

# UC Berkeley

## UC Berkeley Electronic Theses and Dissertations

### Title

Development and implementation of advanced control methods for hybrid simulation

### Permalink

<https://escholarship.org/uc/item/8ff2s95t>

### Author

Kim, Hong

### Publication Date

2011

Peer reviewed|Thesis/dissertation

**Development and implementation of advanced control methods for hybrid simulation**

by

Hong Ki Kim

A dissertation submitted in partial satisfaction of the  
requirements for the degree of  
Doctor of Philosophy

in

Engineering—Civil and Environmental Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Božidar Stojadinović, Chair  
Professor Stephen A. Mahin  
Professor Jon Wilkening

Fall 2011

**Development and implementation of advanced control methods for hybrid simulation**

Copyright 2011  
by  
Hong Ki Kim

## Abstract

Development and implementation of advanced control methods for hybrid simulation

by

Hong Ki Kim

Doctor of Philosophy in Engineering—Civil and Environmental Engineering

University of California, Berkeley

Professor Božidar Stojadinović, Chair

Hybrid simulation is an effective way of testing structures that combines the benefits of a computational analysis and experimental testing techniques. Innovative structures consists of state-of-the-art components and assemblages whose function as a system needs to be tested experimentally. Often times, these components and assemblages push the controller and other testing equipment to its limits. Performing hybrid simulation with the controller in displacement control mode does not always suffice.

Force control, switch control and mixed control methods in hybrid simulation are explored in order to overcome these limitations and provide robust ways of performing hybrid simulation. Force control hybrid simulation is a type of hybrid simulation where the control system is in force control mode. Switch control hybrid simulation is another type of hybrid simulation where the control system switches between displacement control and force control modes. Switch control hybrid simulation is applicable with setups that have only one control degree-of-freedom. Mixed control hybrid simulation is an extension of switch control hybrid simulation where multiple control degrees-of-freedom are switching between control modes independently of each other.

Force control, switch control and mixed control hybrid simulation methods are developed, tested and verified. The motivation for the development of these methods is discussed. New methods are presented and explained. New OpenFresco classes and Simulink/Stateflow models are developed to implement these methods. Then these three alternative control methods are tested using the  $\mu$ -NEES experimental setup at the *nees@berkeley* structural engineering lab using two different configurations. The first configuration is setup with one control degree-of-freedom which contains a relatively stiff specimen. The second configuration extends the first one with two control degrees-of-freedom. The performance of various time integration schemes with these methods are studied. These tests verify and validate the three alternative control methods as well as the new OpenFresco classes and Simulink/Stateflow models.

The test results are evaluated to assess the performance of each control method by comparing them first to the numerical results, then to the displacement control results and finally to each other. The force control methods, in general, provides better results than the displacement control method for the second  $\mu$ -NEES configuration. Switch control hybrid simulation methods provide better results than their force control counterpart for the one degree-of-freedom configuration but not better than the displacement control results. Mixed control hybrid simulation methods do not

provide better results than the force control methods because of the interaction between the two actuator control modes.

This dissertation concludes with a presentation on the direction of future research. Further validation of these methods is required with a very stiff experimental setup and an in-depth error analysis. New parameters for switching need to be explored. Continual development of mixed control methods is suggested. Simulink/Stateflow models using high order polynomials should be tested. Explicit and predictor-corrector time integration schemes should be implemented for the equilibrium force control method.

More than that, I count all things to be loss in view of the surpassing value of knowing  
Christ Jesus my Lord.  
— Apostle Paul (Romans 3:8(a) NASB)

