer is attached to each actuator housing to measure the inertial acceleration of the test article at that location. If control algorithms were to be implemented, these inertial acceleration measurements would be integrated to obtain the inertial velocity (3:34). The QA-1400 has a bandwidth of at least

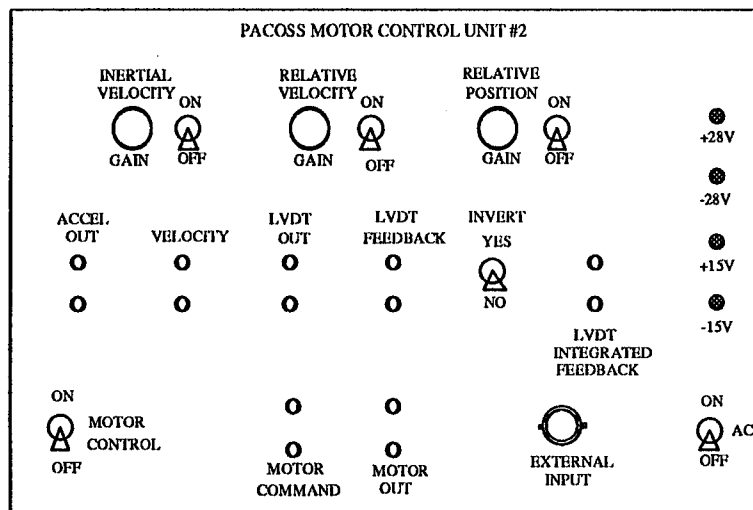


Figure 2.4 Motor Control Unit: Front Panel (6)

100 Hz; therefore, the accelerometer measurement may be treated as a perfect measurement for system identification purposes. Signal processing (i.e., integration of the measurement), if desired, can be supplied by the PACOSS Motor Control Units (9:90).

2.2.3 Motor Control Unit. The Motor Control Unit, shown in Fig. 2.4, provides all of the signal processing and power for an actuator/sensor unit. There are a total of eight control units—one for each actuator/sensor. Each unit is comprised of a linear motor power amplifier, an accelerometer pre-amp, an integrator, and an LVT amplifier, and adjustable potentiometers. Adjustable gain controls and measurement output ports are included on the front panel of the instrument cabinet (9:90,93). The velocity output port on each unit was connected to an analyzer to generate frequency and time response data.

2.3 Optima/3

Built by Systolic Systems Inc., the Optima/3 Real-Time Data Acquisition and Control Processor is a high-speed parallel processor that is designed for data acquisition, estimation and control applications. Comprised of a Sun 3 master processor, eight Systolic parallel

processors, and a data acquisition system, the Optima has 16 input channels and 16 output channels. The Optima's main purpose in this research effort was friction compensation. The sampling rate, signal range, and input gain for each channel were programmed specifically for use in identifying a large, flexible space structure (10:2-1). The Optima/3 has valid sampling rates ranging from 2 to 20,000 Hz with a 2 Hz resolution. This corresponds to sampling periods of 0.05 to 500 msec (10:3-4).

Friction compensation was accomplished using software written in C on the Sun 3/50 host. The LABWARETM download utility was able to download the compiled C code to the Optima/3. The Optima then executes the friction compensation program (10:3-1).

2.4 Additional Hardware and Software

A Tektronix 2642A Personal Fourier Analyzer was used to collect the data and to create the frequency response functions. The 2642A analyzer is a high performance analog data acquisition and signal processing peripheral for use with personal computers. The analyzer processes data at bandwidths up to 200 kHz, with a sample rate 2.56 times the bandwidth. The analyzer, however, can only acquire and process four channels of data (12:1-1). The requirements for this research effort was a minimum of 8 input channels. Therefore, a Tektronix SI 5010 Programmable Scanner was used to increase the number of available input channels from four to sixteen.

There were two software packages that were used to collect and process the data. To collect the data, the FSCAN utility (11) and the Instrument Program (12) from Tektronix were used in conjunction with the Tektronix 2642A Fourier Analyzer. The FSCAN utility, running on a personal computer, interfaces with the SI 5010 scanner and the Instrument Program (IP) to allow data to be taken in a shorter amount of time. The other software package, MATLABTM from MathWorks, Inc., was used in the analysis of the data (13). The ERA algorithm was coded in MATLABTM and was used to generate the state-space model for the PACOSS DTA.

III. Theory

The material presented in this chapter represents the theoretical basis for the development of an accurate state-space model which can be used to design a vibration suppression system. First, the method for determining the amount of Coulomb damping present in the reaction mass actuators will be developed. Knowing the amount of Coulomb damping allows for the removal of this nonlinearity from the frequency response functions. The next section explains the Eigensystem Realization Algorithm which will be used to generate the state-space model from the frequency response functions.

3.1 Coulomb Damping

The reaction mass actuator, acting as the control mechanism for the testbed, is the only moving part on the DTA. Like every other mechanical system, friction is also present in this mechanism. In this case, Coulomb (dry friction) damping is the major friction force at work. Coulomb damping cannot be physically removed from the system. However, by employing a nonlinear compensator, the effects of Coulomb damping can be removed from the frequency response functions.

The RMA, which consists of two extension springs, a reaction mass, and an LVT, is modeled as a single-degree-of-freedom harmonic oscillator. Figure 3.1 shows a schematic of the RMA. The Coulomb damping force, F_C , always opposes the motion of the mass. Therefore, if the velocity of the mass is negative, then the friction is positive and vice versa. The equation of motion for the RMA is

$$m\ddot{x} + F_C \operatorname{sgn}(\dot{x}) + kx = 0 \quad (3.1)$$

where

$$\operatorname{sgn}(\dot{x}) = \frac{\dot{x}}{|\dot{x}|}$$

accounts for the sign change. The initial conditions are $x(0) = x_0$ and $\dot{x}(0) = 0$.

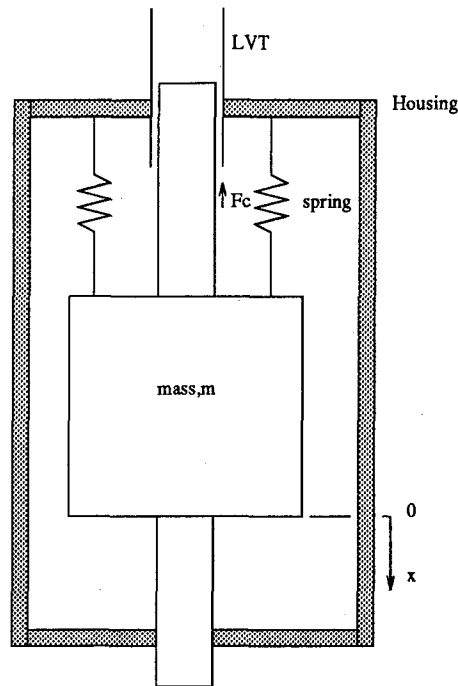


Figure 3.1 Functional Diagram of Reaction Mass Actuator

The piecewise linear solution to Eq. (3.1) is

$$x(t) = A \cos \omega_n t + B \sin \omega_n t - F_C \operatorname{sgn}(\dot{x})/k \quad (3.2)$$

where $\omega_n^2 = k/m$ and A and B are constants. Equation 3.2 is only valid between velocity sign changes. Applying the initial conditions, the solution becomes

$$x(t) = (F_C \operatorname{sgn}(\dot{x})/k + x_0) \cos \omega_n t - F_C \operatorname{sgn}(\dot{x})/k \quad (3.3)$$

The output of the LVT, which is the response that will be used to determine the friction force, F_C , is the velocity of the proof mass relative to the actuator housing. Therefore, the velocity of the proof mass (\dot{x}) is most important. Taking the derivative of $x(t)$, the velocity function is

$$\dot{x}(t) = -(F_C \operatorname{sgn}(\dot{x})/k + x_0) \omega_n \sin \omega_n t \quad (3.4)$$

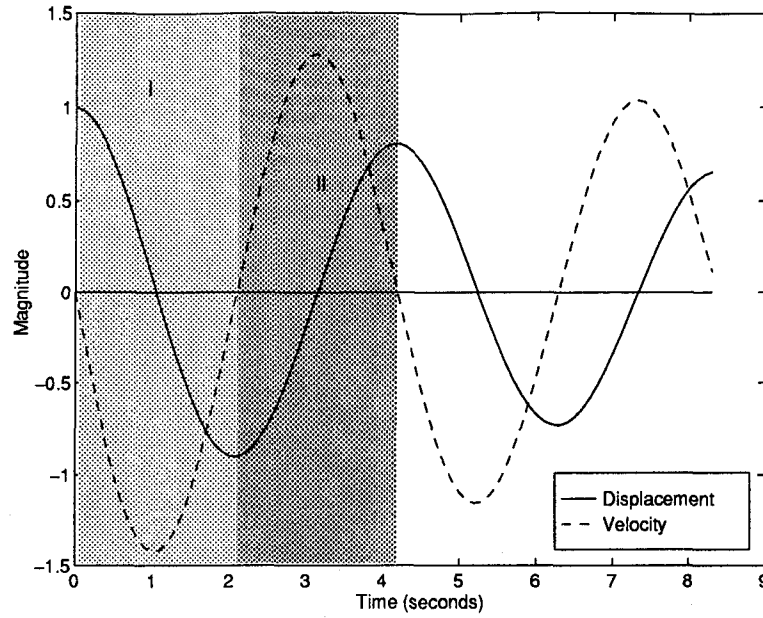


Figure 3.2 Free Decay Response Function

Shown in Fig. 3.2 is a representative free decay response for both displacement and velocity of the actuator. To solve for the Coulomb damping force present in the mechanism, piecewise techniques will be used. The two regions highlighted in Fig. 3.2 will be considered. The first region is where \dot{x} is negative; the second, where \dot{x} is positive.

3.1.1 Case I: $\dot{x} < 0$. Throughout Region I (with the exception of the start and end times), the velocity is negative. Substituting this condition into Eqs. (3.3) and (3.4) yields

$$x_I(t) = (x_0 - F_C/k) \cos \omega_n t + F_C/k \quad (3.5a)$$

$$\dot{x}_I(t) = -(x_0 - F_C/k) \omega_n \sin \omega_n t \quad (3.5b)$$

Now, $x_I(t_1)$ and $\dot{x}_I(t_1)$ must be calculated, where the subscript I denotes the responses in Region I, and t_1 is the time in Region I where $\dot{x} = 0$. From Eq. (3.5b), for \dot{x} to equal 0 then

$$\sin \omega_n t = 0$$

Thus, $t_1 = \pi / \omega_n$. Substituting $t = t_1$ in Eqs. (3.5a) and (3.5b) yields

$$x_I(t_1) = 2 F_C / k - x_0 \quad (3.6a)$$

$$\dot{x}_I(t_1) = -(x_0 - F_C / k) \omega_n \sin \omega_n t_1 = 0 \quad (3.6b)$$

Referencing Fig. 3.2, t_1 is the end time for Region I and the start time for Region II; thus, $x_I(t_1)$ and $\dot{x}_I(t_1)$ become the initial conditions for Region II.

3.1.2 Case II: $\dot{x} > 0$. In Region II, with the exception of the start and end times once again, the velocity is positive. From above, the initial conditions for this region are

$$x_{II}(t_1) = 2 F_C / k - x_0$$

and

$$\dot{x}_{II}(t_1) = 0$$

Using trigonometric formulas and the condition $\dot{x} > 0$, the solutions in this region can be rewritten as

$$x_{II}(t) = A \sin \omega_n(t - t_1) + B \cos \omega_n(t - t_1) - F_C / k \quad (3.7a)$$

$$\dot{x}_{II}(t) = A \omega_n \cos \omega_n(t - t_1) - B \omega_n \sin \omega_n(t - t_1) \quad (3.7b)$$

Applying the initial conditions for Region II, Eq. (3.7) becomes, for $t_1 < t < t_2$,

$$x_{II}(t) = (3 F_C / k - x_0) \cos \omega_n(t - t_1) - F_C / k \quad (3.8a)$$

$$\dot{x}_{II}(t) = -(3 F_C / k - x_0) \omega_n \sin \omega_n(t - t_1) \quad (3.8b)$$

Next, t_2 must be determined. t_2 is the next zero crossing of the velocity function in the region. Again referencing Fig. 3.2, t_2 is found to be

$$t_2 = t_1 + \frac{\pi}{\omega_n}$$

Evaluating Eq. (3.8) at $t = t_2$ yields

$$x_{II}(t_2) = x_0 - 4 F_C / k \quad (3.9a)$$

$$\dot{x}_{II}(t_2) = -(3 F_C / k - x_0) \omega_n \sin \omega_n \left(\frac{\pi}{\omega_n} \right) = 0 \quad (3.9b)$$

3.1.3 Determining the Coulomb Damping Coefficient. Using the peak amplitudes of Eqs. (3.6b) and (3.9b), the decrease in amplitude from $t = t_1$ to $t = t_2$ is calculated. The decrease over every half cycle is found to be

$$(x_0 - F_C / k) \omega_n - (x_0 - 3 F_C / k) \omega_n = (2 F_C / k) \omega_n \quad (3.10)$$

Thus, the amplitude decays by $(4 F_C / k) \omega_n$ every cycle.

Recalling that $\omega_n = 2\pi f_n$ where f_n is the natural frequency in Hz, the decrease in amplitude per cycle, Δy , can be written as

$$\Delta y = \frac{8 F_C \pi}{k} f_n$$

Rearranging, the Coulomb damping force can be determined by

$$F_C = \frac{\Delta y k}{8 \pi f_n} \quad (3.11)$$

In all probability, more than one cycle will be used to determine the coefficient; thus, the more useful formula is

$$F_C = \frac{\Delta y k}{8 \pi f_n (\text{no. of cycles})} \quad (3.12)$$

where Δy is redefined as the decrease in amplitude over the number of cycles to be used in determining the Coulomb damping coefficient.

Using Eq. (3.12), the amount of Coulomb damping in the actuator can be calculated. This is of importance since the friction force can be mathematically removed from the frequency response functions. In removing this nonlinearity from the frequency response functions, it is hoped that a more accurate state-space representation of the testbed will be generated.

3.2 System Identification

The main goal of this thesis is to obtain an accurate mathematical representation of the PACOSS DTA from frequency response functions. The basic problem posed is one of system identification: Given measurement functions $y(k)$, construct a state-space realization such that the functions y are reproduced by the state-variable equations (5:621).

ERA is based upon a singular value decomposition of the block Hankel matrix, which is composed of impulse response data. This modal synthesis technique is capable of identifying the system if there is little or no noise found in the measurements (2:1564).

Consider the discrete-time linear system:

$$x(k+1) = Ax(k) + Bu(k) \quad (3.13a)$$

$$y(k) = Cx(k) + Du(k) \quad (3.13b)$$

where x is an n -dimensional state vector, u an p -dimensional input vector, and y a q -dimensional output vector. The A , B , C , and D matrices have dimensions $(n \times n)$, $(n \times p)$, $(q \times n)$, and $(q \times p)$, respectively. The integer k is the sample indicator. The A matrix represents the dynamics of the system. For flexible structures, it is a representation of the mass, stiffness, and damping properties (5:621). For the system in Eq. (3.13) with unit impulse response, the

time-domain description is given by the function known as the Markov Parameters ¹:

$$Y(k) = CA^{k-1}B \quad (3.14)$$

The ERA technique first forms a block Hankel matrix. As mentioned before, this matrix is composed of the sampled unit impulse response:

$$H(k-1) = \begin{bmatrix} Y(k) & \cdots & Y(k+m_{s-1}) \\ \vdots & \ddots & \vdots \\ Y(k+l_{r-1}) & \cdots & Y(k+l_{r-1}+m_{s-1}) \end{bmatrix} \quad (3.15)$$

where r and s are arbitrary integers satisfying the inequalities $rq \geq n$ and $sp \geq n$, and l_i ($i = 1, 2, \dots, r-1$) and m_j ($j = 1, 2, \dots, s-1$) are arbitrary integers. The singular value decomposition of the $r \times s$ block Hankel matrix is expressed as:

$$H(0) = PDQ^T \quad (3.16)$$

Using the decomposition of the Hankel matrix, the state-space realization, which is discrete-time and reduced-order, is computed:

$$A = D_n^{-1/2} P_n^T H(1) Q_n D_n^{-1/2} \quad (3.17a)$$

$$B = D_n^{1/2} Q_n^T E_p \quad (3.17b)$$

$$C = E_q^T P_n D_n^{1/2} \quad (3.17c)$$

$$D = Y(0) \quad (3.17d)$$

The matrices P_n and Q_n are composed of the first n columns of P and Q from the decomposition. D_n is a diagonal matrix of n non-zero singular values. E_p^T is defined as $[I_p, 0]$ and E_q^T

¹In literature the discrete-impulse response function sequence is commonly referred to as the Markov Parameters.

is defined as $[I_q, 0]$, where I_p and I_q are identity matrices of the order p and q , respectively, and 0 is the zero matrix (2:1565).

For a more in-depth look at the Eigensystem Realization Algorithm, see the original derivation by Juang and Pappa (5:620-627).

IV. Experimental Procedure

The following sections detail the procedures used to obtain and process the data in this research effort. Procedures used to identify the Coulomb damping coefficients and to implement the compensation routine will be discussed. The procedures that were used to acquire the frequency response data, compensated and non-compensated, also will be presented. The method used to validate the ERA routines will also be discussed.

4.1 DTA and Actuator Preparation

In 1992, Capt Scott George completed a finite element analysis of the DTA. To obtain the finite element model of the DTA, 151 accelerometers and the associated instrumentation wiring were connected to the DTA testbed (4:73). The accelerometers and wires remained attached to the structure throughout the next two research efforts. Due to the weight and stiffness of the wires, damping and stiffness were being added to the frequency response data. All of the accelerometers and the instrumentation wiring were removed from the structure, since they were no longer needed. Thus, the damping and stiffness that resulted from the wires was removed from the frequency response functions.

Maintenance was also performed on the eight mass actuators. Due to the fact that the masses had been suspended by the same springs since at least late 1991, the extension springs on the eight RMAs were replaced. Upon replacement, the net spring constant was calculated via bench testing. The net spring constant, k , was found to be equal to 1.0 lb/in. This verified the results obtained by Martin Marietta (9:90). The actuator's shaft, linear velocity transducer, and linear bearings were cleaned and lubricated with a silicone-based lubricant to eliminate friction in the system. This resulted in improved results when free decay tests were accomplished to determine the Coulomb damping coefficients.

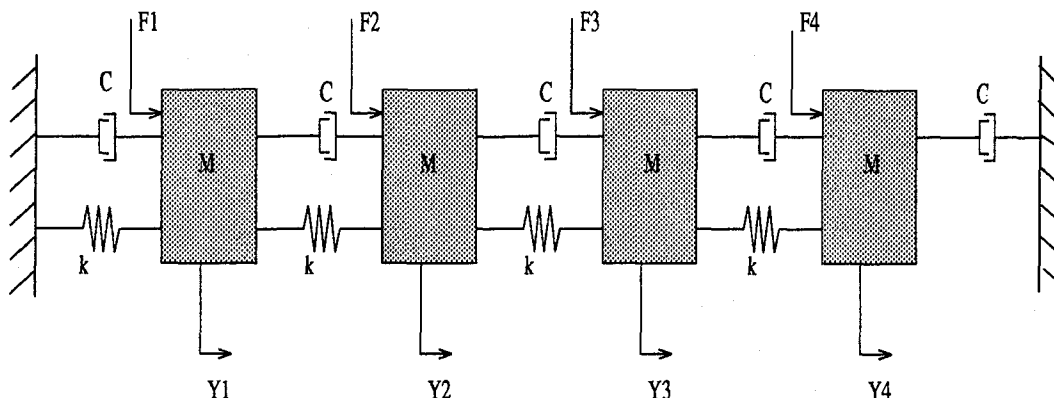


Figure 4.1 Truth Model Diagram

4.2 Validation of ERA Routines

The ERA routines which performed the system identification were written in MATLABTM by Capt. Richard Cobb, a Ph.D. candidate at the Air Force Institute of Technology (1). Several of the routines were modified for use in this thesis. Before using these routines to identify the PACOSS DTA, a smaller scale identification was used to validate the usefulness of the code.

The spring-mass-damper system pictured in Fig. 4.1 was used as the truth model for the validation of the ERA routines. This system was modeled after the system simulator box built by Tektronix for use with the Instrument Program tutorial (12:1-9). The system is composed of four equal masses (1×10^{-3} slugs), interconnected with four springs and five dampers. The four springs have the same spring constant (1×10^6 lb/in) and the dampers have the same damping coefficient (2 lb-sec/in). There are four *force* inputs (F1 through F4) and four *displacement* outputs (Y1 through Y4) in the system.

Three inputs (F1, F2, and F3) and three outputs (Y1, Y2, and Y3) were used to obtain a 3×3 MIMO system. Two cases were used: (1) the truth model without noise and (2) the truth model with a 15% measurement noise. Noise was added to the system to ascertain the accuracy of the algorithm without perfect measurements. The noise was added directly to the FRFs. This is not the case, however, with the PACOSS DTA, where noise is continually

introduced into the system. The rms value of the added noise was obtained by taking 15% of the rms value of the frequency response function. This noise was then added to the FRFs.

Frequency response functions for the truth model system were generated using MATLABTM. The equations of motion and a state-space representation of the system were derived from Fig. 4.1. The state-space representation was discretized and the frequency response functions were calculated using *mkfrf.m*. The 15% noise was added using *addnoise.m* (1).

The ERA routines were then employed to find a state-space realization for the truth model, with and without noise. The parameter settings for the system identification can be found in Appendix J. The undamped natural frequency and damping ratio of the poles of the ERA fit for both systems will be compared with the poles of the truth model. This comparison will be used as the metric to determine the accuracy of the ERA fits.

4.3 Coulomb Damping Identification

As mentioned in Chapter III, free decay was used to determine the amount of Coulomb damping in the eight reaction mass actuators, which provide the disturbances to generate the frequency response functions (14:125). In free decay, the mass is displaced a fixed distance and released. The spring-mass system, shown in Fig. 3.1, will oscillate until the total energy left in the system can no longer overcome gravity and the friction forces present. The data obtained from this method consists of a voltage proportional to the velocity of the proof mass relative to the actuator housing. The voltages, obtained from the LVT OUT port on the motor control unit, were stored by the IP program and converted to ASCII format.

A MATLABTM routine, which is referred to as a *m-file*, was written to analyze the time response data. The *m-file* (shown in Appendix L) plots the data and calculates the frequency, the number of cycles, and the change in amplitude during those cycles. The routine uses Eq. (3.12) and the free decay data to calculate the Coulomb damping coefficient. The net spring constant, k , is equal to 1 lb/in (9:90).

For each actuator, four sets of data were taken. Coulomb damping coefficients were calculated from each set of data. The four coefficients obtained for each actuator were then

averaged to find an approximation of the Coulomb damping coefficient for that actuator. To determine the accuracy of the Coulomb damping coefficients, a Coulomb damping compensator, which was programmed in C code and downloaded to the Optima/3, was used. If the coefficients used are accurate, the compensation force applied should not induced any oscillatory motion since the force applied should be the exact amount to neutralize the Coulomb damping force. In a test to determine the accuracy of the calculated coefficients, the compensation force was the only force applied. The compensation force caused each mass to oscillate; thus, thus, the calculated coefficients were too large and, if used, would result in an unstable system. A method was then devised to adjust the Coulomb damping coefficient for each actuator.

Using the compensator, Fourier analyzer and IP program, free decay was employed to adjust the damping coefficients. When the correct coefficient was used, there was no decrease in the amplitude of the time response function. Viewing the free decay response via the IP program, the coefficient could be adjusted either up or down to provide the desired time response. The C code used to adjust these coefficients can be found in Appendix K. The adjusted coefficients were then programmed into the Coulomb damping compensator and used to remove the effects of Coulomb damping from the frequency response functions.

4.4 *Coulomb Damping Compensation*

Coulomb damping compensation was accomplished by using the Optima/3. A nonlinear compensator was programmed in C code to mathematically remove the effects of Coulomb damping from the frequency response functions.

The *couldamp.c* program can be found in Appendix K. The compensator sampled eight relative velocities for the eight mass actuators 200 times per second. The velocities, which were output by the LVTs and read in by the Optima as voltages, were converted to inches per second. The sign, as well as the magnitude, of the velocity was checked. This is due to the *signum* function in Eq. (3.4). However, a "dead zone" was created at ± 0.2 inches per second. The dead zone was created because the corresponding voltages for velocities below

0.2 inches/second are the same order of magnitude as noise found in the system. Without this dead zone, the compensator would apply Coulomb damping compensation when not needed. It should also be mentioned that 85% of the adjusted Coulomb damping coefficient for each actuator was used. This was done so that the actuators were not overcompensated, which would yield an unstable system.

If the mass' relative velocity was greater than 0.2 inches per second, then a positive friction compensation was applied by sending a voltage proportional to the friction coefficient. For a negative velocity, a negative compensation voltage is applied. The Optima added the compensation voltage to the random noise voltage that was output by the Tektronix 2642A analyzer. This sum was then output to the motor control unit of the excitation actuator.

4.5 Data Acquisition

4.5.1 Frequency Response Functions. The state-space representation of the DTA was identified based only on experimental measurements of the open loop system. Figure 4.2 shows the equipment configuration used to obtain the frequency response functions (FRFs).

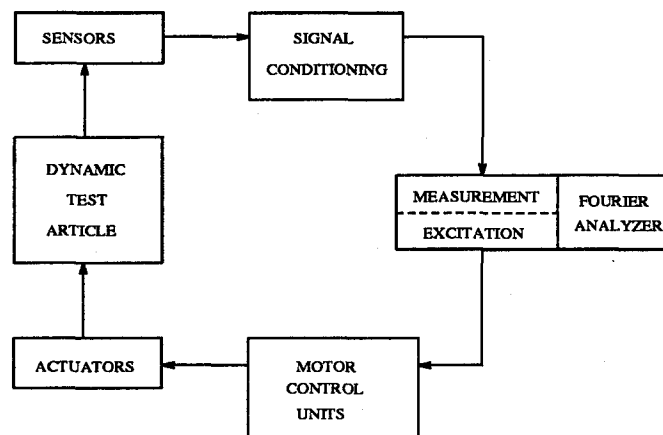


Figure 4.2 Equipment Configuration for Open Loop System

The FRFs were generated by the 2642A Fourier analyzer. The analyzer, outputting a 600 mV random noise with a 20 Hz bandwidth, was used to excite an actuator. The analyzer

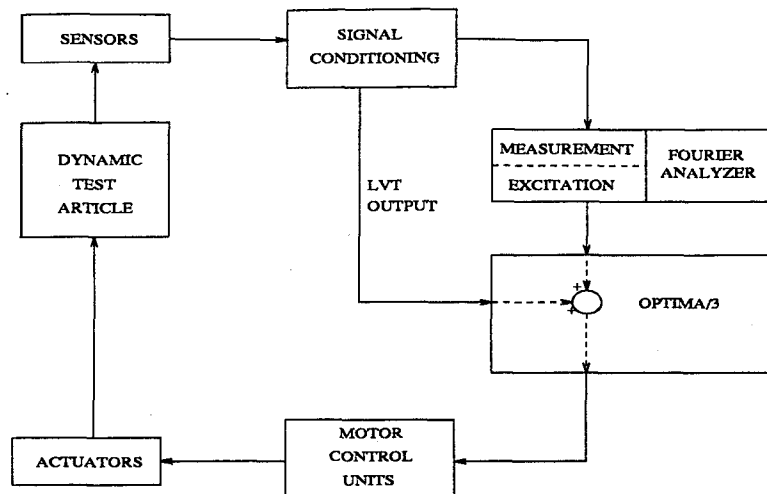


Figure 4.3 Equipment Configuration with Active Friction Compensation

and IP program, in conjunction with the SI 5010 scanner, obtained the inertial acceleration readings from each accelerometer. The analyzer used this data to calculate the corresponding FRFs. This was done seven more times to obtain a complete set of 64 FRFs. The accelerometer readings were sampled 2048 times per frame with a 20 Hz base bandwidth using a Hanning windowing type with no overlap. Hanning windowing, which is recommended for structural vibration data, suppresses leakage of the data. These settings give a sampling rate of 51.2 samples per second and a frequency resolution of 0.025 Hz (12:A-16).

4.5.2 Frequency Response Functions with Friction Compensation. As mentioned earlier, the Coulomb friction force in the actuators introduces nonlinearities into the system. Therefore, active friction compensation was employed to remove the effect of this friction force from the FRFs. Figure 4.3 illustrates the equipment configuration used to obtain this set of frequency response data.

The IP program, scanner, and analyzer settings for this setup were virtually identical. The only difference was that a 300 mV random noise was output from the analyzer to act as a disturbance. The output voltage level was decreased to ensure that, with active friction compensation, the actuator did not saturate and hit the stops.

V. Results

The following sections describe the results of the procedures previously discussed in Chapter IV. The discussion includes the calculated and adjusted Coulomb damping coefficients, comparison of truth model FRFs and the corresponding ERA fit, results of the ERA algorithm when applied to the FRFs obtained from the PACOSS DTA, and results of a comparison between FRFs with and without active friction compensation as well as FRFs obtained by Lt. Matheson in 1992. It should be noted that the measured data was acceleration frequency response functions. One integration was performed on the data to obtain velocity frequency response functions. The results shown in this chapter are the ERA fits to the velocity FRFs.

Throughout this chapter, the terms *collocated* and *non-collocated* are used. *Collocated* is used to describe frequency response functions when the measurement is obtained from the same point where the excitation force is applied. For example, a frequency response function for Measurement 1 Excitation Actuator 1 is a *collocated* FRF since the measurement was taken at Actuator 1 and the force was also applied there. *Non-collocated* is used to describe FRFs when the measurement is obtained at a different location than where the excitation force was applied. An example of a *non-collocated* FRF is Measurement 1 Excitation Actuator 2.

5.1 Coulomb Damping Coefficients

Table 5.1 shows the Coulomb damping coefficients that were derived experimentally and the adjusted coefficients, which were used in the compensation program. During the course of the research, it was found that these coefficients change depending on the temperature and humidity.

As shown in Table 5.1, there is a big discrepancy in the coefficients for Actuators 9 and 10. This is explained by the fact that these two actuators are mounted horizontally. The linear bearings used in the eight actuators are not designed for horizontal use. Thus, when free decay was used, only one cycle occurred before the motion died out resulting in a rather

Table 5.1 Coulomb Damping Coefficients

Actuator No.	Calculated Coefficient (lb _f)	Adjusted Coefficient (lb _f)
1	0.0591	0.0325
2	0.0565	0.0500
3	0.0549	0.0250
4	0.0353	0.0100
5	0.0574	0.0150
6	0.0518	0.0500
9	0.2824	0.0325
10	0.3188	0.0175

large coefficient. For this reason, the actuators were adjusted. The more accurate, adjusted coefficients were used in the Coulomb damping compensator.

5.2 ERA Fit of the Truth Model

As mentioned in Chapter IV, a smaller MIMO system, the truth model shown in Fig. 4.1, was identified first to validate the ERA routines that were used in the identification of the PACOSS DTA. A 3 x 3 MIMO system utilizing *force* inputs F1 through F3 and *displacement* outputs Y1 through Y3, with and without noise, was used as the validation model.

5.2.1 Truth Model without Noise. Shown in Fig. 5.1 and 5.2 are two FRFs and their corresponding ERA fits. Figure 5.1, the transfer function and the corresponding fit for displacement *Y1* and force input *F2* is representative of the remaining non-collocated output measurement and force input pairs. Figure 5.2 shows the collocated transfer function and its corresponding fit for displacement *Y2* and force input *F2*. The ERA fit for the collocated transfer function shown in Fig. 5.2 is also representative of the remaining collocated transfer functions. As shown in Table 5.2, the algorithm, in the case with no noise, is able to accurately identify the system.

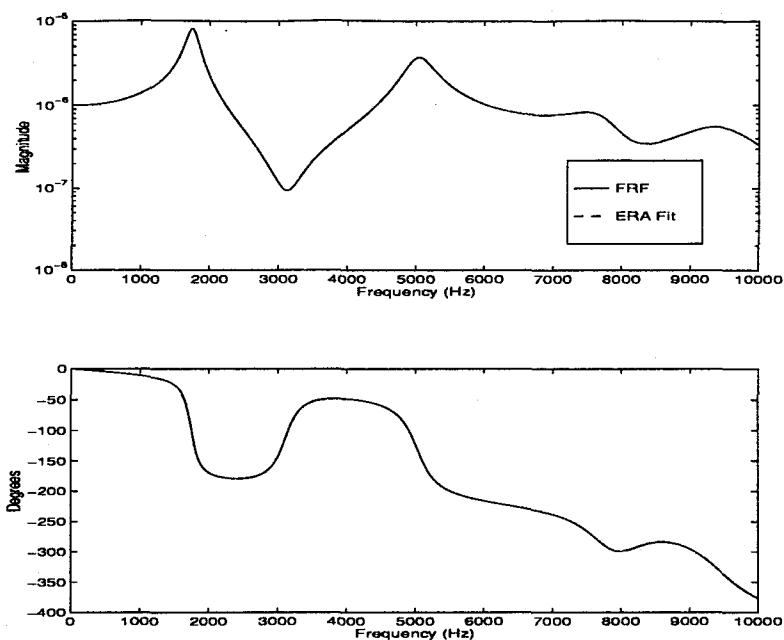


Figure 5.1 Displacement Y1 Force Input F2 for the Truth Model with No Noise

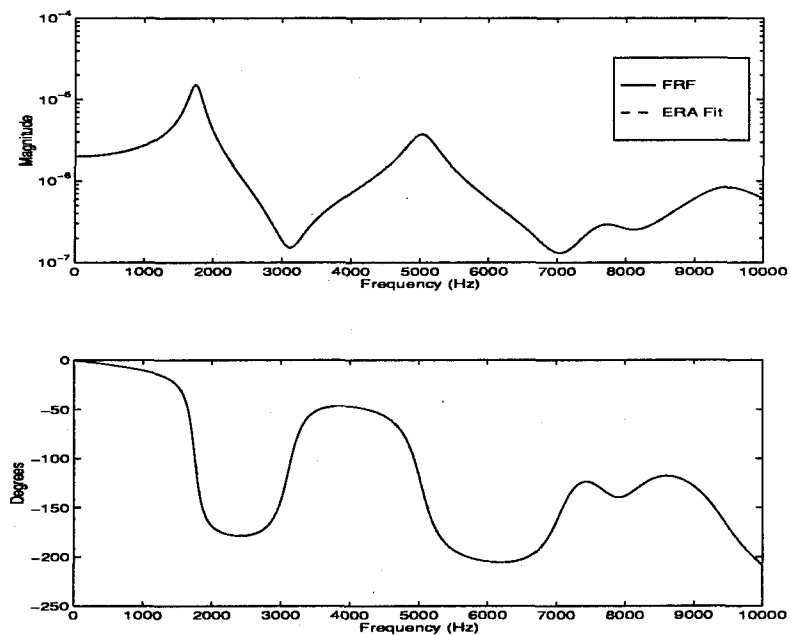


Figure 5.2 Displacement Y2 Force Input F2 for the Truth Model with No Noise

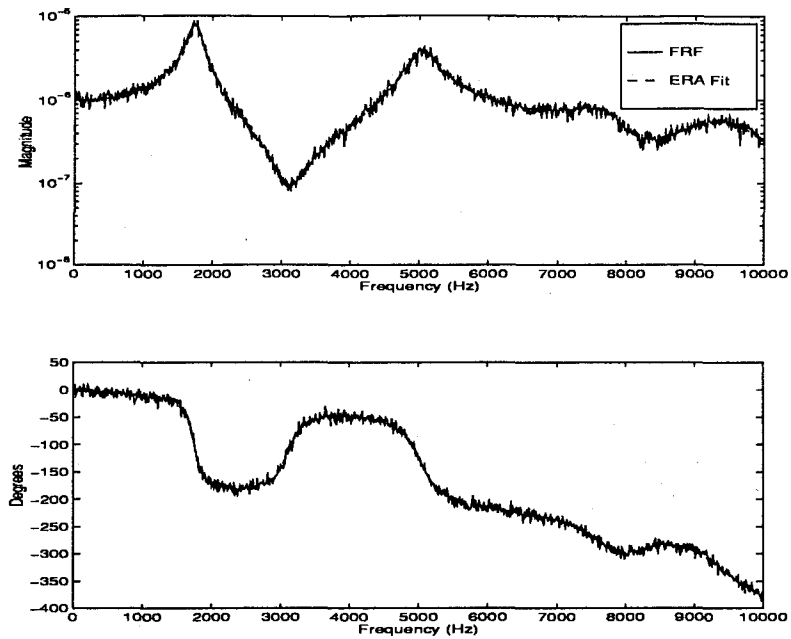


Figure 5.3 Displacement Y1 Force Input F2 for the Truth Model with Noise

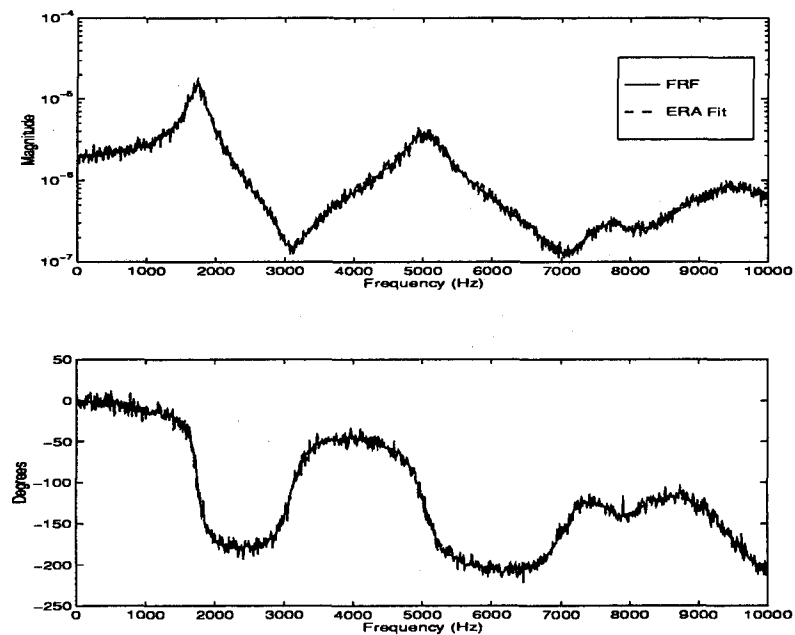


Figure 5.4 Displacement Y2 Force Input F2 for the Truth Model with Noise

5.2.2 Truth Model with Noise. To determine how well the Eigensystem Realization Algorithm could handle noise, a 15% noise was added to the frequency response functions obtained from MATLABTM. Figure 5.3 shows the transfer function and the corresponding ERA fit for displacement $Y1$ and force input $F2$, while Fig. 5.4 shows the transfer function for displacement $Y2$ and input $F2$ and its corresponding fit. The ERA fits shown for these transfer functions are characteristic of the remaining frequency response functions. Even in the presence of noise, the ERA algorithm is able to accurately identify the truth model. As shown in Table 5.2, the ERA algorithm is able to obtain an accurate model of the system even with 15% noise included in the frequency response functions.

Table 5.2 Truth Model Results

Mode No.	TRUTH MODEL		ERA FIT with No Noise		ERA FIT with Noise	
	Frequency (Hz)	ζ (%)	Frequency (Hz)	ζ (%)	Frequency (Hz)	ζ (%)
1	1748.64	5.02	1748.64	5.02	1747.70	4.98
2	5031.80	4.22	5031.80	4.22	5031.66	4.28
3	7709.75	5.22	7709.77	5.22	7708.31	5.31
4	9458.39	6.03	9458.44	6.03	9457.83	6.03

5.3 ERA Fit of the PACOSS DTA

Based upon the results of the ERA fit on the truth model, it was determined that the Eigensystem Realization Algorithm could identify the poles and zeros of the larger system. The state-space representation of the PACOSS DTA obtained using the friction-compensated system is not shown due to the size of the system. 80 states were identified under 14 Hz. The resulting A , B , C , and D matrices were 80 x 80, 80 x 8, 8 x 80, and 8 x 8, respectively. The resulting D matrix is a zero matrix.

The results presented in this section are the result of identifying the PACOSS DTA with active friction compensation. All transfer functions shown are for inertial acceleration outputs. Figure 5.5 shows the results for a non-collocated sensor and excitation actuator. For

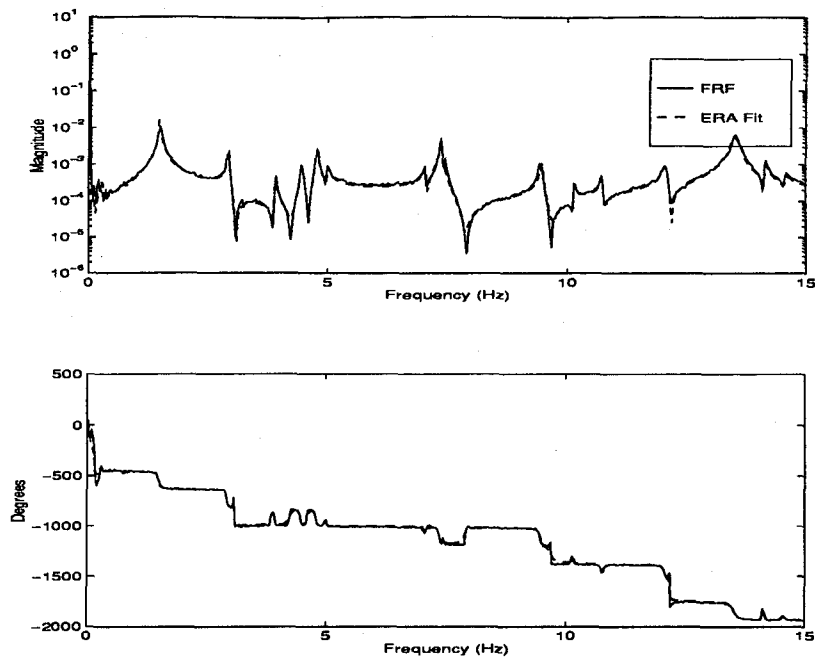


Figure 5.5 Non-located Pair: Measurement 1 Excitation Actuator 2

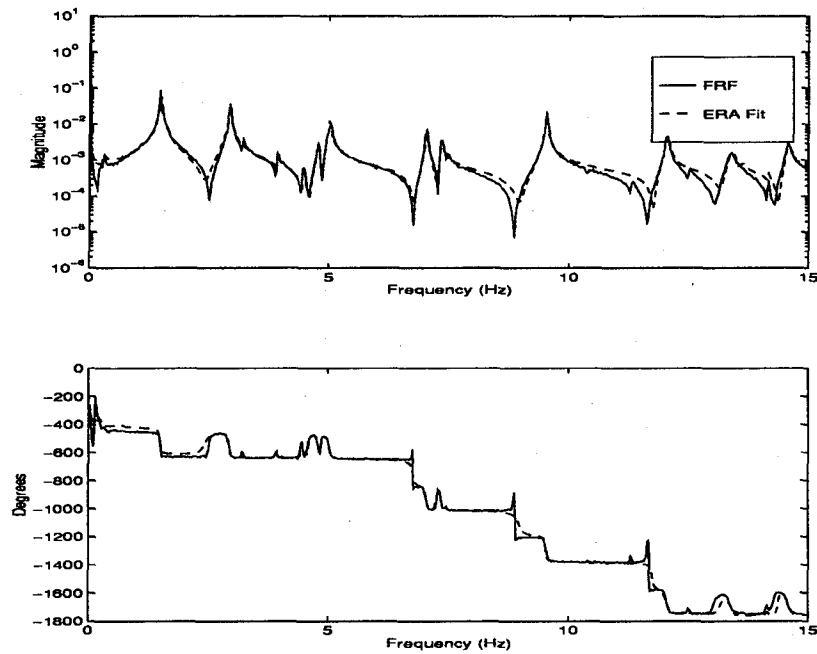


Figure 5.6 Collocated Pair: Measurement 6 Excitation Actuator 6

this particular FRF, the measurement was taken at Actuator 1 with Actuator 2 acting as the disturbance input. This fit is representative of the non-collocated FRFs. Figure 5.6 shows the ERA fit for a collocated sensor and actuator. A trend was noticed in the identification of this structure: the ERA fit for the non-collocated sensor/actuator pairs are better than those of the collocated pairs. Figures 5.5 and 5.6 show two of the better fits that were obtained from the ERA algorithm. Not all of the ERA fits were of this quality. Figure 5.7 shows the frequency response function for Measurement 1 Excitation Actuator 6 and its corresponding ERA fit. This FRF/ERA fit was the worst in the set of 64 transfer functions; however, it should be noted that it is not possible to fit every FRF exactly since a nonlinear system is being identified using a linear scheme. The resulting 64 FRFs and their corresponding ERA fits are shown in Appendices A-H.

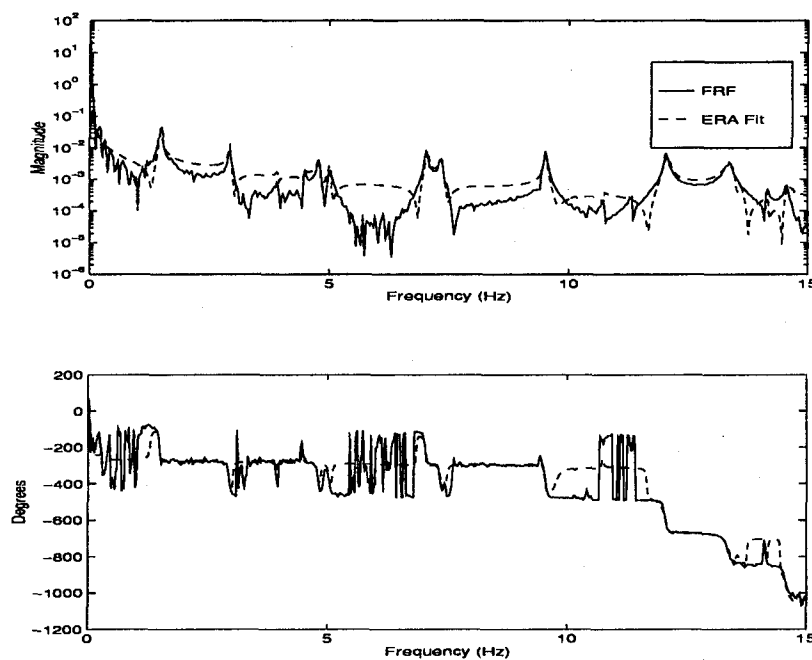


Figure 5.7 Measurement 1 Excitation Actuator 6

5.4 Friction Compensation: Is it worth it?

One of the goals of this thesis was to calculate the Coulomb damping coefficients and apply active friction compensation to the PACOSS DTA. This goal was achieved. However, the question remains: Was friction compensation worth the trouble to calculate the coefficients and implement the compensation program? The answer is yes. The following paragraphs give a more in-depth look at why friction compensation should be used.

First, however, the measured FRFs of both the compensated and non-compensated system are compared. FRFs from each system are shown in Figs. 5.8 and 5.9. Figure 5.8 compares both FRFs for a non-collocated sensor/actuator pair while Fig. 5.9 compares two collocated FRFs.