# Contents

<b>Contents</b>	<b>ii</b>
<b>List of Tables</b>	<b>vi</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Algorithms</b>	<b>xviii</b>
<b>List of Symbols and Acronyms</b>	<b>xix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Components of hybrid simulation . . . . .	2
1.1.1 Computational driver . . . . .	2
1.1.2 Physical transfer system . . . . .	2
1.1.3 Middleware . . . . .	3
1.2 Motivation . . . . .	3
1.3 Organization and scope . . . . .	4
<b>2 Displacement control hybrid simulation</b>	<b>5</b>
2.1 Introduction . . . . .	5
2.2 Computational driver . . . . .	5
2.3 Physical transfer system . . . . .	6
2.4 OpenFresco middleware hybrid simulation framework . . . . .	8
2.4.1 Two-Node Link experimental element . . . . .	9
2.4.2 Local experimental site . . . . .	10
2.4.3 One Actuator experimental setup . . . . .	10
2.4.4 xPC-Target experimental control . . . . .	11
2.5 Sequential Processes of DC hybrid simulation . . . . .	12
2.6 Advantages and limitations of DC hybrid simulation . . . . .	18
<b>3 Force control hybrid simulation</b>	<b>19</b>
3.1 Introduction . . . . .	19
3.1.1 Problem definition . . . . .	19
3.1.2 Motivation . . . . .	20
3.1.3 Previous research . . . . .	23

3.2	Methods for acquiring the trial force . . . . .	25
3.2.1	Compatibility methods . . . . .	26
3.2.1.1	Tangent-based compatibility methods . . . . .	26
3.2.1.2	Krylov subspace-based compatibility method . . . . .	30
3.2.2	Equilibrium method . . . . .	31
3.2.3	Summary of methods for acquiring the trial forces . . . . .	34
3.3	Implementation . . . . .	34
3.3.1	Implementation of compatibility methods . . . . .	35
3.3.2	Implementation of equilibrium method . . . . .	39
3.3.3	Simulink/Stateflow model . . . . .	41
3.3.4	OpenFresco force experimental control . . . . .	43
3.4	Experimental results and verification . . . . .	44
3.4.1	1-DOF $\mu$ -NEES experimental setup and results . . . . .	44
3.4.1.1	Hybrid model and OpenFresco configuration . . . . .	47
3.4.1.2	Linear FC hybrid simulation . . . . .	47
3.4.1.3	Nonlinear FC hybrid simulation . . . . .	51
3.4.2	2-DOF $\mu$ -NEES experimental setup and results . . . . .	56
3.4.2.1	Hybrid model and OpenFresco configuration . . . . .	59
3.4.2.2	Linear FC hybrid simulation . . . . .	60
3.4.2.3	Nonlinear FC hybrid simulation . . . . .	66
3.4.3	Summary of experimental results and verification . . . . .	74
<b>4</b>	<b>Switch control hybrid simulation</b>	<b>75</b>
4.1	Introduction . . . . .	75
4.1.1	Problem definition . . . . .	75
4.1.2	Motivation . . . . .	76
4.1.3	Previous research . . . . .	77
4.2	Methods for SC hybrid simulation . . . . .	77
4.2.1	Trial force based switching strategy . . . . .	78
4.2.2	Secant stiffness based switching strategy . . . . .	80
4.3	Implementation . . . . .	81
4.3.1	Compatibility methods for switching . . . . .	81
4.3.2	Equilibrium methods for switching . . . . .	82
4.3.3	Simulink/Stateflow model . . . . .	83
4.4	Experimental results and verification . . . . .	88
<b>5</b>	<b>Mixed control hybrid simulation</b>	<b>94</b>
5.1	Introduction . . . . .	94
5.1.1	Problem definition . . . . .	94
5.1.2	Motivation . . . . .	95
5.1.3	Previous research . . . . .	95
5.2	Methods for MC hybrid simulation . . . . .	95
5.3	Implementation . . . . .	96
5.3.1	OpenFresco MC experimental control . . . . .	96