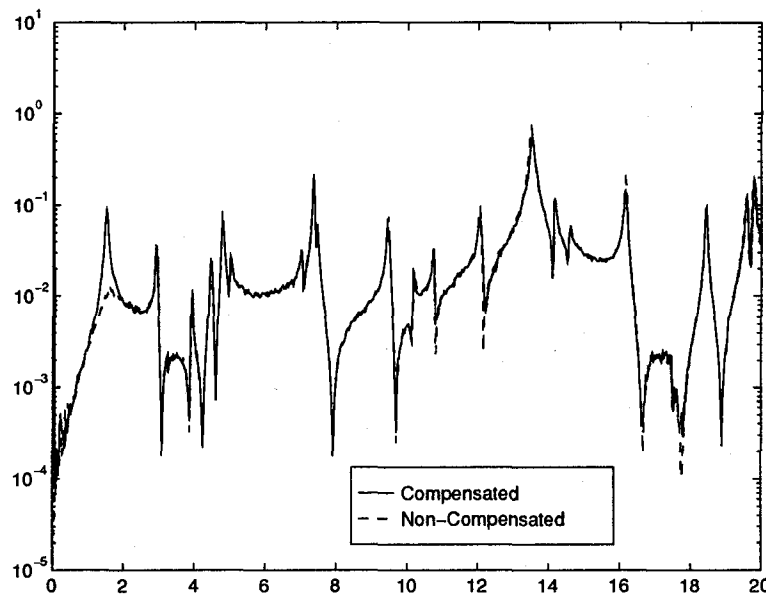


Figure 5.8 FRFs for Measurement 1 Actuator 2

Referencing both of these figures, one can see that the only difference between the FRF of the compensated system and the FRF of the non-compensated system is the actuator mode at 1.5 Hz. Of course, this is expected since the friction compensation removes the nonlinear damping from that mode. What is surprising, however, is that the actuator mode is the **only**

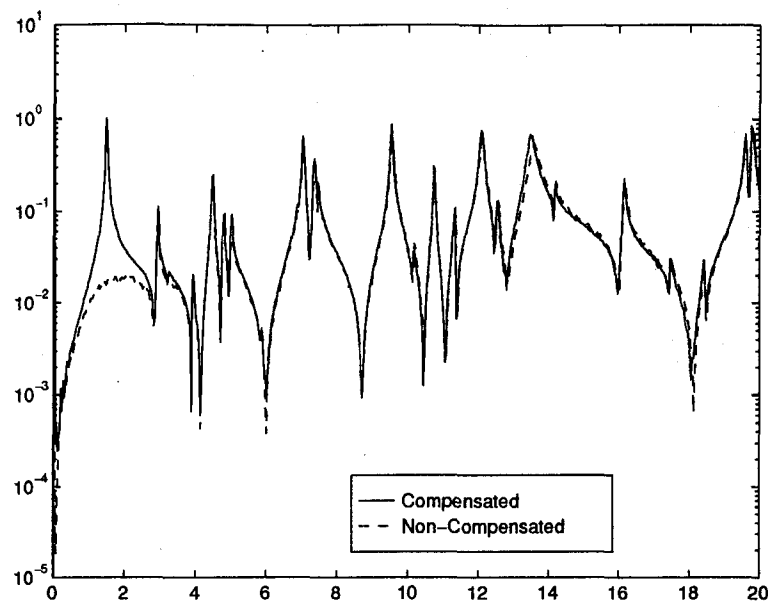


Figure 5.9 FRFs for Measurement 4 Actuator 4

mode that changes. Prior to this comparison, it was thought that many of the low frequency modes would be altered by the friction compensation. This was not the case.

Due to the fact that both systems resulted in almost identical FRFs, it was assumed that friction compensation was not worth the effort and that the ERA fit for the non-compensated frequency response functions would be the same as those that contained the friction compensation. Again, this was not the case. Looking at Figs. 5.10 and 5.11, one can see that the ERA fit for the compensated system did an overall better job identifying the poles and zeros of the DTA. This can be seen in the identification of the zeros especially. The zeros identified in the compensated system are deeper than those found in the non-compensated system. Overall, the ERA fit for the compensated system is better than the fit for the non-compensated system.

A reason for the difference in the ERA fits is that the ERA algorithm produces a state-space, or linear, model. Thus, nonlinearities in a system produce larger errors in the fit. Removing nonlinearities, such as Coulomb damping, helps to produce a linear system to be

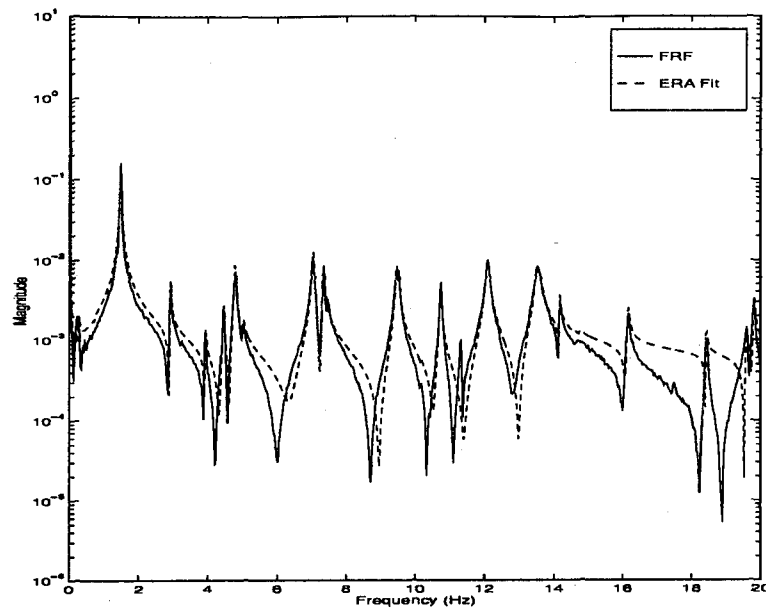


Figure 5.10 ERA Fit for Compensated System: Measurement 1 Actuator 1

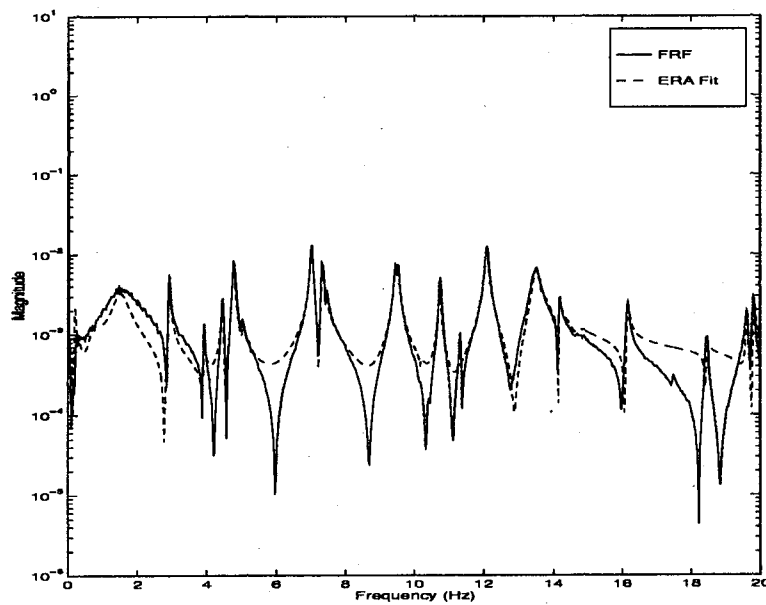


Figure 5.11 ERA Fit for Non-Compensated System: Measurement 1 Actuator 1

used for system identification. Using the compensated system, then, produces a “more linear” model which is more accurately identified by the ERA algorithm.

5.5 Effects of Instrumentation Wire Removal

In 1992, Lt. Chad Matheson, using transfer function matching, constructed a model of the PACOSS DTA. However, the test article at that time was outfitted with instrumentation wiring for 151 accelerometers. In order to reduce damping and the number of nonlinearities in the system, the wiring and the sensors were removed so that an accurate model could be obtained. After obtaining the frequency response data, it was compared to the frequency response data obtained by Lt. Chad Matheson. Figure 5.12 shows the frequency response function for Measurement 2 Excitation Actuator 1 obtained from the compensated system and that obtained by Lt. Matheson for the same sensor/actuator pair.

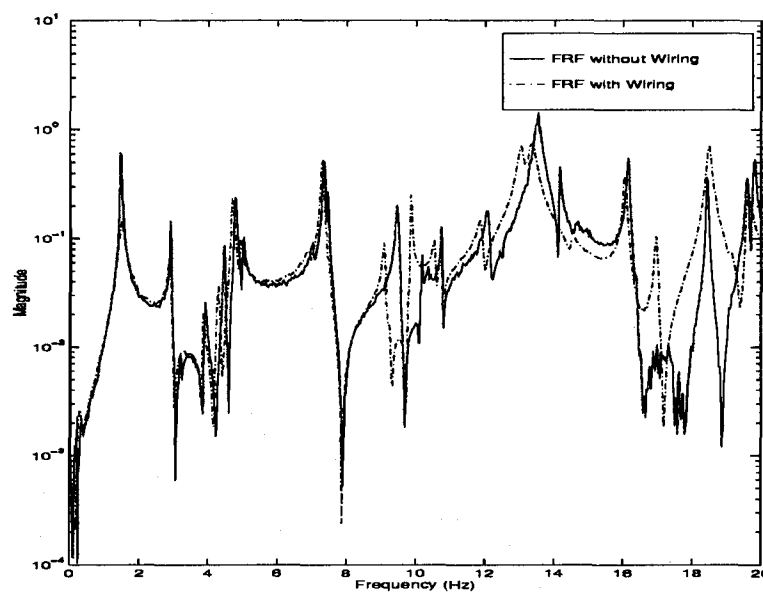


Figure 5.12 Comparison of FRFs

Up to 9 Hz, the two response functions match rather well. Starting at 9 Hz, the poles and zeros of the two systems no longer match. The poles and zeros not only occur at

different frequencies, but their amplitudes also vary. Figure 5.12 shows that the removal of the instrumentation wires has a pronounced effect upon the system. The frequency response functions, both compensated and non-compensated, obtained in the course of this research are reproducible and have been reproduced several times.

VI. Recommendations and Conclusions

This thesis had two primary goals: (1) Obtain an accurate state-space model of the PACOSS Dynamic Test Article and (2) Identify the Coulomb damping in each actuator and compensate for this nonlinearity in the frequency response functions. The two primary goals of this thesis have been achieved. The state-space representation of the system's dynamics for the first 40 modes is accurate. As seen in Chapter V, the Coulomb damping factor in each actuator was identified, and the friction compensation was successful, as demonstrated by the undamped actuator mode in the compensated system.

As demonstrated, the ERA algorithm worked well considering the dynamics of the structure being identified is nonlinear. It was noticed that the system identified by ERA did not fit the collocated sensor/actuator pairs as well as the non-collocated pairs. The reason for this discrepancy is not known and needs to be investigated further. Overall, the ERA algorithm did a good job of identifying the poles in the system, which is important for stability reasons.

The ERA algorithm performed better on the friction-compensated system. This is expected since the compensated system is "more linear" than the non-compensated system due to the removal of the damped actuator mode. Thus, the friction-compensated state-space model should be used for control design. The friction compensation can be taken into account in the design of the controller. This makes the application of optimal control strategies simpler since the structure's dynamics can be considered linear.

As depicted in Chapter V, the removal of outside influences when performing system identification is of the utmost importance. This is especially true of large space structures. Insulated from the environment by the air-bearing suspension system, the DTA was in a simulated zero-gravity environment. However, in earlier efforts to model the test article, extraneous wiring (not in use at the time) was attached to the DTA. These wires, adding nonlinear effects, caused many of the system's modes to change frequency and amplitude. When obtaining an accurate model of a system on which one plans to employ active control strategies, the structure should be identified in the same configuration in which it will be

utilized. This was the case in this identification and the removal of the instrumentation wiring proved valuable in obtaining an accurate model of the DTA.

The ERA algorithm was validated first on the truth model, a linear system. It provided an excellent fit of the truth model, even when a 15% noise was added to the frequency response functions. However, the same results were not obtained from the more complex PACOSS DTA. There are several reasons for this discrepancy. First, the truth model is a linear system, whereas the DTA is a nonlinear system. Since the model being developed is linear, it is expected it would identify the linear system better than the nonlinear system. Also, the noise in the truth model did not obscure the shape of the FRF as sometimes happens in the DTA. Thus, the ERA routine had a much easier time identifying the modes in the truth model. Noise was added to the truth model's frequency response functions after they had already been obtained. Noise has many ports of entry into the frequency response functions of the DTA, especially when active Coulomb damping compensation is being applied.

The ERA routine used for this system identification was written in MATLABTM. A Sparc 10 with dual 55 MHz processors was also used. The Sparc 10 provided a needed increase in speed and random access memory, over the Sparc 2, in identifying the 8 x 8 MIMO system with 80 states. The identification process was limited to the first 40 modes due to the limitation on computer memory. If a machine with more memory had been used, the number of modes identified could have been increased. If the amount of memory available could be increased, the identification could be extended up to 20 Hz at a minimum.

After obtaining the frequency response functions, it took more than one run of the ERA routine to acquire an acceptable model of the DTA. The method first used in determining the number of states to be identified was to count the number of peaks in the FRF. This method led to an inaccurate and erroneous model. In utilizing this method, 40 states were identified, whereas 80 states should have been identified below 14 Hz. This was remedied when it was discovered that there were 40 modes (i.e., 80 first-order states) below the cutoff frequency of 14 Hz. This was determined by looking at the finite element model generated by Capt. Scott George (4). Learning from this mistake, it is recommended that a modal analysis of some type

be completed prior to utilizing the Eigensystem Realization Algorithm. Doing so will save time and result in a more accurate description of the systems low frequency dynamics.

As can be seen in the results of the ERA fit of the DTA, ERA fits the poles in the system rather well, but is much more inaccurate in fitting the zeros. Therefore, it is recommended that parameter optimization be applied to this system. The A , B , C , and D matrices identified by the ERA algorithm could be used as starting points in the time domain based parameter optimization routines to improve the system response fits. MATLAB's System Identification toolbox contains some parameter optimization routines that could be used to possibly obtain a more accurate state-space representation of the PACOSS DTA's low frequency dynamics.

Appendix A. State-Space Model FRFs of PACOSS DTA

This appendix is divided into eight separate appendices with each appendix representing a column of the transfer function matrix. These appendices contain the output (a pictorial representation) of the *frffilt.m* routine.

This appendix contains the eight measurements resulting from the exciting of Actuator 1 with a random noise.

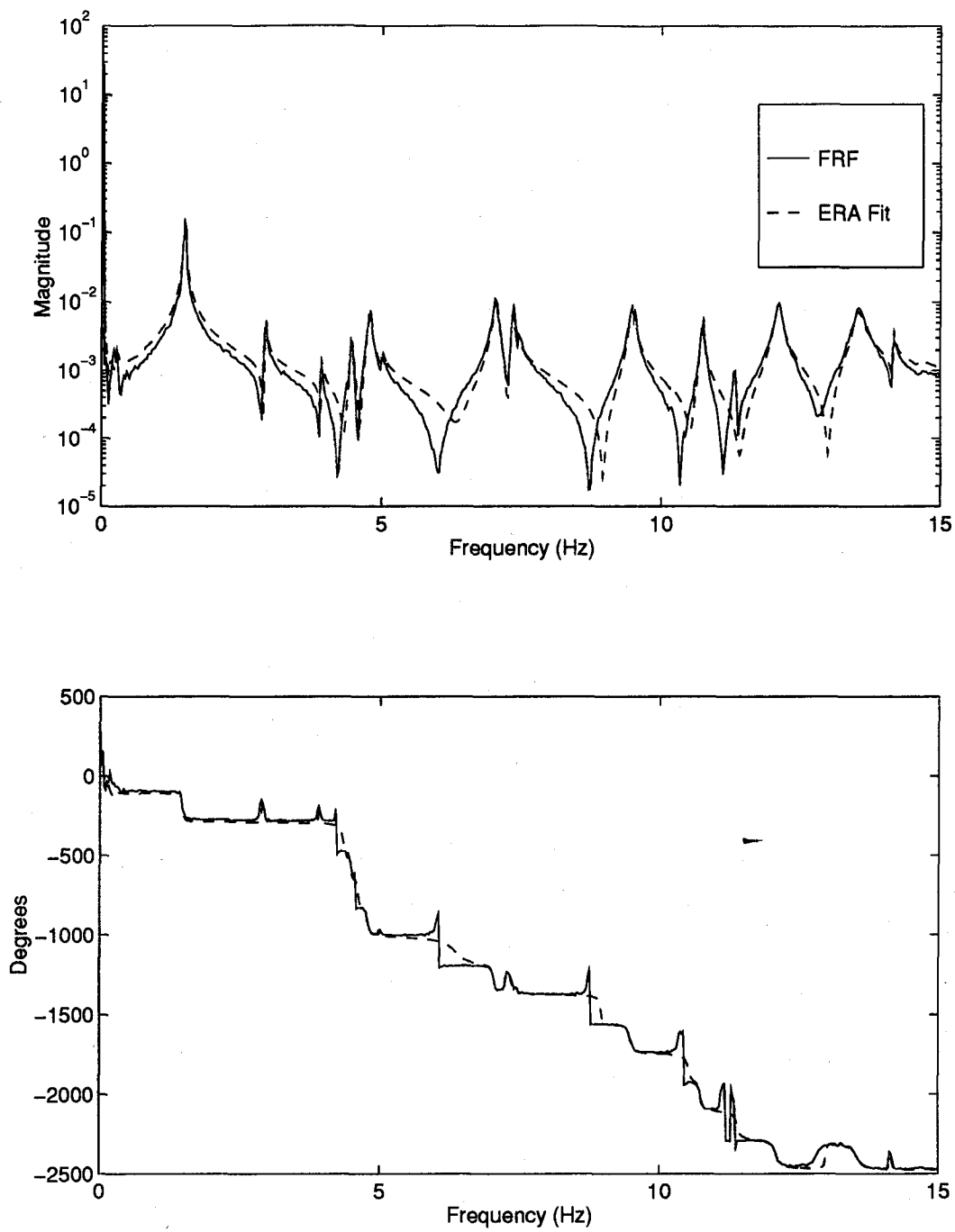


Figure A.1 Measurement 1 Excitation Actuator 1

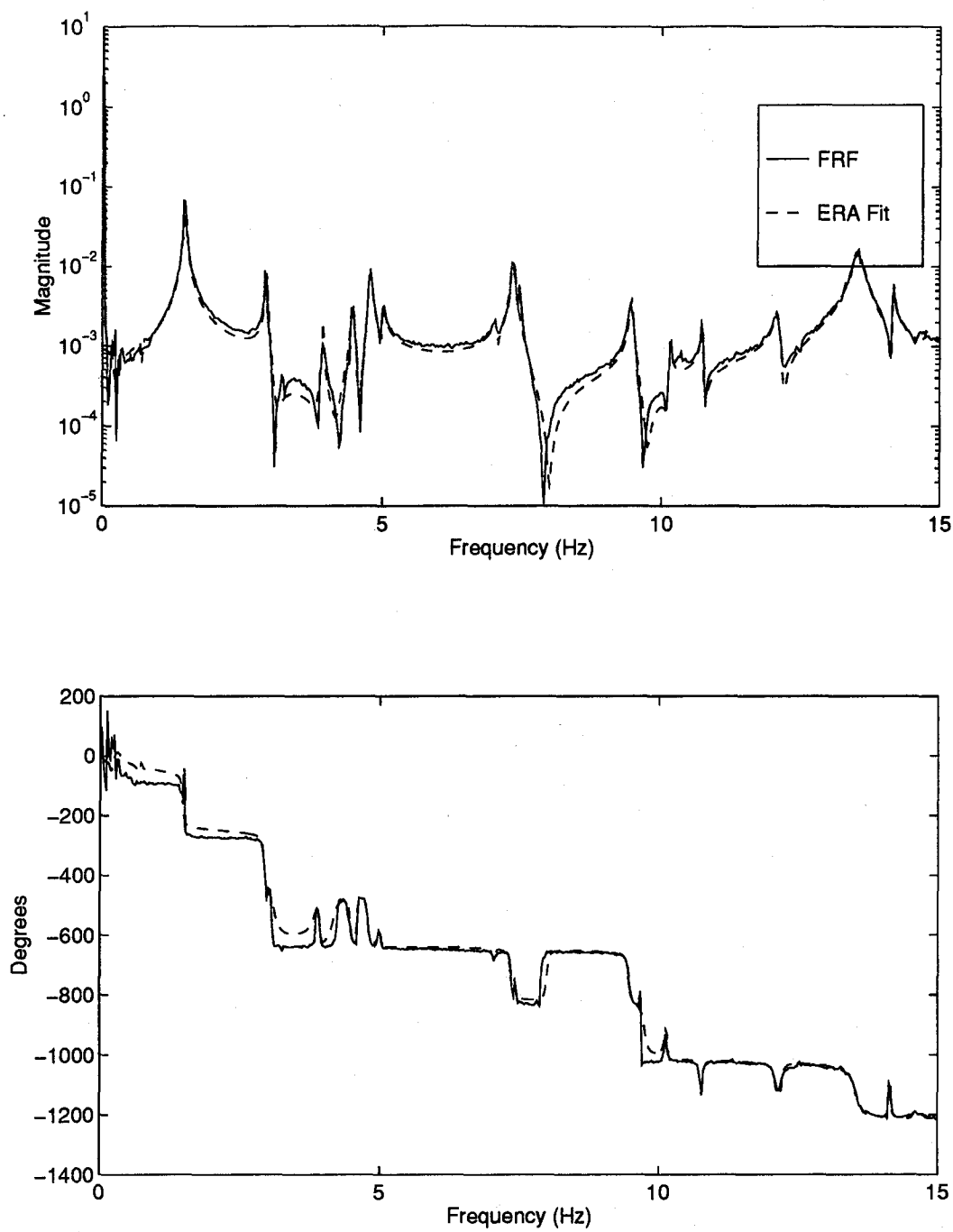


Figure A.2 Measurement 2 Excitation Actuator 1

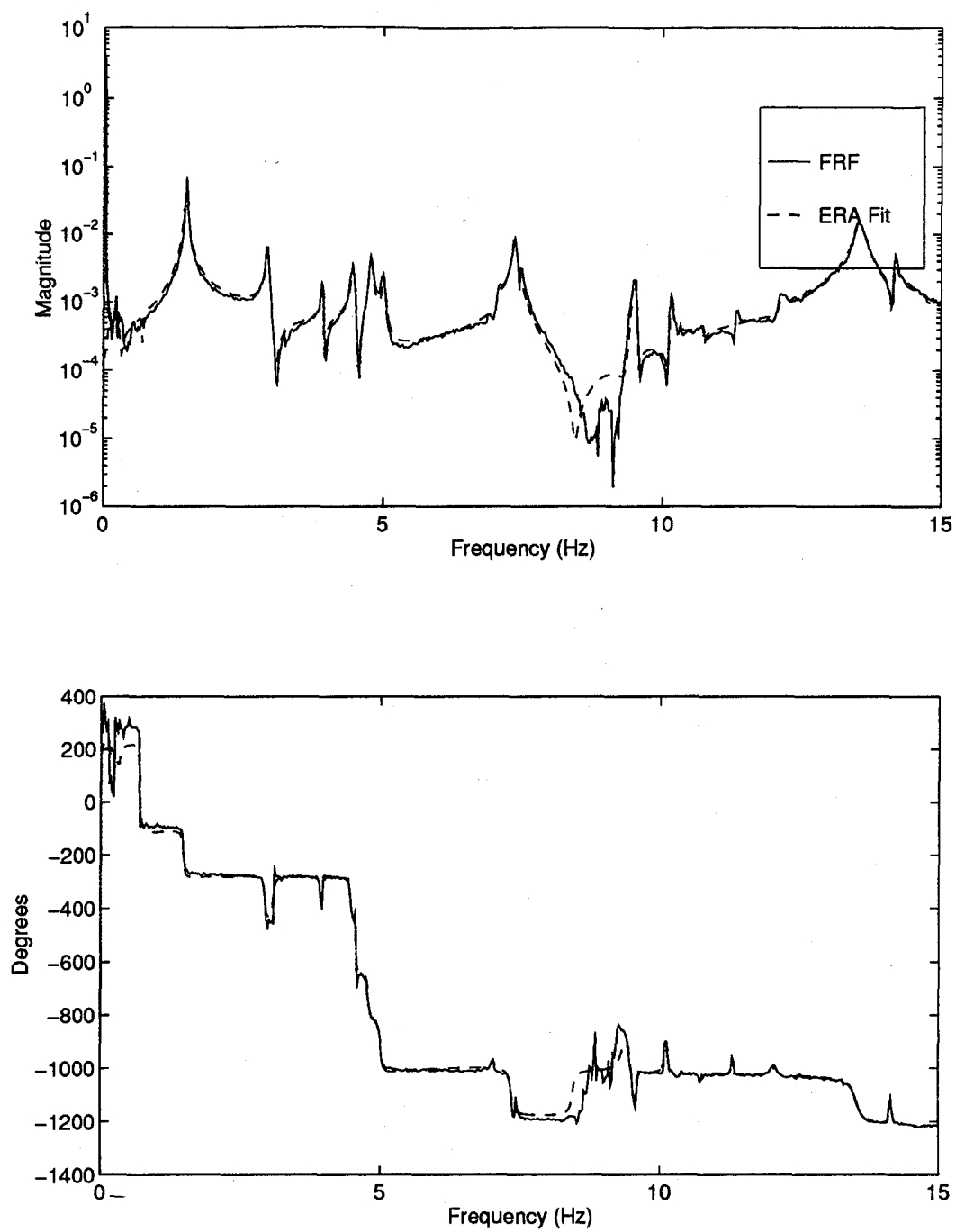


Figure A.3 Measurement 3 Excitation Actuator 1

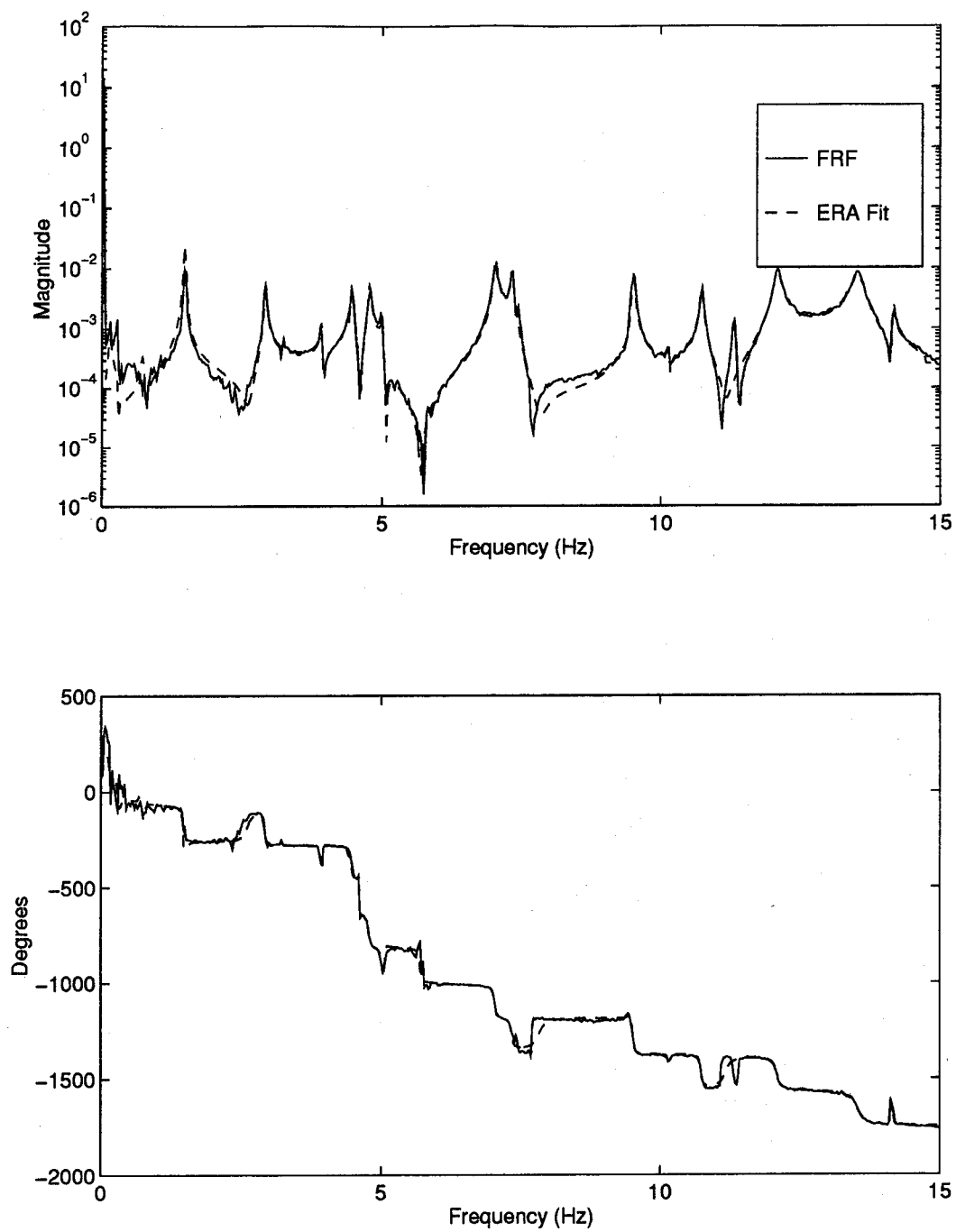


Figure A.4 Measurement 4 Excitation Actuator 1

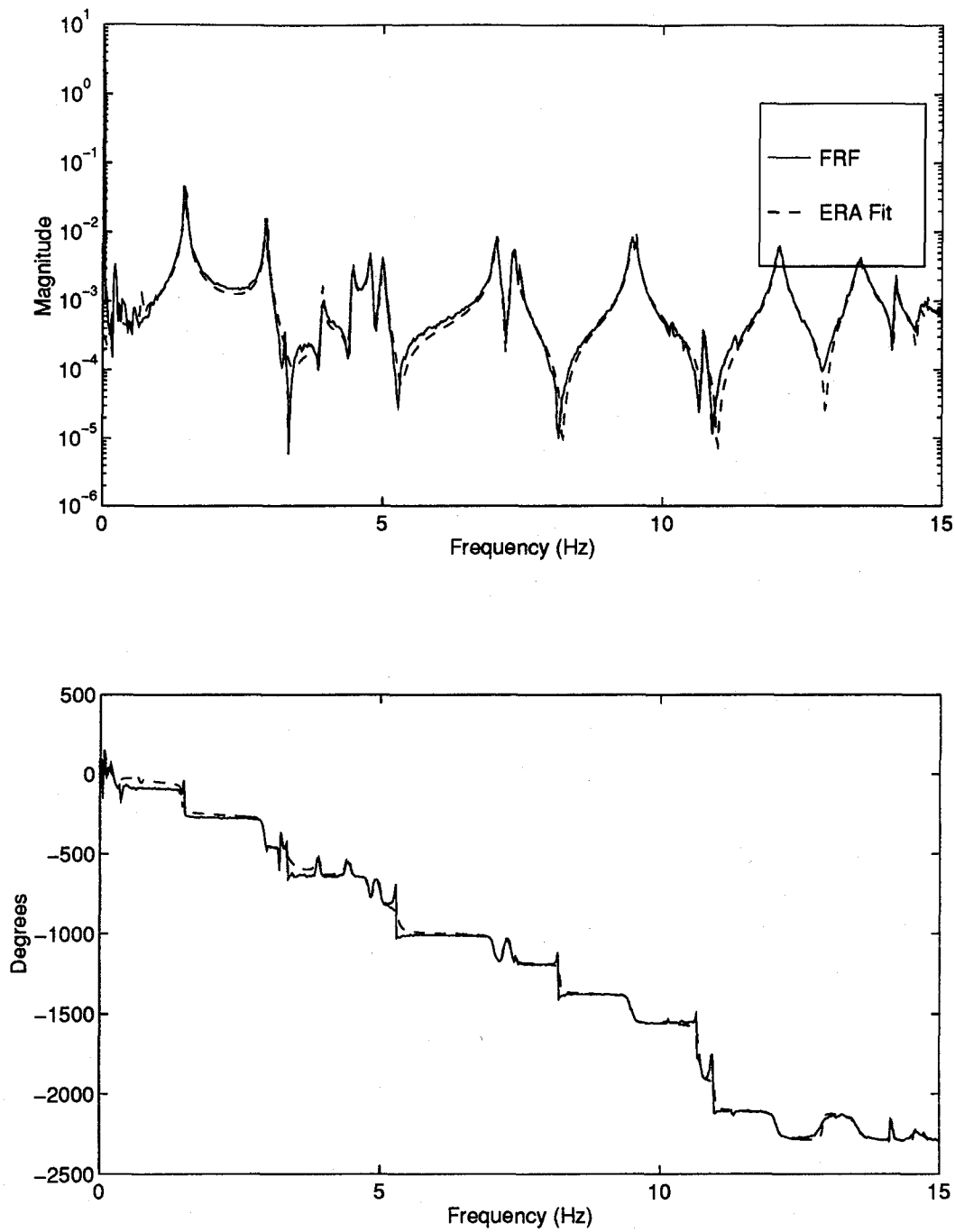


Figure A.5 Measurement 5 Excitation Actuator 1

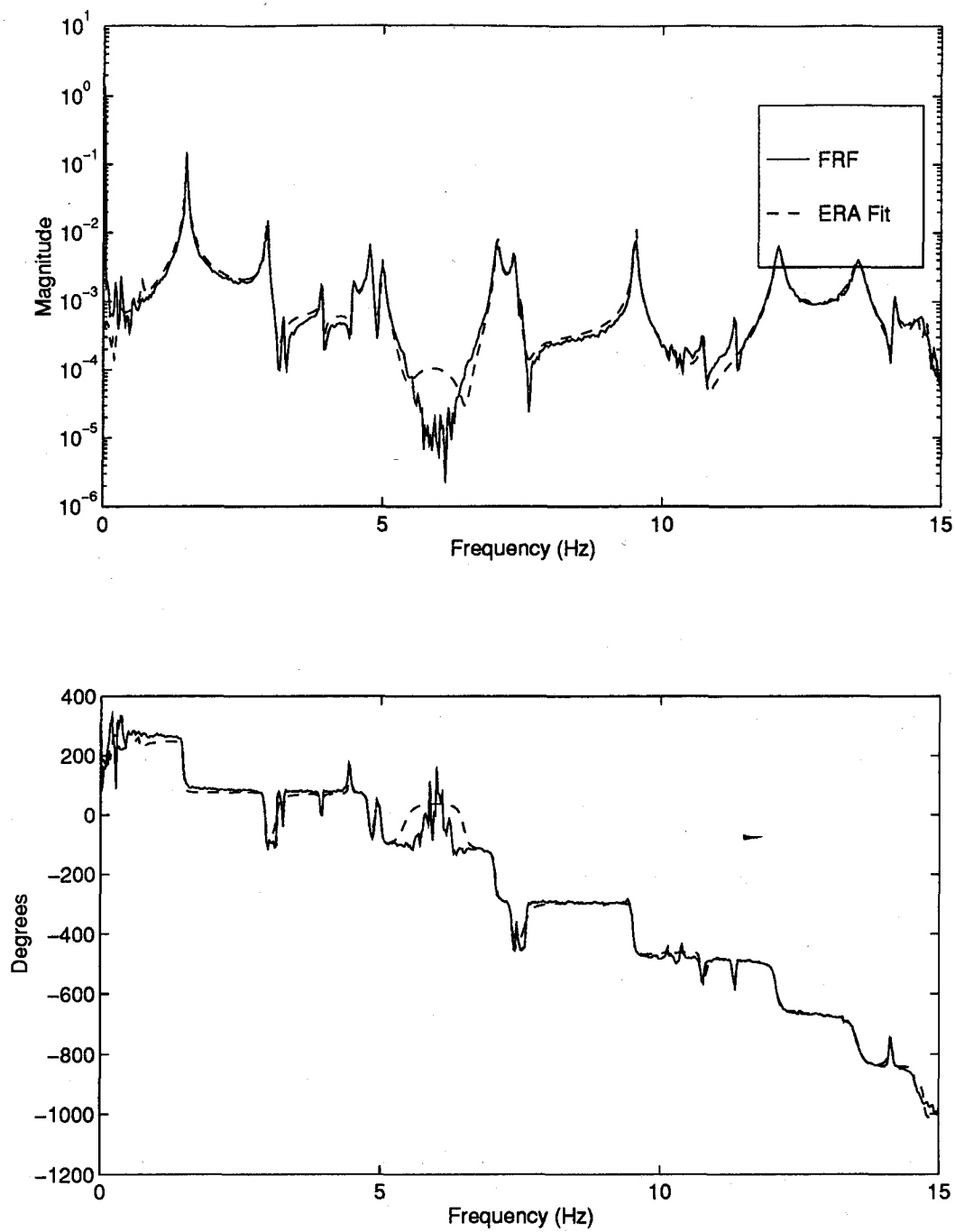


Figure A.6 Measurement 6 Excitation Actuator 1

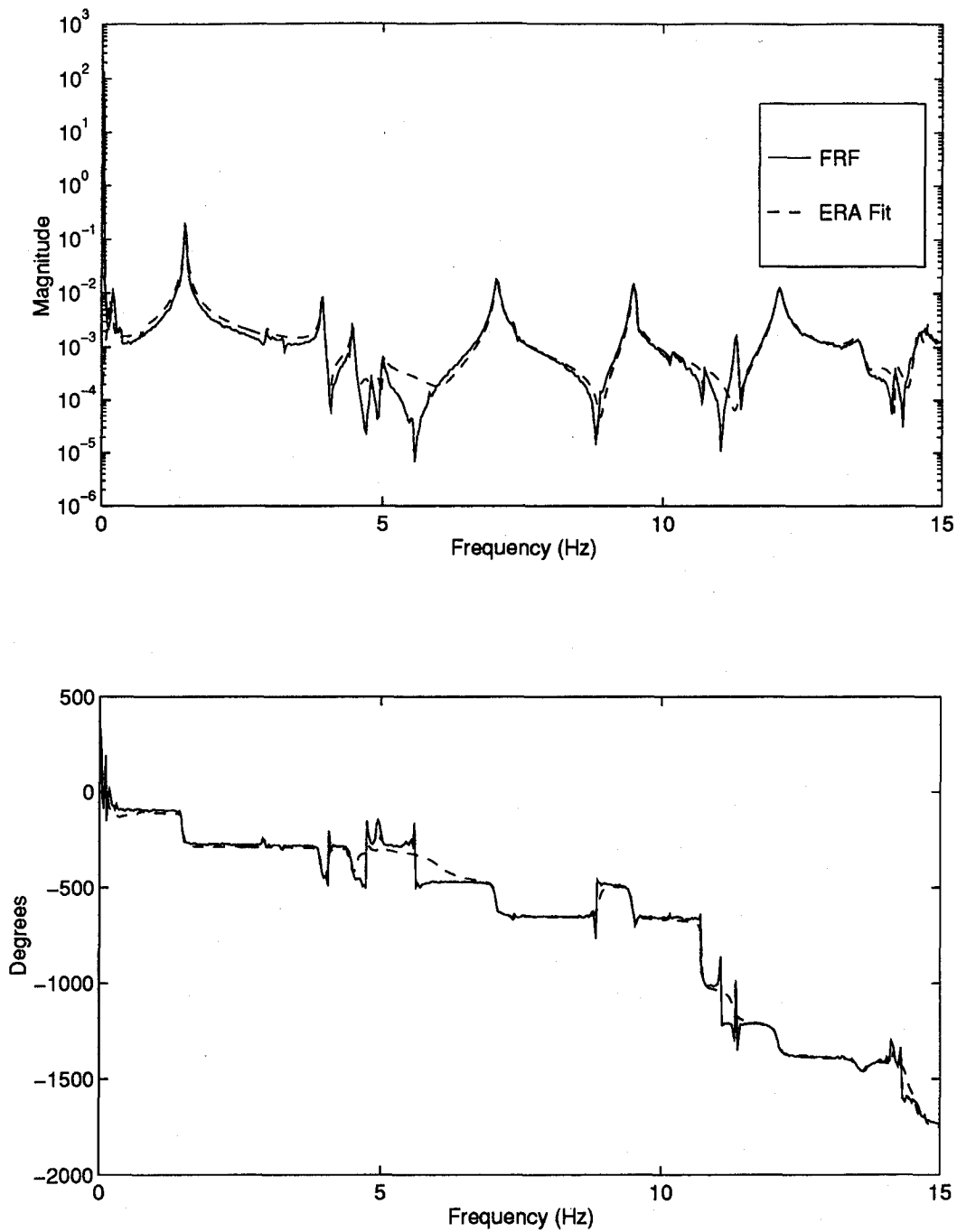


Figure A.7 Measurement 7 Excitation Actuator 1

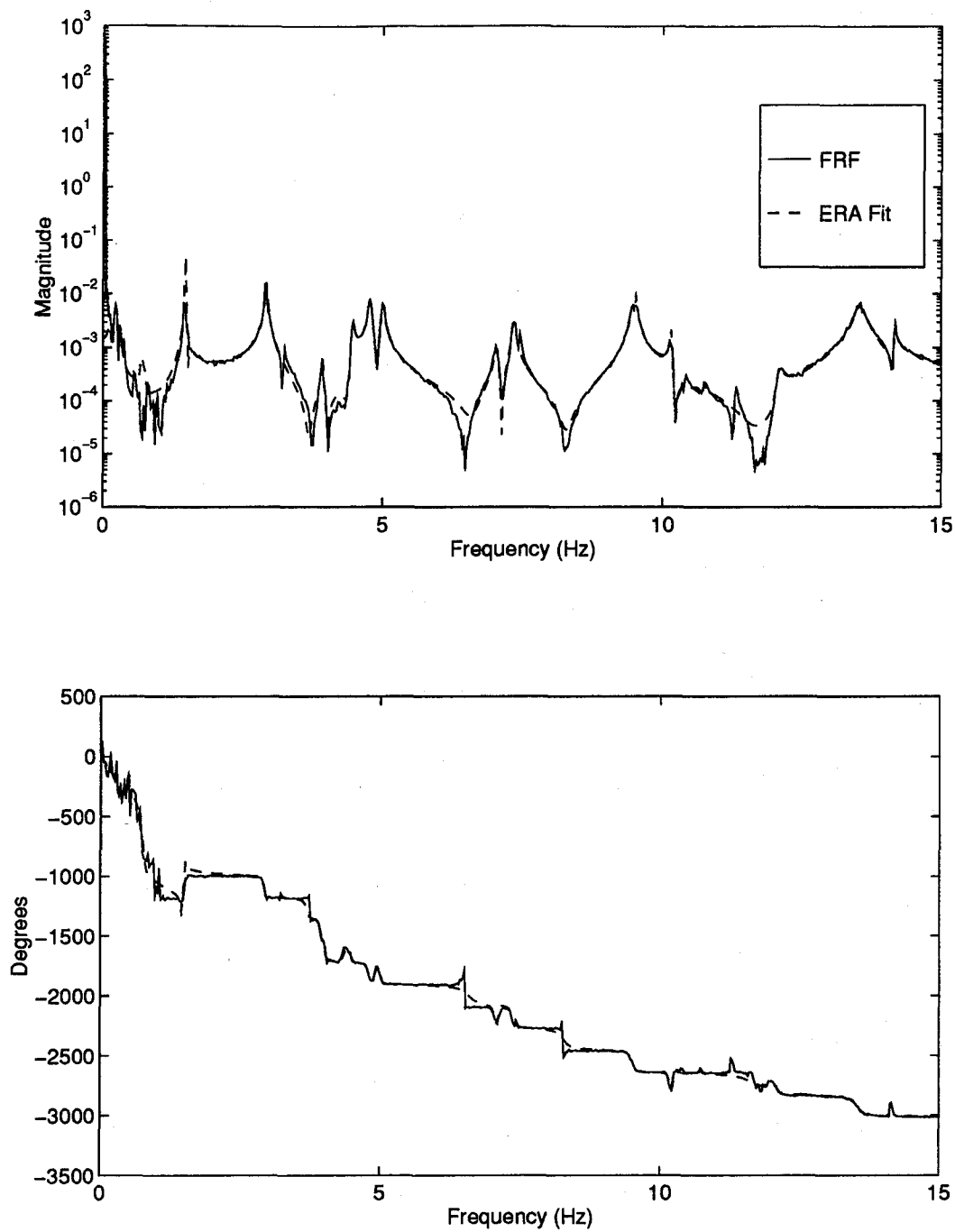


Figure A.8 Measurement 8 Excitation Actuator 1

Appendix B. State-Space Model FRFs of PACOSS DTA (Cont.)

The eight plots in this appendix are the measurements taken when Actuator 2 was excited with a random noise.