5.3.2	Simulink/Stateflow model . . . . .	97
5.4	Experimental results and verification . . . . .	102
5.4.1	MC hybrid simulation trial force based switching . . . . .	102
5.4.2	MC hybrid simulation with secant stiffness based switching . . . . .	106
5.4.3	Summary of experimental results . . . . .	113
<b>6</b>	<b>Conclusion</b>	<b>114</b>
6.1	Force Control . . . . .	114
6.2	Switch Control . . . . .	115
6.3	Mixed Control . . . . .	115
6.4	Direction for future work . . . . .	116
6.4.1	Further verification . . . . .	116
6.4.2	Using power for switching . . . . .	116
6.4.3	Further investigation of MC hybrid simulation . . . . .	117
6.4.4	Simulink/Stateflow models . . . . .	118
6.4.5	EFC method with other time integration schemes . . . . .	119
6.4.6	Summary . . . . .	119
	<b>Bibliography</b>	<b>120</b>
<b>A</b>	<b>OpenFresco for FC, SC and MC hybrid simulation</b>	<b>123</b>
A.1	OpenFresco software architecture . . . . .	123
A.1.1	OpenFresco UML classes diagram . . . . .	124
A.2	OpenFresco classes . . . . .	127
A.2.0.1	Experimental Element . . . . .	127
A.2.0.2	Experimental Sites . . . . .	127
A.2.0.3	Experimental Setups . . . . .	127
A.2.0.4	Experimental Controls . . . . .	128
A.3	OpenFresco code for force control hybrid simulation . . . . .	128
<b>B</b>	<b>Tuning of <math>\mu</math>-NEES experimental setup</b>	<b>132</b>
B.1	Tuning of MTS-STs Channel 3: Bottom actuator . . . . .	132
B.2	Tuning of MTS-STs Channel 4: Top actuator . . . . .	136
<b>C</b>	<b>FC hybrid simulation experimental results</b>	<b>138</b>
C.1	1-DOF setup: Linear hybrid simulation . . . . .	138
C.2	1-DOF setup: Nonlinear hybrid simulation . . . . .	147
C.3	2-DOF setup: Linear hybrid simulation . . . . .	152
C.4	2-DOF setup: Nonlinear hybrid simulation . . . . .	170
<b>D</b>	<b>SC hybrid simulation experimental results</b>	<b>189</b>

<b>E</b>	<b>MC hybrid simulation experimental results</b>	<b>206</b>
E.1	Point switching hybrid simulation . . . . .	206
E.1.1	NME time integration scheme . . . . .	206
E.1.2	$\alpha$ time integration scheme . . . . .	215
E.1.3	NMF time integration scheme . . . . .	224
E.1.4	NMR time integration scheme . . . . .	233
E.2	Secant switching hybrid simulation . . . . .	240
E.2.1	NME time integration scheme . . . . .	240
E.2.2	$\alpha$ time integration scheme . . . . .	249
E.2.3	NMF time integration scheme . . . . .	258
E.2.4	NMR time integration scheme . . . . .	267

# List of Tables

2.1	OpenFresco configuration for the one-bay-frame example . . . . .	9
3.1	Properties of soft and stiff one-bay-frame models . . . . .	20
3.2	Time integration schemes and their parameters . . . . .	47
3.3	Properties of one-bay-frame model for 1-DOF $\mu$ -NEES setup . . . . .	47
3.4	Absolute errors of linear numerical simulation results of one-bay-frame OpenSees model . . . . .	50
3.5	Experimental absolute displacement error at FEA level for 1-DOF linear hybrid simulation. . . . .	50
3.6	Experimental absolute force error at FEA level for 1-DOF linear hybrid simulation. . . . .	51
3.7	Absolute errors at control system level for 1-DOF linear simulation . . . . .	51
3.8	Absolute errors of nonlinear numerical simulation results of one-bay-frame OpenSees model . . . . .	52
3.9	Absolute displacement error at FEA level for 1-DOF nonlinear hybrid simulation. . . . .	55
3.10	Absolute force error at FEA level for 1-DOF nonlinear hybrid simulation. . . . .	55
3.11	Absolute error at control system level for 1-DOF nonlinear simulation. . . . .	56
3.12	Properties of 2-DOF hybrid model for 2-DOF $\mu$ -NEES setup . . . . .	59
3.13	OpenFresco configuration for 2-DOF setup . . . . .	59
3.14	Absolute errors of linear numerical simulation results of 2-DOF OpenSees model . . . . .	63
3.15	Node 1 absolute displacement error at FEA level for 2-DOF linear hybrid simulation. . . . .	63
3.16	Node 2 absolute displacement errors at FEA level for 2-DOF linear hybrid simulation. . . . .	64
3.17	Node 1 absolute force errors at FEA level for 2-DOF linear hybrid simulation. . . . .	64
3.18	Node 2 absolute force error at FEA level for 2-DOF linear hybrid simulation. . . . .	64
3.19	Bottom actuator absolute error at control system level for 2-DOF linear simulation . . . . .	65
3.20	Top actuator absolute error at control system level for 2-DOF linear simulation . . . . .	65
3.21	Absolute errors of nonlinear numerical simulation results of 2-DOF OpenSees model . . . . .	69
3.22	Node 1 absolute displacement error at FEA level for 2-DOF nonlinear hybrid simulation. . . . .	69
3.23	Node 2 absolute displacement errors at FEA level for 2-DOF nonlinear hybrid simulation. . . . .	70
3.24	Node 1 absolute force errors at FEA level for 2-DOF nonlinear hybrid simulation. . . . .	70
3.25	Node 2 absolute force errors at FEA level for 2-DOF nonlinear hybrid simulation. . . . .	70
3.26	Bottom actuator absolute error at control system level for 2-DOF nonlinear hybrid simulation . . . . .	71