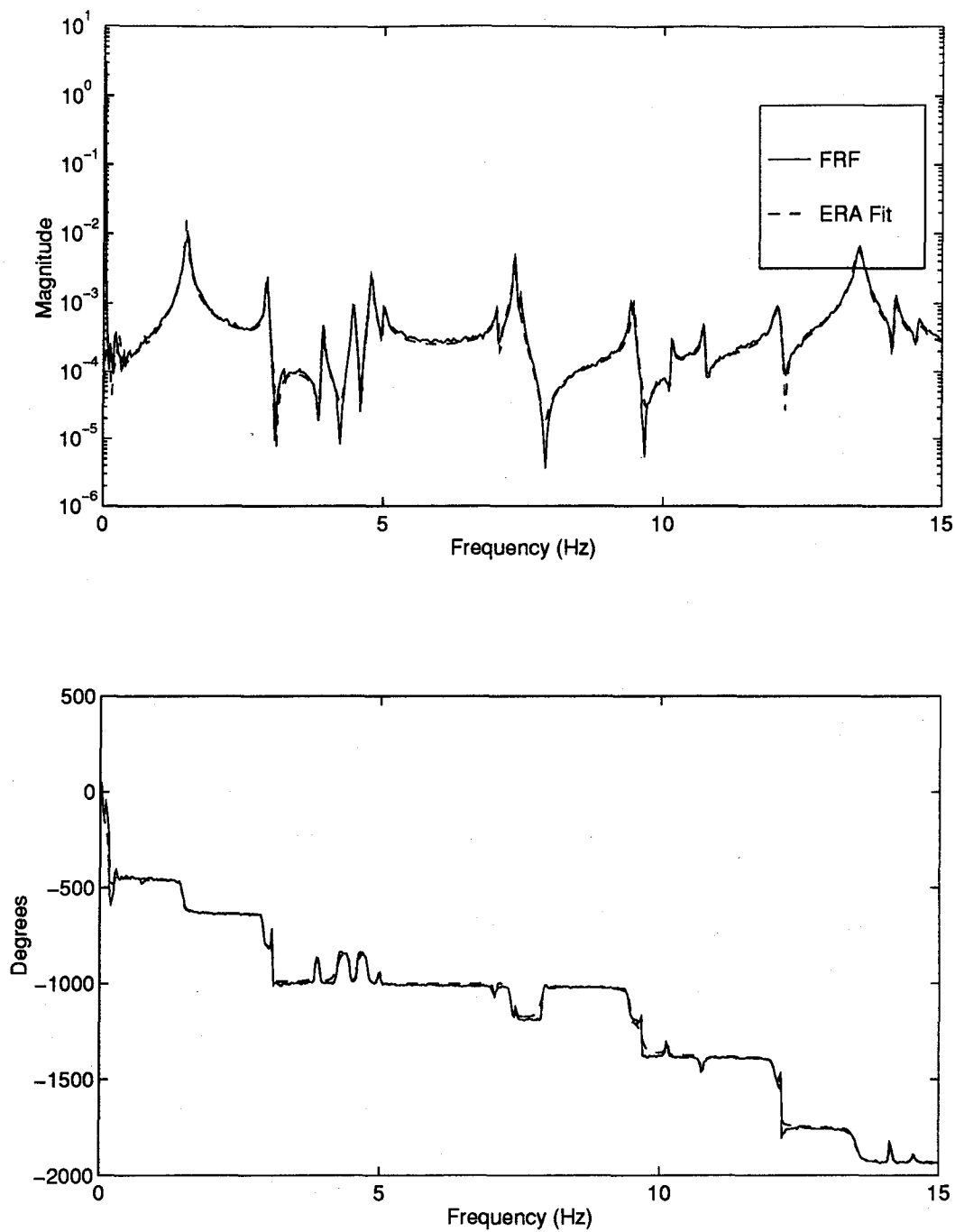


Figure B.1 Measurement 1 Excitation Actuator 2

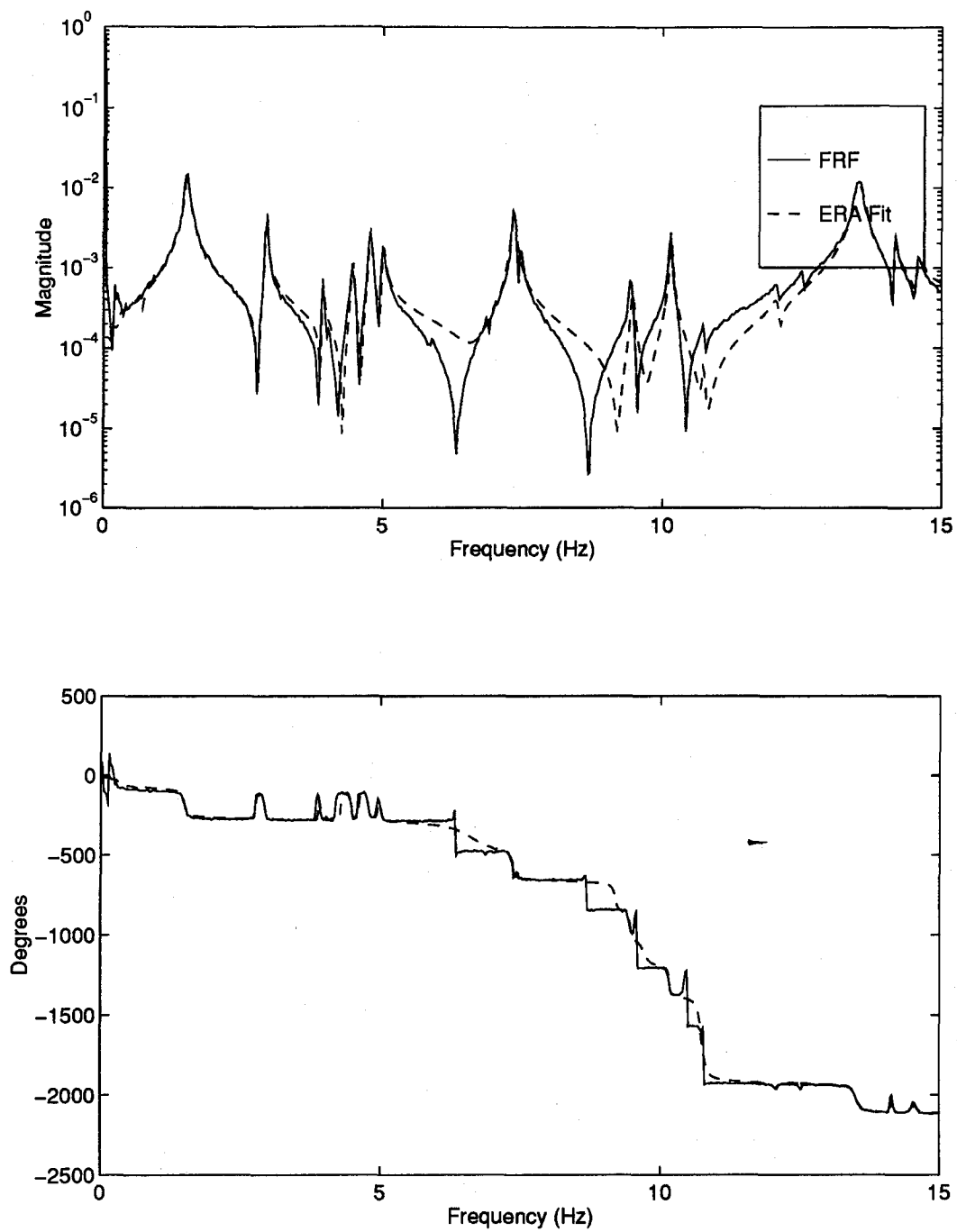


Figure B.2 Measurement 2 Excitation Actuator 2

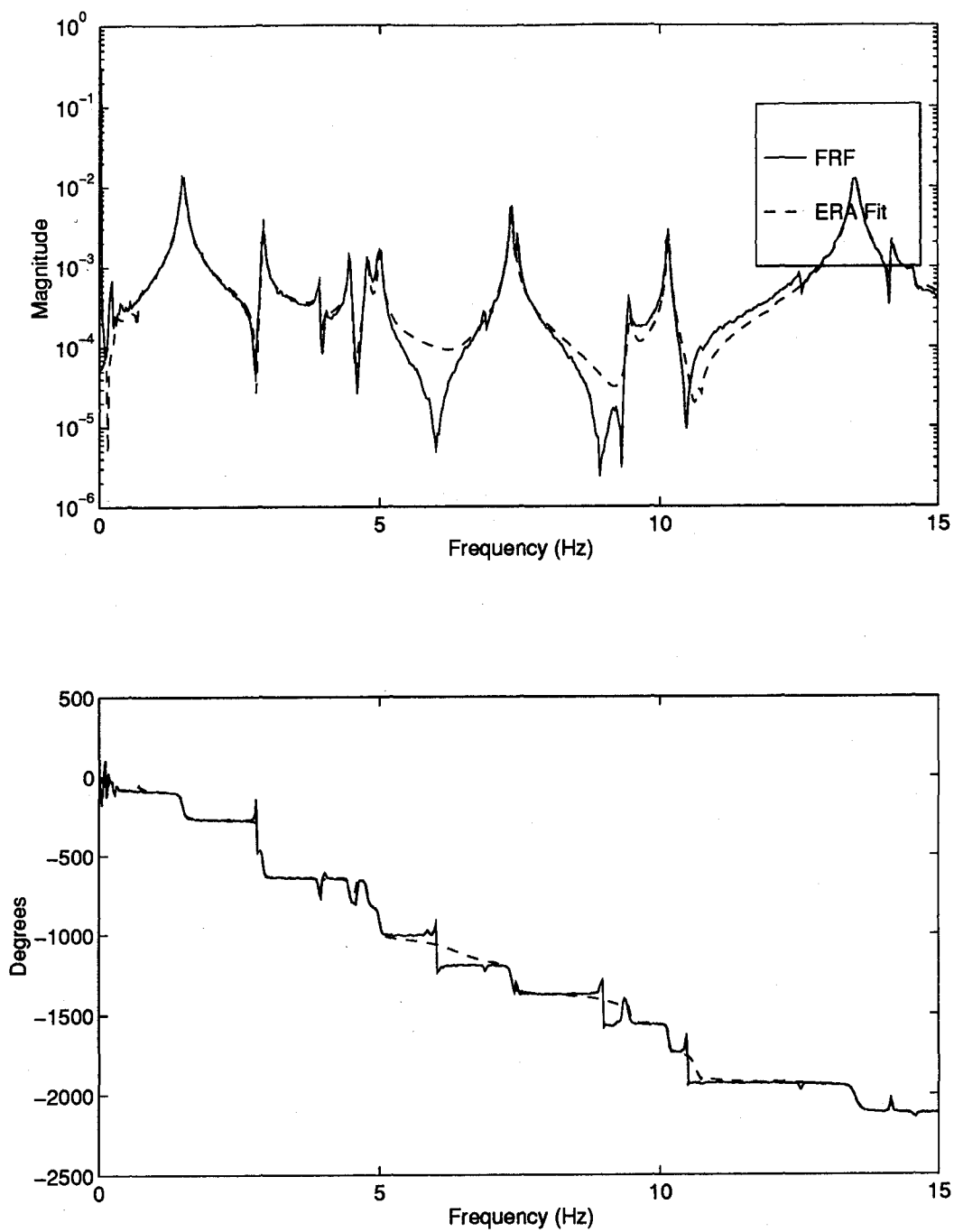


Figure B.3 Measurement 3 Excitation Actuator 2

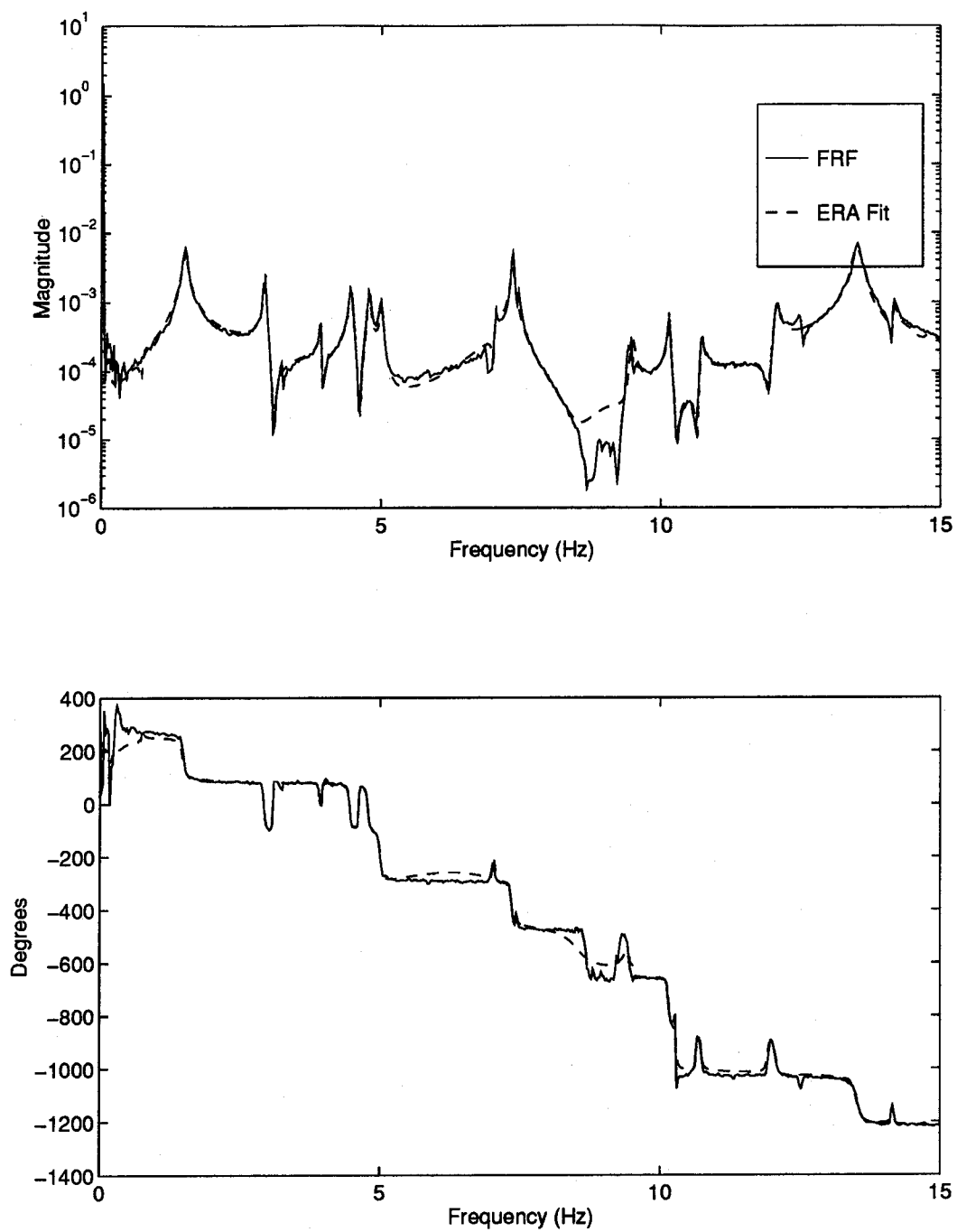


Figure B.4 Measurement 4 Excitation Actuator 2

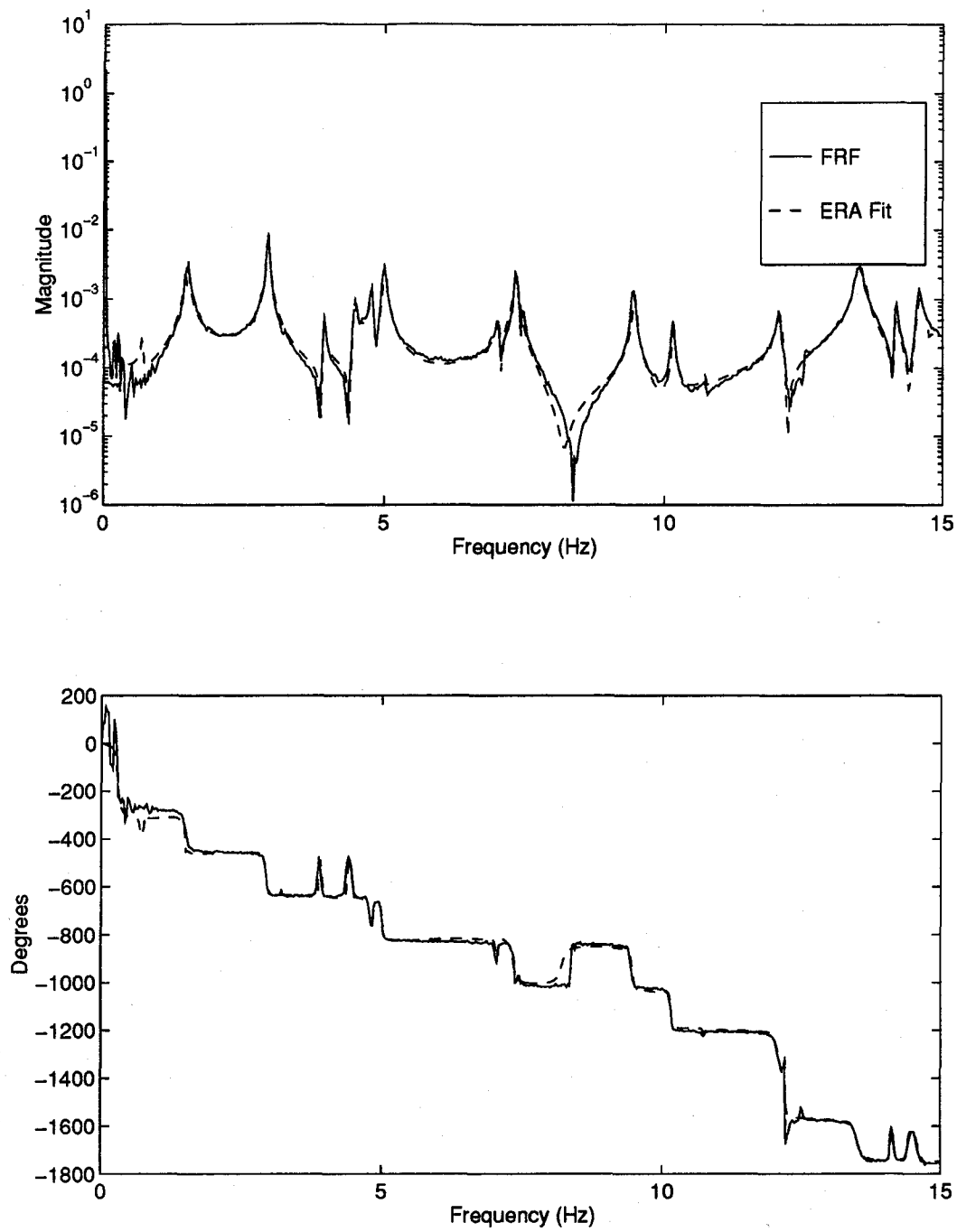


Figure B.5 Measurement 5 Excitation Actuator 2

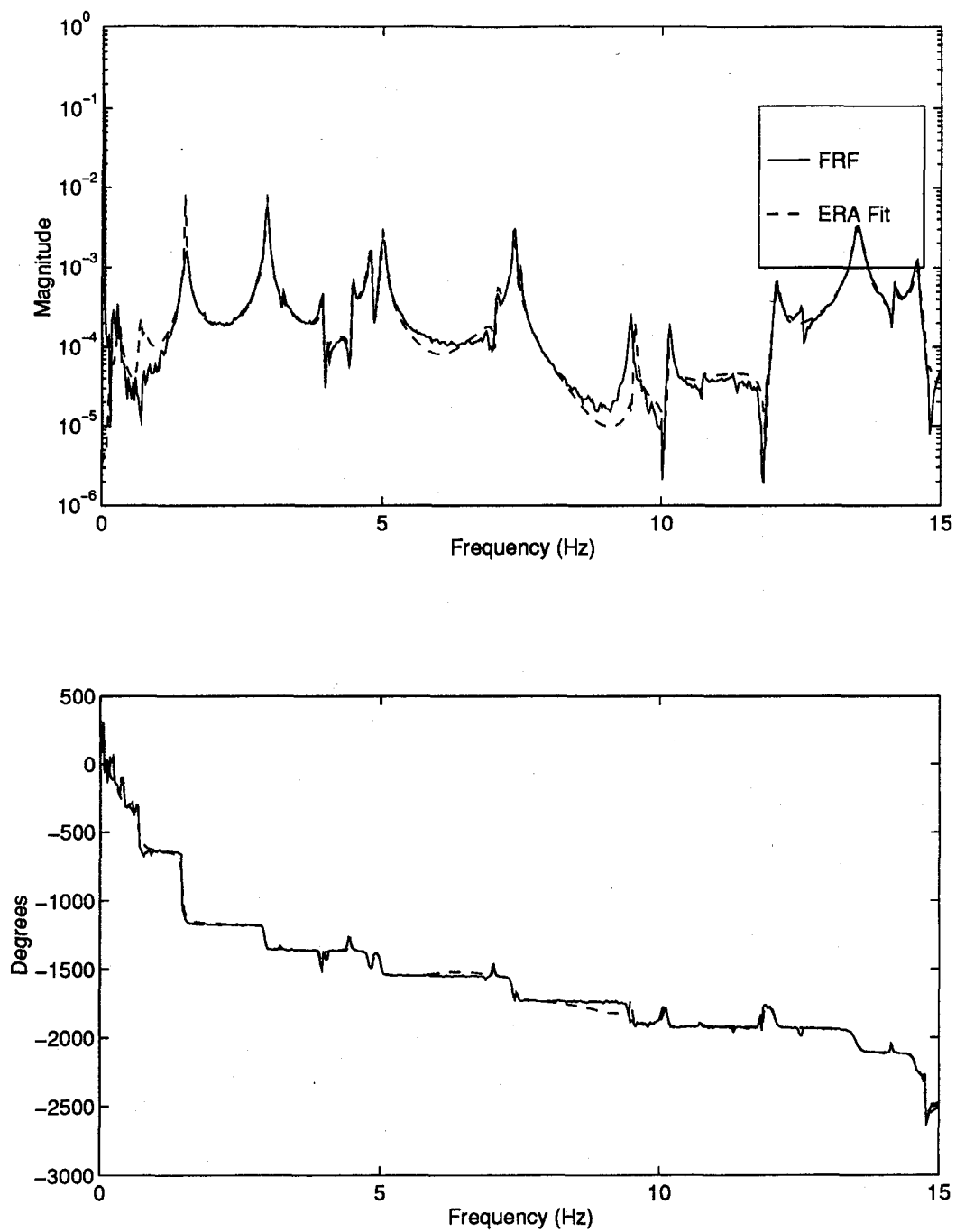


Figure B.6 Measurement 6 Excitation Actuator 2

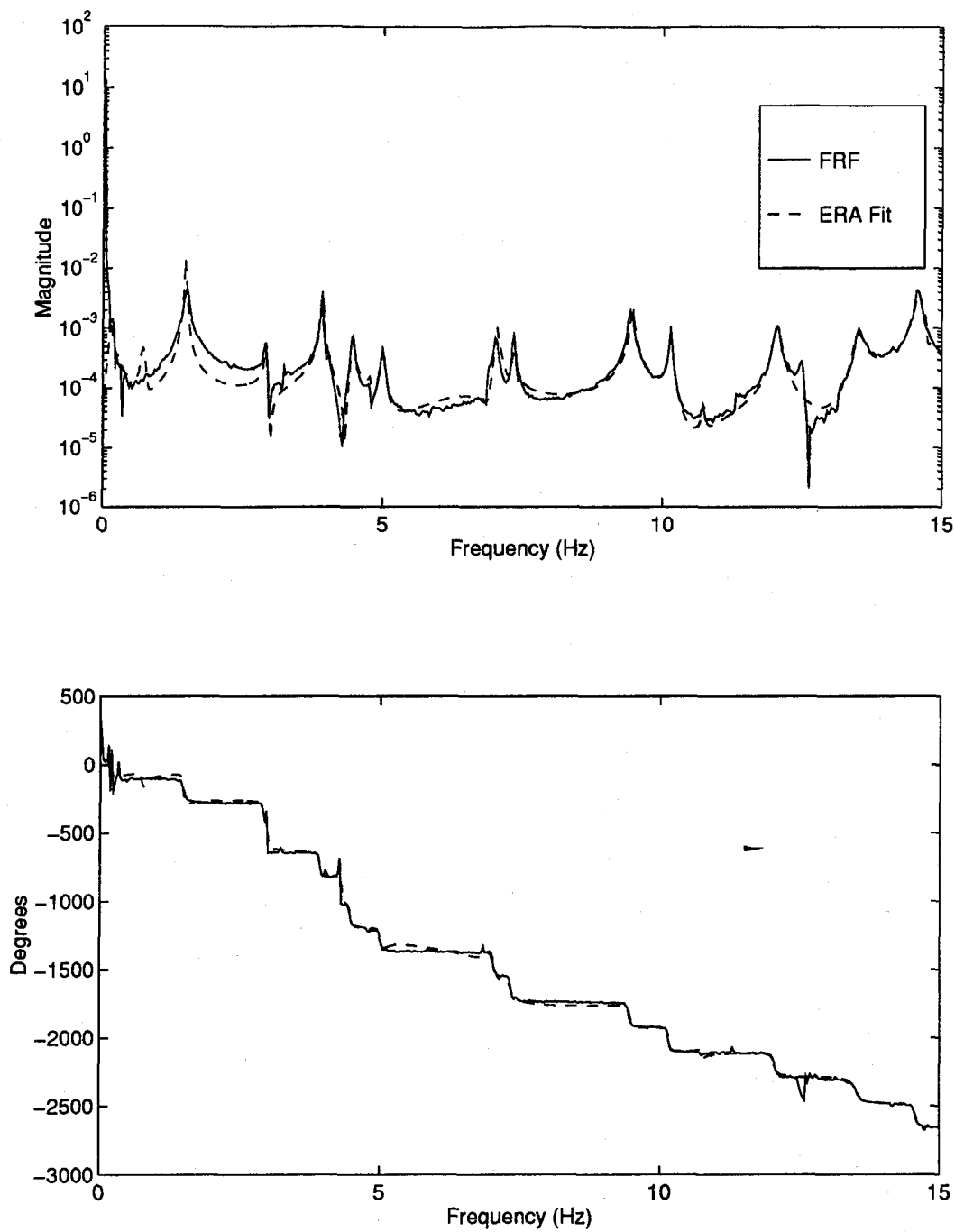


Figure B.7 Measurement 7 Excitation Actuator 2

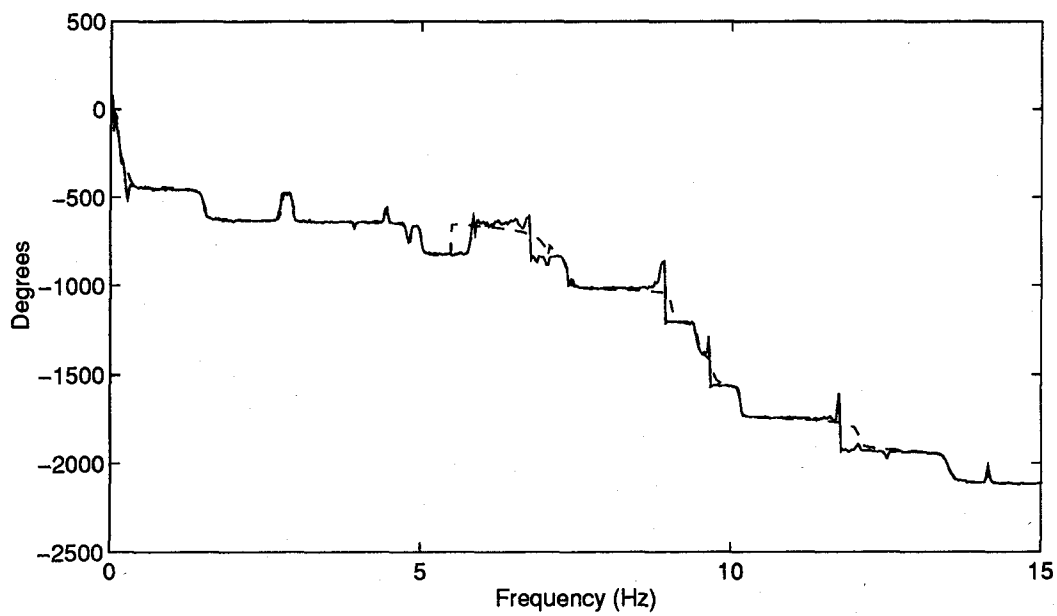
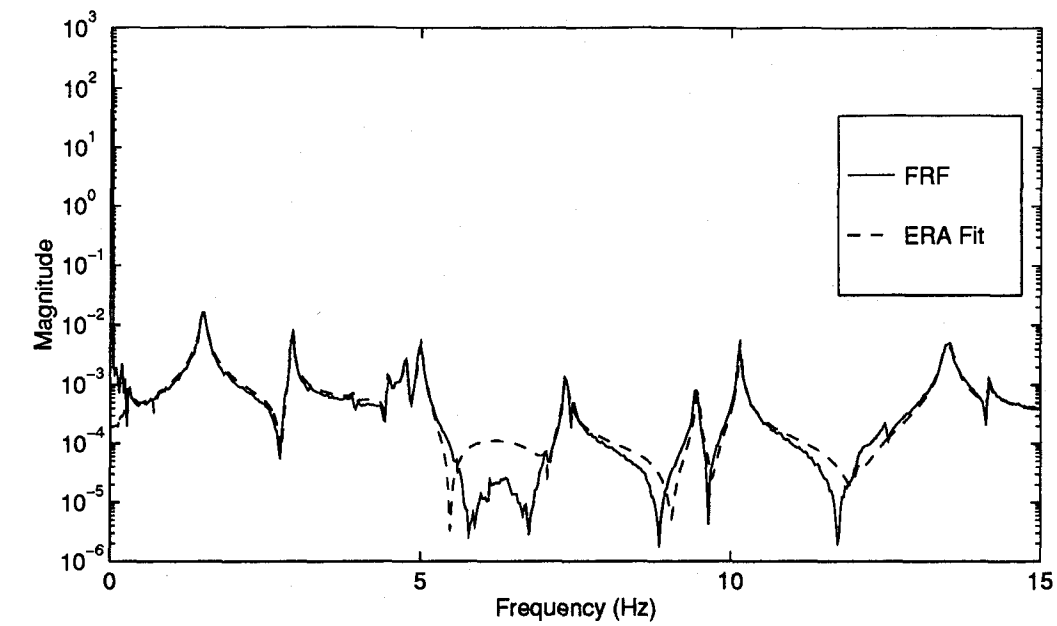


Figure B.8 Measurement 8 Excitation Actuator 2

Appendix C. State-Space Model FRFs of PACOSS DTA (Cont.)

The eight plots in this appendix are the measurements taken when Actuator 3 was excited with a random noise.

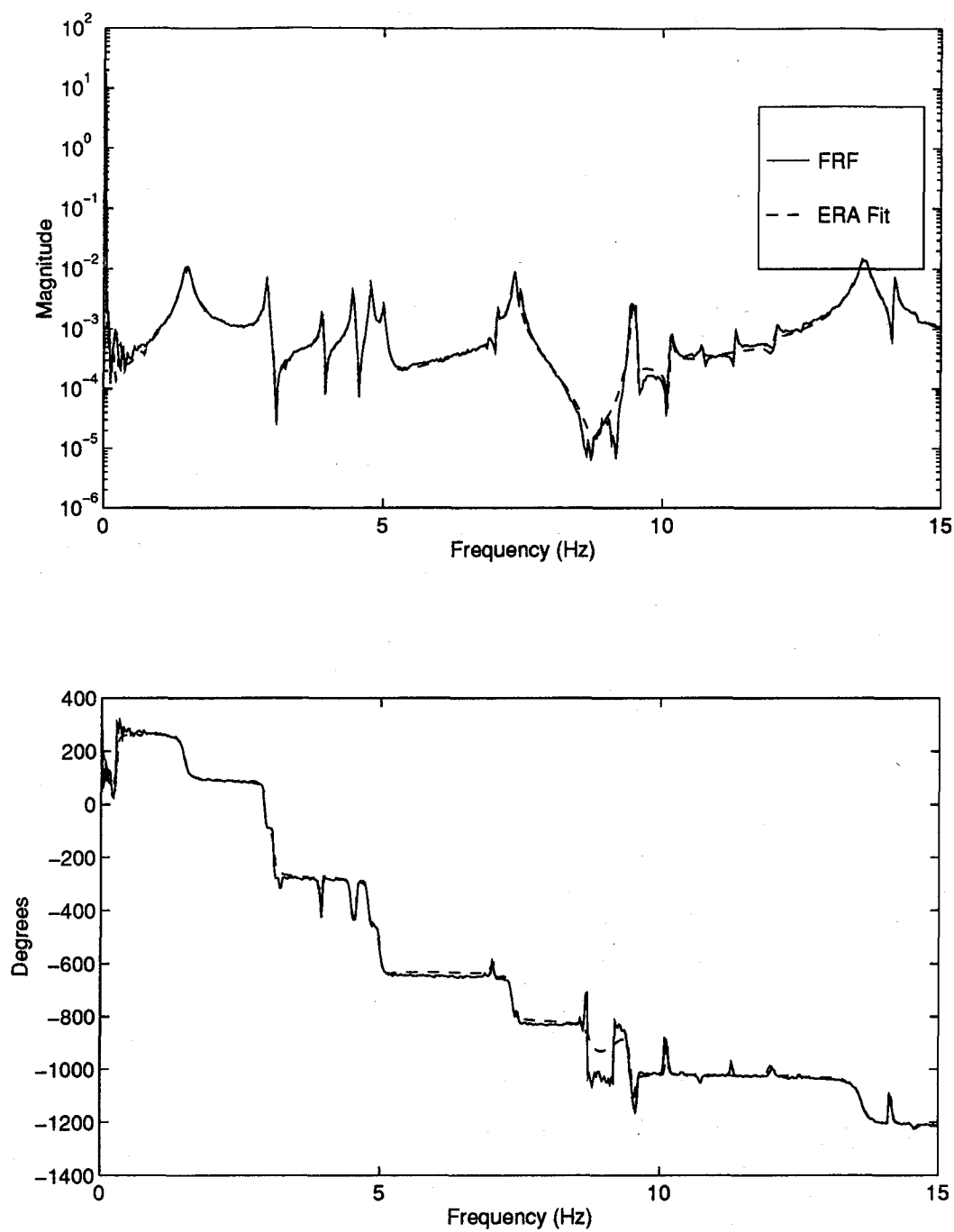


Figure C.1 Measurement 1 Excitation Actuator 3

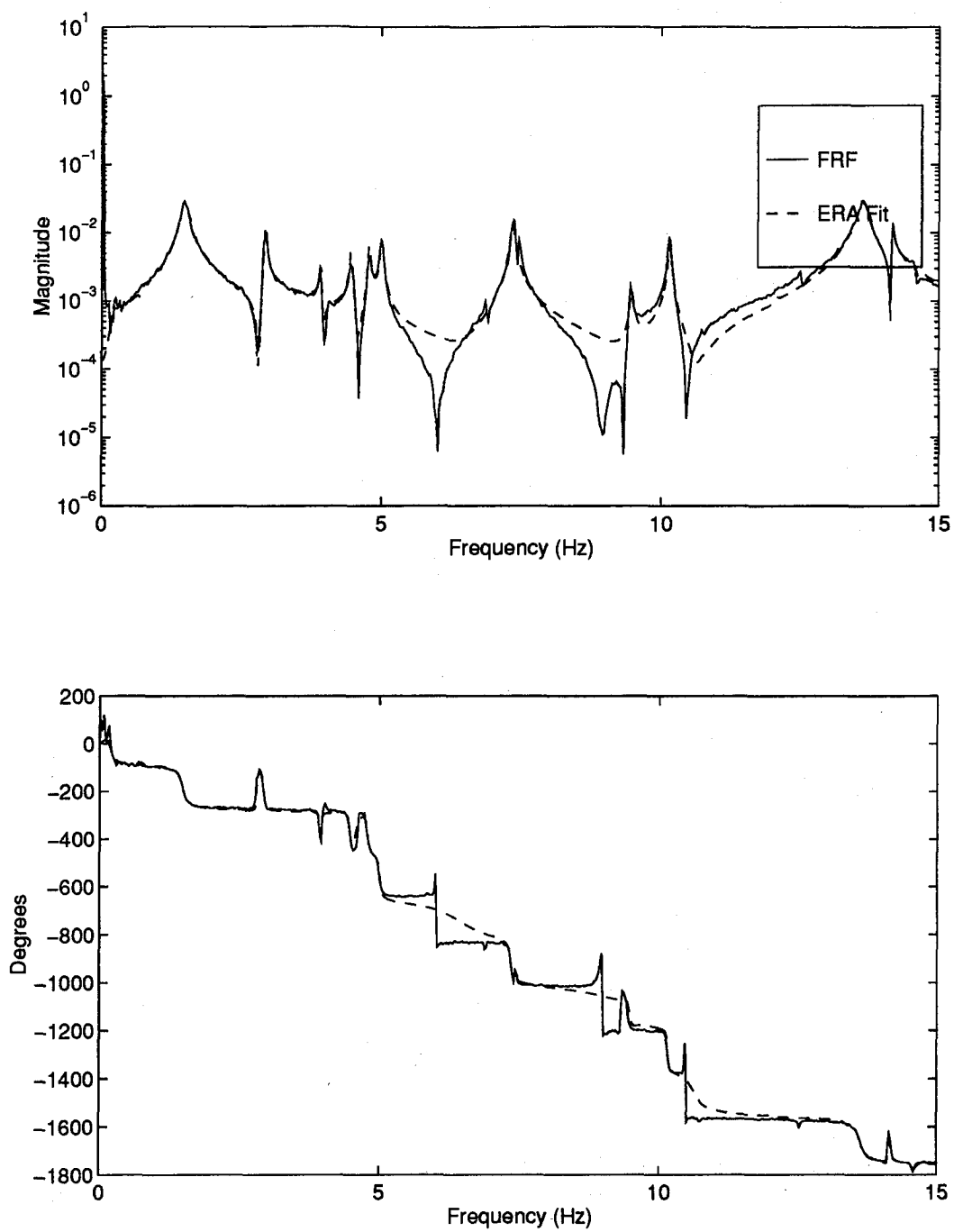


Figure C.2 Measurement 2 Excitation Actuator 3

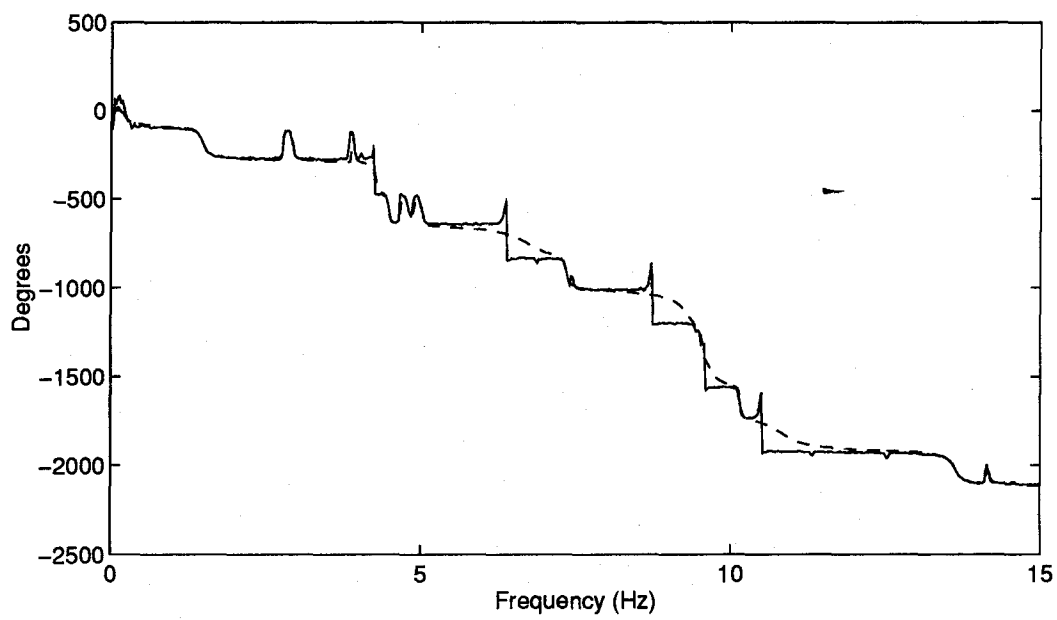
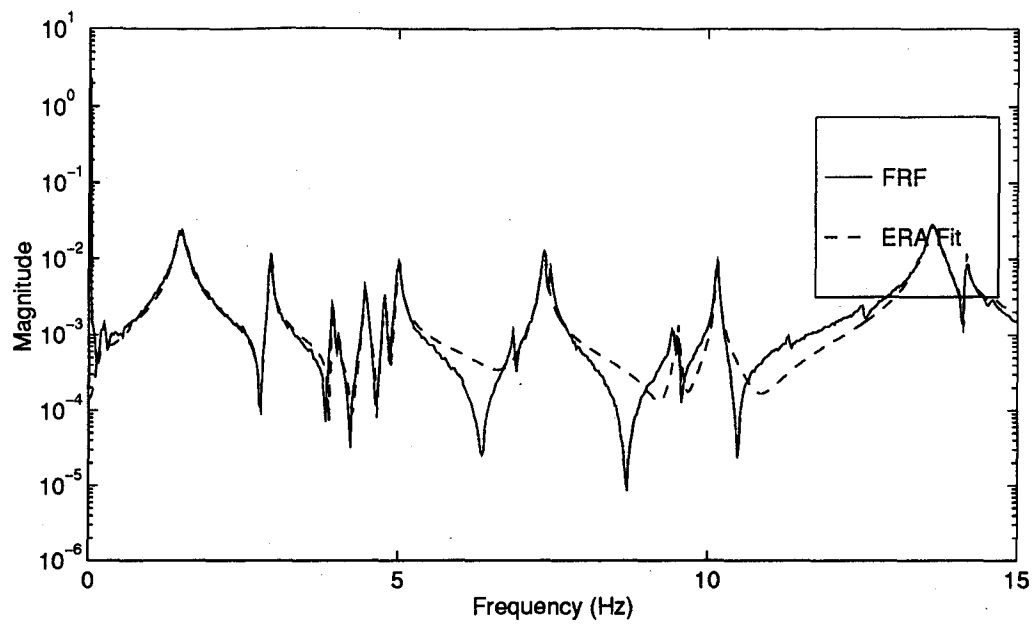


Figure C.3 Measurement 3 Excitation Actuator 3

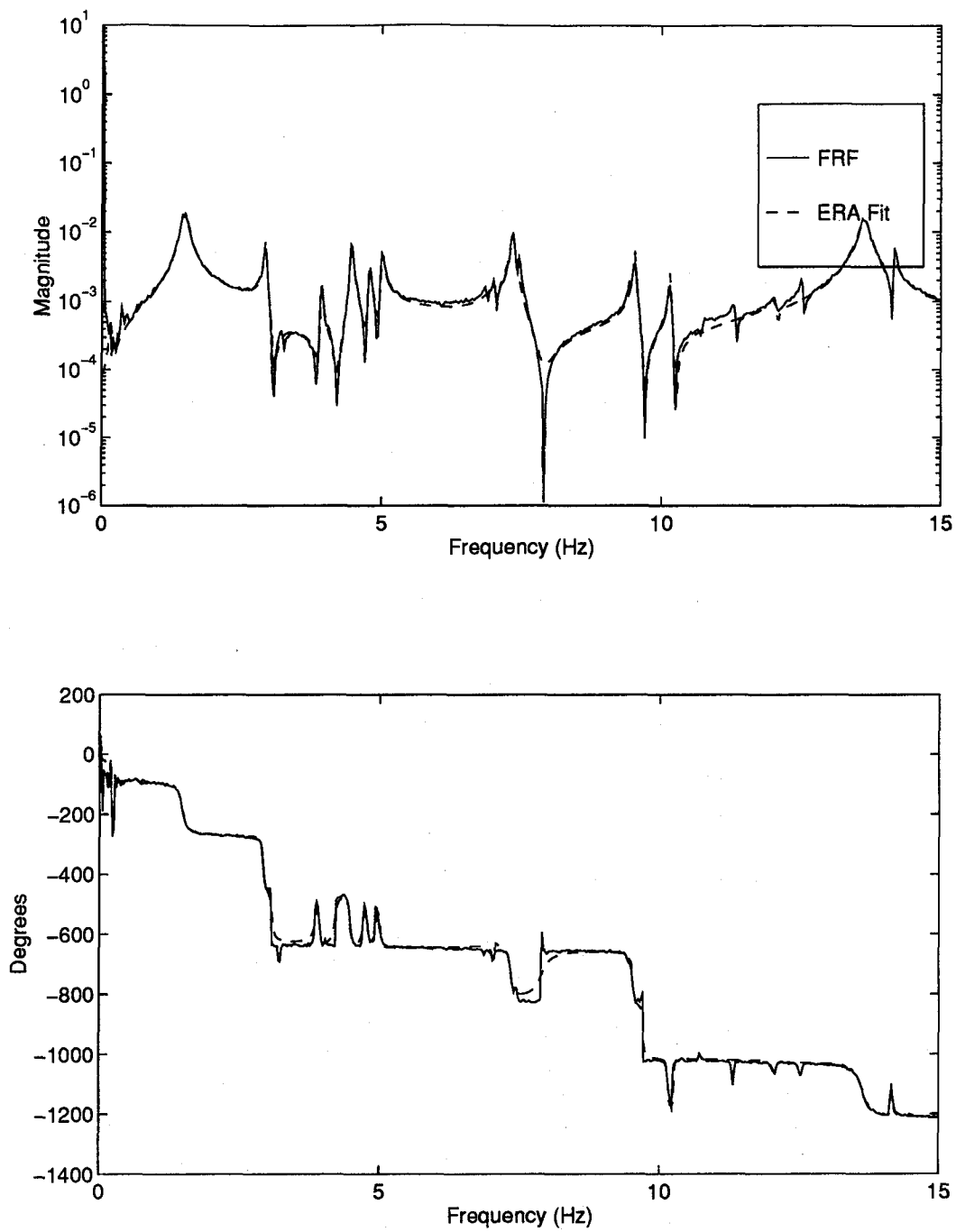


Figure C.4 Measurement 4 Excitation Actuator 3

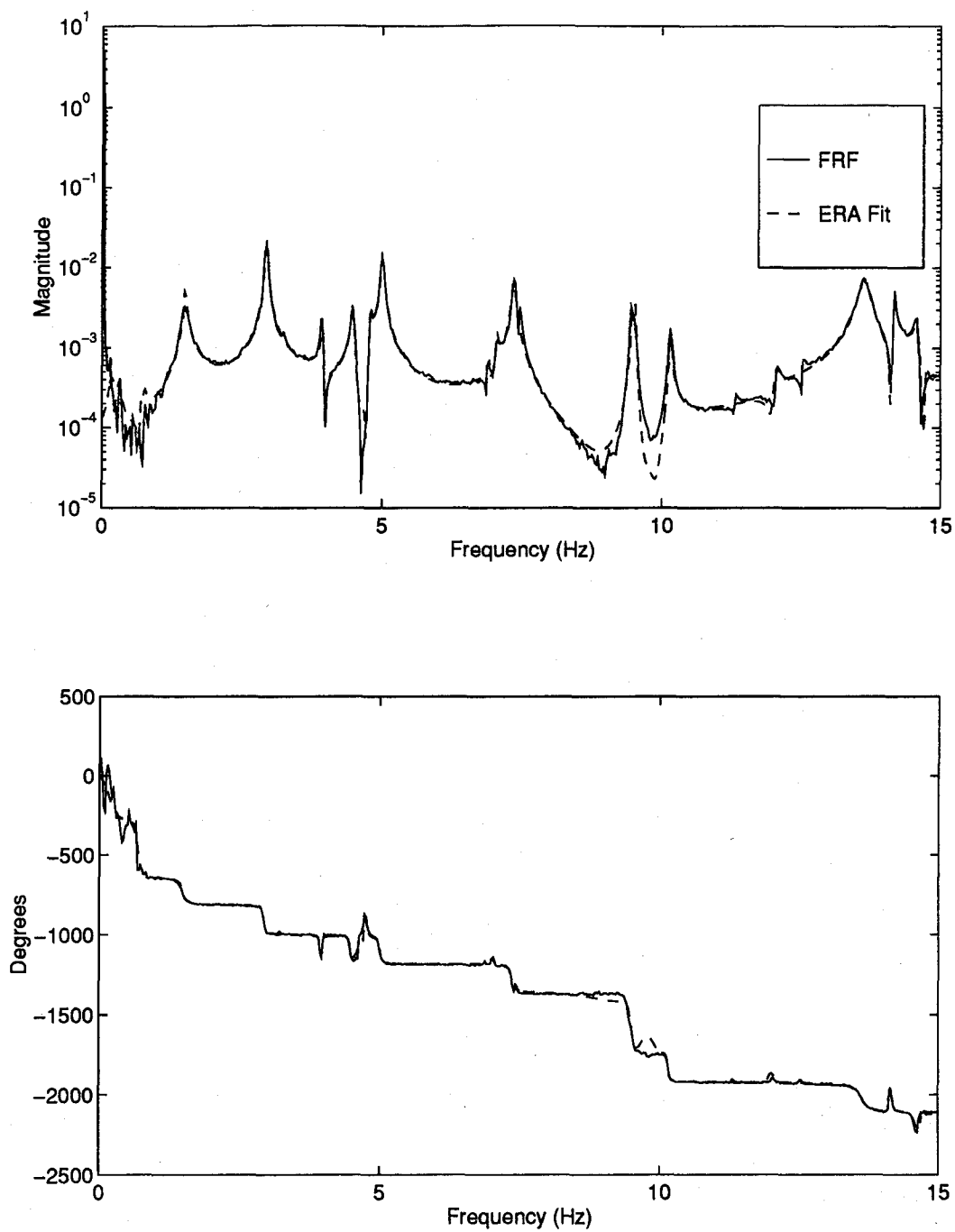


Figure C.5 Measurement 5 Excitation Actuator 3

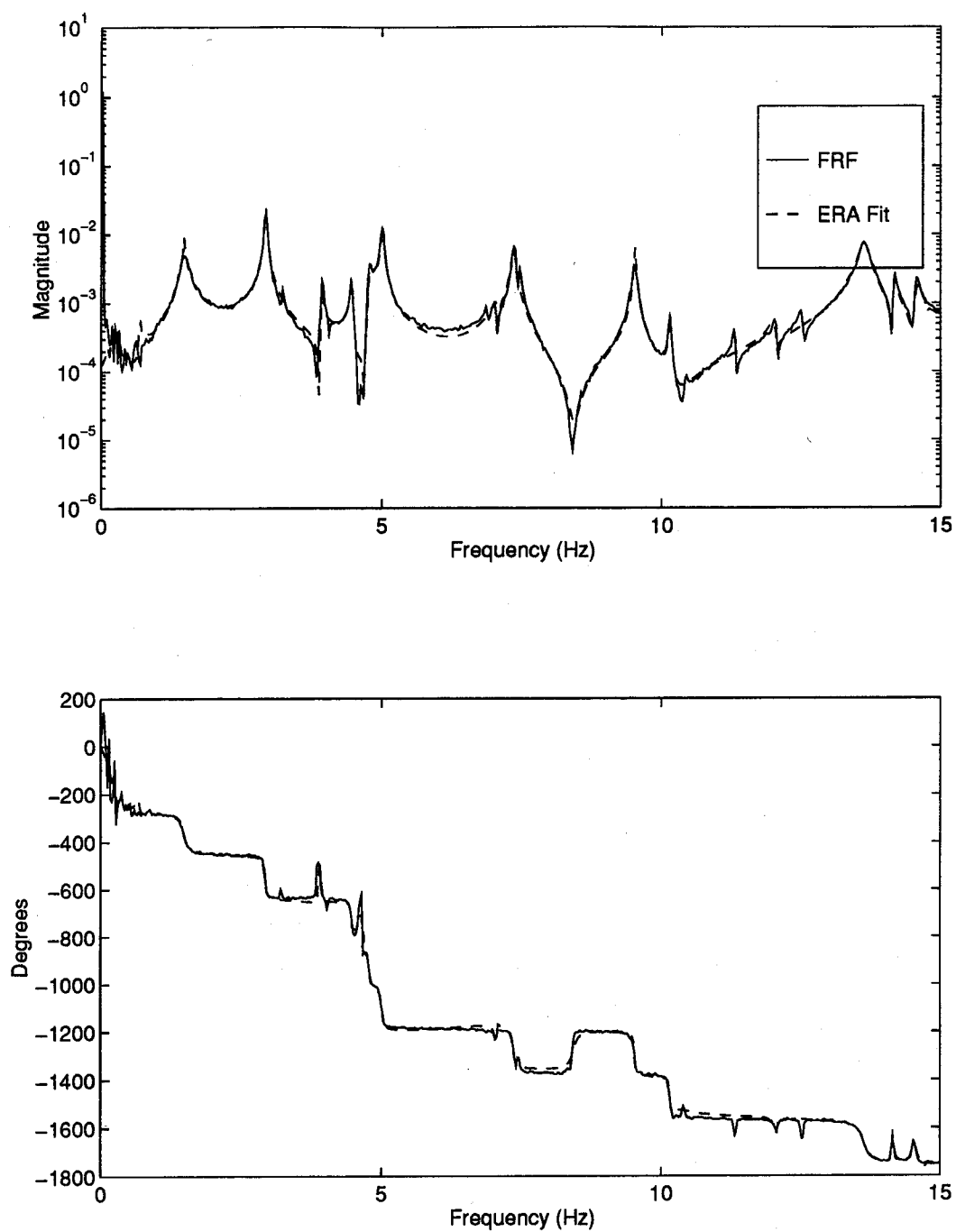


Figure C.6 Measurement 6 Excitation Actuator 3

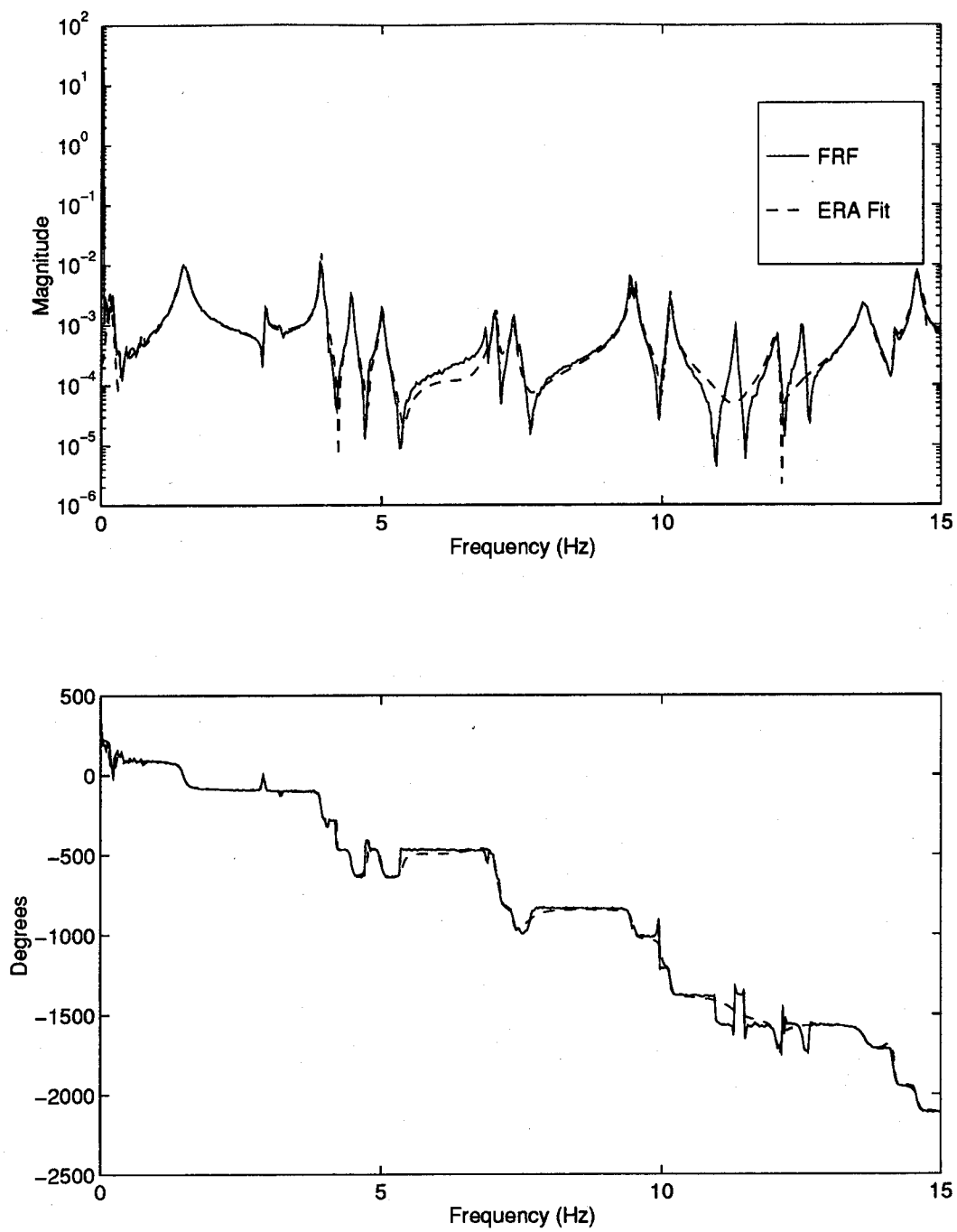


Figure C.7 Measurement 7 Excitation Actuator 3

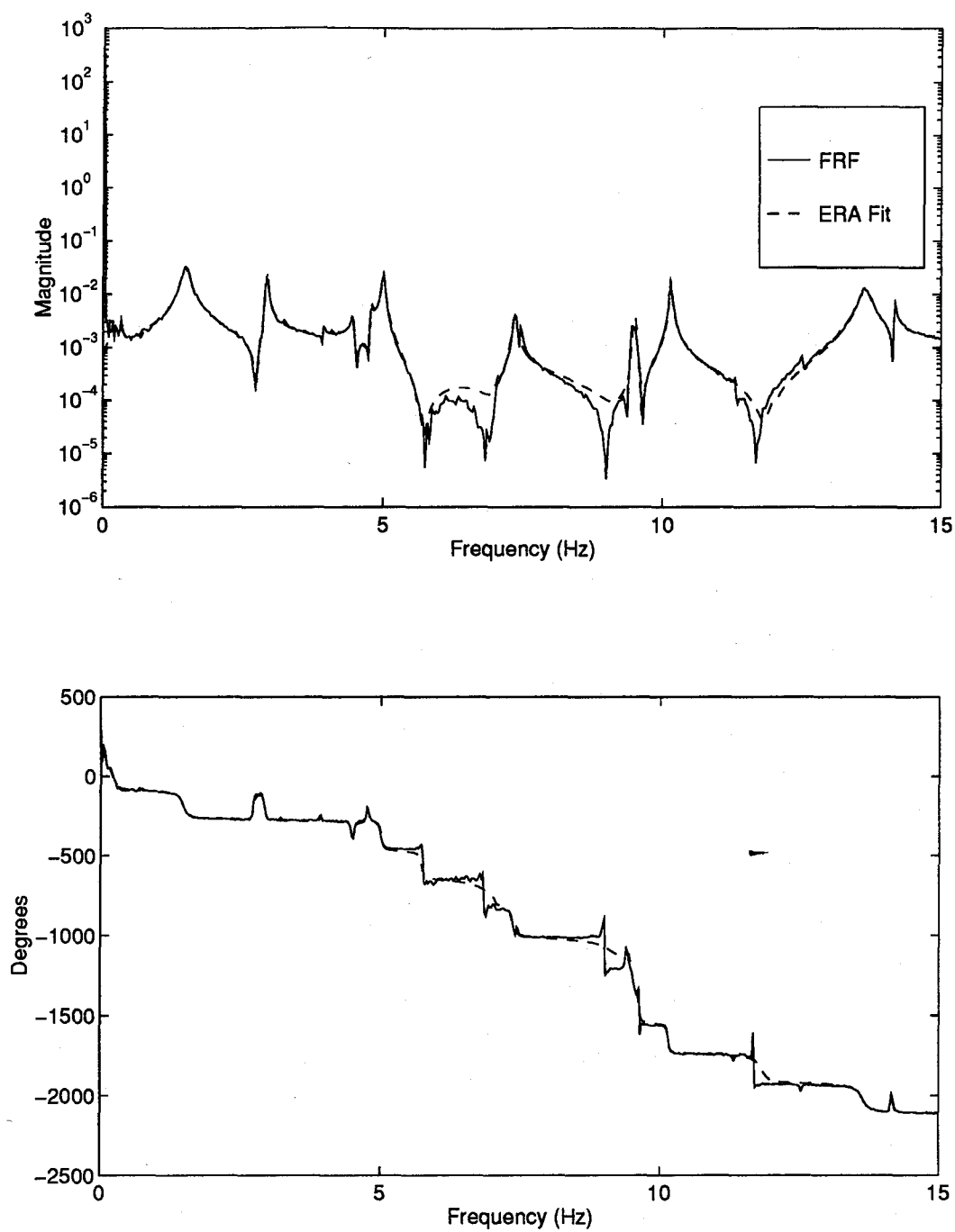


Figure C.8 Measurement 8 Excitation Actuator 3

Appendix D. State-Space Model FRFs of PACOSS DTA (Cont.)

The eight plots in this appendix are the measurements taken when Actuator 4 was excited with a random noise.

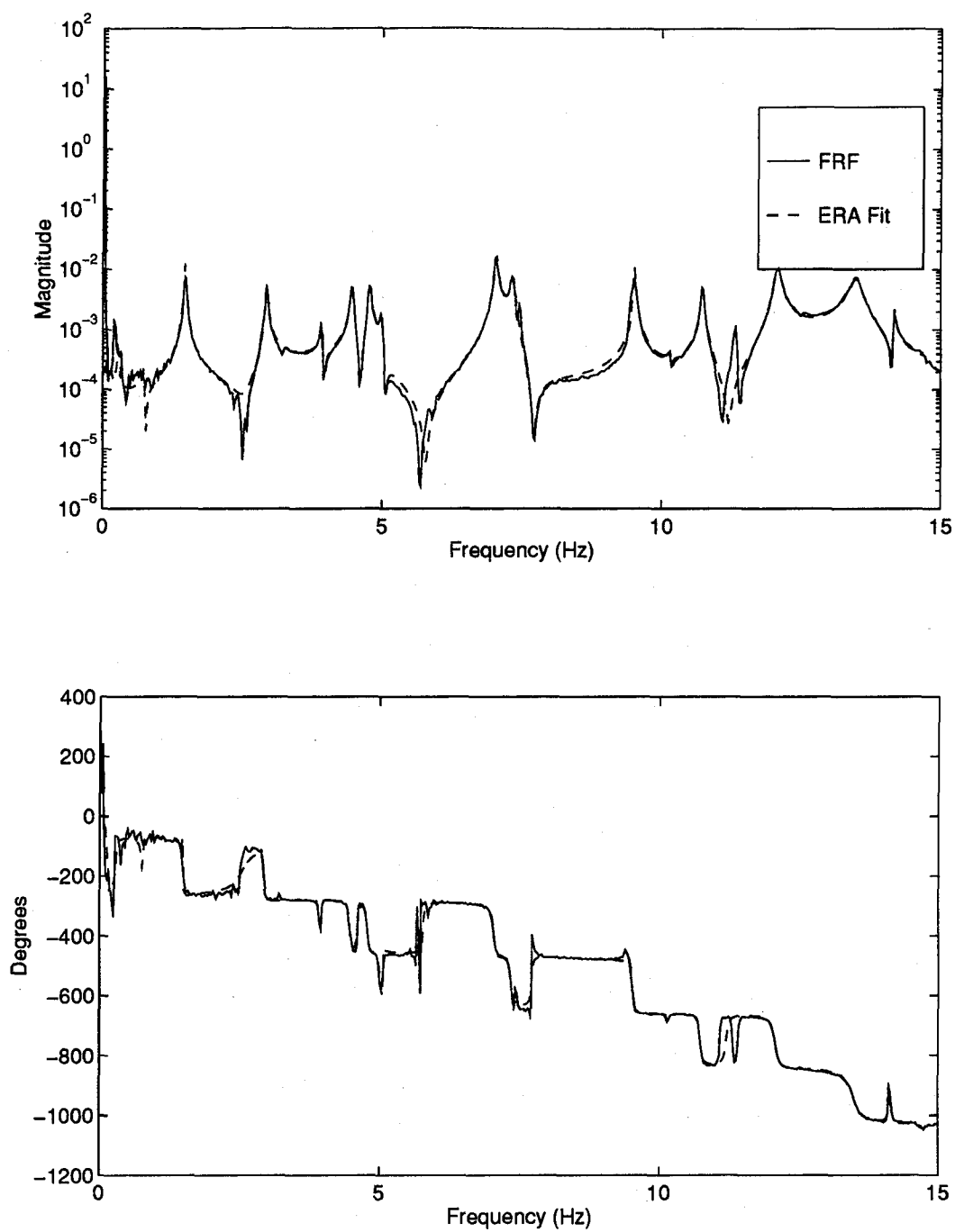


Figure D.1 Measurement 1 Excitation Actuator 4

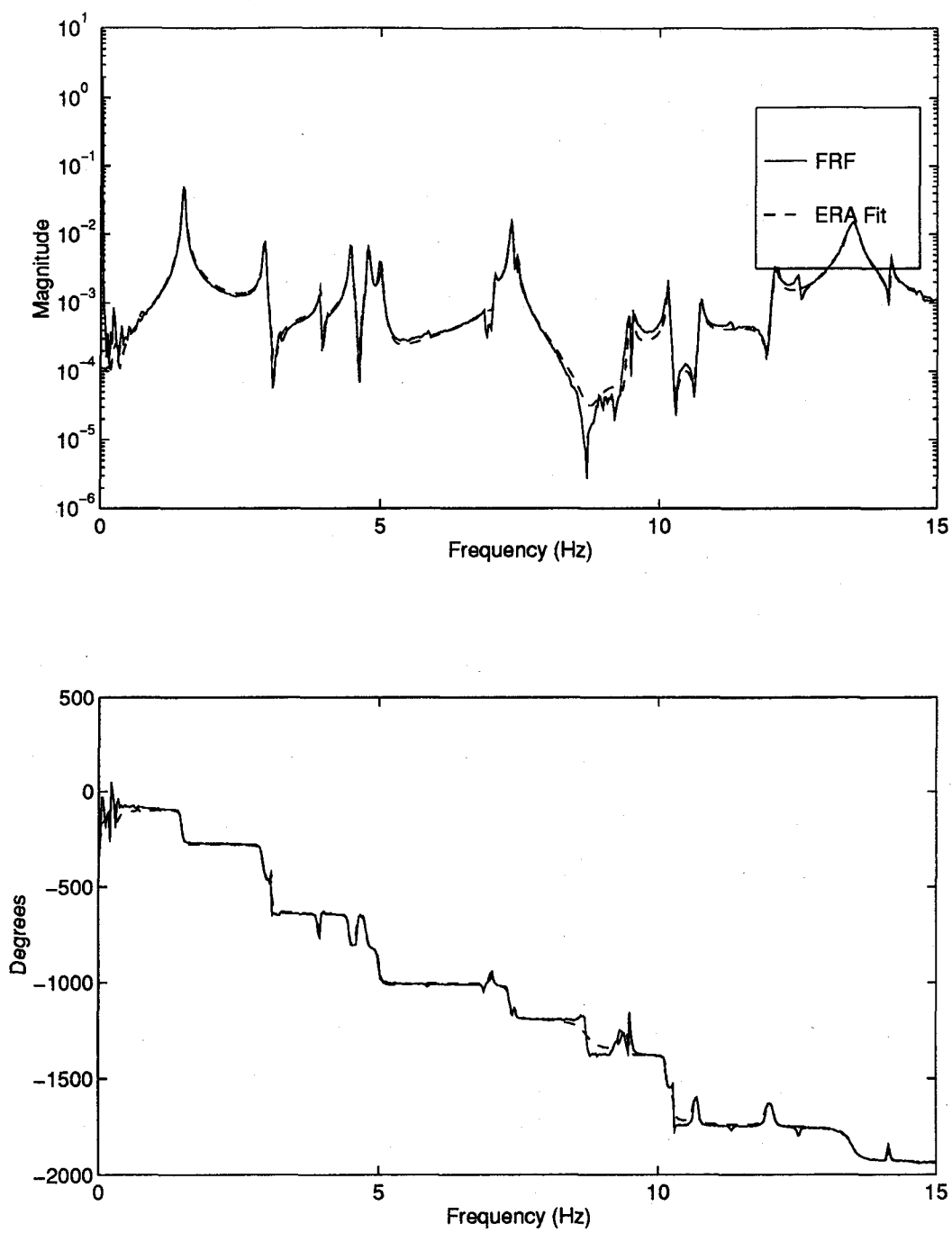


Figure D.2 Measurement 2 Excitation Actuator 4

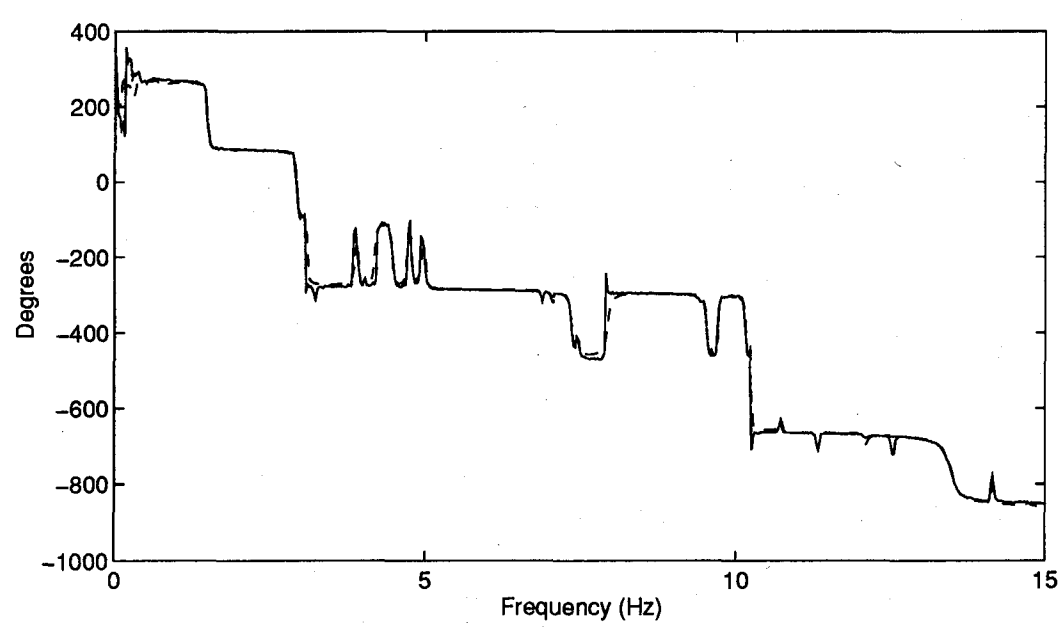
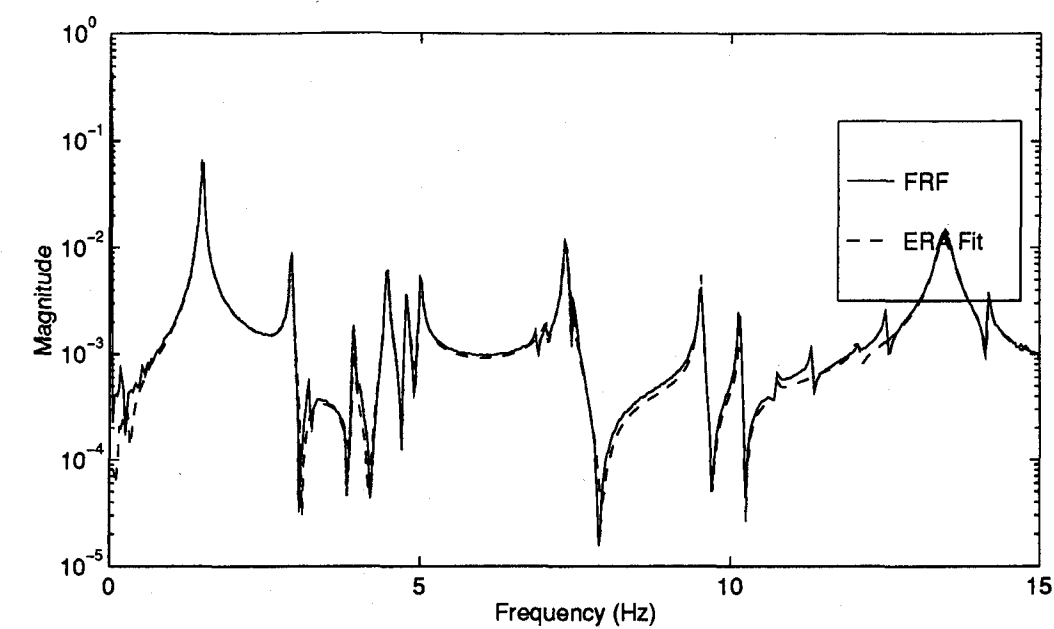


Figure D.3 Measurement 3 Excitation Actuator 4

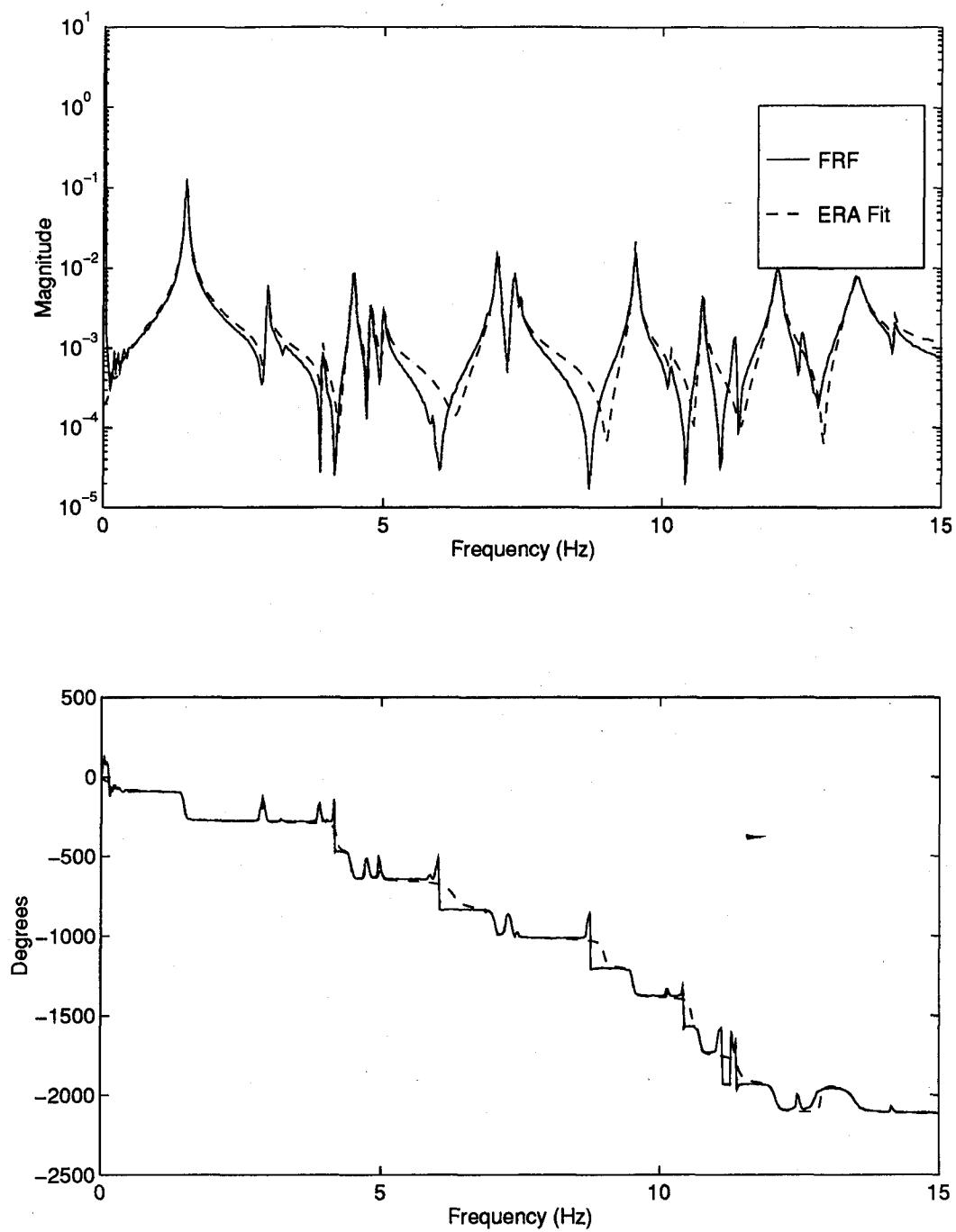


Figure D.4 Measurement 4 Excitation Actuator 4

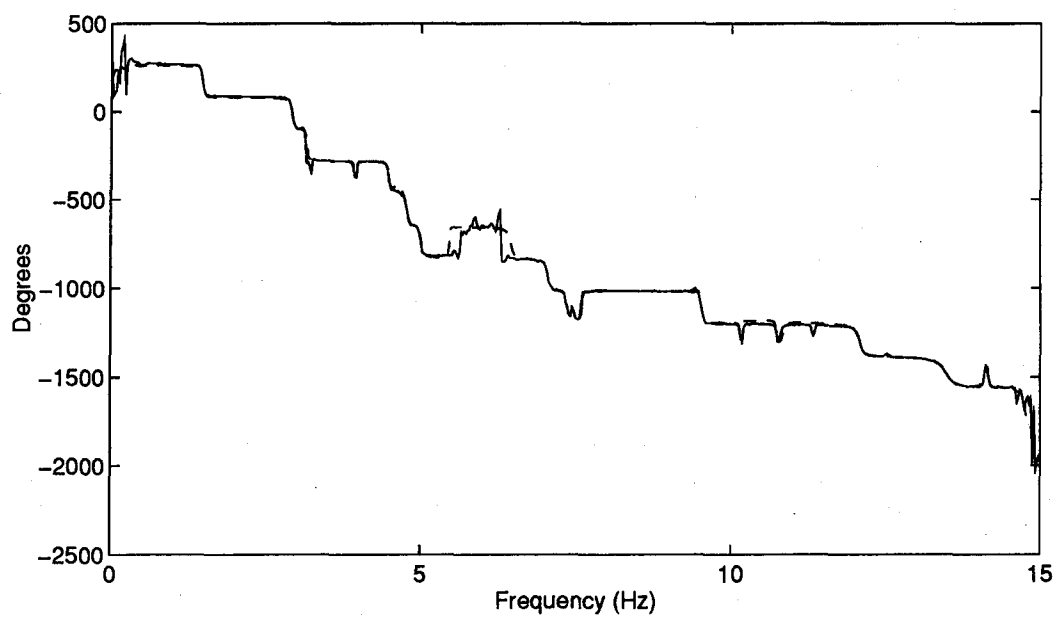
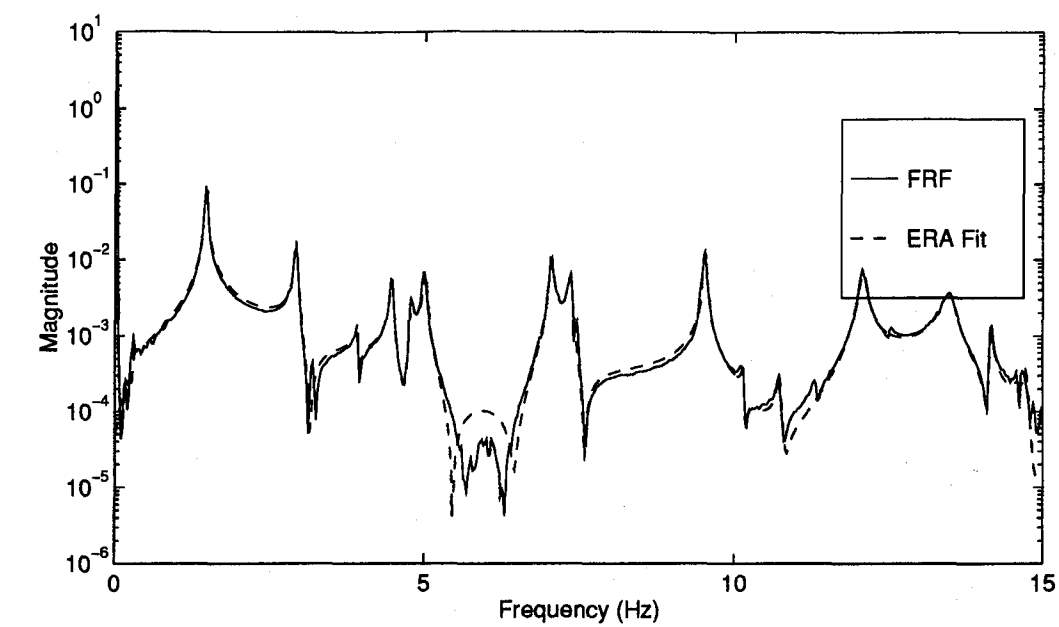


Figure D.5 Measurement 5 Excitation Actuator 4

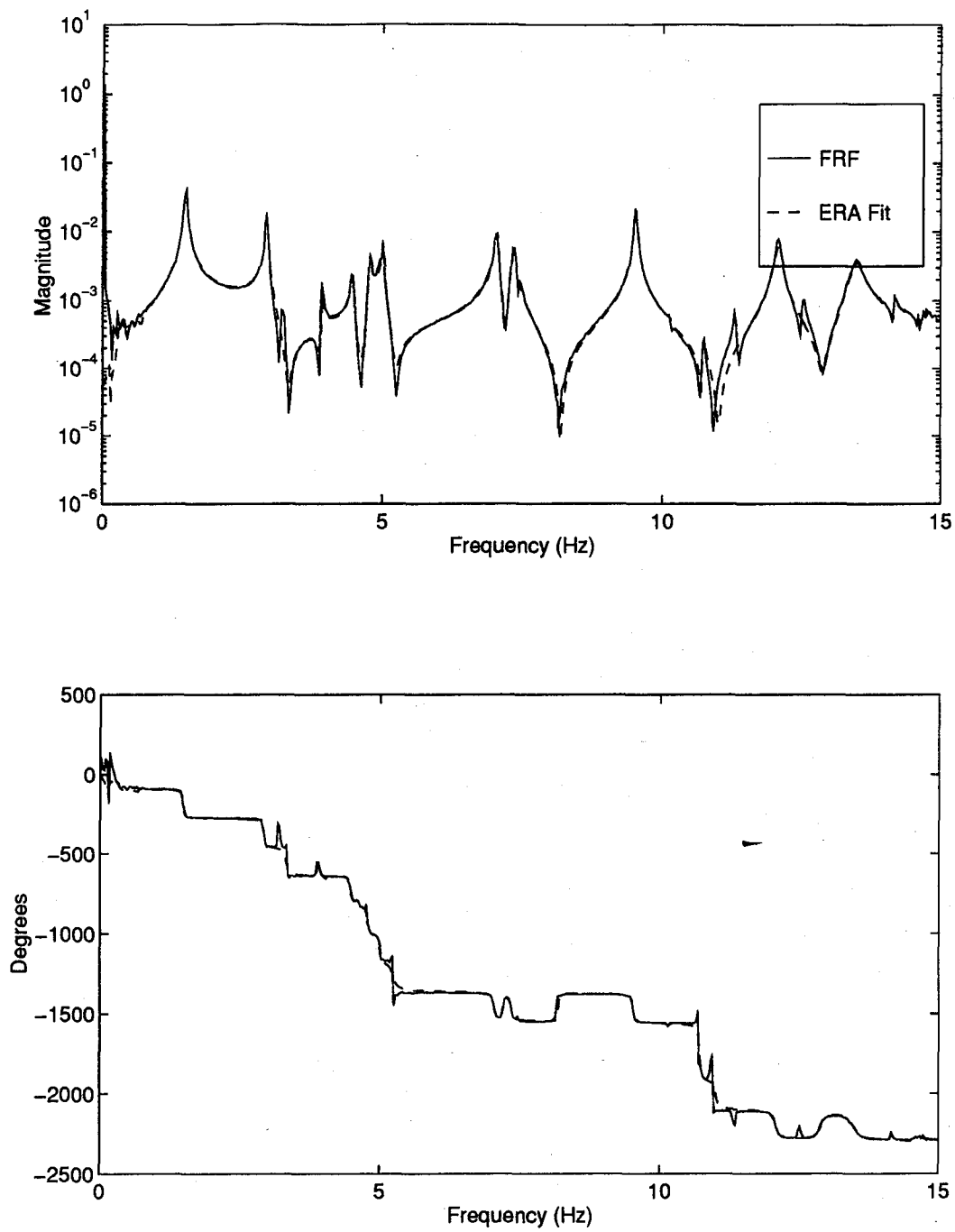


Figure D.6 Measurement 6 Excitation Actuator 4

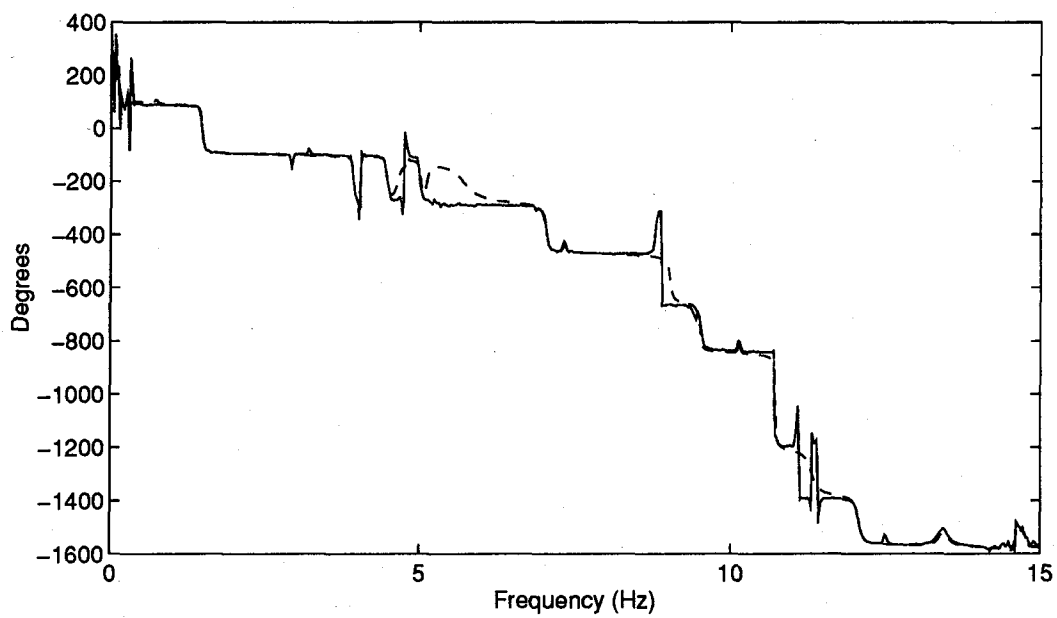
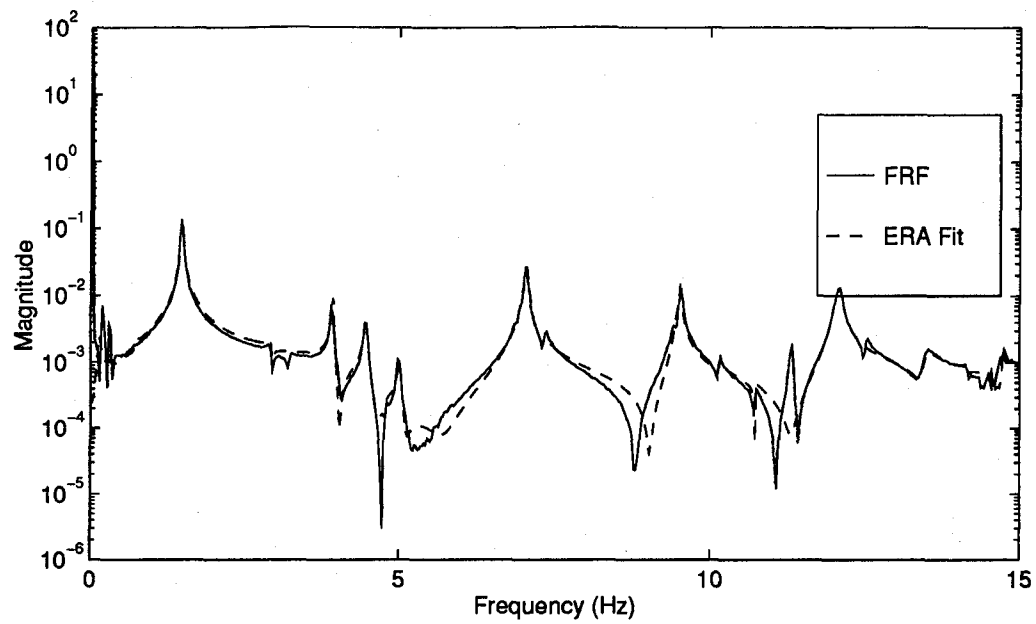


Figure D.7 Measurement 7 Excitation Actuator 4

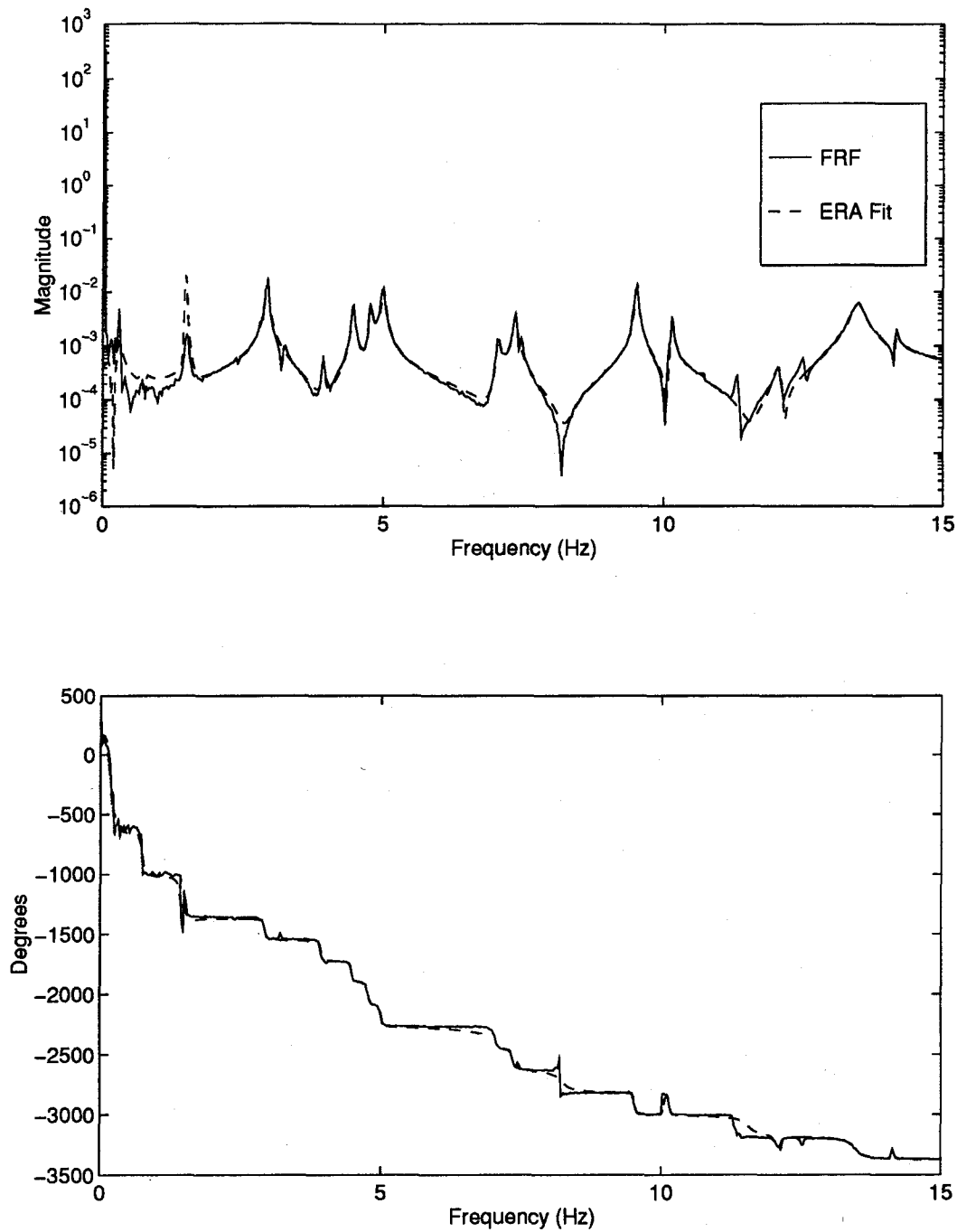


Figure D.8 Measurement 8 Excitation Actuator 4

Appendix E. State-Space Model FRFs of PACOSS DTA (Cont.)

The eight plots in this appendix are the measurements taken when Actuator 5 was excited with a random noise.