3.27	Bottom actuator absolute error at control system level for 2-DOF nonlinear hybrid simulation . . . . .	71
4.1	Absolute displacement error at FEA level for 1-DOF nonlinear SC hybrid simulation.	89
4.2	Absolute force error at FEA level for 1-DOF nonlinear SC hybrid simulation. . . .	90
4.3	Absolute error at control system level for 1-DOF nonlinear SC hybrid simulation. .	93
5.1	Node 1 absolute displacement error at FEA level for 2-DOF PMC nonlinear hybrid simulation. . . . .	103
5.2	Node 2 absolute displacement errors at FEA level for 2-DOF nonlinear PMC hybrid simulation. . . . .	103
5.3	Node 1 absolute force error at FEA level for 2-DOF nonlinear PMC hybrid simulation. . . . .	103
5.4	Node 2 absolute force errors at FEA level for 2-DOF nonlinear PMC hybrid simulation. . . . .	104
5.5	Bottom actuator absolute error at control system level for 2-DOF nonlinear PMC hybrid simulation . . . . .	104
5.6	Top actuator absolute error at control system level for 2-DOF nonlinear PMC hybrid simulation . . . . .	105
5.7	Node 1 absolute displacement error at FEA level for 2-DOF SMC nonlinear hybrid simulation. . . . .	106
5.8	Node 2 absolute displacement errors at FEA level for 2-DOF nonlinear SMC hybrid simulation. . . . .	107
5.9	Node 1 absolute force error at FEA level for 2-DOF nonlinear SMC hybrid simulation. . . . .	107
5.10	Node 2 absolute force error at FEA level for 2-DOF nonlinear SMC hybrid simulation. . . . .	107
5.11	Bottom actuator absolute error at control system level for 2-DOF nonlinear SMC hybrid simulation . . . . .	108
5.12	Top actuator absolute error at control system level for 2-DOF nonlinear SMC hybrid simulation . . . . .	108
B.1	Absolute error at control system level for 1-DOF nonlinear simulation. . . . .	135
B.2	Absolute error at control system level for 1-DOF nonlinear simulation. . . . .	136