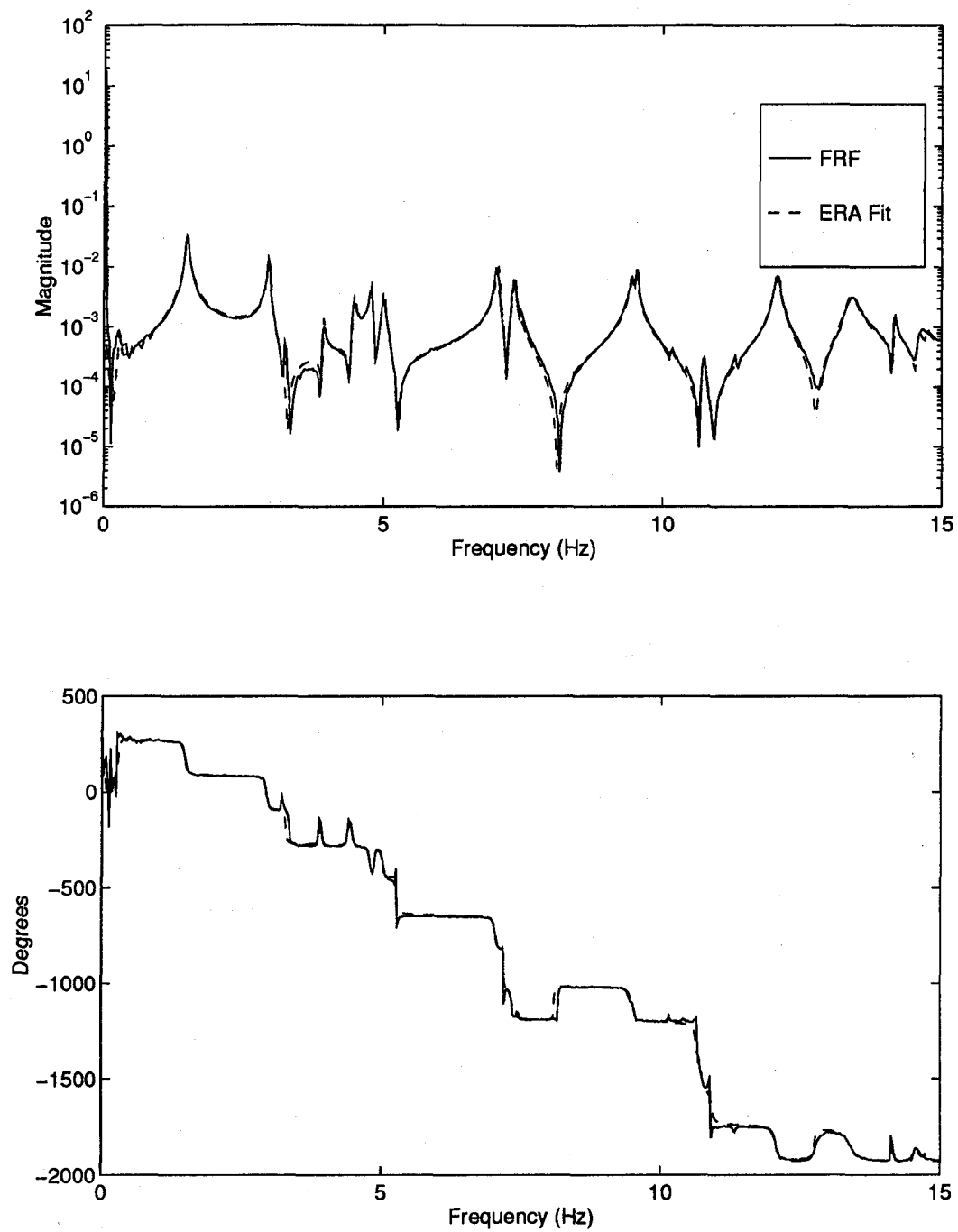


Figure E.1 Measurement 1 Excitation Actuator 5

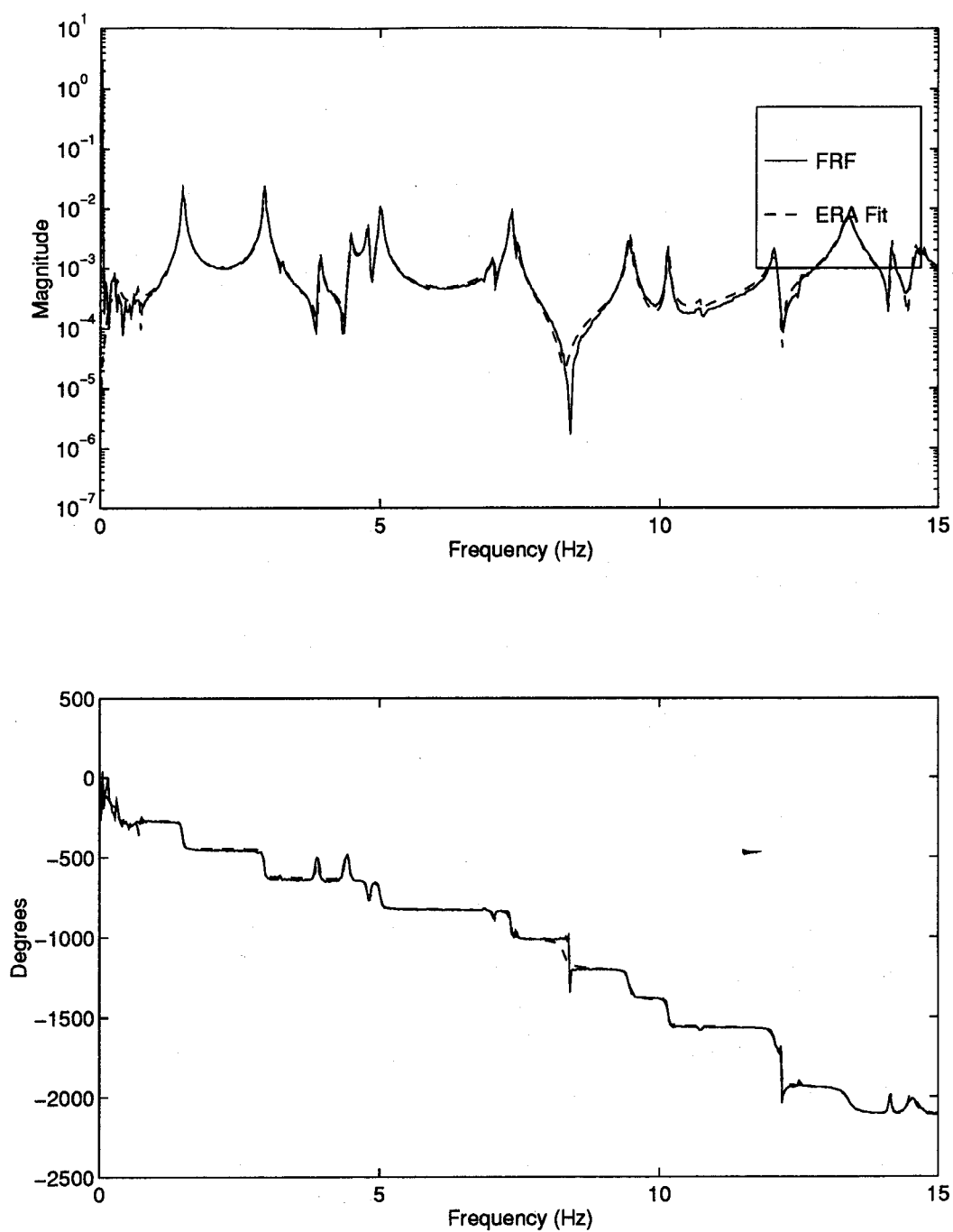


Figure E.2 Measurement 2 Excitation Actuator 5

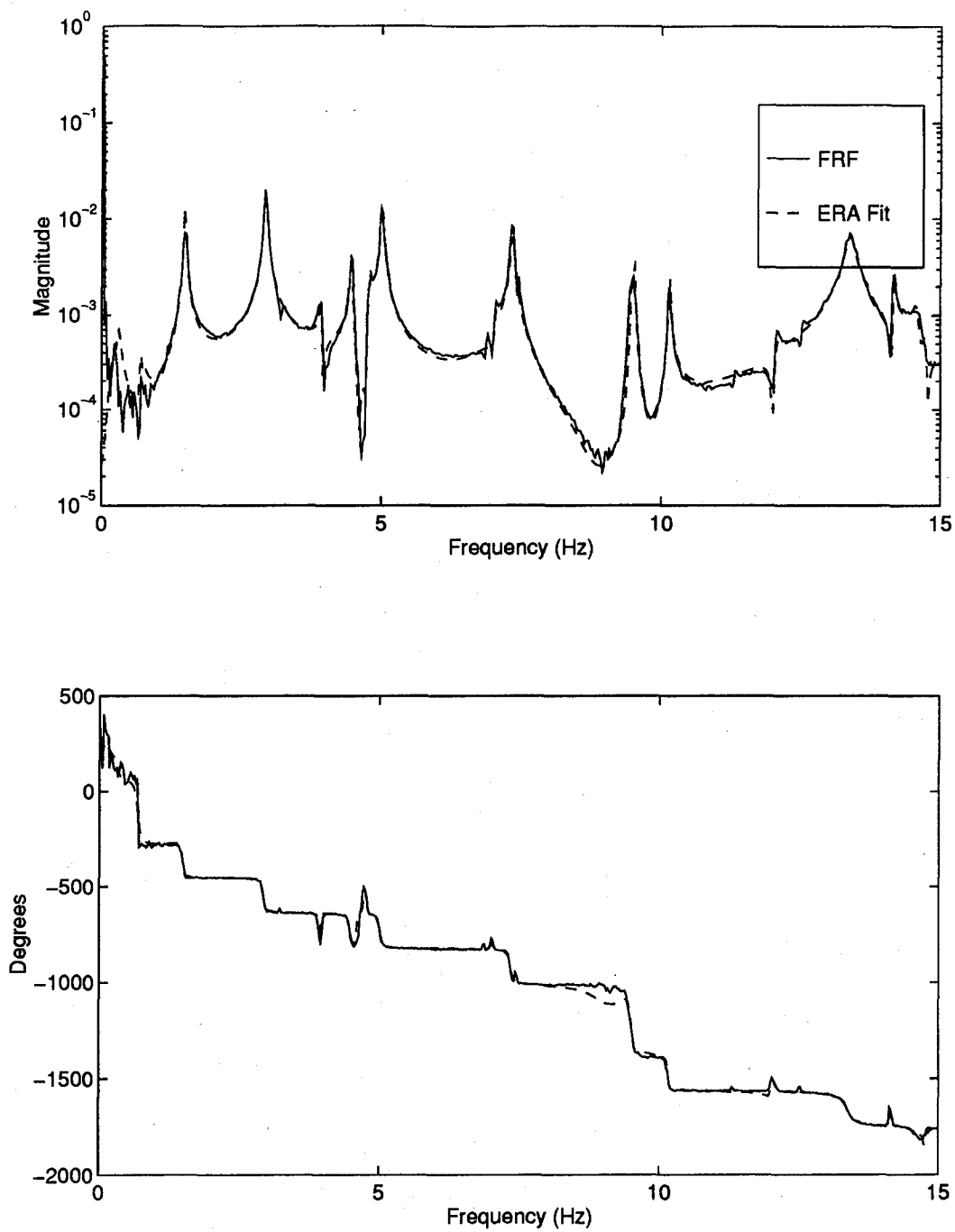


Figure E.3 Measurement 3 Excitation Actuator 5

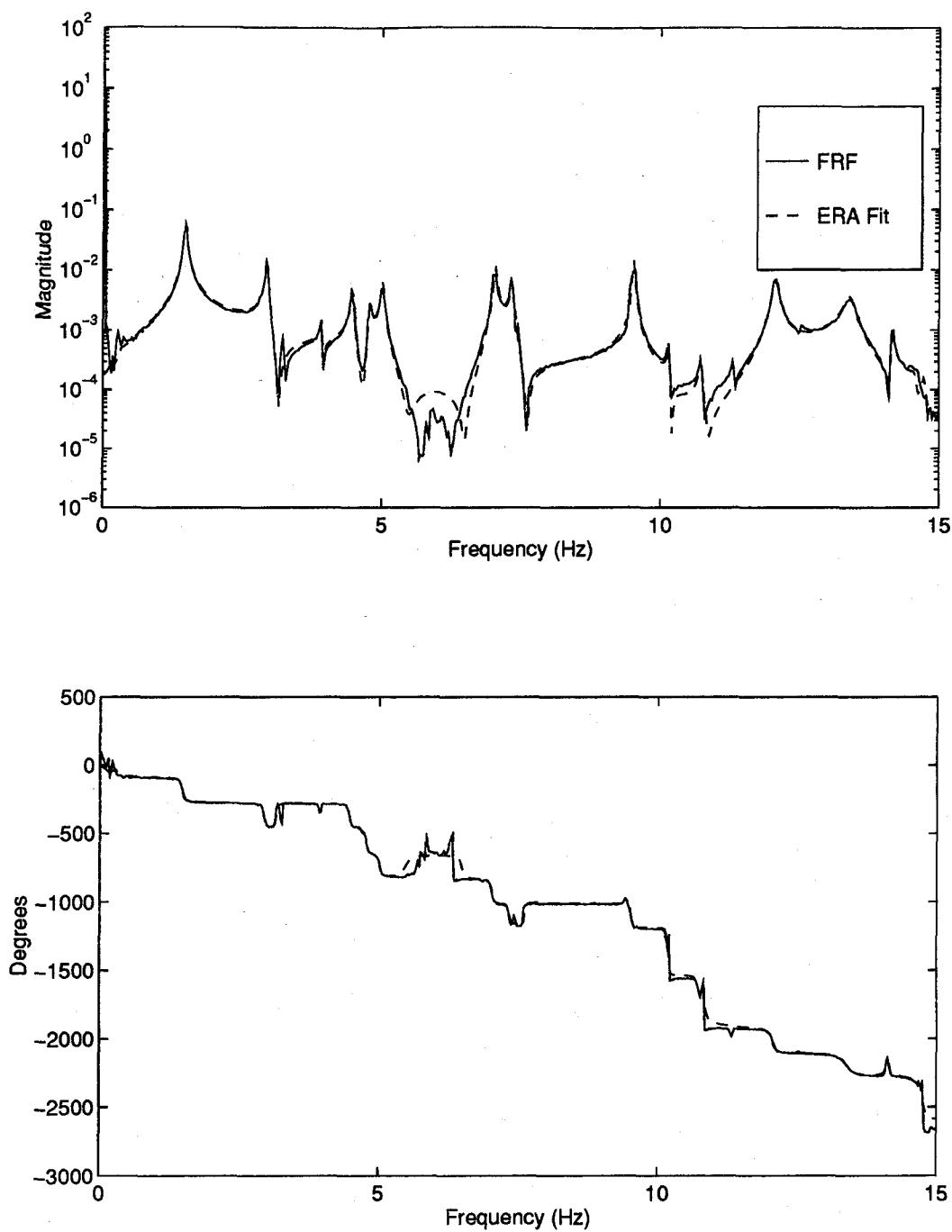


Figure E.4 Measurement 4 Excitation Actuator 5

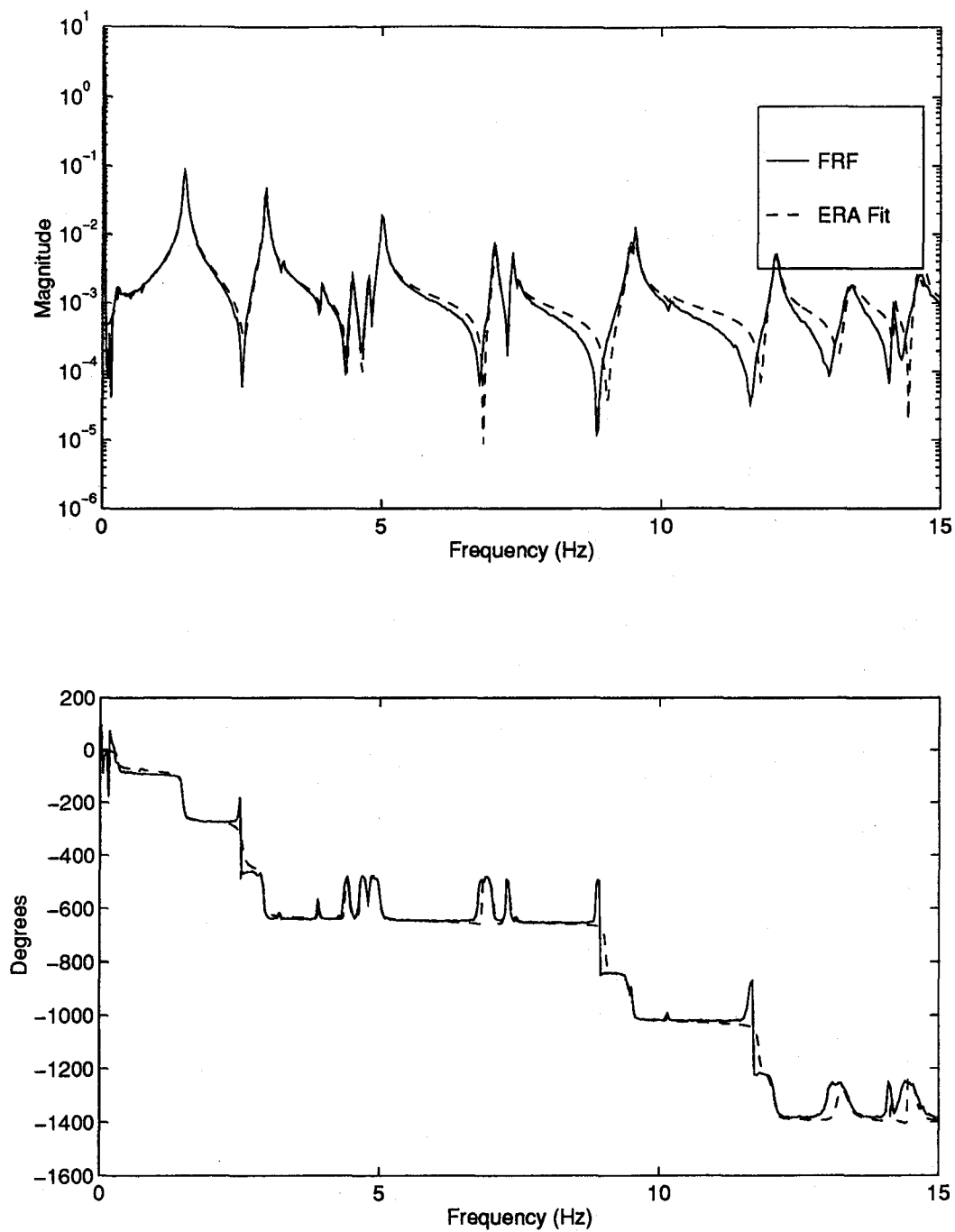


Figure E.5 Measurement 5 Excitation Actuator 5

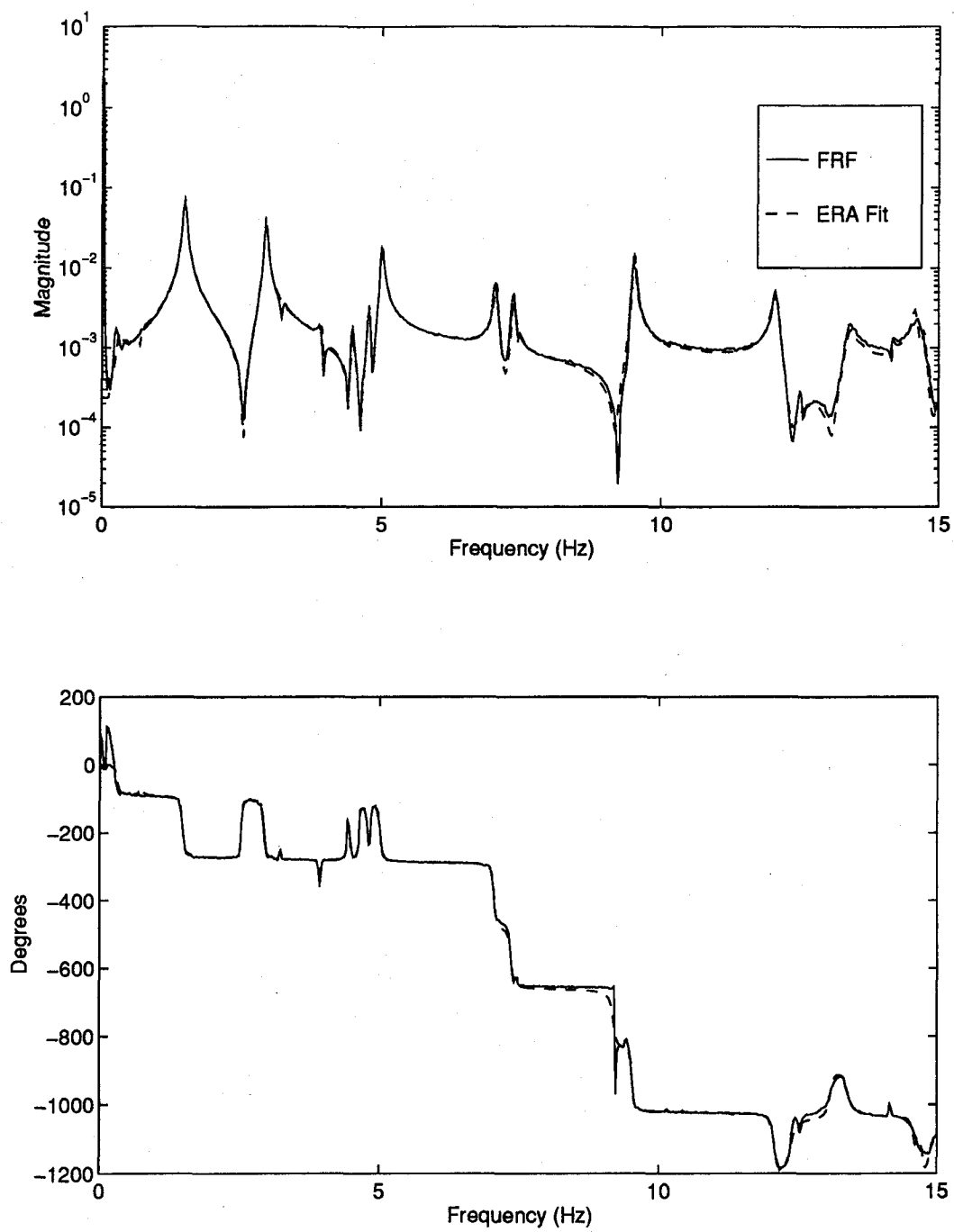


Figure E.6 Measurement 6 Excitation Actuator 5

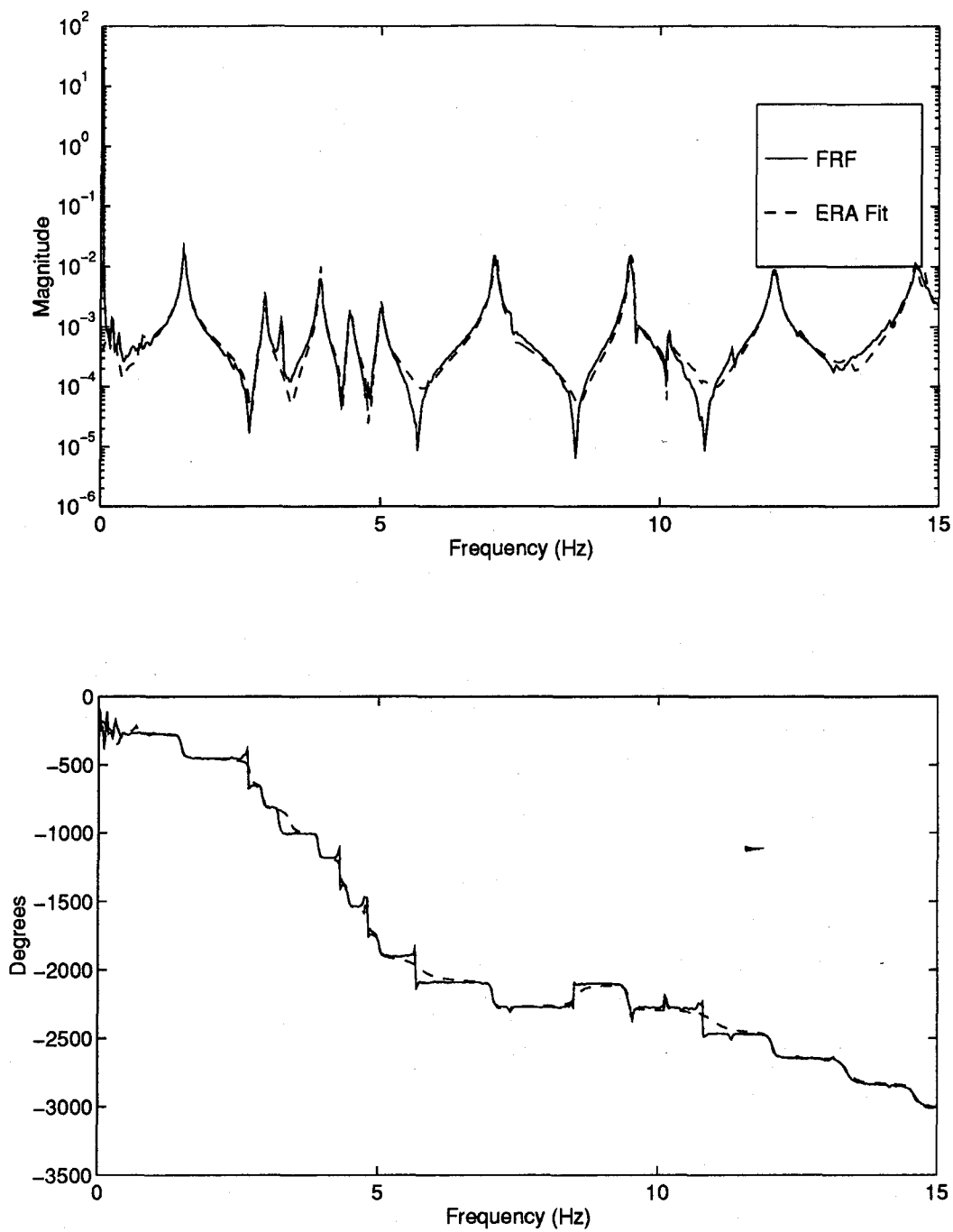


Figure E.7 Measurement 7 Excitation Actuator 5

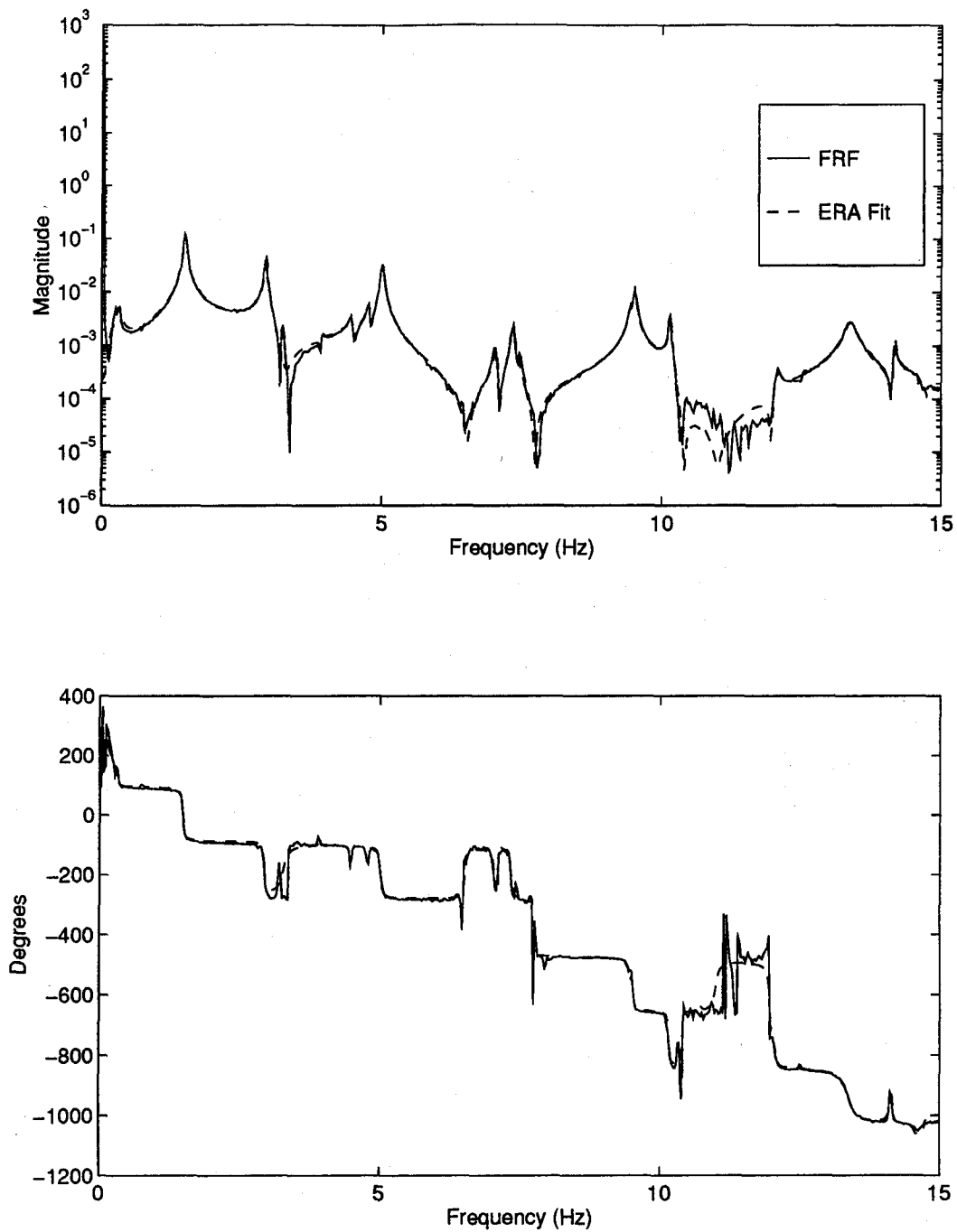


Figure E.8 Measurement 8 Excitation Actuator 5

Appendix F. State-Space Model FRFs of PACOSS DTA (Cont.)

The eight plots in this appendix are the measurements taken when Actuator 6 was excited with a random noise.

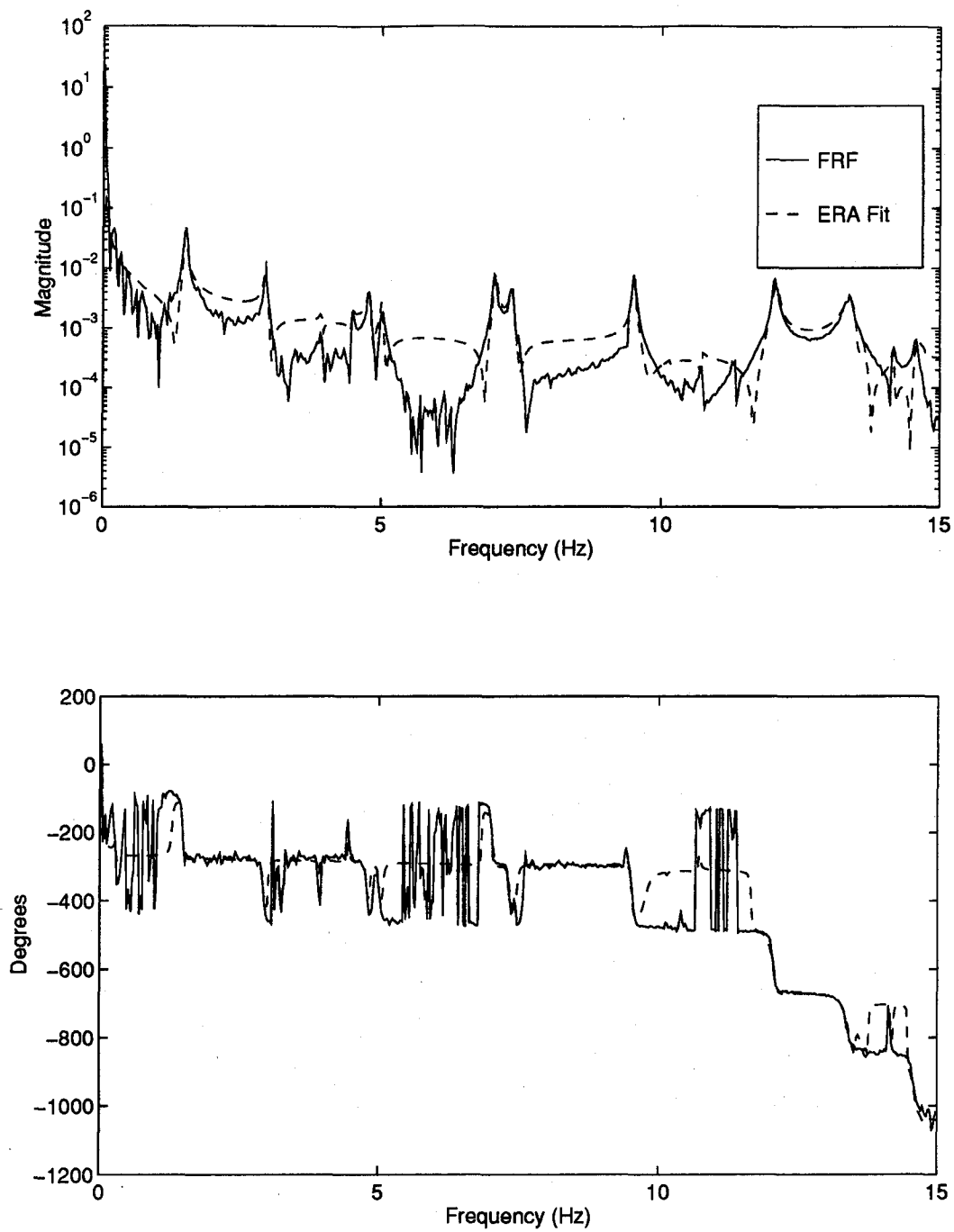


Figure F.1 Measurement 1 Excitation Actuator 6

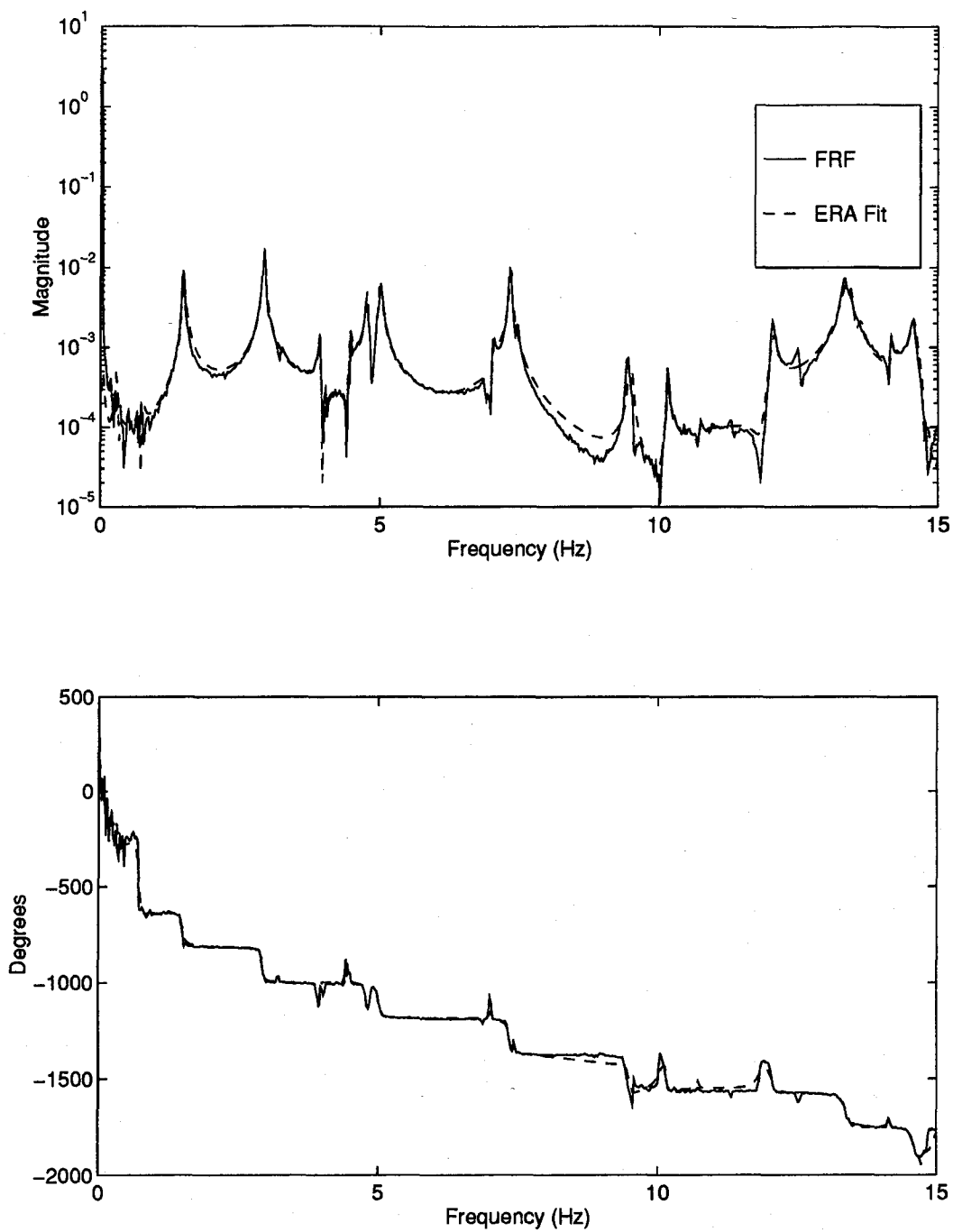


Figure F.2 Measurement 2 Excitation Actuator 6

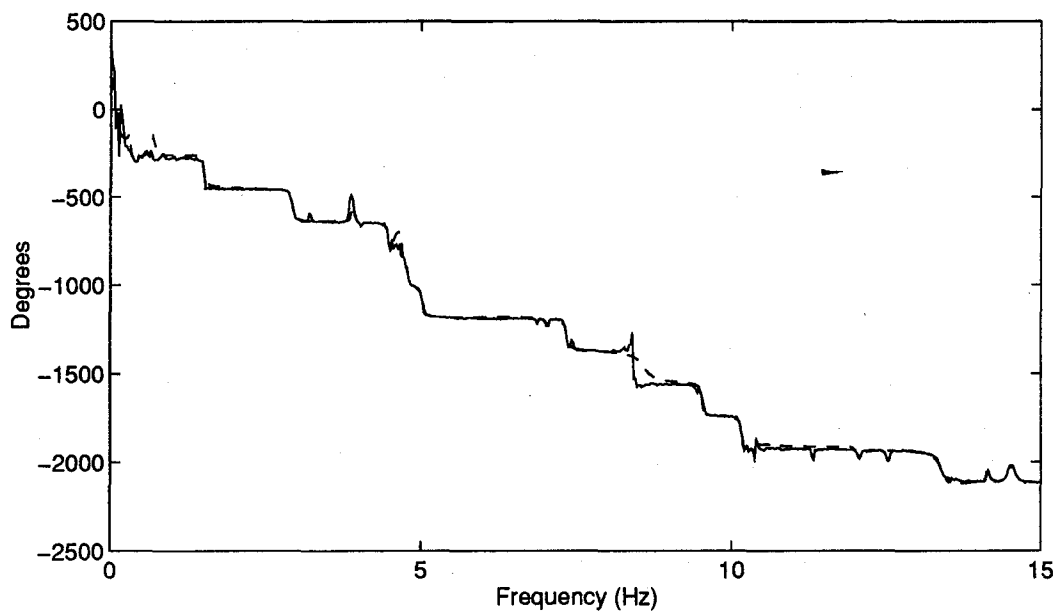
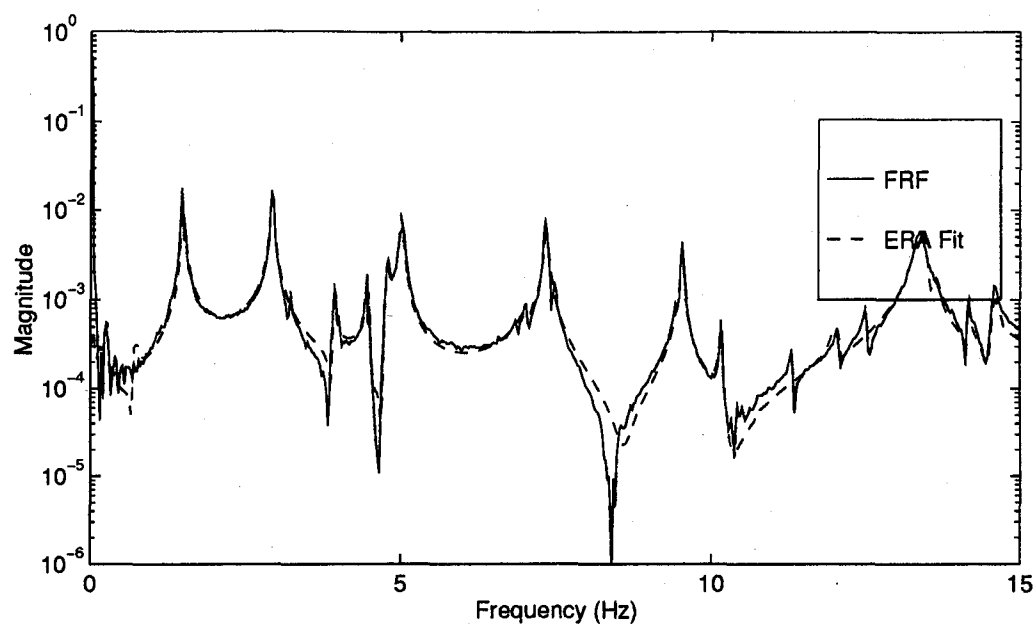


Figure F.3 Measurement 3 Excitation Actuator 6

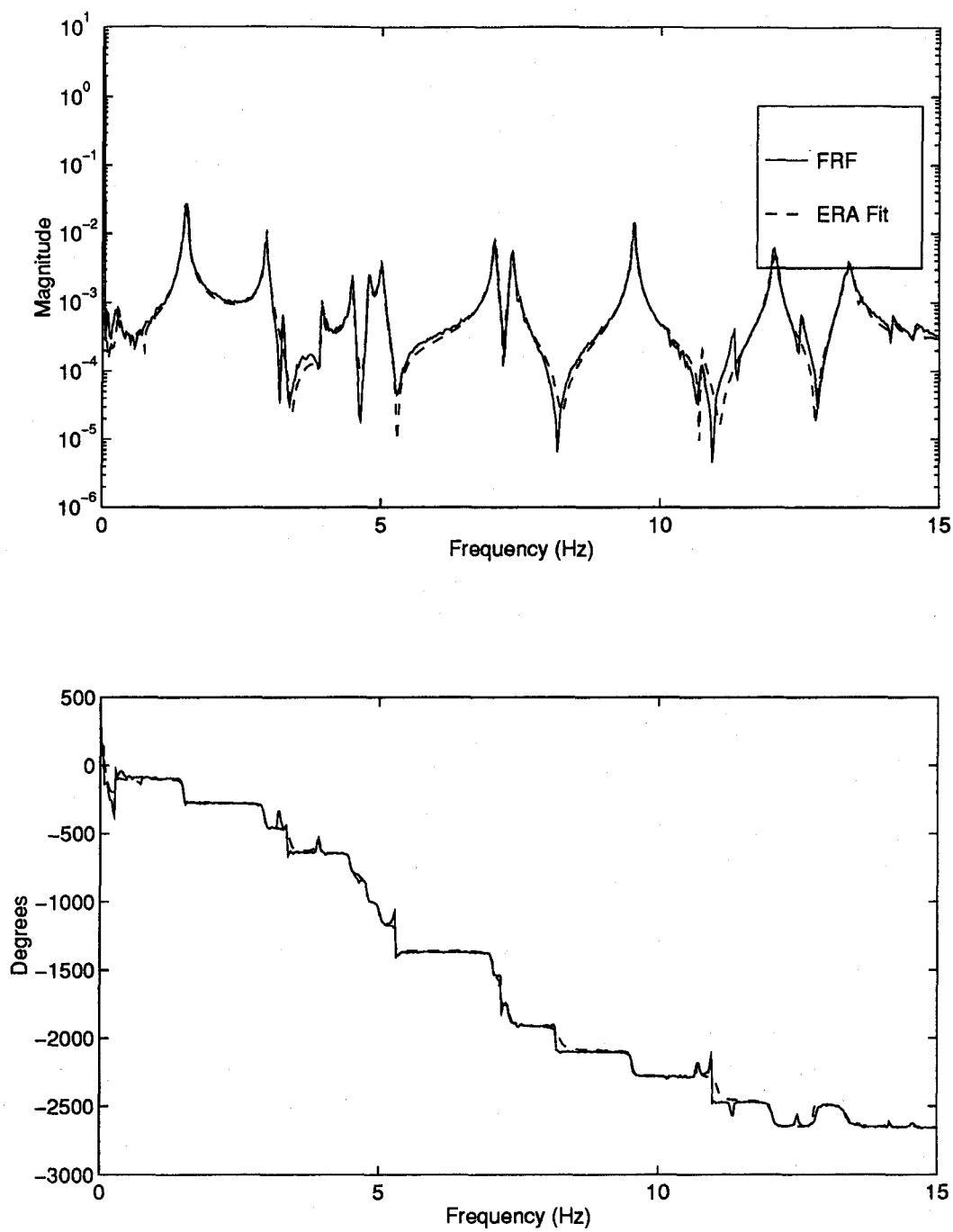


Figure F.4 Measurement 4 Excitation Actuator 6

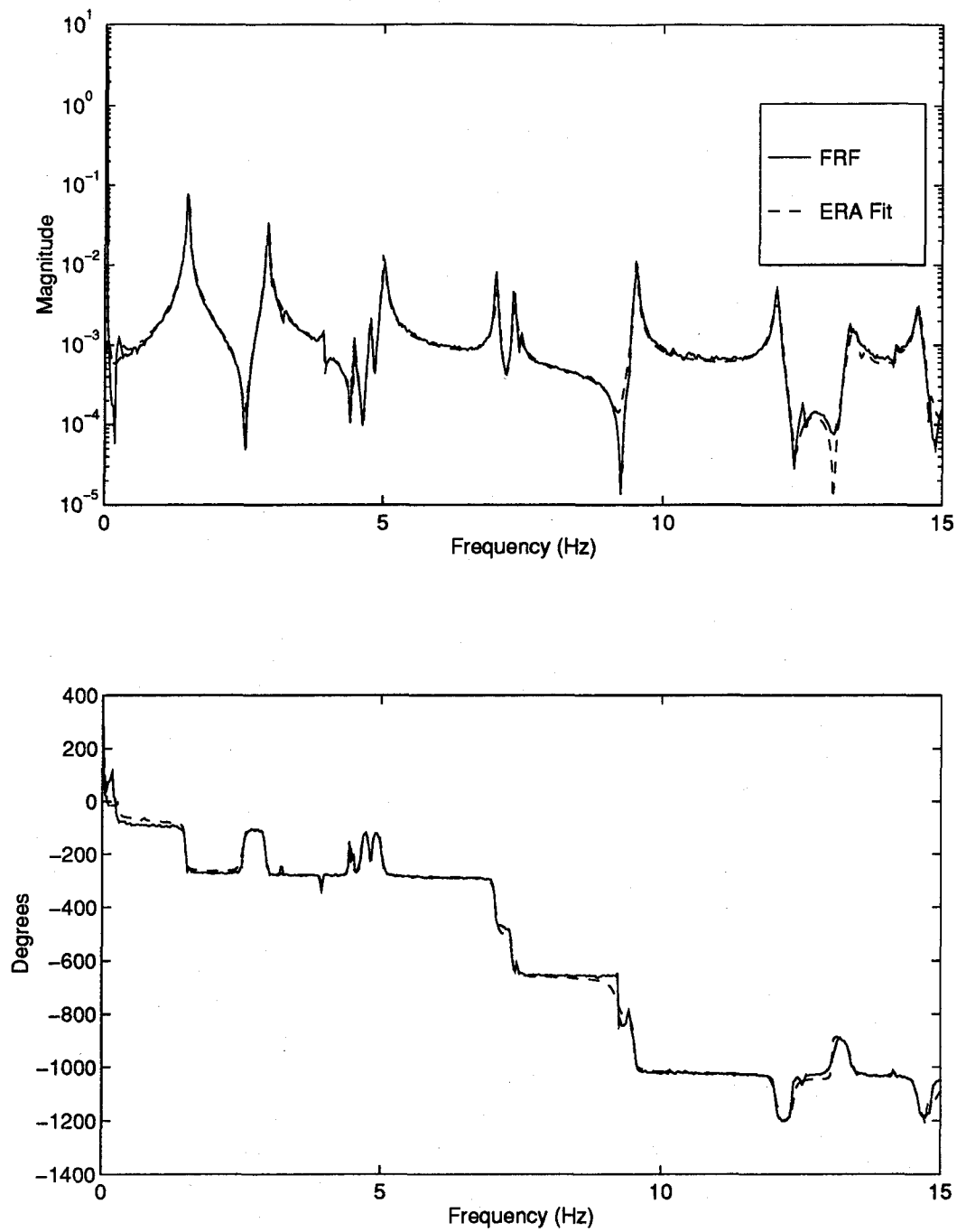


Figure F.5 Measurement 5 Excitation Actuator 6

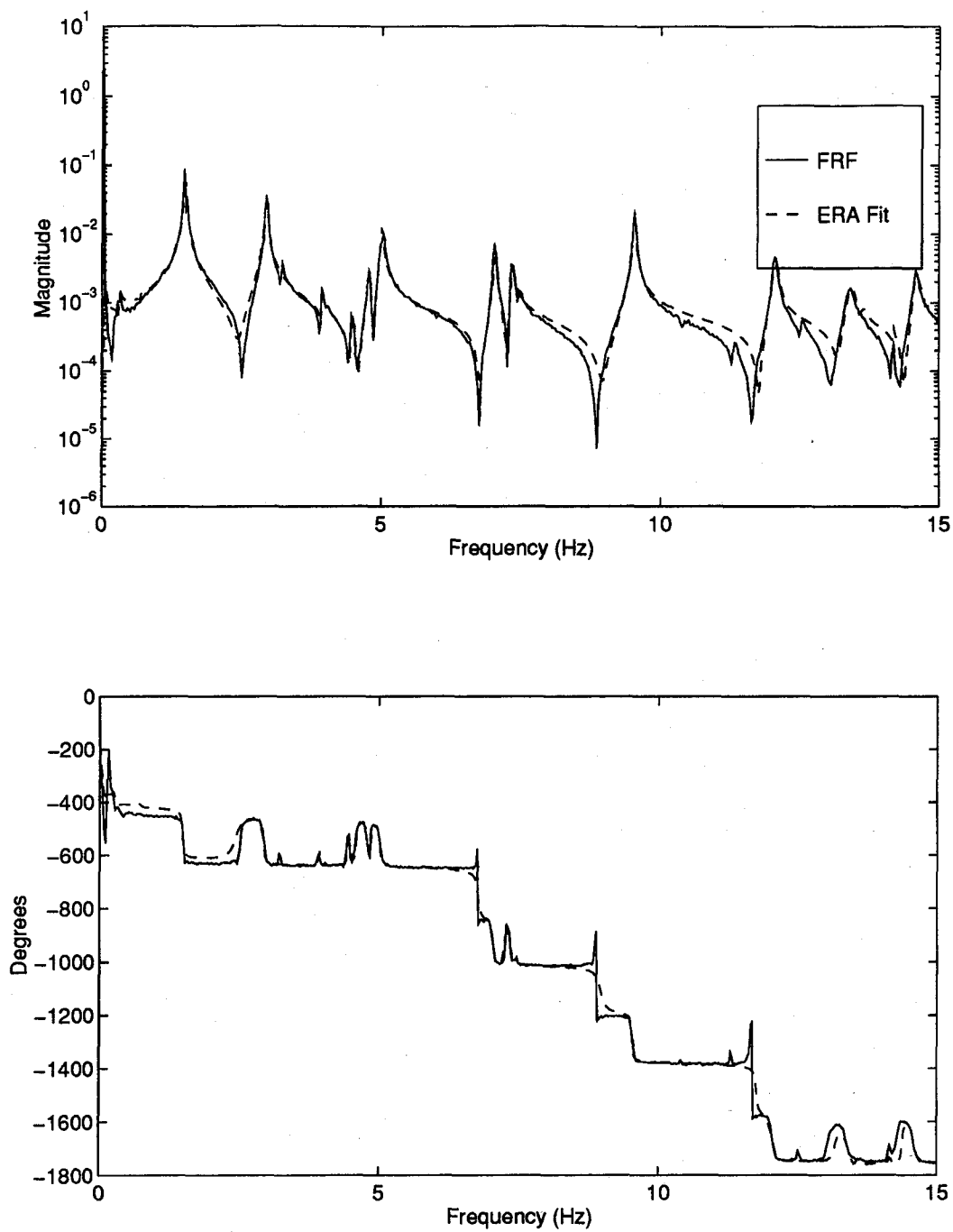


Figure F.6 Measurement 6 Excitation Actuator 6

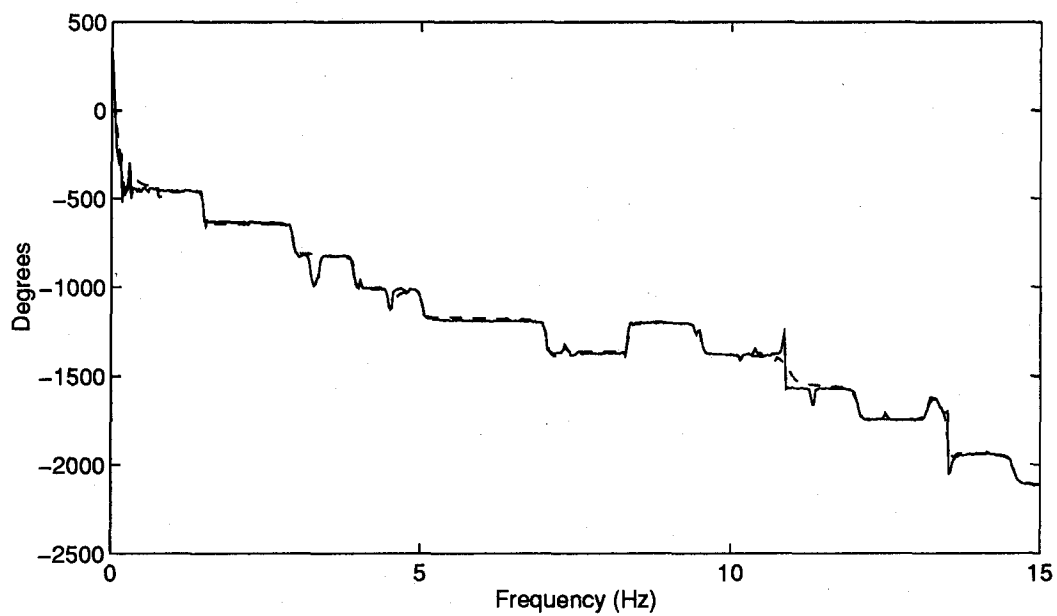
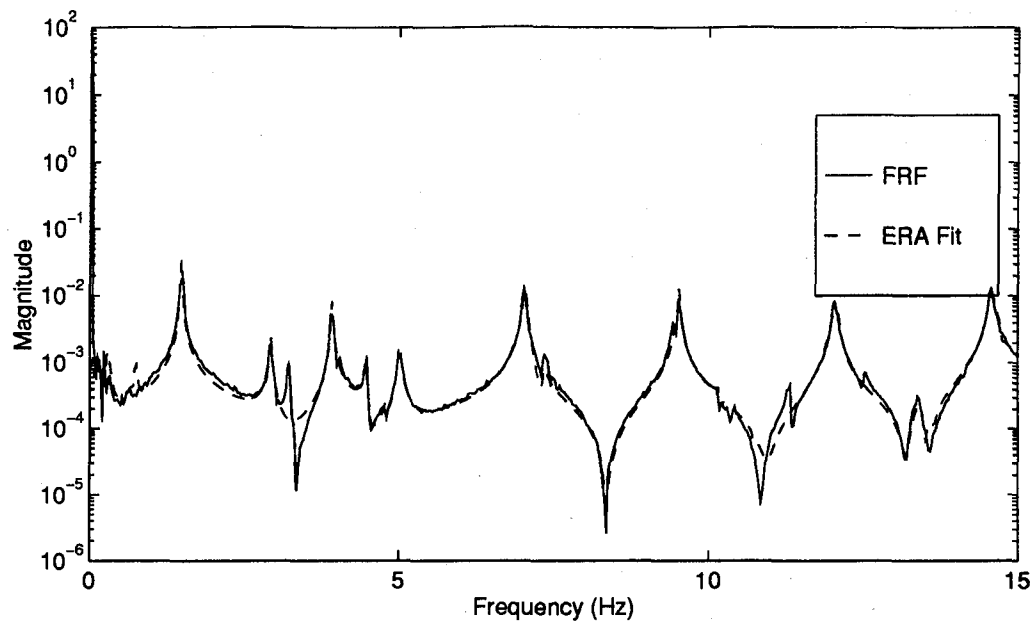


Figure F.7 Measurement 7 Excitation Actuator 6

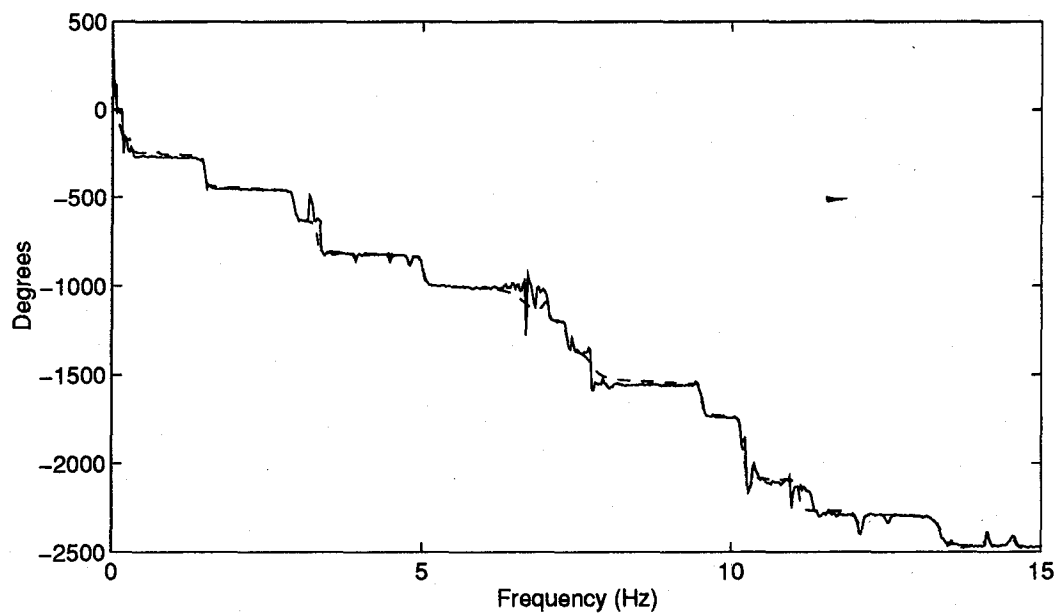
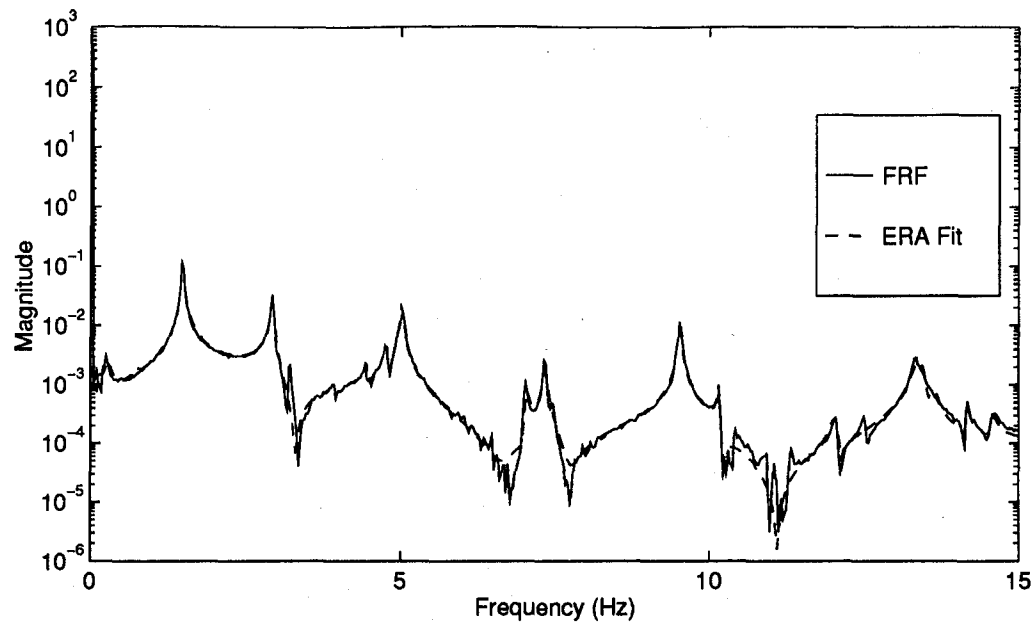


Figure F.8 Measurement 8 Excitation Actuator 6

Appendix G. State-Space Model FRFs of PACOSS DTA (Cont.)

The eight plots in this appendix are the measurements taken when Actuator 7 was excited with a random noise.

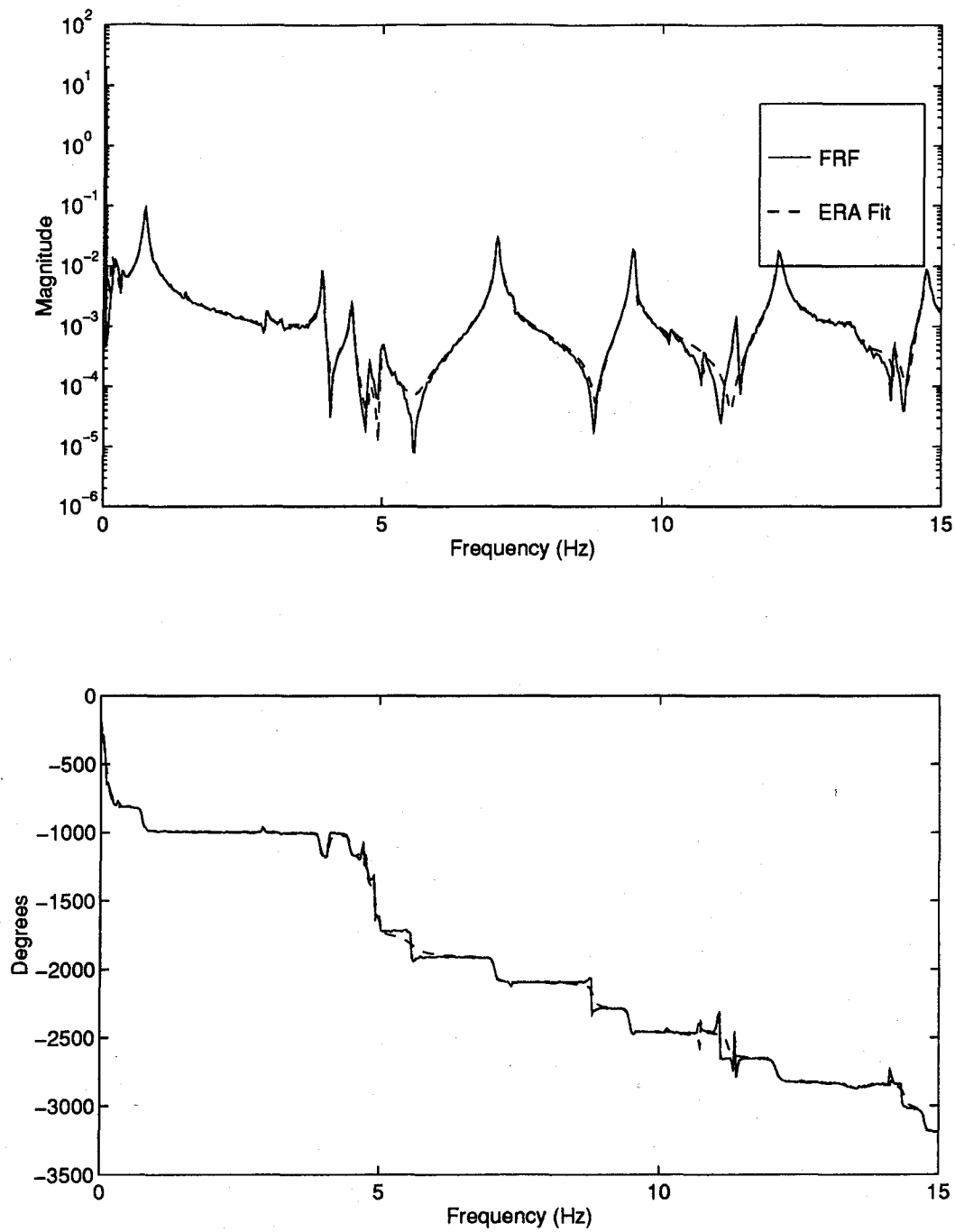


Figure G.1 Measurement 1 Excitation Actuator 7

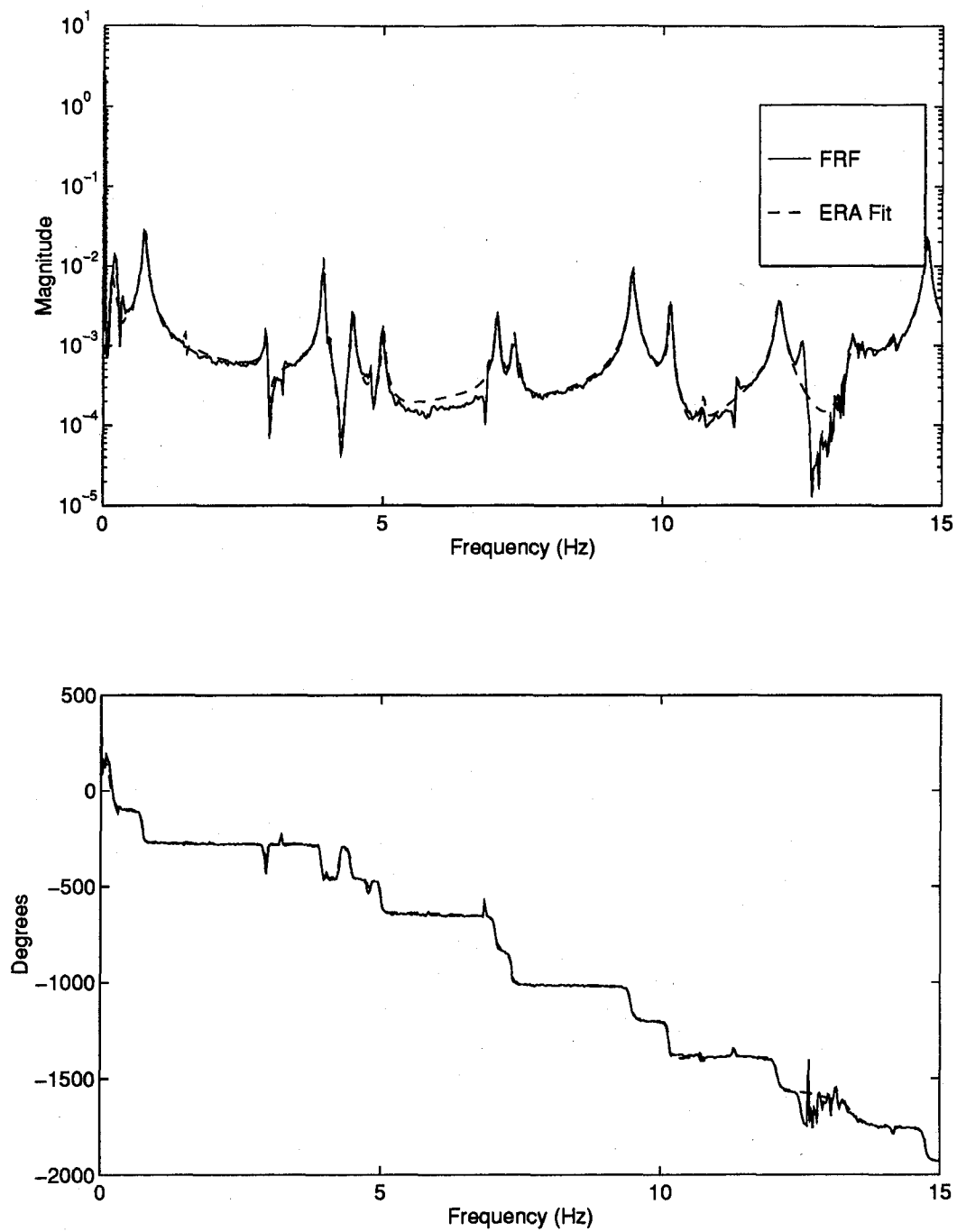


Figure G.2 Measurement 2 Excitation Actuator 7

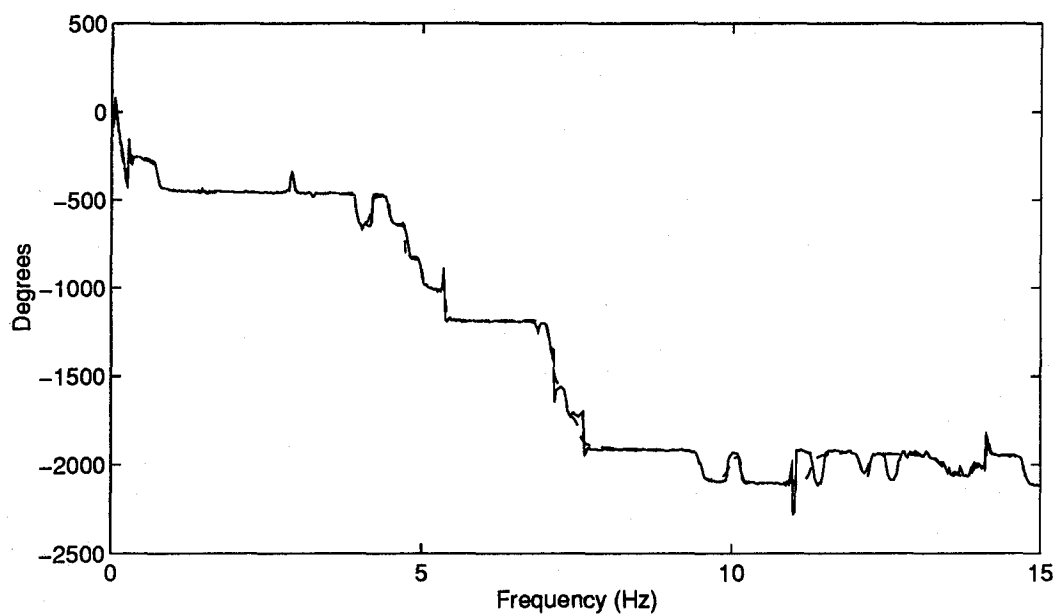
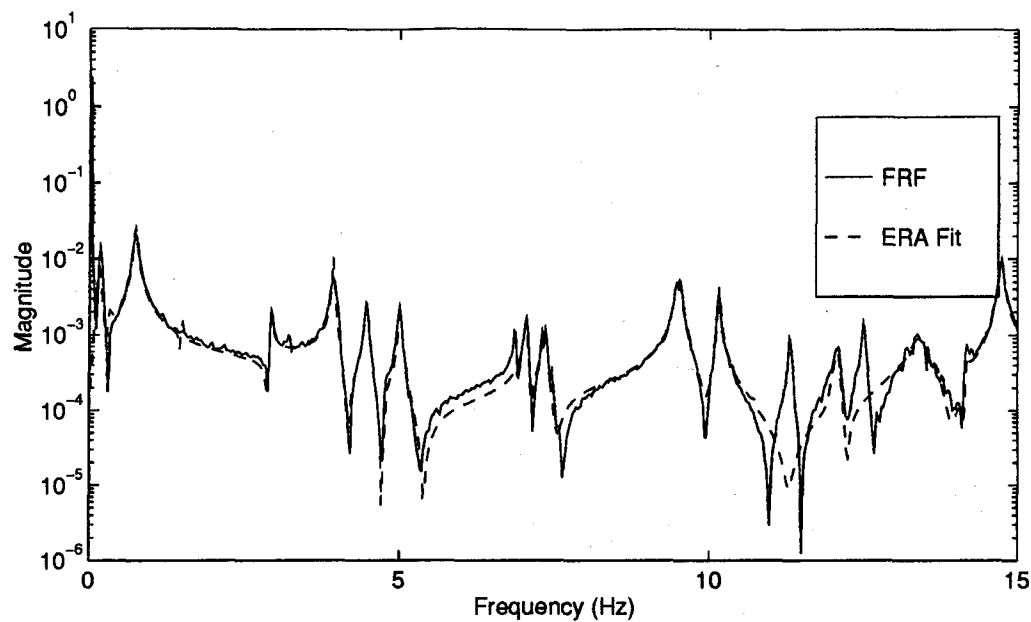


Figure G.3 Measurement 3 Excitation Actuator 7

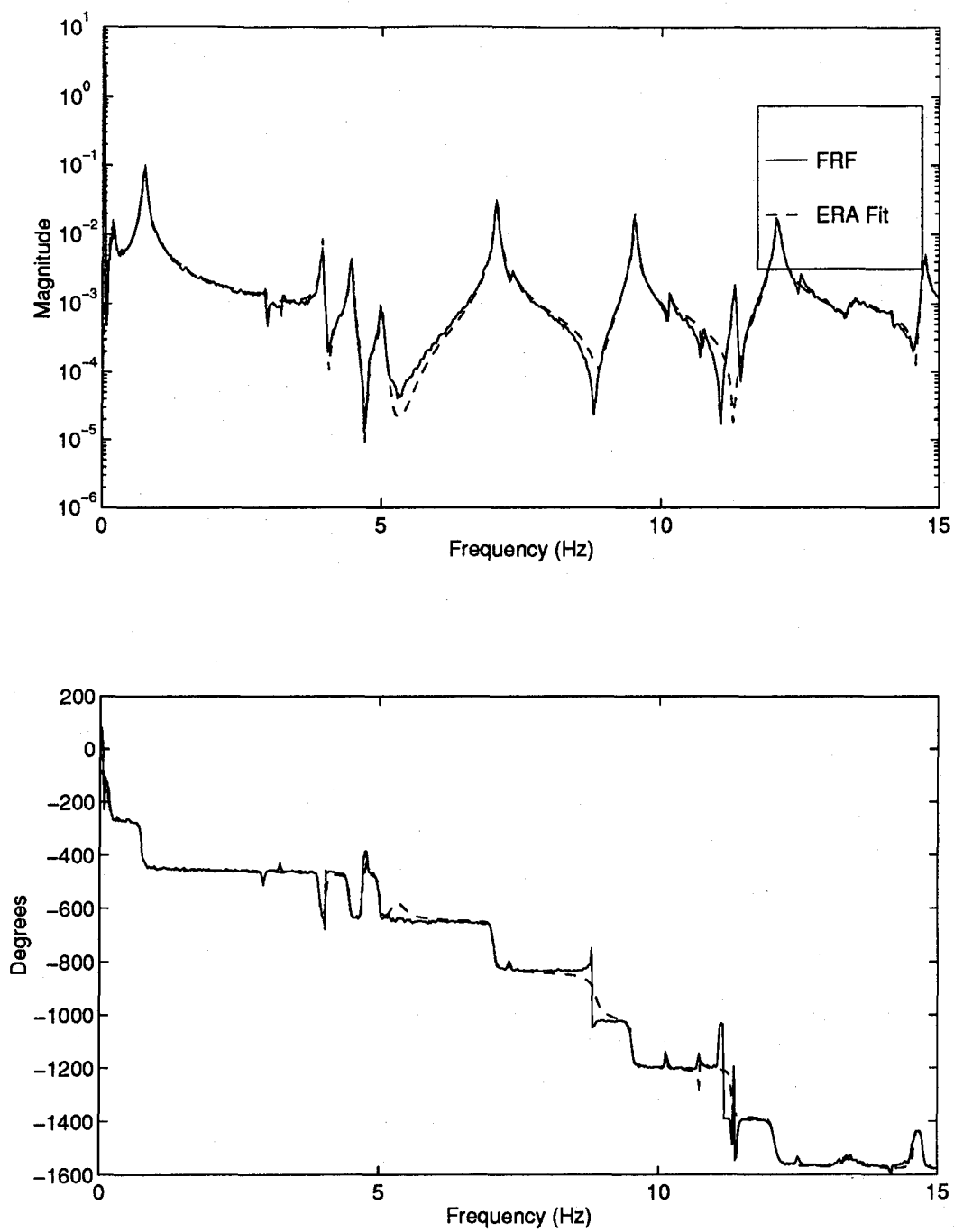


Figure G.4 Measurement 4 Excitation Actuator 7

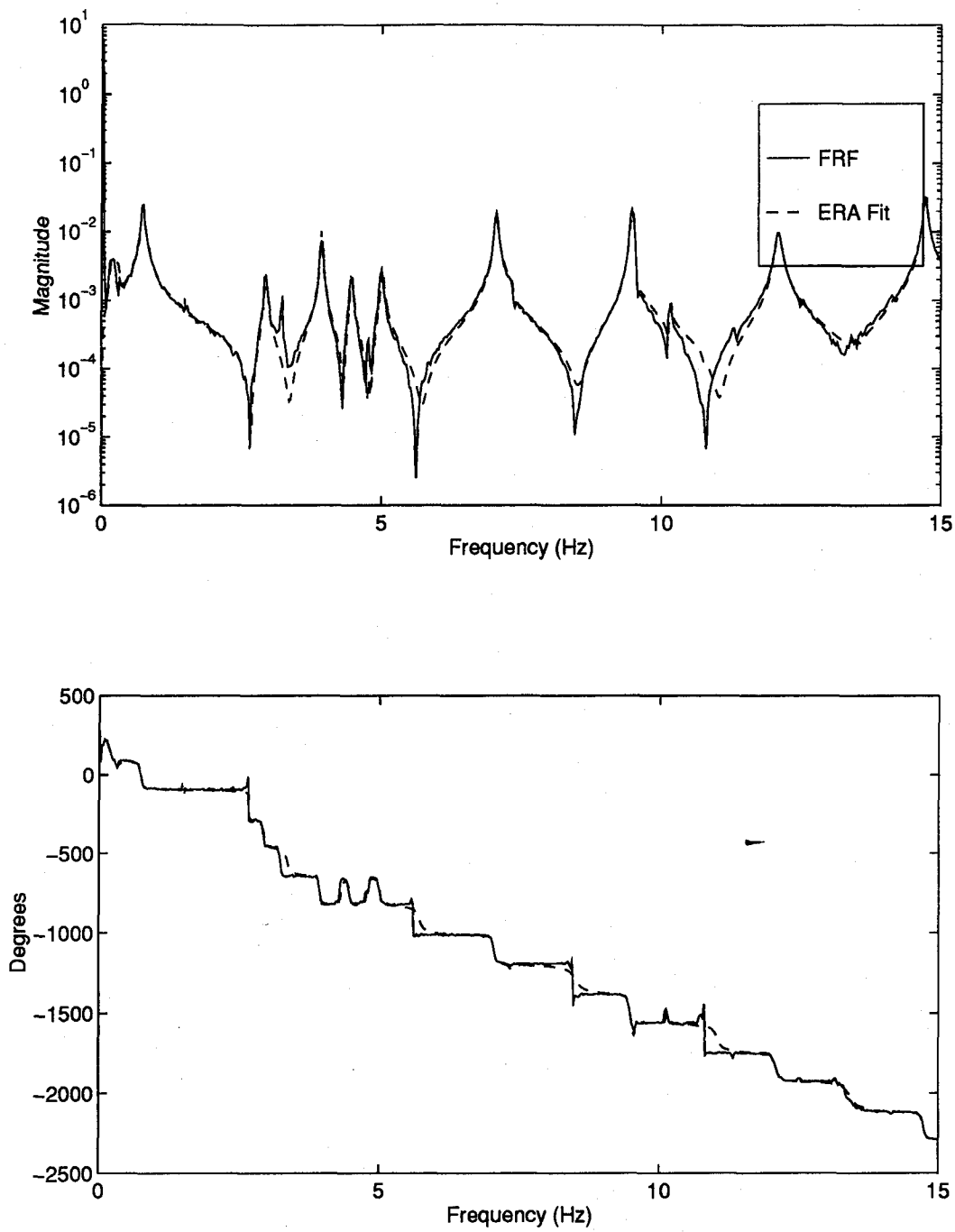


Figure G.5 Measurement 5 Excitation Actuator 7

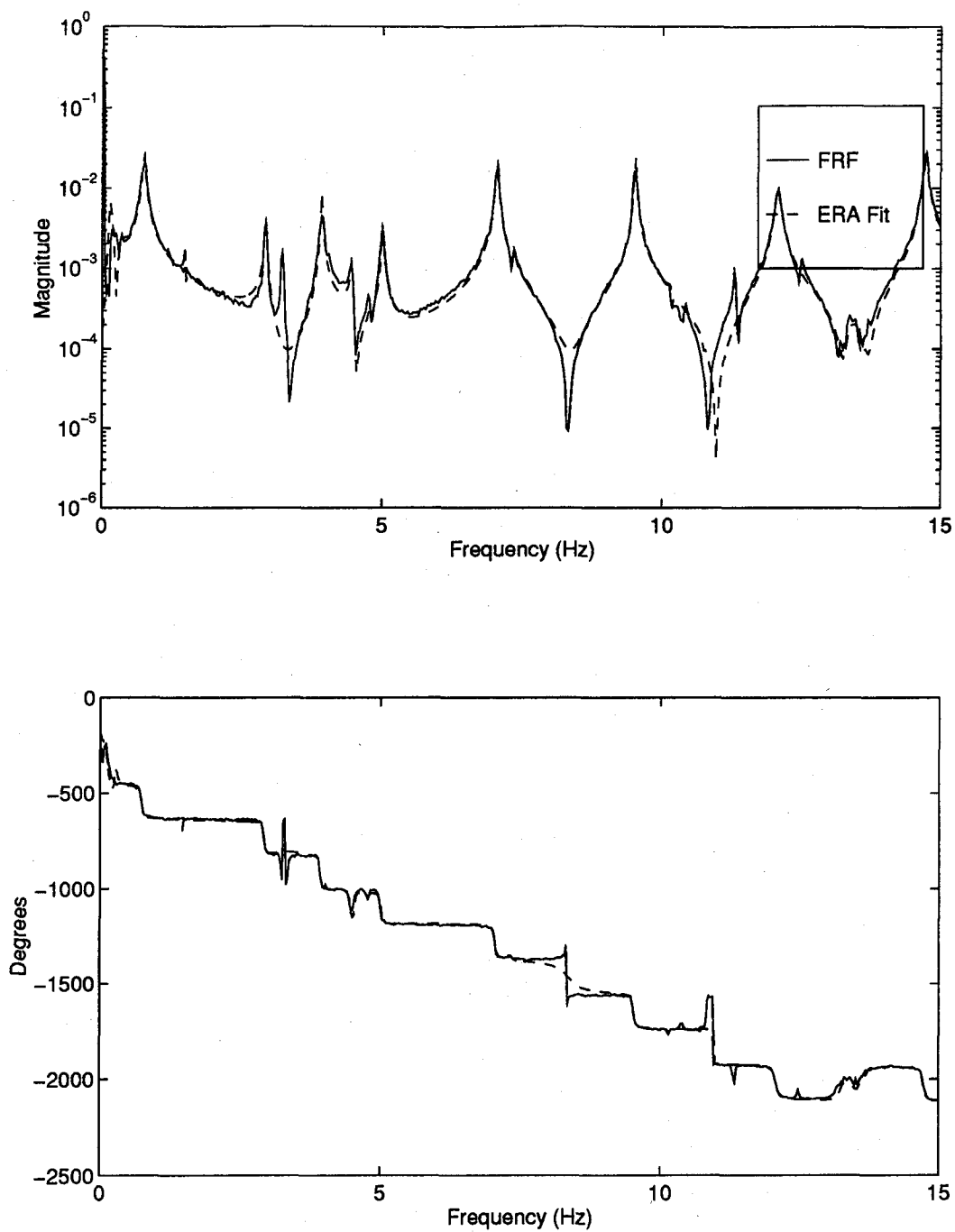


Figure G.6 Measurement 6 Excitation Actuator 7

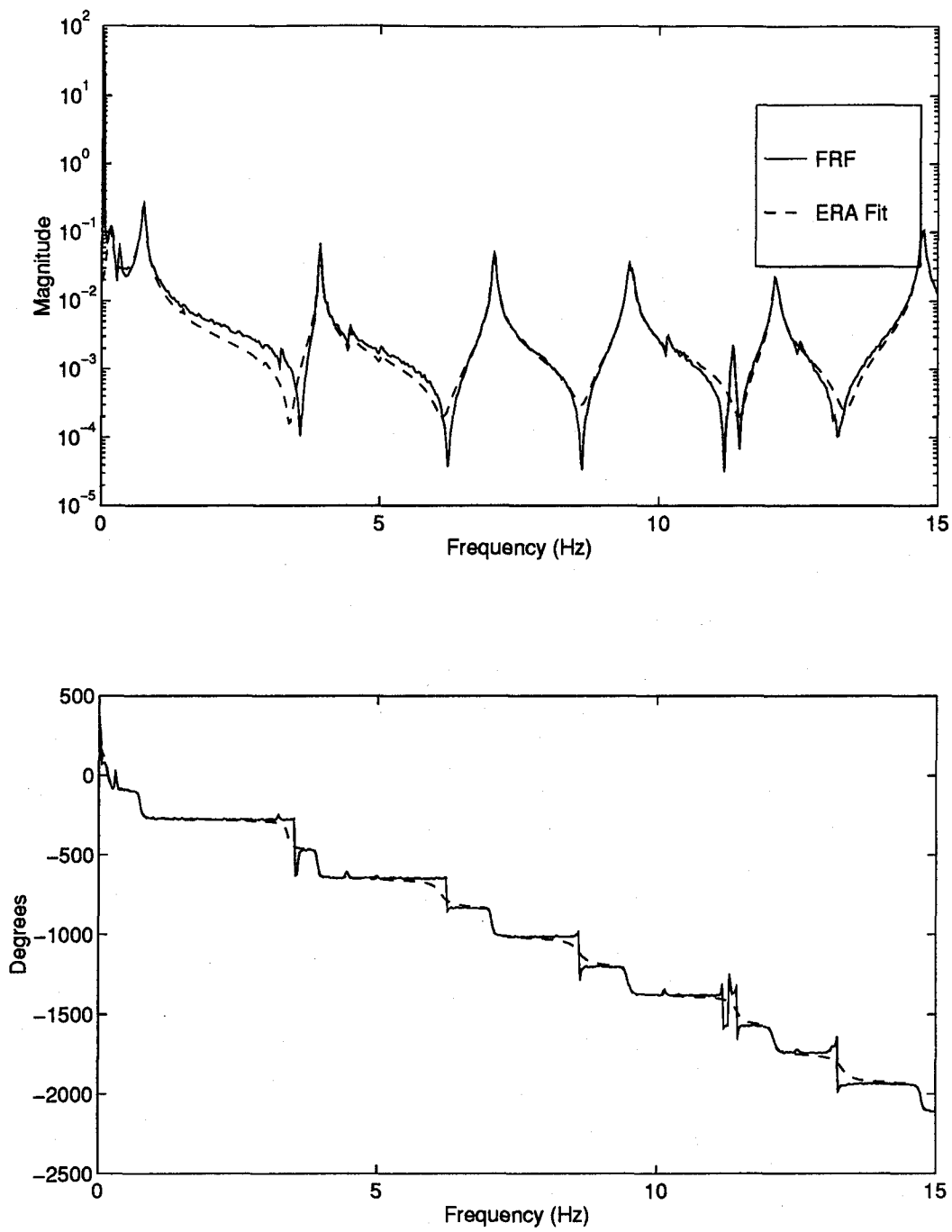


Figure G.7 Measurement 7 Excitation Actuator 7

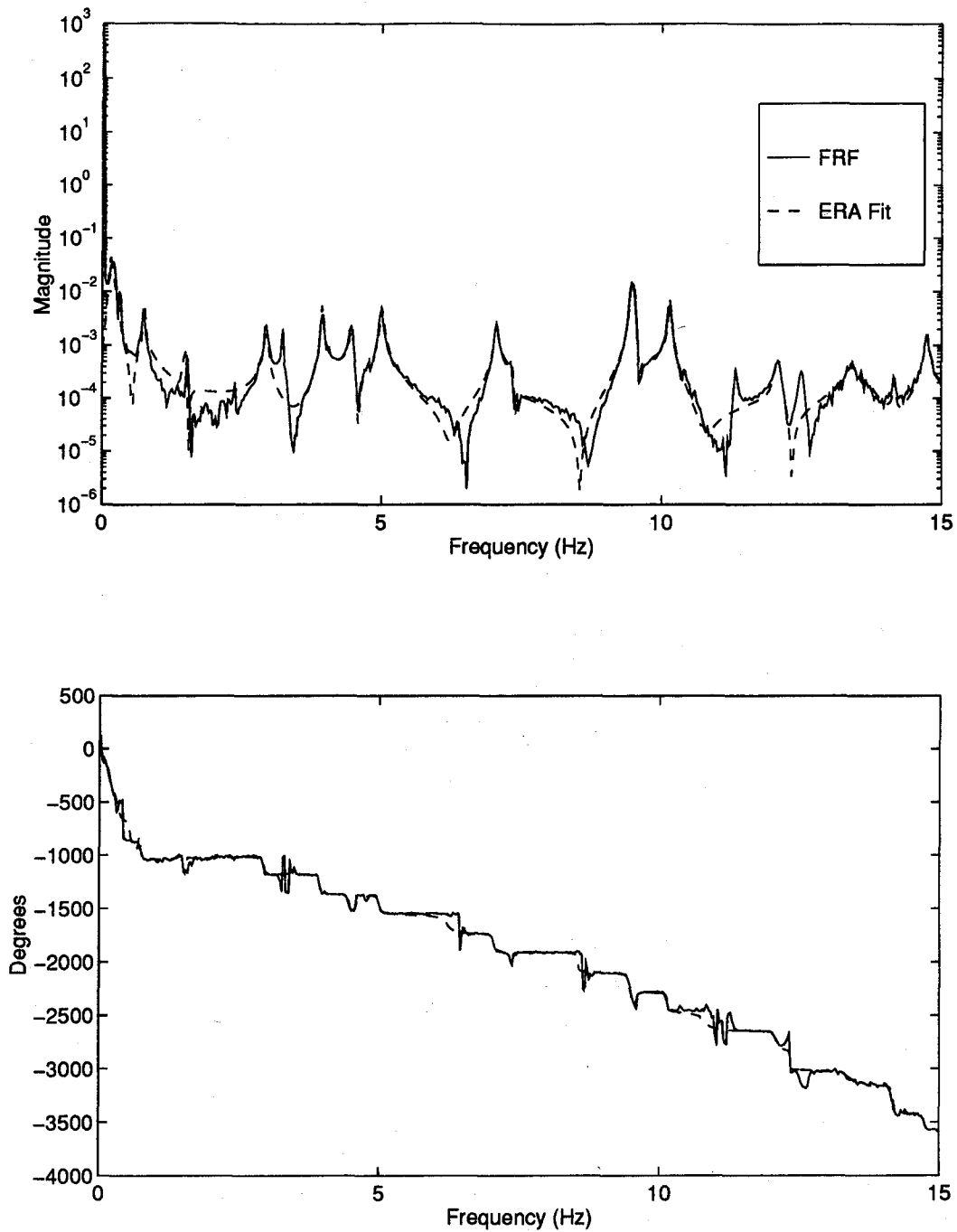


Figure G.8 Measurement 8 Excitation Actuator 7

Appendix H. State-Space Model FRFs of PACOSS DTA (Cont.)

The eight plots in this appendix are the measurements taken when Actuator 8 was excited with a random noise.