# List of Figures

2.1	One-bay-frame model for DC hybrid simulation . . . . .	6
2.2	Control loop in DC hybrid simulation . . . . .	7
2.3	$\mu$ -NEES experimental setup at <i>nees@berkeley</i> . . . . .	8
2.4	OpenFresco abstract classes . . . . .	9
2.5	OpenFresco Two-Node Link experimental element . . . . .	10
2.6	OpenFresco One Actuator experimental setup . . . . .	11
2.7	OpenFresco One Actuator experimental setup transformation . . . . .	11
2.8	OpenFresco UML sequence diagram for setting the trial response . . . . .	13
2.9	OpenFresco UML sequence diagram for obtaining the resisting force . . . . .	14
2.10	Force-deformation curve showing the $\alpha$ OS approximation . . . . .	17
3.1	The El Centro ground motion. . . . .	21
3.2	Element 1 trial displacement-time history plots for Model 1 and 2. . . . .	21
3.3	Element 1 measured force-time history plots for Model 1 and 2. . . . .	21
3.4	Implementation of the compatibility methods. . . . .	35
3.5	Interface for the <i>ExperimentalSignalFilter</i> class. . . . .	36
3.6	Extended OpenFresco UML class diagram. . . . .	37
3.7	Interface for the <i>ExperimentalTangentStiff</i> class. . . . .	38
3.8	SimuLink model for FC hybrid simulation. . . . .	41
3.9	xPC Target Hybrid Controller (F1F1). . . . .	42
3.10	StateFlow predictor-corrector model for FC hybrid simulation. . . . .	43
3.11	1-DOF $\mu$ -NEES experimental setup schematic. . . . .	45
3.12	1-DOF $\mu$ -NEES experimental setup. . . . .	46
3.13	1-DOF $\mu$ -NEES experimental setup clevis and coupon configuration. . . . .	46
3.14	Linear numerical simulation results of displacement-time history using NMI, NME, $\alpha$ OS, NMF and NMR for one-bay-frame OpenSees model. . . . .	48
3.15	Linear numerical simulation results of force-time history using NMI, NME, $\alpha$ OS, NMF and NMR for one-bay-frame OpenSees model. . . . .	49
3.16	OpenSees model of the 1-DOF $\mu$ -NEES setup for numerical simulation. . . . .	52
3.17	Nonlinear numerical simulation results of displacement-time history using NMI, NMF and NMR for one-bay-frame OpenSees model . . . . .	53
3.18	Nonlinear numerical simulation results of force-time history using NMI, NMF and NMR for one-bay-frame OpenSees model. . . . .	54

3.19	Tangent stiffness estimation of the 1-DOF Setup during nonlinear hybrid simulation using NMF . . . . .	56
3.20	2-DOF $\mu$ -NEES experimental setup schematic. . . . .	57
3.21	2-DOF $\mu$ -NEES experimental setup and 2-DOF hybrid model. . . . .	58
3.22	2-DOF $\mu$ -NEES experimental setup top clevis and coupon configuration. . . . .	58
3.23	2-DOF OpenSees numerical model for numerical simulation . . . . .	60
3.24	Linear numerical simulation results of displacement-time history using NMI, NME, $\alpha$ OS, NMF and NMI for 2-DOF OpenSees model . . . . .	61
3.25	Linear numerical simulation results of force-time history using NMI, NME, $\alpha$ OS, NMF and NMI for 2-DOF OpenSees model . . . . .	62
3.26	Nonlinear numerical simulation results of displacement-time history using NMI, NME, $\alpha$ OS, NMF and NMI for the 2-DOF OpenSees model. . . . .	67
3.27	Nonlinear numerical simulation results of force-time history using NMI, NME, $\alpha$ OS, NMF and NMI for the 2-DOF OpenSees model. . . . .	68
3.28	Tangent stiffness values for nonlinear 2-DOF hybrid simulation using CFC:BFGS. . . . .	72
3.29	Tangent stiffness values for nonlinear 2-DOF hybrid simulation using CFC:Intrinsic . . . . .	73
4.1	Hysteresis curve of High Damping Rubber Bearing. . . . .	76
4.2	SC hybrid simulation components and their processes. . . . .	81
4.3	SimuLink model for SC. . . . .	84
4.4	xPC Target Hybrid Controller (S1S1). . . . .	85
4.5	StateFlow predictor-corrector model for SC hybrid simulation. . . . .	87
4.6	Plot of control mode and trial forces vs. time for PSC with NME. . . . .	90
4.7	Plot of control mode and secant stiffness value vs. time for SSC with NME. . . . .	91
4.8	Plot of control mode and trial forces vs. time for EPSC with NMR. . . . .	91
4.9	Plot of control mode and secant stiffness value vs. time for ESSC with NMR. . . . .	91
5.1	MC hybrid simulation components and their processes. . . . .	97
5.2	SimuLink model for MC hybrid simulation. . . . .	98
5.3	xPC Target Hybrid Controller (M1M1). . . . .	99
5.4	StateFlow predictor-corrector model for MC hybrid simulation. . . . .	101
5.5	Plot of control mode and trial forces vs. time of the bottom actuator for PMC:Broyden with NME. . . . .	105
5.6	Plot of control mode and trial forces vs. time of the bottom actuator for PMC:BFGS with $\alpha$ OS. . . . .	106
5.7	Plot of control mode and trial forces vs. time of the bottom actuator for PMC:Broyden with NME. . . . .	110
5.8	Plot of control mode and trial forces vs. time of the bottom actuator for PMC:BFGS with $\alpha$ OS. . . . .	111
5.9	Plot of control mode and trial forces vs. time of the bottom actuator for PMC:BFGS with $\alpha$ OS. . . . .	112
6.1	Plot of power and trial forces vs. time for NME. . . . .	117
6.2	Proposed method for MC hybrid simulation . . . . .	118