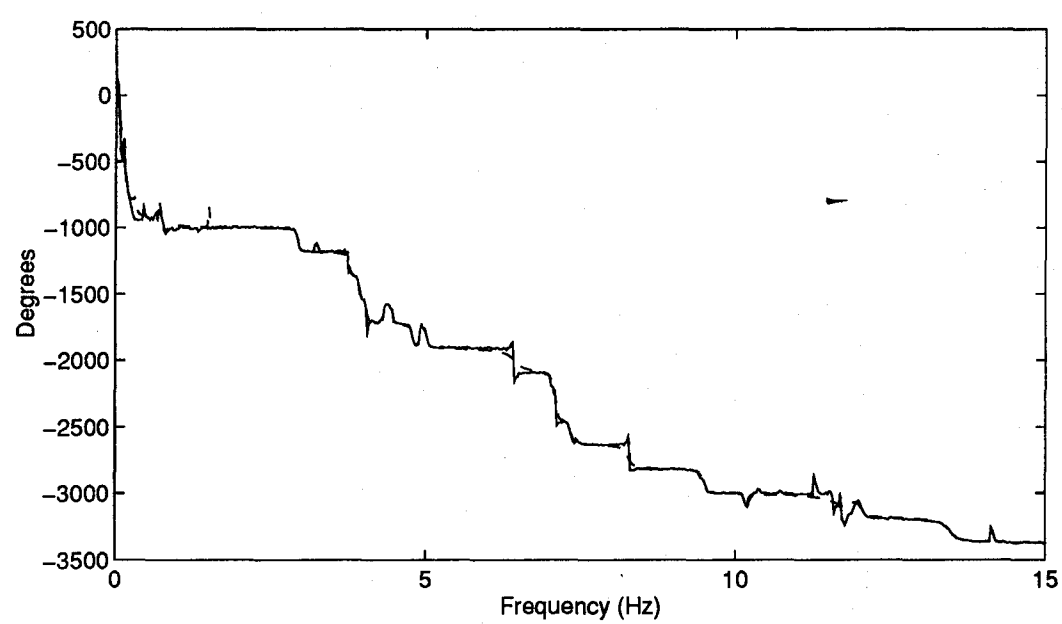
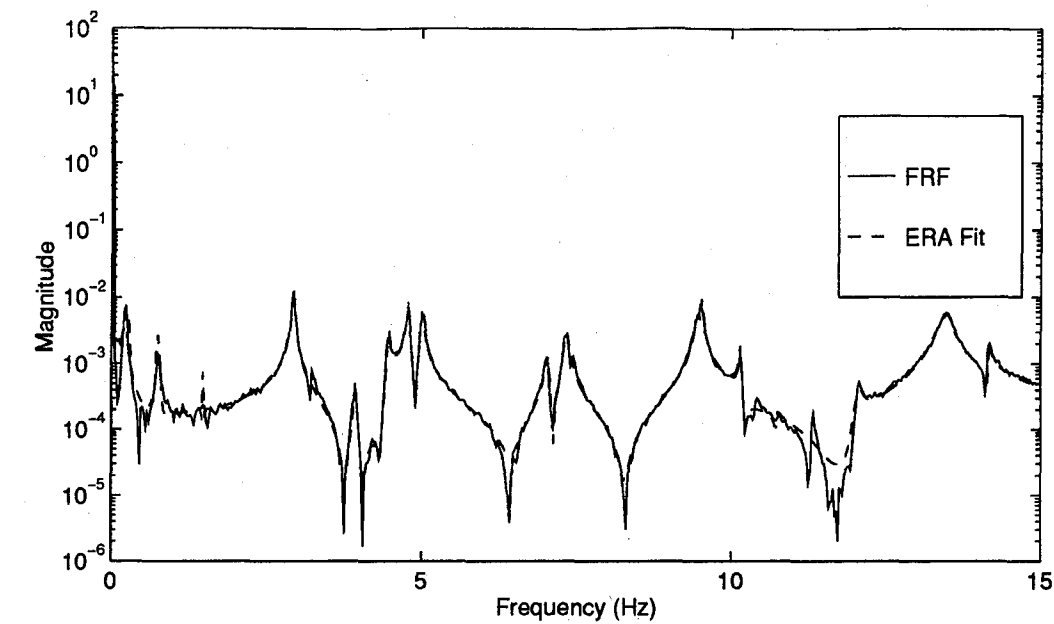


Figure H.1 Measurement 1 Excitation Actuator 8

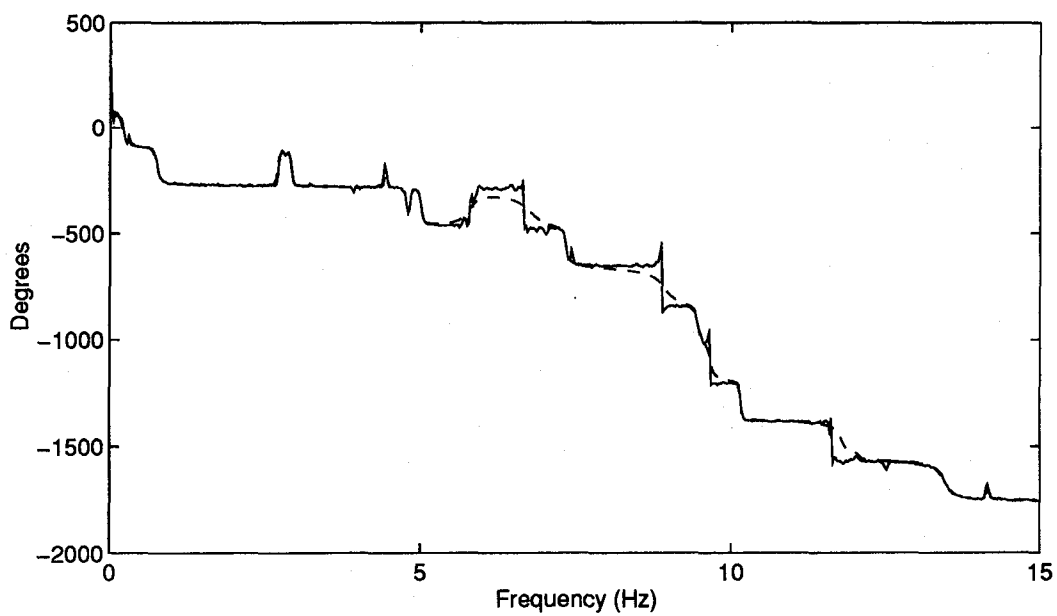
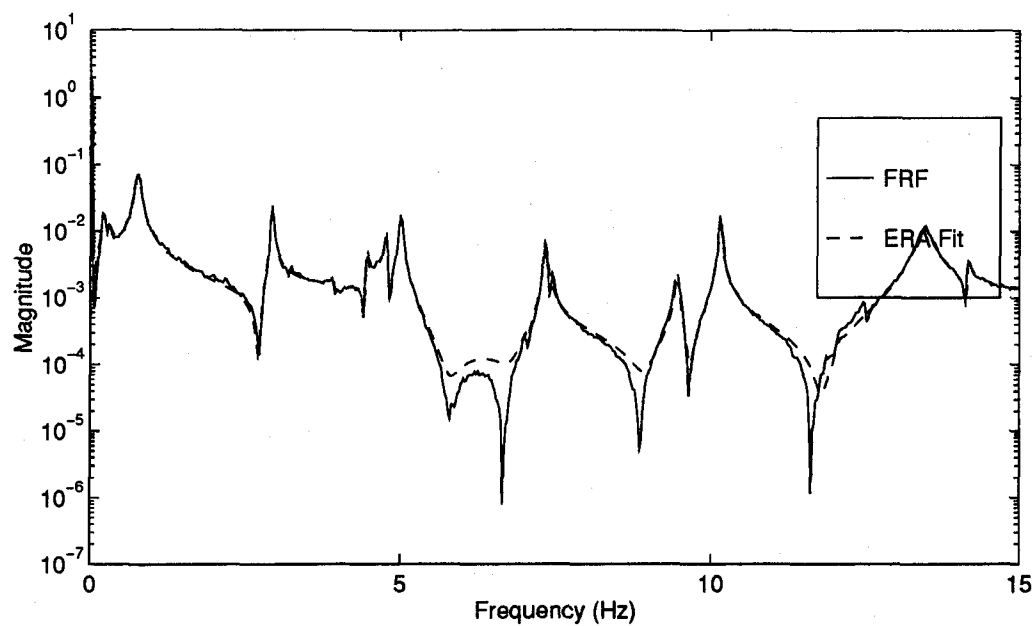


Figure H.2 Measurement 2 Excitation Actuator 8

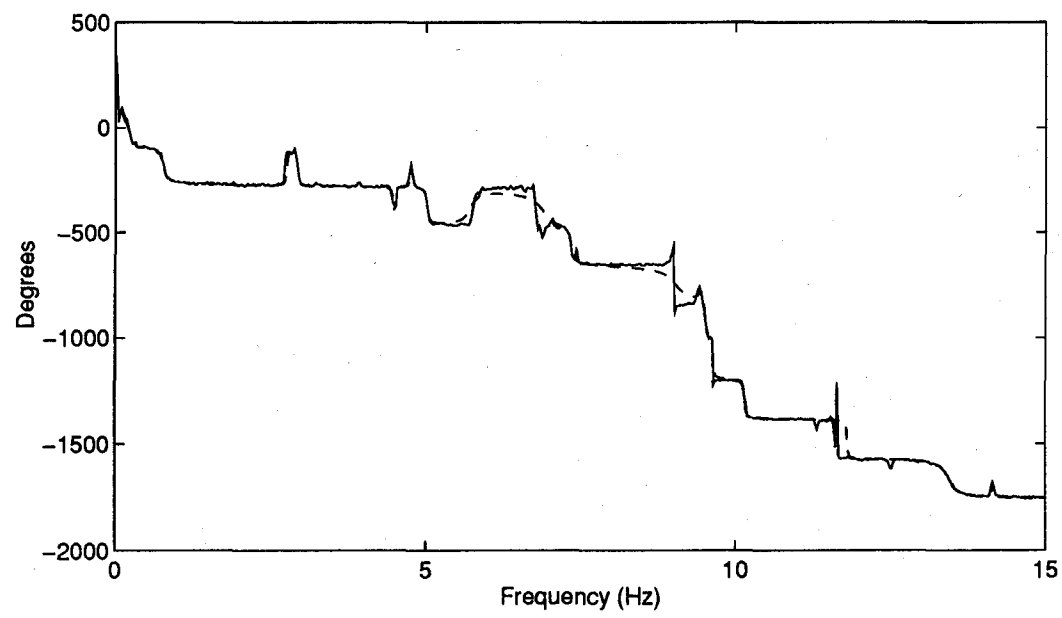
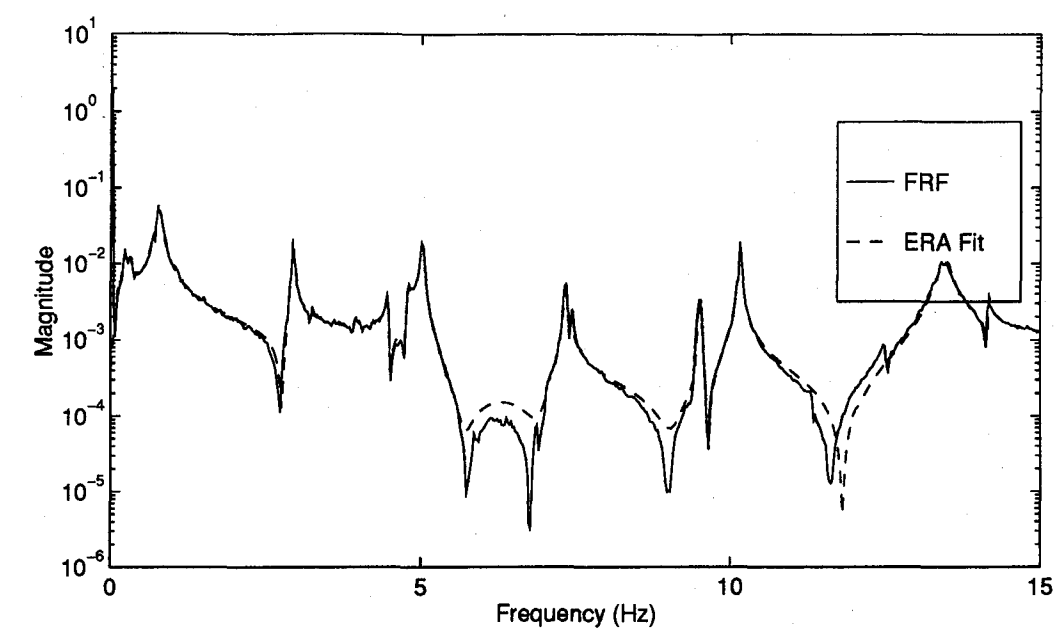


Figure H.3 Measurement 3 Excitation Actuator 8

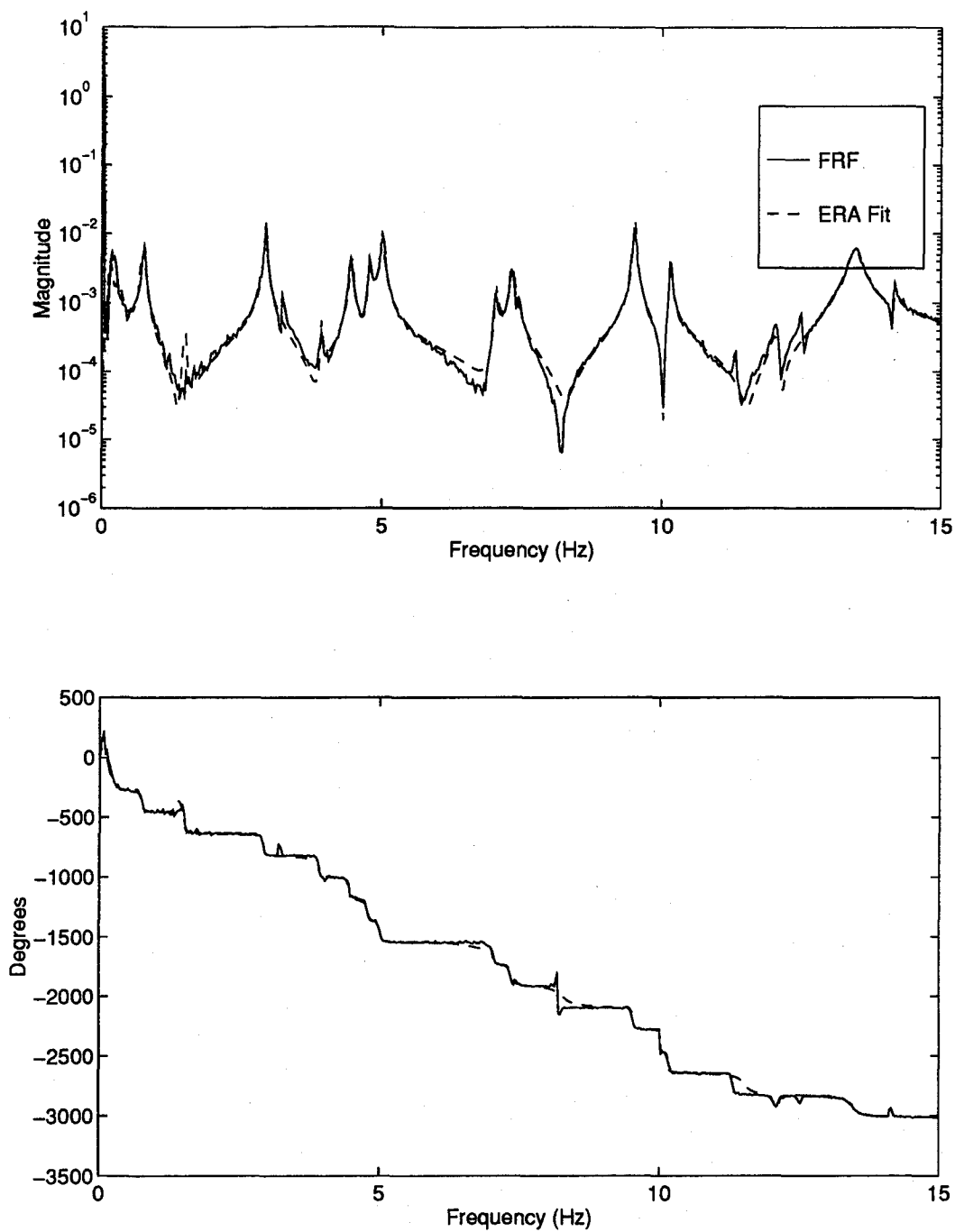


Figure H.4 Measurement 4 Excitation Actuator 8

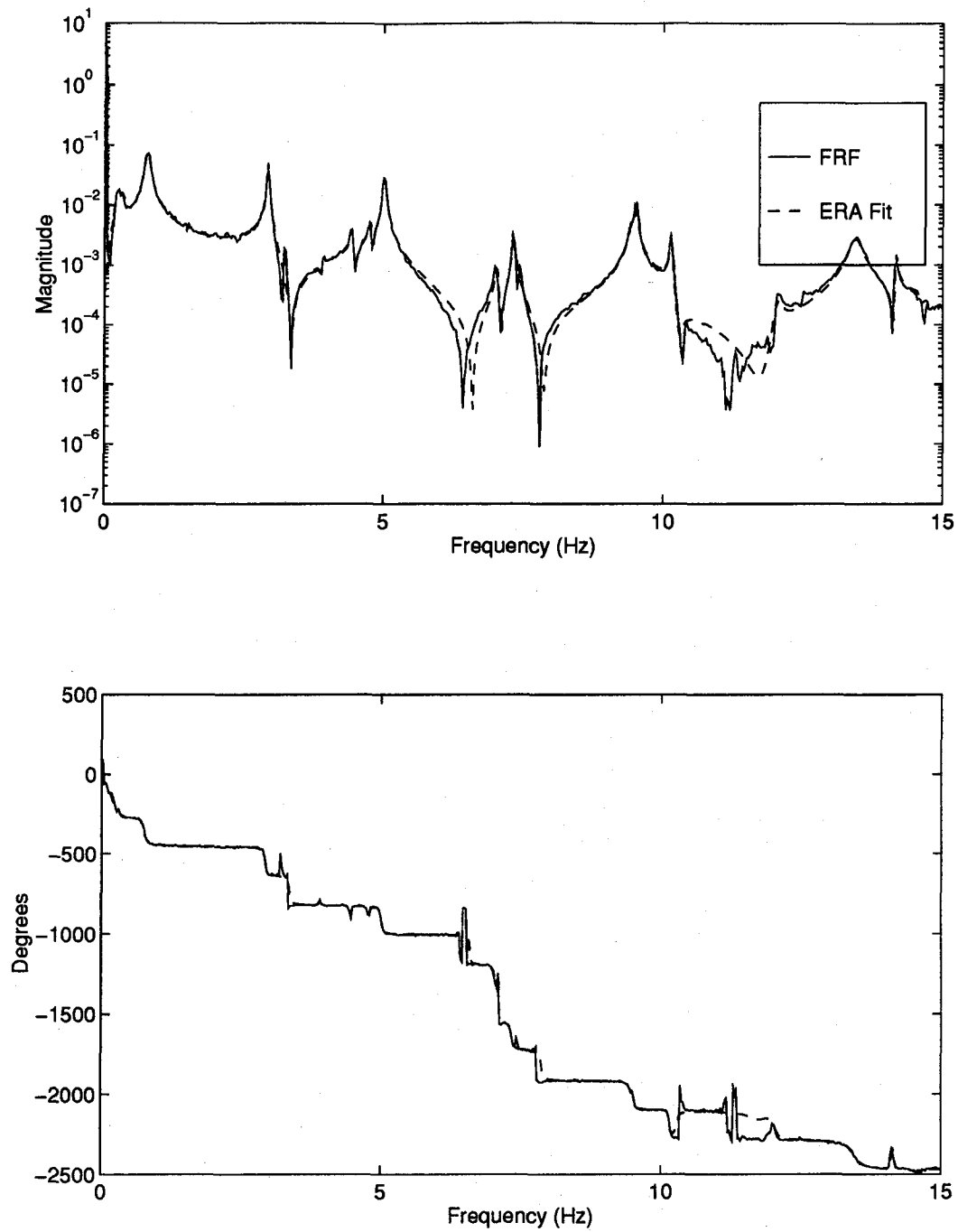


Figure H.5 Measurement 5 Excitation Actuator 8

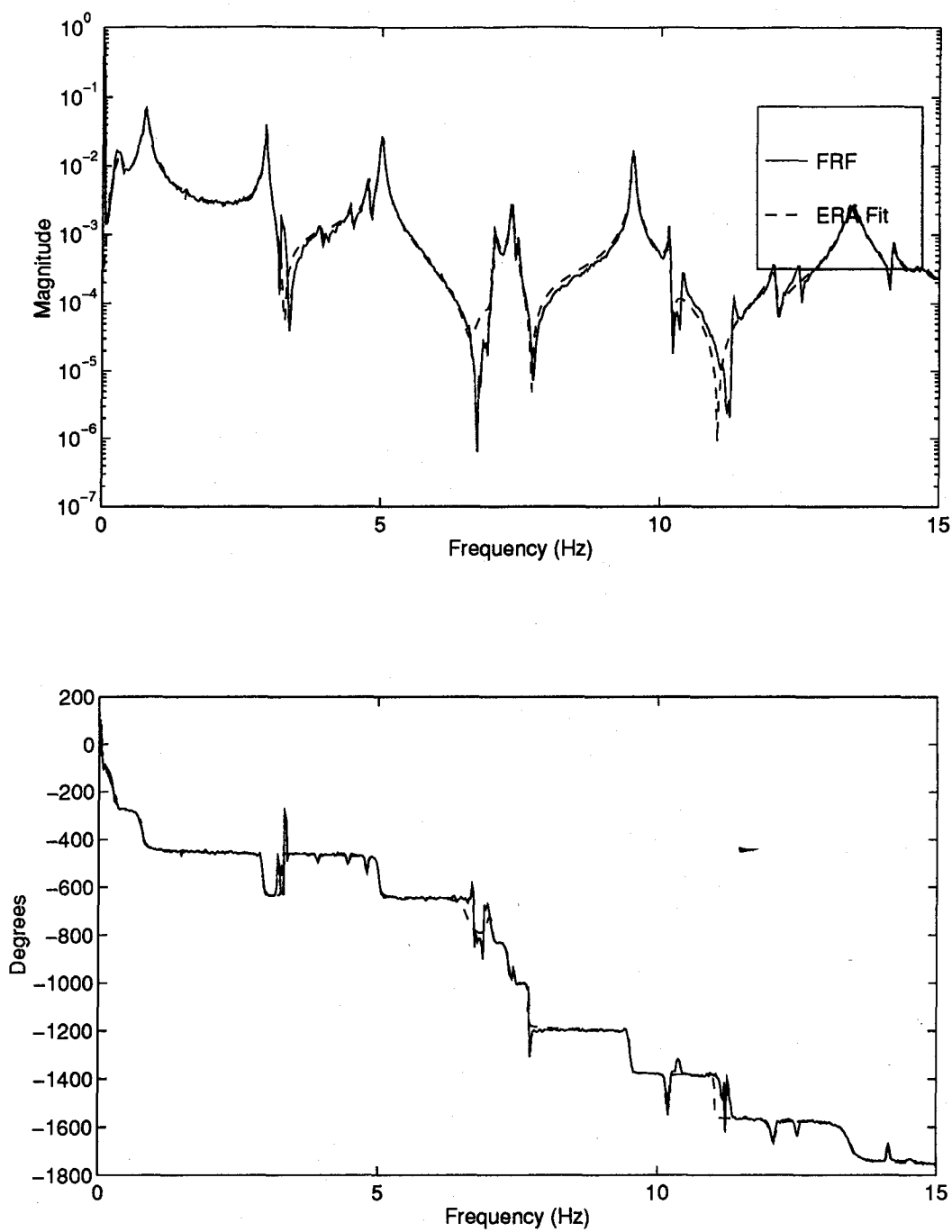


Figure H.6 Measurement 6 Excitation Actuator 8

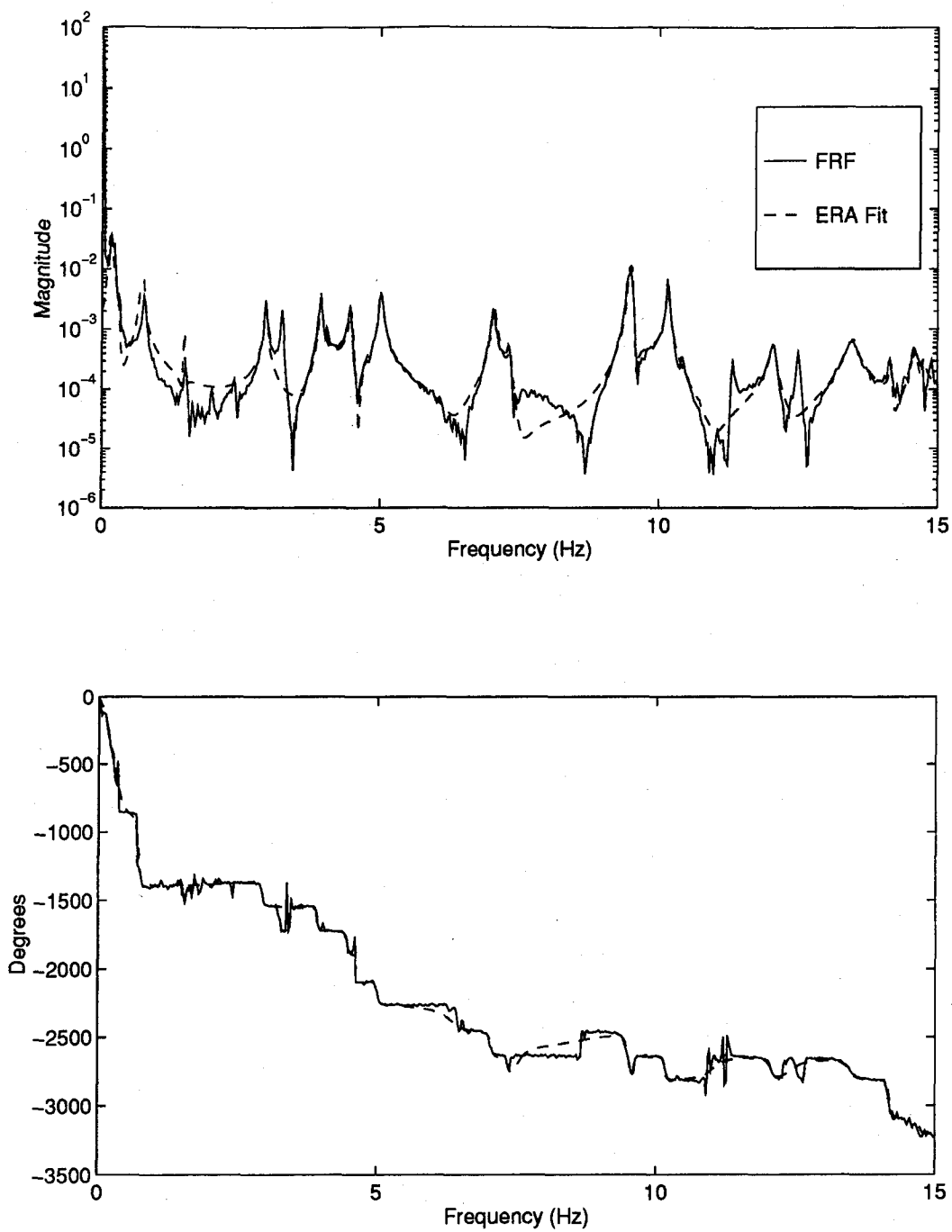


Figure H.7 Measurement 7 Excitation Actuator 8

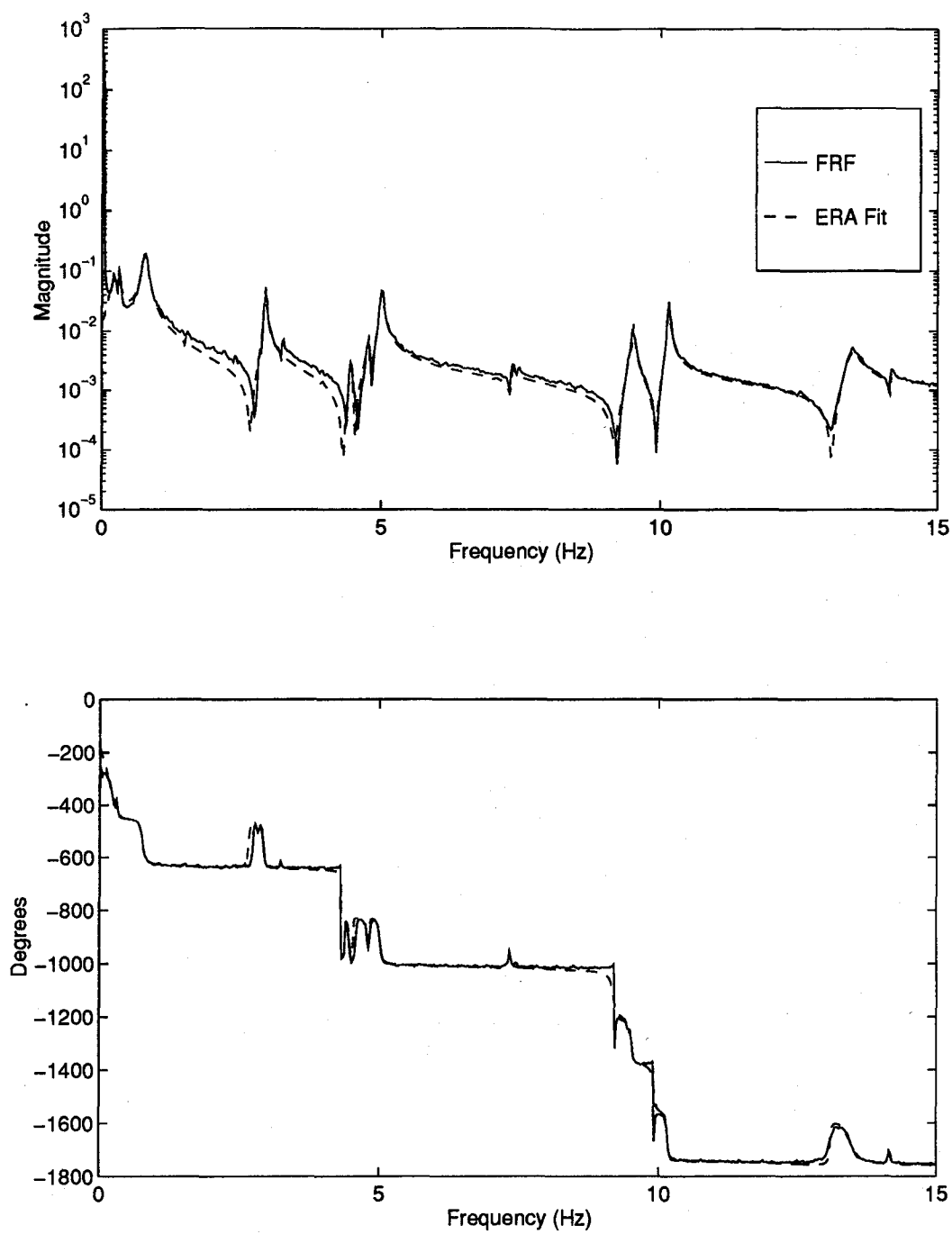


Figure H.8 Measurement 8 Excitation Actuator 8

Appendix I. State-Space Model FRFs of Truth Model

The plots seen in this appendix show the measured FRFs of the truth model, with and without noise, and their corresponding FRFs generated by the ERA-produced state-space model.