A.1	OpenFresco Multi-Tier Architecture . . . . .	124
A.2	OpenFresco class diagram for DC hybrid simulation . . . . .	125
A.3	OpenFresco class diagram for FC, SC and MC hybrid simulation . . . . .	126
A.4	ESFTangForceConverter class definition. . . . .	129
A.5	ESFKylovForceConverter class definition. . . . .	130
A.6	ECxPCtargetForce class definition. . . . .	131
B.1	Displacement control tuning of 1-DOF $\mu$ -NEES experimental setup. . . . .	132
B.2	Absolute value of the displacement error of 1-DOF $\mu$ -NEES experimental setup. . . . .	133
B.3	Tracking and hysteresis plots of 1-DOF $\mu$ -NEES experimental setup in displacement control. . . . .	133
B.4	Force control tuning of 1-DOF $\mu$ -NEES experimental setup. . . . .	134
B.5	Absolute value of the force error of 1-DOF $\mu$ -NEES experimental setup. . . . .	134
B.6	Tracking and hysteresis plots of 1-DOF $\mu$ -NEES experimental setup in force control. . . . .	135
B.7	Displacement control tuning of 2-DOF $\mu$ -NEES experimental setup. . . . .	136
B.8	Absolute value of the displacement error of 1-DOF $\mu$ -NEES experimental setup. . . . .	137
B.9	Tracking and hysteresis plots of 1-DOF $\mu$ -NEES experimental setup in displacement control. . . . .	137
C.1	Displacement- and force-time history plots for linear FC hybrid simulation using NME with 1-DOF setup. . . . .	139
C.2	Command and measured displacement plots from STS controller for linear DC hybrid simulation using NME with 1-DOF setup. . . . .	140
C.3	Command and measured force plots from STS controller for linear CFC hybrid simulation using NME with 1-DOF setup. . . . .	140
C.4	Displacement- and force-time history plots for linear FC hybrid simulation using $\alpha$ OS with 1-DOF setup. . . . .	141
C.5	Command and measured displacement plots from STS controller for linear DC hybrid simulation using $\alpha$ OS with 1-DOF setup. . . . .	142
C.6	Command and measured force plots from STS controller for linear CFC hybrid simulation using $\alpha$ OS with 1-DOF setup. . . . .	142
C.7	Displacement- and force-time history plots for linear FC hybrid simulation using NMF with 1-DOF setup. . . . .	143
C.8	Command and measured displacement plots from STS controller for linear DC hybrid simulation using NMF with 1-DOF setup. . . . .	144
C.9	Command and measured force plots from STS controller for linear CFC hybrid simulation using NMF with 1-DOF setup. . . . .	144
C.10	Displacement- and force-time history plots for linear FC hybrid simulation using NMR with 1-DOF setup. . . . .	145
C.11	Command and measured displacement plots from STS controller for linear DC hybrid simulation using NMR with 1-DOF setup. . . . .	146
C.12	Command and measured force plots from STS controller for linear CFC hybrid simulation using NMR with 1-DOF setup. . . . .	146

C.13	Command and measured force plots from STS controller for linear EFC hybrid simulation using NMR with 1-DOF setup. . . . .	147
C.14	Displacement- and force-time history plots for nonlinear FC hybrid simulation using NMF with 1-DOF setup. . . . .	148
C.15	Command and measured displacement plots from STS controller for nonlinear DC hybrid simulation using NMF with 1-DOF setup. . . . .	149
C.16	Command and measured force plots from STS controller for nonlinear CFC hybrid simulation using NMF with 1-DOF setup. . . . .	149
C.17	Displacement- and force-time history plots for nonlinear FC hybrid simulation using NMR with 1-DOF setup. . . . .	150
C.18	Command and measured displacement plots from STS controller for nonlinear DC hybrid simulation using NMR with 1-DOF setup. . . . .	151
C.19	Command and measured force plots from STS controller for nonlinear EFC hybrid simulation using NMR with 1-DOF setup. . . . .	151
C.20	Nodal displacement-time history plots for linear FC hybrid simulation using NME with 2-DOF setup. . . . .	153
C.21	Nodal force-time history plots for linear FC hybrid simulation using NME with 2-DOF setup. . . . .	154
C.22	Command and measured displacement plots from STS controller for linear DC simulation using NME with 2-DOF setup. . . . .	155
C.23	Command and measured force plots from STS controller for linear FC simulation using NME with 2-DOF setup. . . . .	156
C.24	Displacement-time history plots for linear FC hybrid simulation using $\alpha$ OS with 2-DOF setup. . . . .	157
C.25	Force-time history plots for linear FC hybrid simulation using $\alpha$ OS with 2-DOF setup. . . . .	158
C.26	Command and measured displacement plots from STS controller for linear DC simulation using $\alpha$ OS with 2-DOF setup. . . . .	159
C.27	Command and measured force plots from STS controller for linear FC simulation using $\alpha$ OS with 2-DOF setup. . . . .	160
C.28	Nodal displacement-time history plots for linear FC hybrid simulation using NMF with 2-DOF setup. . . . .	161
C.29	Nodal force-time history plots for linear FC hybrid simulation using NMF with 2-DOF setup. . . . .	162
C.30	Command and measured displacement plots from STS controller for linear DC simulation using NMF with 2-DOF setup. . . . .	163
C.31	Command and measured force plots from STS controller for linear FC simulation using NMF with 2-DOF setup. . . . .	164
C.32	Nodal displacement-time history plots for linear FC hybrid simulation using NMR with 2-DOF setup. . . . .	165
C.33	Nodal force-time history plots for linear FC hybrid simulation using NMR with 2-DOF setup. . . . .	166
C.34	Command and measured displacement plots from STS controller for linear DC simulation using NMR with 2-DOF setup. . . . .	167



C.35	Command and measured force plots from STS controller for linear FC simulation using NMR-CFC with 2-DOF setup. . . . .	168
C.36	Command and measured force plots from STS controller for linear FC simulation using NMR-EFC with 2-DOF setup. . . . .	169
C.37	Nodal displacement-time history plots for nonlinear FC hybrid simulation using NME with 2-DOF setup. . . . .	171
C.38	Nodal force-time history plots for nonlinear FC hybrid simulation using NME with 2-DOF setup. . . . .	172
C.39	Command and measured displacement plots from STS controller for nonlinear DC simulation using NME with 2-DOF setup. . . . .	173
C.40	Command and measured force plots from STS controller for nonlinear FC simulation using NME with 2-DOF setup. . . . .	174
C.41	Displacement-time history plots for nonlinear FC hybrid simulation using $\alpha$ OS with 2-DOF setup. . . . .	175
C.42	Force-time history plots for nonlinear FC hybrid simulation using $\alpha$ OS with 2-DOF setup. . . . .	176
C.43	Command and measured displacement plots from STS controller for nonlinear DC simulation using $\alpha$ OS with 2-DOF setup.. . . .	177
C.44	Command and measured force plots from STS controller for nonlinear FC simulation using $\alpha$ OS-CFC:BFGS with 2-DOF setup . . . . .	178
C.45	Command and measured force plots from STS controller for nonlinear FC simulation using $\alpha$ OS=CFC:Intrinsic with 2-DOF setup . . . . .	179
C.46	Nodal displacement-time history plots for nonlinear FC hybrid simulation using NMF with 2-DOF setup. . . . .	180
C.47	Nodal force-time history plots for nonlinear FC hybrid simulation using NMF with 2-DOF setup. . . . .	181
C.48	Command and measured displacement plots from STS controller for nonlinear DC simulation using NMF with 2-DOF setup. . . . .	182
C.49	Command and measured force plots from STS controller for nonlinear FC simulation using NMF with 2-DOF setup. . . . .	183
C.50	Nodal displacement-time history plots for nonlinear FC hybrid simulation using NMR with 2-DOF setup. . . . .	184
C.51	Nodal force-time history plots for nonlinear FC hybrid simulation using NMR with 2-DOF setup. . . . .	185
C.52	Command and measured displacement plots from STS controller for nonlinear DC simulation using NMR with 2-DOF setup. . . . .	186
C.53	Command and measured force plots from STS controller for nonlinear FC simulation using NMR-CFC:BFGS with 2-DOF setup. . . . .	187
C.54	Command and measured force plots from STS controller for nonlinear FC simulation using NMR-CFC:Intrinsic with 2-DOF setup. . . . .	188
D.1	Displacement- and force-time history plots for nonlinear SC hybrid simulation using NME with 1-DOF setup. . . . .	190

D.2	Command and measured displacement plots from STS controller for nonlinear DC hybrid simulation using NME with 1-DOF setup. . . . .	191
D.3	Command and measured displacement plots from STS controller for nonlinear PSC hybrid simulation using NME with 1-DOF setup. . . . .	192
D.4	Command and measured displacement plots from STS controller for nonlinear SSC hybrid simulation using NME with 1-DOF setup. . . . .	193
D.5	Displacement- and force-time history plots for nonlinear SC hybrid simulation using $\alpha$ OS with 1-DOF setup. . . . .	194
D.6	Command and measured displacement plots from STS controller for nonlinear DC hybrid simulation using $\alpha$ OS with 1-DOF setup. . . . .	195
D.7	Command and measured displacement plots from STS controller for nonlinear PSC hybrid simulation using $\alpha$ OS with 1-DOF setup. . . . .	196
D.8	Command and measured displacement plots from STS controller for nonlinear SSC hybrid simulation using $\alpha$ OS with 1-DOF setup. . . . .	197
D.9	Displacement- and force-time history plots for nonlinear SC hybrid simulation using NMF with 1-DOF setup. . . . .	198
D.10	Command and measured displacement plots from STS controller for nonlinear PSC hybrid simulation using NMF with 1-DOF setup. . . . .	199
D.11	Command and measured displacement plots from STS controller for nonlinear SSC hybrid simulation using NMF with 1-DOF setup. . . . .	200
D.12	Displacement- and force-time history plots for nonlinear SC hybrid simulation using NMR with 1-DOF setup. . . . .	201
D.13	Command and measured displacement plots from STS controller for nonlinear PSC hybrid simulation using NMR with 1-DOF setup. . . . .	202
D.14	Command and measured displacement plots from STS controller for nonlinear SSC hybrid simulation using NMR with 1-DOF setup. . . . .	203
D.15	Command and measured displacement plots from STS controller for nonlinear EPSC hybrid simulation using NMR with 1-DOF setup. . . . .	204
D.16	Command and measured displacement plots from STS controller for nonlinear ESSC hybrid