I.1 Truth Model without Noise

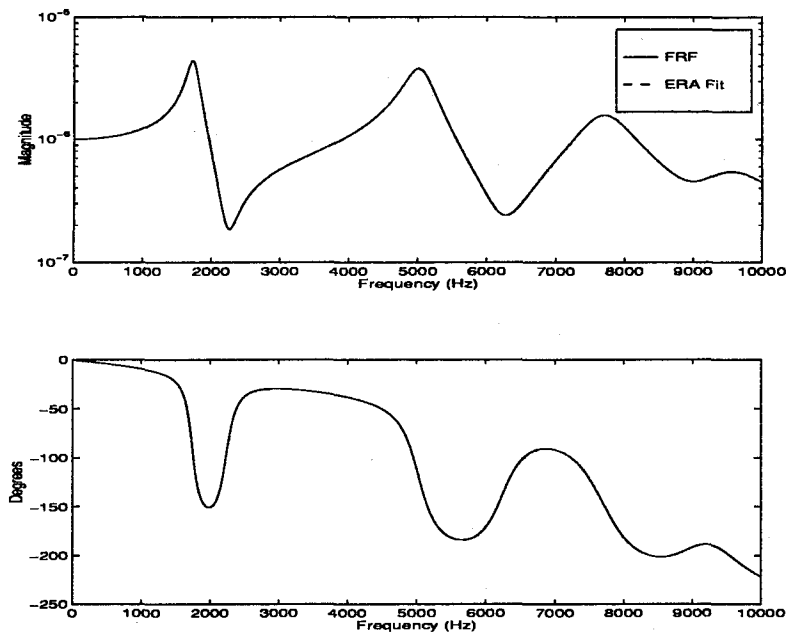


Figure I.1 Displacement Y1 Force Input F1

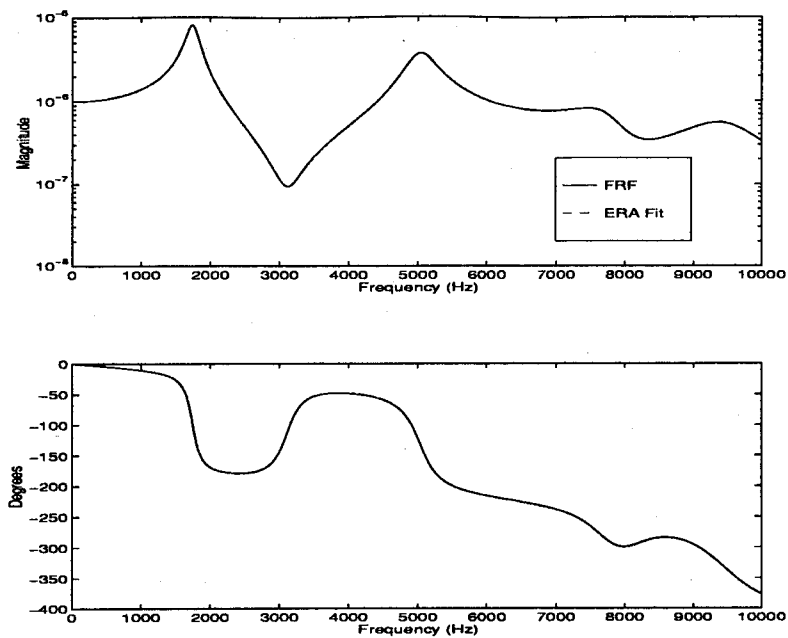


Figure I.2 Displacement Y2 Force Input F1

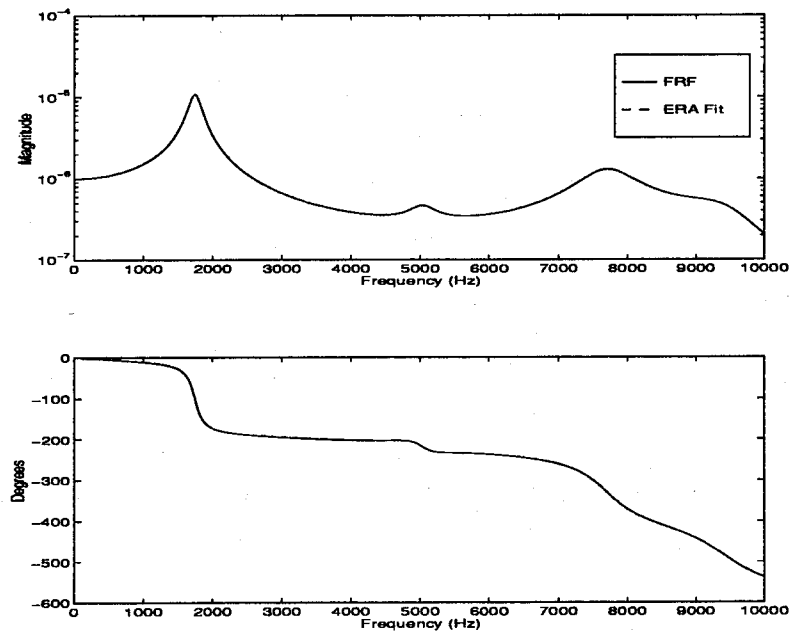


Figure I.3 Displacement Y3 Force Input F1

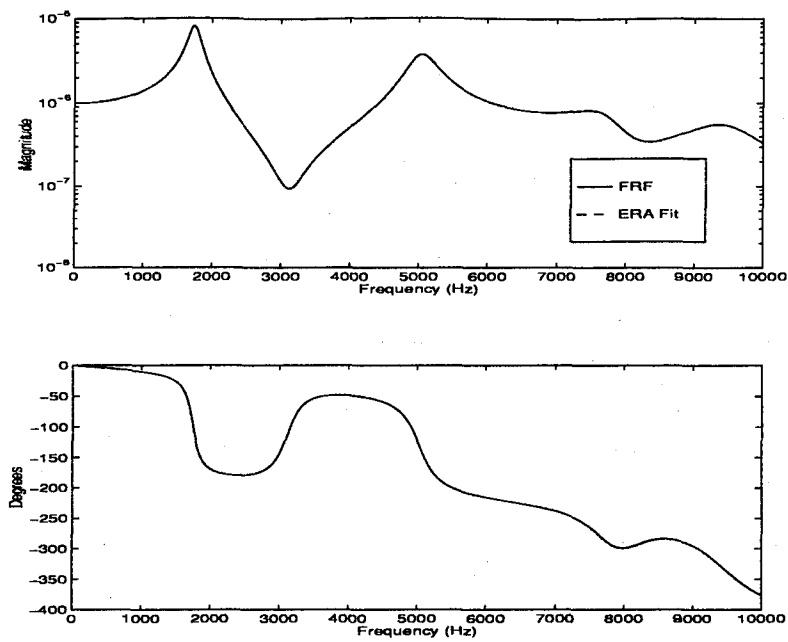


Figure I.4 Displacement Y1 Force Input F2

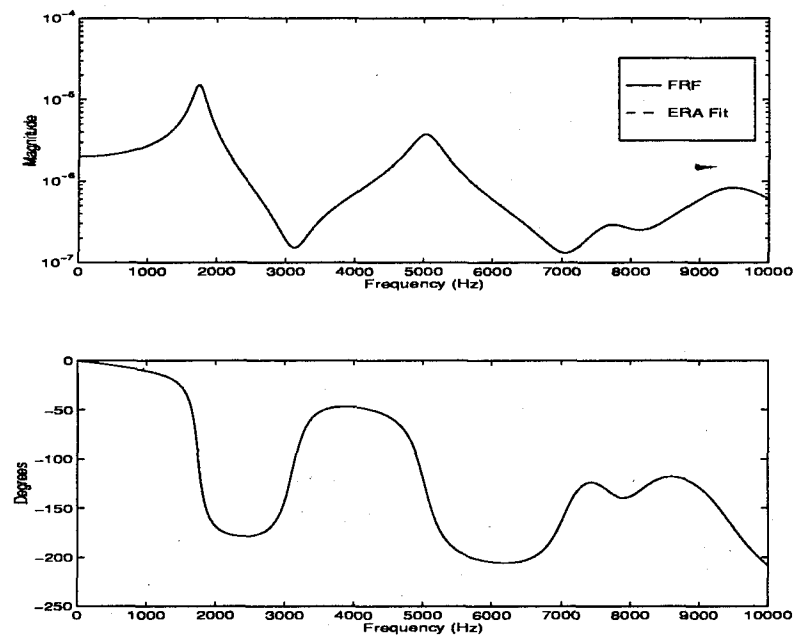


Figure I.5 Displacement Y2 Force Input F2

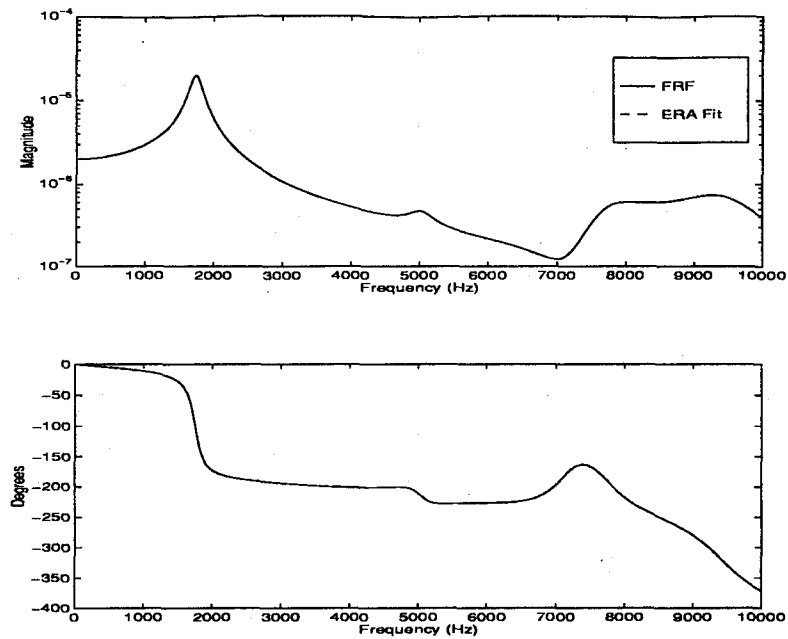


Figure I.6 Displacement Y3 Force Input F2

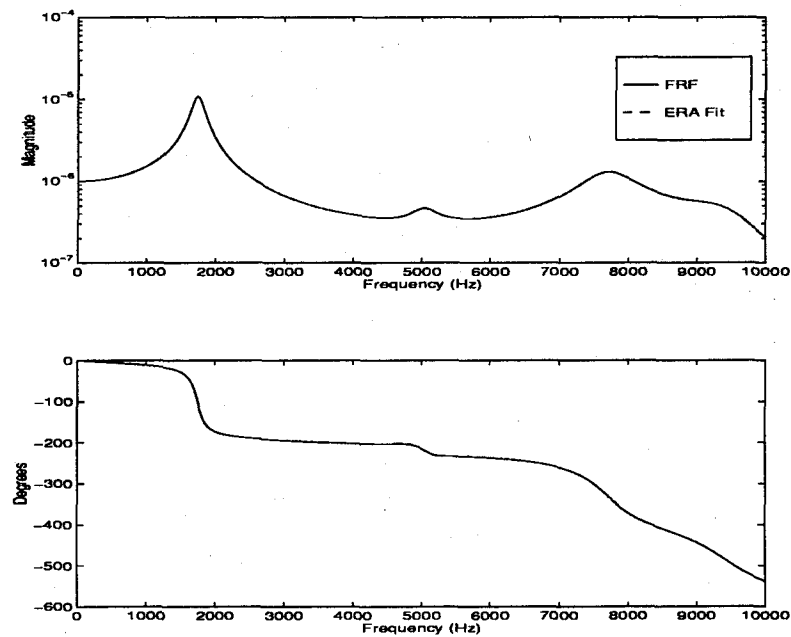


Figure I.7 Displacement Y1 Force Input F3

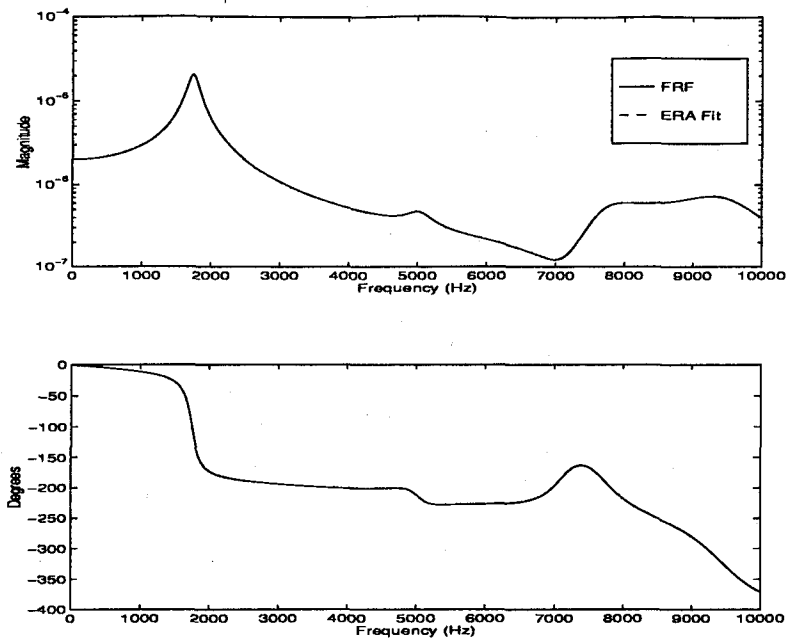


Figure I.8 Displacement Y2 Force Input F3

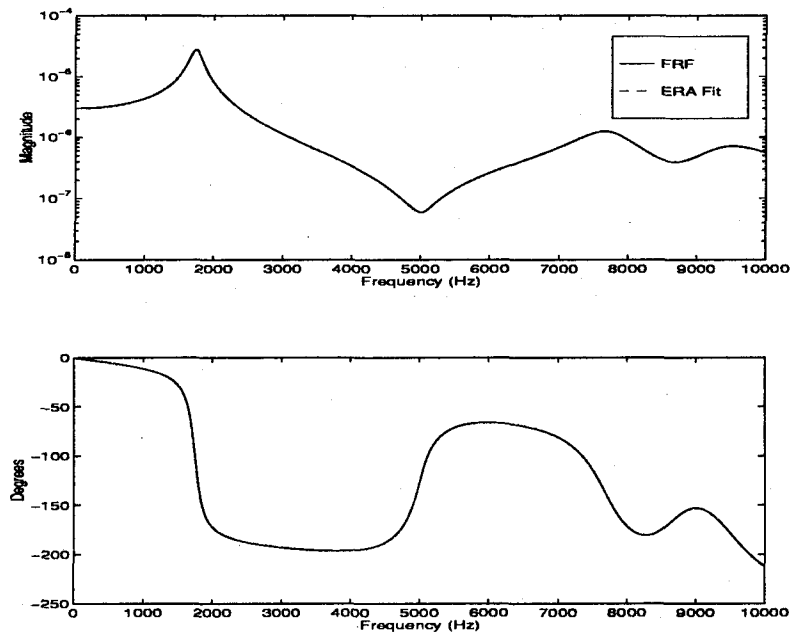


Figure I.9 Displacement Y3 Force Input F3

I.2 Truth Model with Noise

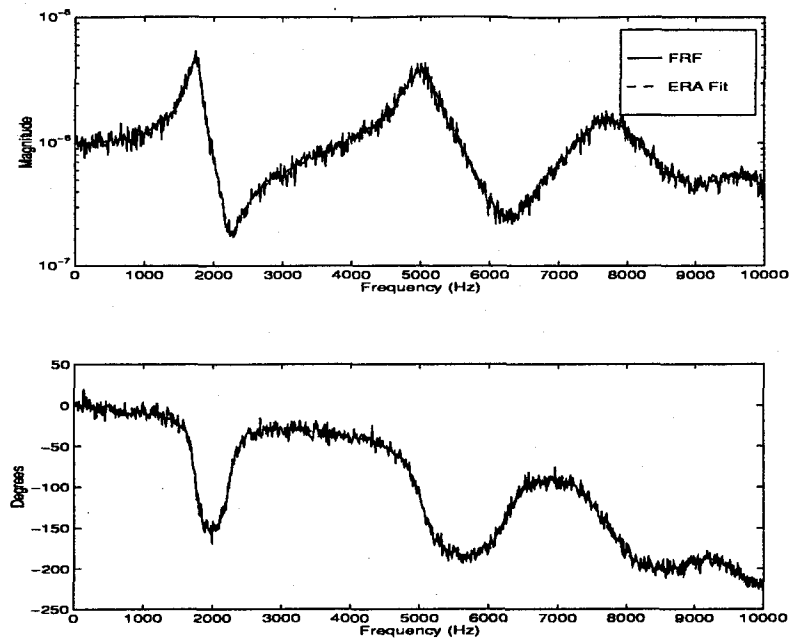


Figure I.10 Displacement Y1 Force Input F1

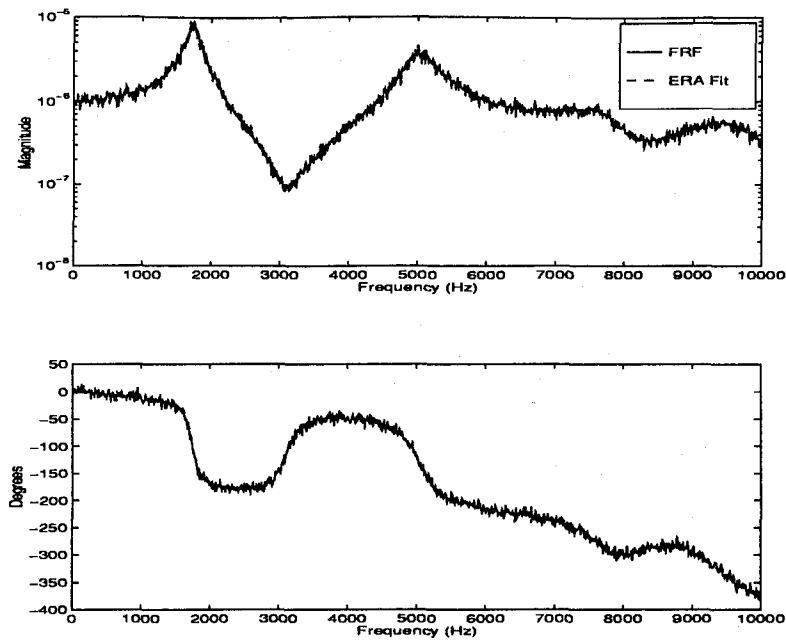


Figure I.11 Displacement Y2 Force Input F1

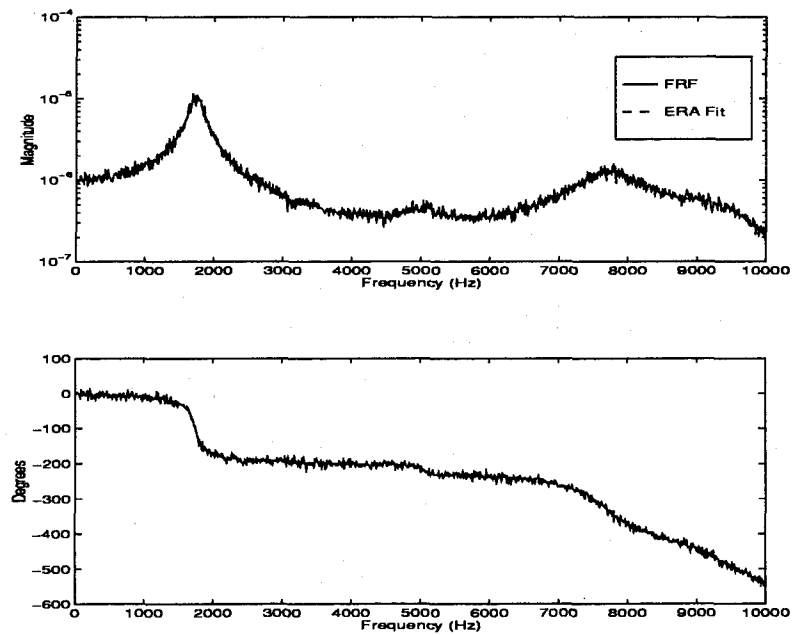


Figure I.12 Displacement Y3 Force Input F1

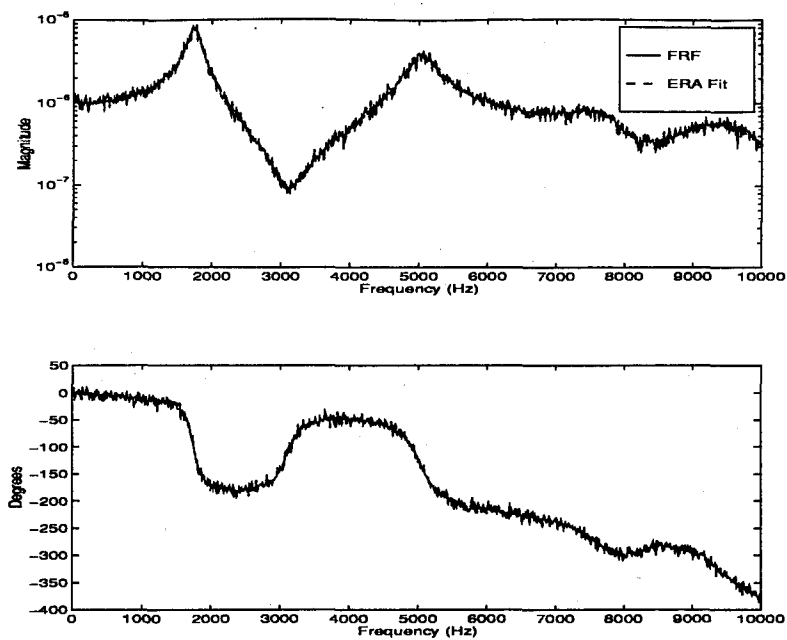


Figure I.13 Displacement Y1 Force Input F2

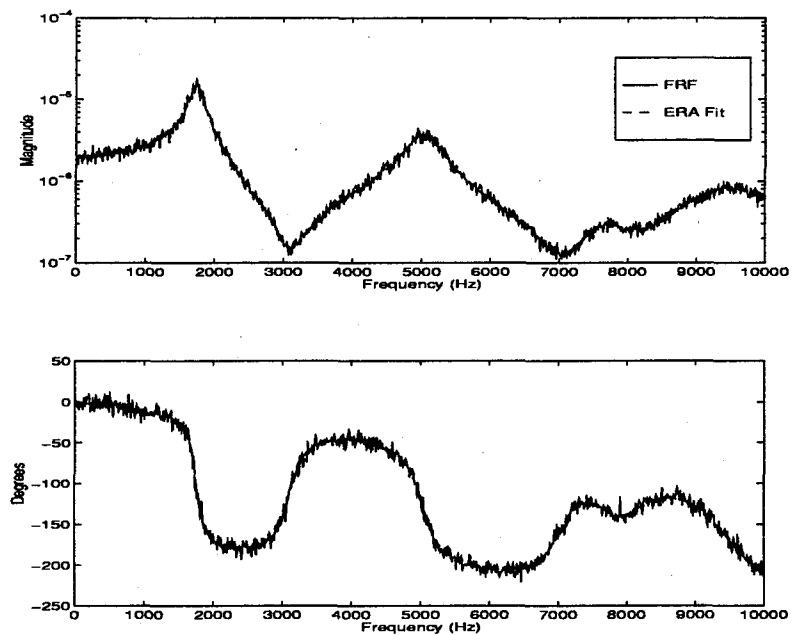


Figure I.14 Displacement Y2 Force Input F2

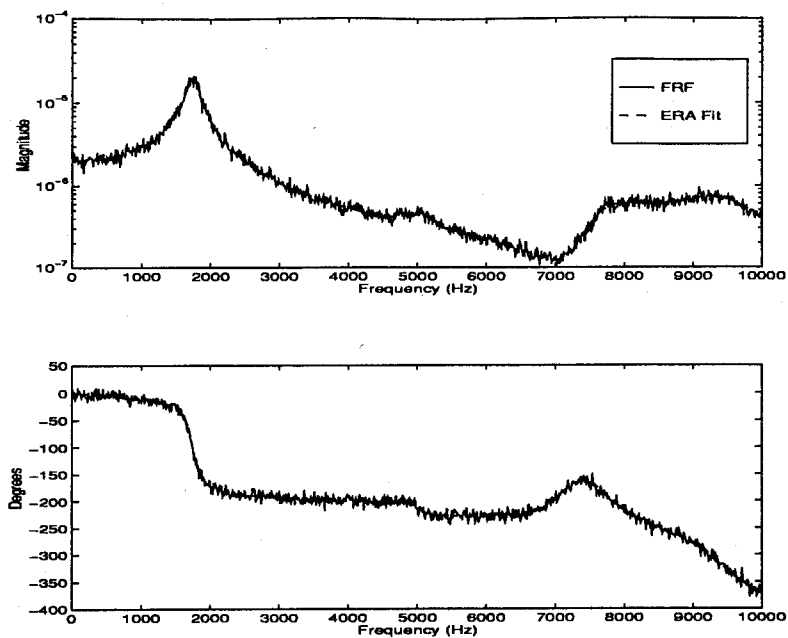


Figure I.15 Displacement Y3 Force Input F2

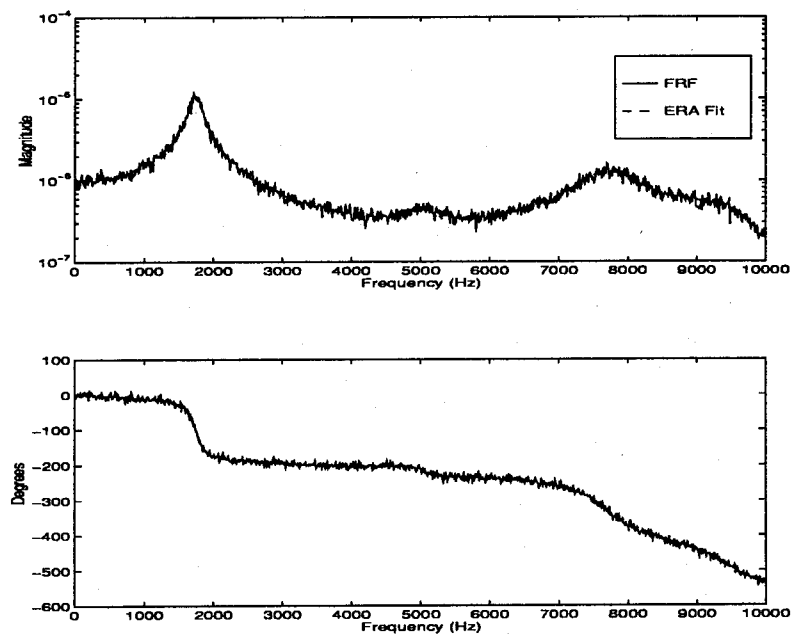


Figure I.16 Displacement Y1 Force Input F3

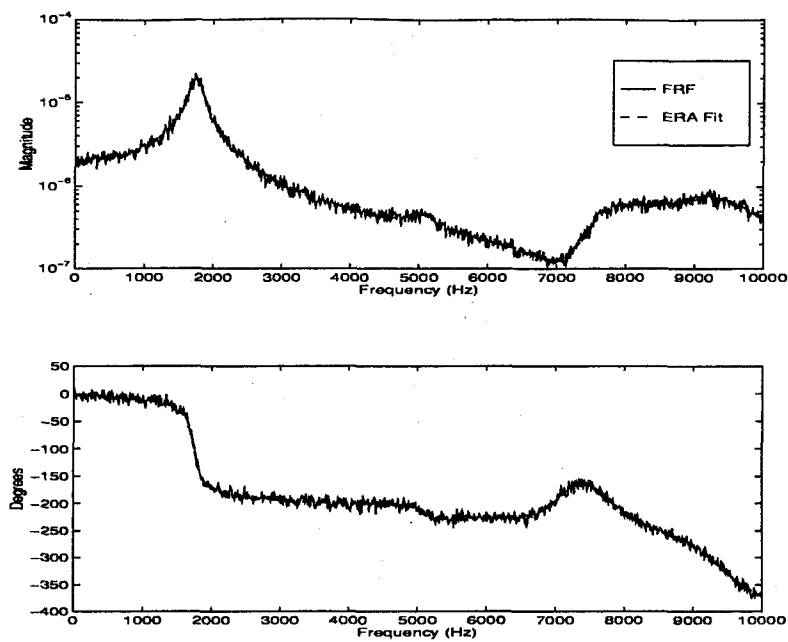


Figure I.17 Displacement Y2 Force Input F3

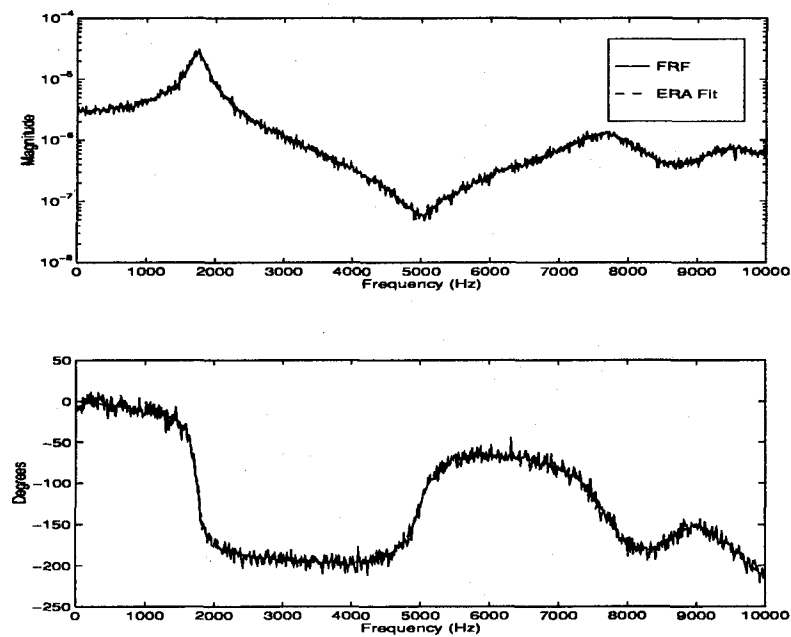


Figure I.18 Displacement Y3 Force Input F3

Appendix J. Computer Routines

This appendix contains an explanation of each of the MATLABTM ERA routines and how they are used. Also, the parameter settings used to identify the PACOSS DTA and the truth model are presented.

J.1 ERA Routines

The ERA routines used for this identification can be found in Appendix L. These routines, as mentioned earlier, were written by Capt. Richard Cobb. The following sections give a brief overview of each routine. The last section shows the settings used to generate the results shown in Appendices A-H and the corresponding A , B , C , and D matrices.

J.1.1 frfinter.m. This function integrates the frequency response functions. The data obtained by the IP program was acceleration transfer functions. When designing control systems, acceleration is seldom used; therefore, the frequency response functions must be integrated. It follows then that the number of integrations to be performed depends on the choice of using either velocity or displacement FRFs in the control system design.

J.1.2 mirror.m. This function was written in order to produce a symmetric frequency response function given an asymmetric response function. An FRF obtained from the IP program (after ASCII conversion) is input into this *m-file*. The FRF input into ERA routines is composed of $(n/2) + 1$ complex points. This points are for the positive frequencies. In order to perform an inverse fast Fourier transform (IFFT), an n point symmetric function is needed. This function mirrors the positive frequencies into negative frequencies; thus producing an n point symmetric function. After the mirroring has been accomplished, the IFFT can be performed on the data.

J.1.3 mimoera.m. Given the impulse responses, number of sensors, number of actuators, and number of states, this function produces the A , B , C and D matrices for the

system. The impulse responses are obtained by taking the inverse fast Fourier transform of the output of *mirror.m*.

J.1.4 weave.m and mimohank.m. *weave.m* weaves the individual impulse response functions into the proper form to be utilized by *mimohank.m*.

mimohank.m constructs the block Hankel matrix for a MIMO system. This routine needs the output of *weave.m* (the combined impulse functions) and the number of sensors in the system. The output of this system is the block Hankel matrix for use in *mimoera.m*. A singular value decomposition will be performed on a portion of this Hankel matrix.

J.1.5 parm.m. This routine is a utility function. If *p* is the parameters vector, typing **parm(p)** in the MATLABTM command window produces output on the screen showing the current parameter settings and explains each parameter. By typing **parm**, the *help* file for *parm.m*, which explains each parameter, is displayed on the screen.

J.1.6 frffilt.m. This routine encompasses all of the other routines. This function calls *mirror.m*, performs the inverse Fourier transform, and (after some filtering) calls *mimoera.m*. This routine needs the FRF matrix, the frequency vector (FreqV), and the parameters vector. The output of this function is the *A*, *B*, *C*, and *D* matrices as well as FRF/ERA plots.

J.2 Parameter Settings

The next two sections give the parameter settings used in the identification of the PACOSS DTA and the identification of the truth model.

J.2.1 PACOSS DTA. The system to be identified had 8 inputs and 8 outputs. Looking to identify the modes below 14 Hz, the number of states was set to 80. The *lh* factor, one of the parameters which determines the size of the Hankel matrix, was set to 5 to lengthen the time series used to generate the Hankel matrix. One integration was performed on the data. The original data was acceleration frequency response data. It was determined that one

integration to obtain velocity response data was sufficient to determine the dynamics of the system. Two integrations could have been performed; however, in the interest of conserving CPU time, only one integration was performed. No filtering was used. This was accomplished by setting the low pass filter cutoff parameter to -1.0. Due to the fact that acceleration data was being used and that initial accelerations are usually rather large, it was found that the first three data points needed to be set to zero. Figures were plotted using a separate routine shown in Appendix L. A discrete model was used. The number of sample points was set to 1024. It was also decided that a singular value cutoff was not needed. These parameter settings were input into the parameters vector, which is called by the ERA routines to generate the state-space model of the PACOSS DTA found in this document.

J.2.2 Truth Model. The parameter settings used for the truth model are not much different from those used in the identification of the test article. Only those that differ will be discussed here.

The truth model system has eight states with 3 inputs and 3 outputs. No integrations were used since the output of the system was displacement. The *lh* factor was set to 4 to increase the size of the Hankel matrix since the system was small. Also, no zeros were placed at the beginning of the data sets. The remaining settings were the same as those listed for the PACOSS DTA.

In order to generate the state-space model and provide for user-interface, a routine (*erasetup.m*) was written which sets up the FRF matrix, clears memory to speed processing, and saves the workspace. Another function (*plotfrfs.m*) was written which loads the workspace, plots the FRFs and their corresponding ERA fits, and saves them automatically to a file. The code for both of these routines is shown in Appendix L.

Appendix K. C Code Used in Friction Compensation

K.1 C Code for Coulomb Damping Compensator

This section contains the code use to implement the Coulomb damping compensator.

```

/*****
*
*          COULOMB DAMPING COMPENSATOR
*
*   Author: Capt Robert Daryl Woods   Date: 29 Aug 1994
*
*       This routine was written to compensate for the Coulomb
*       Damping present in each actuator. The Coulomb damping
*       coefficients have already been determined through experi-
*       mentation and tweaked using the Optima to ensure the
*       compensation voltage being applied does not start the
*       actuator. The program has to be recompiled and downloaded
*       each time the disturbance actuator changes.
*
*          USAGE
*
*       The program is compiled using the couldamp.cmd script
*       file. This file looks like:
*
*       cc -c -O -f68881 -o couldamp.o couldamp.c
*       ld -N -o couldamp couldamp.o convrt16.o qc.l std.l stdio.l
*
*       The file is than downloaded to the Optima using:
*
*           download couldamp
*
*****/

/*          DEFINITIONS          */
#define M 9

/*          INCLUDE STATEMENTS          */
#include "stdio.h"
#include "math.h"
```

```

float *APalloc();

/*                      MAIN PROGRAM                      */
main()
{
    float *b,*C,*c_force,*offset,f_out;
    float dt,zone,incal,outcal,endtime,endcount,in_gain,out_gain;

    int i,ii,j,k,l;

/*                      Dimensionalize Variables          */
    APInit(0);
    b      = APalloc (M);
    C      = APalloc (M);
    c_force = APalloc (M);
    offset  = APalloc (M);

/*                      Set Offsets for each Channel      */
    offset[0] = 0.051;      /* Offset for Channel 1 */
    offset[1] = 0.010;      /* Offset for Channel 2 */
    offset[2] = 0.071;      /* Offset for Channel 3 */
    offset[3] = 0.071;      /* Offset for Channel 4 */
    offset[4] = 0.062;      /* Offset for Channel 5 */
    offset[5] = 0.064;      /* Offset for Channel 6 */
    offset[6] = 0.049;      /* Offset for Channel 7 */
    offset[7] = 0.046;      /* Offset for Channel 8 */
    offset[8] = 0.036;      /* Offset for Channel 9 */

/*                      Set Coulomb Damping Amounts       */
    c_force[0] = 0.0325;    /* Actuator 1 LVT */
    c_force[1] = 0.0500;    /* Actuator 2 LVT */
    c_force[2] = 0.0250;    /* Actuator 3 LVT */
    c_force[3] = 0.0100;    /* Actuator 4 LVT */
    c_force[4] = 0.0150;    /* Actuator 5 LVT */
    c_force[5] = 0.0500;    /* Actuator 6 LVT */
    c_force[6] = 0.0325;    /* Actuator 9 LVT */
    c_force[7] = 0.0175;    /* Actuator 10 LVT */
    c_force[8] = 0.00;

/*                      Assign Conversions and Variables  */
    incal      = 1./0.6;    /* Convert input Volts -> in/sec */

```

```

outcal  = 1./0.2;          /* Convert output lbs -> Volts */
dt      = 0.005;          /* Sample Time */
zone    = 0.2;            /* Dead Zone level (+/- in/sec) */
endtime = 3600;           /* End of Data Taking */
endcount = endtime/dt;    /* End of Loop */
in_gain = -0.0003;        /* Gain on input (ATD) Board */
out_gain = 3333.333;      /* Gain on Output (DTA) Board */

/* ----- */
/* -      Intialize cards with dt (second) sample time
/* ----- */
if (CVT_INIT(dt,3)) {
    printf("Sample Time Out of Range\n");
    exit(0);
}

/* ----- */
/* -      Start Conversion Clock
/* ----- */
CVT_STRT();

/* ----- */
/* -      Start For Loop
/* ----- */
for (i=0;i <= endcount; i++) {          /* Start loop */

/* ----- */
/* -      Get Analog Input
/* ----- */
if (CVT_ATD(&C[0], &C[1], &C[2], &C[3], /* First Card Data */
            &C[4], &C[5], &C[6], &C[7], /* Second Card Data */
            &C[8], &C[9], &C[10], &C[11])) { /* Third Card Data */
    printf("Overrun Error \n");
    exit(0);
}

/* ----- */
/* -      Multiply input by Input gain & subtract offset
/* ----- */
for (j=0; j<=8; j++) {
    C[j] = (C[j] * in_gain) - offset[j];
}

/* ----- */
/* -      Convert 1st 8 inputs to Velocity

```

```

* ----- */
for (l=0; l<=7; l++) {
    C[l] = C[l] * incal;
}
/* ----- *
* -          SIGNUM CALCULATIONS
* ----- */
for (ii=0; ii<=7; ii++) {
    f_out = 0.;
    if (C[ii] > zone) f_out = c_force[ii];
    if (C[ii] < -zone) f_out = -c_force[ii];
    b[ii] = f_out * 0.9;          /* 90% of friction force */
}
/* ----- *
* -          Convert lbs to Voltage
* ----- */
for (k=0; k<=7; k++) {
    b[k] = b[k] * outcal * out_gain;
}

/* ----- *
* -          Add compensation to random input from Analyzer
* ----- */
b[8] = (C[8] * out_gain) + b[0];

/* ----- *
* -          Send Analog Output
* ----- */
if (CVT_DTA(&b[0], &b[1], &b[2], &b[3],      /* First Card Data */
           &b[4], &b[5], &b[6], &b[7],      /* Second Card Data */
           &b[8], &b[9], &b[10], &b[11])) { /* Third Card Data */
    printf("Overrun Error \n");
    exit(0);
}
}
/*      End Loop      */
/*      End Program   */

```

K.2 Coulomb Damping "Tweaking" Routine

This routine was used to "tweak" the calculated Coulomb damping coefficients. The compensated output is displayed on the IP program to determine the accuracy of the friction coefficient for each actuator. Depending on the output, the coefficient for that actuator is adjusted either up or down and the routine is re-compiled, downloaded, and executed again on the Optima/3. This cycle continues until all of the coefficients are correct. This may have to be accomplished just prior to taking data for a compensated system since the coefficients seem to be affected by the temperature and humidity changes.

```
/******  
*  
*           Friction Tweaking Routine  
*  
*   Author: Capt Robert Daryl Woods       Date: 23 Aug 1994  
*  
*   This routine was written to tweak the dry friction  
*   constants on Actuators 9 & 10. It can also be used on the  
*   other 6 actuators. The LVT output will be looked at to  
*   determine the friction constant. The free decay method  
*   (lifting mass to a preset height and releasing) will be  
*   employed. The compensated output will be looked at on the  
*   IP display to determine the accuracy of the coefficient.  
*   The coefficient is then adjusted up or down as appropriate.  
*  
*****/  
  
/*           DEFINITIONS           */  
#define M 9  
  
/*           INCLUDE STATEMENTS           */  
#include "stdio.h"  
#include "math.h"  
  
float *APalloc();  
  
/*           MAIN PROGRAM           */  
main()  
{  
    float *b,*C,*c_force,*offset,f_out;
```



```

float dt,zone,incal,outcal,endtime,endcount,in_gain,out_gain;

int i,ii,j,k,l;

/*          Dimensionalize Variables          */
APInit(0);
b      = APAlloc (M);
C      = APAlloc (M);
c_force = APAlloc (M);
offset = APAlloc (M);

/*          Set Offsets for each Channel          */
offset[0] = 0.051;      /* Offset for Channel 1 */
offset[1] = 0.010;      /* Offset for Channel 2 */
offset[2] = 0.071;      /* Offset for Channel 3 */
offset[3] = 0.071;      /* Offset for Channel 4 */
offset[4] = 0.062;      /* Offset for Channel 5 */
offset[5] = 0.064;      /* Offset for Channel 6 */
offset[6] = 0.049;      /* Offset for Channel 7 */
offset[7] = 0.046;      /* Offset for Channel 8 */
offset[8] = 0.036;      /* Offset for Channel 9 */

/*          Initial Coulomb Damping Coefficients          */
c_force[0] = 0.0591;    /* Actuator 1 LVT */
c_force[1] = 0.0565;    /* Actuator 2 LVT */
c_force[2] = 0.0549;    /* Actuator 3 LVT */
c_force[3] = 0.0353;    /* Actuator 4 LVT */
c_force[4] = 0.0574;    /* Actuator 5 LVT */
c_force[5] = 0.0518;    /* Actuator 6 LVT */
c_force[6] = 0.2824;    /* Actuator 9 LVT */
c_force[7] = 0.3188;    /* Actuator 10 LVT */

/*          Assign Conversions and Variables          */
incal    = 1./0.6;      /* Convert input Volts -> in/sec */
outcal   = 1./0.2;      /* Convert output lbs -> Volts */
dt       = 0.005;      /* Sample Time */
zone     = 0.2;         /* Dead Zone level (+/- in/sec) */
endtime  = 3600;        /* End of Data Taking */
endcount = endtime/dt;  /* End of Loop */
in_gain  = -0.0003;     /* Gain on input (ATD) Board */
out_gain = 3333.333;    /* Gain on Output (DTA) Board */

```

```

/* ----- *
* -      Intialize cards with dt (second) sample time
* ----- */
if (CVT_INIT(dt,3)) {
    printf("Sample Time Out of Range\n");
    exit(0);
}
/* ----- *
* -      Start Conversion Clock
* ----- */
CVT_STRT();

/* ----- *
* -      Start For Loop
* ----- */
for (i=0; i <= endcount; i++) {          /* Start loop */

/* ----- *
* -      Get Analog Input
* ----- */
if (CVT_ATD(&C[0], &C[1], &C[2], &C[3], /* First Card Data */
            &C[4], &C[5], &C[6], &C[7], { /*Second Card Data */
    printf("Overrun Error \n");
    exit(0);
}

/* ----- *
* -      Multiply input by Input gain & subtract offset
* ----- */
for (j=0; j<=7; j++) {
    C[j] = (C[j] * in_gain) - offset[j];
}

/* ----- *
* -      Convert 1st 8 inputs to Velocity
* ----- */
for (l=0; l<=7; l++) {
    C[l] = C[l] * incal;
}

/* ----- *
* -      SIGNUM CALCULATIONS
* ----- */
for (ii=0; ii<=7; ii++) {
    f_out = 0.;

```

```

        if (C[ii] > zone) f_out = c_force[ii];
        if (C[ii] < -zone) f_out = -c_force[ii];
        b[ii] = f_out;
    }
    /* ----- *
    * -           Convert lbs to Voltage
    * ----- */
    for (k=0; k<=7; k++) {
        b[k] = b[k] * outcal * out_gain;
    }

    /* ----- *
    * -           Send Analog Output
    * ----- */
    if (CVT_DTA(&b[0], &b[1], &b[2], &b[3], /* First Card Data */
               &b[4], &b[5], &b[6], &b[7])) { /* Second Card Data */

        printf("Overrun Error \n");
        exit(0);
    }
} /* End Loop */
} /* End Program */

```

Appendix L. MATLABTM Code

This appendix is composed of three sections. The first section contains the *m-files* which compose the ERA algorithm. The next section contains *m-files* which load the data and implement Capt. Cobb's routines with the proper parameters. The last section contains the interactive routine used to calculate the Coulomb damping coefficients.

L.1 MATLABTM ERA Code

This section contains *m-files* written to perform the Eigensystem Realization Algorithm. The first section contains the actual ERA routine. These files were written by Capt. Richard Cobb, a Ph.D. candidate at the Air Force Institute of Technology. A description of each *m-file* is given in Appendix J.

L.1.1 frffilt.m.

```
function [a,b,c,d,Ts]=frffilt(FRF,FreqV,paramtrs)
%
% [a,b,c,d,Ts]=filtfrf(FRF,FreqV,paramtrs)
%   FRF=[frf11 frf21 ... frf12 frf22 .....]
%   frfij = frequency response function at sensor i from input j
%   FreqV = frequency vector in Hz, starting at 0 Hz.
%   paramtrs = parameter vector as defined below
%
%   paramtrs=[nmeas,ninput,lh,ns,ni,ford,Hzbreak,nz,fn,cord,cut]
%
%   nmeas = # of sensors
%   ninput = # of actuators
%   lh is a factor 2 < lh < length(FRF)/ns to determine size of
%   Hankel matrix typically 3 - 8. The resulting Hankel matrix
%   is nxn where n is approximately ns*nmeas*ninput*lh/2
%   ns = # number of states for model
%   ni = # of integrations on data (set to 1 for accel to velocity)
%   ford = butterworth filter order, ignored if break is negative
%   Hzbreak = cutoff for low pass filter of impulse response in Hz
%           set break < 0 for no filtering
%   nz = # of zeros at low freq end of spectrum
```

```

%   fn = figure number for first plot (<0 to surpress plots)
%   cord = 1 for continuous model or 2 for discrete model
%   cut = 1 to prompt for singular value cutoff (using figure fn)
%   ndpts = number of original sample points
%           (ie 2^N, 512,1024,2048) set to zero for no padding
%
%                               rcobb 15/feb/94
lenfrf=size(FRF,1);
if length(FreqV) - lenfrf ~= 0
    error('incompatable length FRF and Freq vectors')
end
nmeas=paramtrs(1); ninput=paramtrs(2); lh=paramtrs(3);
ns=paramtrs(4); ni=paramtrs(5); ford=paramtrs(6);
Hzbreak=paramtrs(7); nz=paramtrs(8); fn=paramtrs(9);
cord=paramtrs(10); cut=paramtrs(11); ndpts=paramtrs(12);

% Integrate frf to get vel/force or disp/force xfer
% function
for i=1:ni
    FRF=frfinter(FRF,FreqV*2*pi);
end
% pad frf with zeros to match sample rate
if ndpts ~= 0
    FRF=[FRF;zeros(ndpts+1-lenfrf,size(FRF,2))];
end
% compute sample rate of data
Ts=1/((FreqV(2)-FreqV(1))*(size(FRF,1)-1)*2);
% set low freq end to zero
if nz > 0
    for i=1:nmeas*ninput
        FRF(1:nz,i)=zeros(nz,1);
    end
end
%compute impulse response and filter data
for i=1:nmeas*ninput
    G(:,i)=mirror(FRF(:,i));
end
h=real(ifft(G));
if Hzbreak > 0
    [B,A]=butter(ford,Hzbreak*2*Ts);
    for i=1:nmeas*ninput
        h(:,i)=filtfilt(B,A,h(:,i));
    end
end

```

```

end
end
%
[a,b,c,d]=mimoera(h(2:ns*lh+1,:),nmeas,ninput,ns,fn,cut);
%
if cord == 1
    [a,b]=d2c(a,b,Ts); Ts=[];
    % cut identified poles and zeros above max frequency
    % [a,b,c,d]=cutzeros(a,b,c,d,max(FreqV));
end
if fn > 0 % print plots for positive figure number
    for i=1:ninput
        if cord == 1
            [mag,phase]=bode(a,b,c,d,i,FreqV*2*pi);
        else
            [mag,phase]=undbode(a,b,c,d,Ts,i,FreqV*2*pi);
        end
        figure(fn-1+i),subplot(2,1,1), ...
            semilogy(FreqV,abs(FRF(1:lenfrf,nmeas*(i-1)+1: ...
                nmeas*(i-1)+nmeas)),FreqV,mag)
        figure(fn-1+i),subplot(2,1,2), ...
            plot(FreqV,(angle(FRF(1:lenfrf,nmeas*(i-1)+1: ...
                nmeas*(i-1)+nmeas)))*180/pi,FreqV,phase)
    end
end
end
%eof

```

L1.2 mimohank.m.

```
function H=mimohank(Y,nmeasure)
%
% H=mimohank(Y, # of sensors)
%
% This routine constructs the block Hankel matrix for a
% MIMO system. Y is the output of WEAVE (combined impulse
% functions)
%
% rcobb 3/feb/94
%
nin=nmeasure;
% correct length of Y to avoid diagonal conflicts
[x,ncol]=size(Y);
Y=Y(1:fix(x/(4*nin))*nin*4,:);
[x,ncol]=size(Y);
nrows=x/nin;
nrblock=nrows/2;
ncblock=nrows/2;
H=zeros(nrblock*nin,ncblock*ncol);
for j=0:ncblock-1
    H(1:nrblock*nin,ncol*j+1:ncol*j+ncol)=Y(j*nin+1: ...
                                                j*nin+nin*nrblock,1:ncol);
end
%eof
```

L.1.3 mimoera.m.

```
function [amod,bmod,cmmod,dmod]=mimoera(h,nmeas,ninput,nstates,fn,cut)
%
%   [a,b,c,d]=mimoera([h1 h2 .....],# of sensors,# of actuators,
%                                   # of states,cut)
%   where
%       hi = impulse response to input i
%       hi has as many columns as there are sensors
%
%       fn,cut = if cut = 1 then a singular value plot will be
%               displayed in figure(fn) and prompt the user
%
%                                   rcobb 15/feb/94
%
if nmeas*ninput == 1
Y=h;
else
Y=weave(h,nmeas,ninput);
end
H=mimohank(Y,nmeas);
n=size(H,1)/nmeas;
Ho=H(1:(n-1)*nmeas,1:(n-1)*ninput);
H1=H(1:(n-1)*nmeas,ninput+1:n*ninput);
disp(sprintf('SVD of a %6i by %6i matrix in progress',size(Ho)))
[U,S,V]=svd(Ho,0);

%[U2,psi,V2]=svdra(Ho,nstates,3);
%[U4,S,V4]=svd(psi);
%U=U2*U4;
%V=V2*V4;

s=diag(S);
if length(s) < nstates
error('increase length of time series')
elseif length(s) < 2*nstates
disp('Warning, increase length of time series')
end
% plot singular values and prompt for cutoff
if cut ==1
figure(fn)
clf; semilogy(s)
```



```

xlabel('Select the Singular Value Cutoff (# of states)')
ylabel('log of sv')
nstates=round(myginput);nstates=nstates(1);
disp(sprintf('%5i states selected',nstates))
end
s=s(1:nstates);
u=U(:,1:nstates);
v=V(:,1:nstates);
Ep=[eye(ninput) zeros(ninput,(n-2)*ninput)]';
Eq=[eye(nmeas) zeros(nmeas,(n-2)*nmeas)];
srt=sqrt(s);
amod=diag(1./srt)*u'*H1*v*diag(1./srt);
bmod=diag(srt)*v'*Ep;
cmode=Eq*u*diag(srt);
dmod=zeros(nmeas,ninput);
%eof

```

L.1.4 frfinter.m.

```
function [frfv]=frfinter(frf,w)
%
%           [frfv]=frfinter(frf,w)
%
%       This function, which integrates an frf=[frf1 frf2 ...],
%       is used to convert accelerometer (or velocity) measurements
%       to velocity (or position) measurements. W is the frequency
%       vector in rad/sec.
%
%                                           rcobb 2/feb/94
[m,n]=size(frf);
for k=1:n
    for i=2:m
        frfv(i,k)=frf(i,k)/(sqrt(-1)*w(i));
    end
end

for k=1:n
    if w(1)==0,
        frfv(1,k)=0;
    else
        frfv(1,k)=frf(1,k)/(sqrt(-1)*w(i));
    end
end

%eof
```

L.1.5 mirror.m.

```
function X=mirror(Y)
%
%               X=mirror(Y)
%
%   This function produces an 'n' point symmetric FFT from
%   an (n/2+1) point asymmetric FFT.
%
%                               R. Cobb, 1/Feb/94
n=size(Y,1);

for i=1:n-2
    Ym(n-i) = Y(i+1);
end

X=[Y;Ym(2:n-1)'];
% eof
```

L.1.6 weave.m.

```
function Y=weave(X,nmeas,nin)
%
%   Y=weave([h11 h21... h12 h22...],# of sensors,# of actuators)
%
%   This function weaves the impulse response vectors together
%   for use by the MIMO Hankel algorithm where
%
%       h(ij) = impulse response to input j from sensor i
%
%   see MIMOHANK ,MIMOERA
%
%                                     rcobb 3/feb/94
nr=size(X,1);
Y=zeros(nr*nmeas,nin);
for i=0:nr-1
for k=0:nin-1
for j=0:nmeas-1
    Y(i*nmeas+1+j,k+1)=X(i+1,k*nmeas+j+1);
end
end
end
%eof
```

L.1.7 parm.m.

```
function parm(p)
%   paramtrs=[nmeas,ninput,lh,ns,ni,ford,Hzbreak,nz,fn, ...
%               cord,cut,ndpts]
%
%   nmeas = # of sensors
%   ninput = # of actuators
%   lh is a factor  $2 < lh < \text{length}(\text{FRF})/\text{ns}$  to determine size of
%   of Hankel matrix typically 3 - 8. The resulting Hankel
%   matrix is nxn where  $n = \text{ns} * \text{nmeas} * \text{ninput} * \text{lh} / 2$ 
%   ns = # number of states for model
%   ni = # of integrations on data (set to 1 for accel to vel.)
%   ford = butterworth filter order, ignored if break is negative
%   Hzbreak = cutoff for low pass filter of impulse response
%   in Hz (set break < 0 for no filtering)
%   nz = # of zeros at low freq end of spectrum
%   fn = figure number for first plot (<0 to surpress plots)
%   cord = 1 for continuous model or 2 for discrete model
%   cut = 1 to prompt for singular value cutoff (using Fig. fn)
%   ndpts = number of original sample points
%   (ie  $2^N$ , 512,1024,2048) set to zero for no padding
if nargin == 0
help parm
else
disp(sprintf('%5i = # of sensors',p(1)))
disp(sprintf('%5i = # of actuators',p(2)))
disp(sprintf('%5i = lh is a factor  $2 < lh < \text{length}(\text{FRF})/\text{ns}$  to ...
determine size of hankel matrix ',p(3)))
disp(sprintf('%5i = # number of states for model',p(4)))
disp(sprintf('%5i = # of integrations on data (set to 1 for ...
accel to velocity)',p(5)))
disp(sprintf('%5i = butterworth filter order, ignored if break ...
is negative',p(6)))
disp(sprintf('%5.1f = cutoff for low pass filter of impulse ...
response in Hz set break < 0 for no ...
filtering',p(7)))
disp(sprintf('%5i = # of zeros at low freq end of spectrum',p(8)))
disp(sprintf('%5i = figure number for first plot (<0 to ...
suppress plots)',p(9)))
disp(sprintf('%5i = 1 for continuous model or 2 for discrete ...
model',p(10)))
```

```
disp(sprintf('%5i = 1 to prompt for singular value cutoff ...  
            (using figure fn)',p(11)))  
disp(sprintf('%5i = number of original sample points set to ...  
            zero for no padding',p(12)))  
end  
%eof
```

L.2 MATLABTM ERA Setup Routine

The routines presented in this section were written to setup the FRF vector, to clear variables and vectors not needed for computation, and to start *frffilt.m* to obtain the state space model.

L.2.1 *erasetup.m*.

```
% -----  
%                                     ERASETUP  
% -----  
%      This routine ERASETUP runs the ERA algorithm. It allows  
% (prompts) the user to input the individual parameters which  
% make up the parameters vector (p) needed by FRFFILT. It  
% assembles the parameters into vector p, loads the transfer  
% functions (xfer**'s) using FRFBUILD and starts the ERA routine  
% in the FRFFILT function.  
%                                     Daryl Woods  
% -----  
%  
% Prompt user to pick parameters %  
%  
nmeas = input('Number of measurements: ');  
ninput = input('Number of actuators: ');  
lh      = input('Input lh (determines size of Hankel matrix): ');  
ns      = input('Number of states for the model: ');  
ni      = input('Number of intergrations on the data: ');  
ford    = input('Order of Butterworth Filter: ');  
Hzbrk   = input('Cutoff for the Low-Pass Filter ...  
              (<0 for no filtering): ');  
nz      = input('Number of zeros at low frequency end of ...  
              spectrum: ');  
fn      = input('Figure number for first plot (<0 to suppress ...  
              plots): ');  
cord    = input('1-Continuous or 2-Discrete: ');  
cut     = input('Set equal to 1 for SV cutoff: ');  
ndpts   = input('Number of original points: ');  
%  
% Setup parameters vector %  
p = [nmeas,ninput,lh,ns,ni,ford,Hzbrk,nz,fn,cord,cut,ndpts];  
%  
% Prompt user to see the Parameters vector %
```

```

disp('Would you like to see the parameters vector (p)');
yorn = input('1 for Yes/2 for No ');
%
if yorn == 1;
    parm(p)
end
%
% Clear individual parameters since they are in p vector %
clear yorn nmeas ninput lh ns ni ford Hzbrk nz fn cord;
clear cut ndpts;
%
% Load in workspace with FRFs and build FRF matrix %
load xfers3;      % Loads FRFs and FreqV %
frfbuilt;         % Place FRFs into FRF matrix %
clearxfer;        % Clears individual FRFs to save memory %
%
% Call frffilt to start ERA routine%
[a,b,c,d,ts] = frffilt(FRF,FreqV,p);
%
% End-Of-File

```


L.2.2 frfbuild.m.

```
% -----  
%                               FRFBUILD  
% -----  
%   This routine builds the frequency response function  
%   (FRF) matrix for use in the FRFFILT function.  
%                               Daryl Woods  
% -----  
  
FRF=[];  
  
for j=1:p(2)    % Input index %  
    for i=1:p(1) % Measurement index %  
        FRF = eval(['[FRF,xfer',int2str(i),int2str(j),']']);  
    end  
end  
end
```

L.2.3 clearxfers.

```
% -----  
%                                     CLEARXFERS  
% -----  
%  
%      Since the frequency response functions have been  
% placed in the FRF matrix, the FRFs will be deleted to  
% save on memory. At the end of the program, l and k  
% (the counters) are also cleared.  
%                                     Daryl Woods  
% -----  
for k = p(2);  
    for l = p(1);  
        eval(['clear Xfer',int2str(l),int2str(k)]);  
    end  
end
```

L.2.4 plotfrfs.m.

```
% -----  
%                               PLOTFRFS  
% -----  
%  
%       This routine plots each FRF and its  
%       corresponding ERA fit on a separate plot and saves  
%       these plots to an EPS file for use with LaTeX.  
%                               Daryl Woods  
% -----  
lenfrf = length(FRF);           % Find length of FRF %  
FRFv = frfinter(FRF,FreqV*2*pi); % Integrate FRF matrix %  
k = 8;                          % k = 8 for DTA %  
%k = 3;                         % k = 3 for truth model %  
  
for i = 1:k;  
    [mag,phase] = dbode(a,b,c,d,ts,i,FreqV*2*pi);  
  
    for j = 1:k;  
        figure(j+(k*(i-1))), ...  
            subplot(211),semilogy(FreqV(1:lenfrf), abs(FRFv(1: ...  
                lenfrf),j+(k*(i-1)))),'-',FreqV(1: ...  
                lenfrf),mag(1:lenfrf,j),'--'), ...  
                ylabel('Magnitude'), ...  
                xlabel('Frequency (Hz)'), ...  
                legend('-', 'FRF', '--', 'ERA Fit'), ...  
            subplot(212),plot(FreqV(1:lenfrf),unwrap(angle(FRFv ...  
                (1:lenfrf,j+(k*(i-1)))))*180/pi,'-', ...  
                FreqV(1:lenfrf),phase(1:lenfrf,j), ...  
                '--'),ylabel('Degrees'), ...  
                xlabel('Frequency (Hz)');  
        orient tall;  
        eval(['print -deps /rwoods/latex/figs/figure', ...  
            int2str(j),int2str(i),'.eps'])  
    end  
end
```

L.3 Friction Coefficient Calculation Routine

```
% -----  
%                               FC.M  
% -----  
% This M-File plots the free-decay of the actuator and  
% calculates the friction coefficient for the actuator.  
% Data was obtained via the Instrument Program and con-  
% verted to Matlab format using PCMATLNK. The two vectors  
% used are 2048x1; however, not all of this data is needed.  
% Therefore, a lower and upper end of usable data is  
% specified. The user is prompted to pick to peaks over  
% which the amplitude change is calculated. The Coulomb  
% friction coefficient is then calculated and output to  
% the screen.  
%                               Daryl Woods  
% -----  
min=200;      % Lower End of Range  
max=400;      % Upper End of Range  
%  
% Plot Time vs Data for that range  
plot(TimeV(min:max),Timei3(min:max)./(0.6))  
%  
% Prompt user for number of points to be fit  
n = input('Number of Points to be Fit?');  
cycles = (n - 1);  
[x,y] = ginput(n); % User must click on points  
%  
deltaY = y(1)-y(n); % Get delta Y  
%  
% Calculate rps (frequency in radians/sec)  
deltaX = x(n) - x(1);  
freq = cycles / deltaX;  
rps = 2 * pi * freq;  
%  
k = 1.0000;    % Spring Constant (lb/in)  
%  
FC = (k * deltaY) / (4 * cycles * rps);  
disp('Coulomb Friction is:'),FC
```

Bibliography

1. Cobb, Capt. Richard. "MATLAB ERA Routines." Computer Code, September 1994.
2. Crassidis, John L. and others. "Robust Identification and Vibration Suppression of a Flexible Structure," *AIAA Guidance, Navigation and Control Conference*, 1560-1570 (1993).
3. Gehling, Russell N. and others. *Passive and Active Control of Space Structures: Final Report, November 1988-April 1991*. Contract F33615-82-C-3222, Denver CO: Martin Marietta Astronautics Group, June 1991.
4. George, Capt Scott E. *Modal Analysis and Modeling of a Lightly Damped Large Space Structure*. MS thesis, AFIT/GA/ENY/92J-01, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, June 1992.
5. Juang, Jer-Nan and Richard S. Pappa. "An Eigensystem Realization Algorithm for Modal Parameter Identification and Model Reduction," *Journal of Guidance and Control*, 8:620-627 (July 1985).
6. Matheson, Lt Chad T. *Vibration Suppression in Large Flexible Space Structures using Active Control Techniques*. MS thesis, AFIT/GAE/ENY/92D-13, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1992.
7. Morgenthaler, Daniel R. *Passive and Active Control of Space Structures Volume 1: Damping Design Methodology, Final Report, September 1987-November 1988*. Contract F33615-82-C-3222, Denver CO: Martin Marietta Astronautics Group, September 1990.
8. Nash, Capt Anthony R. *Modeling of a Large Undamped Space Structure using Time Domain Techniques*. MS thesis, AFIT/GA/ENY/93D-07, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1993.
9. *Passive and Active Control of Space Structures: Design and Fabrication of the PACOSS Dynamic Test Article*. Contract F33615-82-C-3222, Denver CO: Martin Marietta Astronautics Group, April 1988.
10. Systolic Systems, Inc., San Jose, CA. *Optima/3 Real-Time Nonlinear Control and Data Acquisition System Operations Manual*, November 1989.
11. Tektronix, Inc., Campbell, CA. *2600 Analyzer Application Library*, October 1990.
12. Tektronix, Inc., Campbell, CA. *2600 Analyzer User's Guide*, October 1990.
13. The Math Works Inc., Natick, MA. *MATLAB Reference Guide* (4.0 Edition), 1990.
14. Tse, Francis S. and others. *Mechanical Vibrations* (Second Edition). Boston: Allyn and Bacon, Inc., 1978.

Vita

Robert Daryl Woods [REDACTED] After graduating from Shelby County High School in May 1985, he attended the United States Air Force Academy. He graduated in May 1989 with a Bachelor of Science degree in Astronautical Engineering. His first assignment was the 67th Strategic Missile Squadron at Ellsworth AFB, South Dakota. During his tour of duty, he held the positions of Deputy Missile Combat Crew Commander; Instructor, Deputy Missile Combat Crew Commander; and Missile Combat Crew Commander. In May 1993, he entered the School Of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio to pursue a Master of Science degree in Astronautical Engineering.

Permanent address: [REDACTED]

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE December 1994	3. REPORT TYPE AND DATES COVERED Master's Thesis
4. TITLE AND SUBTITLE A STATE-SPACE MODEL OF A LARGE, LIGHTLY DAMPED SPACE STRUCTURE USING THE EIGENSYSTEM REALIZATION ALGORITHM			5. FUNDING NUMBERS
6. AUTHOR(S) Robert Daryl Woods Capt, USAF			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Wright-Patterson AFB OH 45433-6583			8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GA/ENY/94D-2
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) L. Kevin Slimak PL/VTs Kirtland AFB, NM 87117-5776			10. SPONSORING / MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited			12b. DISTRIBUTION CODE
13. ABSTRACT (Maximum 200 words) <p style="text-align: center;">Abstract</p> <p>Large, lightweight space structures of the future will require state-of-the-art vibration suppression systems. To design such a system, it is necessary to have a mathematical model that adequately describes the motion of the system. Coulomb damping compensation was utilized to remove a known nonlinearity from the frequency response functions obtained from the Passive and Active Control of Space Structures (PACOSS) Dynamic Test Article (DTA). Via the Eigensystem Realization Algorithm (ERA), the frequency response functions were then used to generate a mathematical representation of the structure's dynamics. Two models were obtained. The first model, without friction compensation, acted as a baseline. The second was obtained using friction compensation. Comparing the models, it was determined that active friction compensation was worthwhile and resulted in a more accurate mathematical description of the system.</p>			
14. SUBJECT TERMS Large Space Structures, Structural Vibration, Flexible Spacecraft, System Identification			15. NUMBER OF PAGES 164
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL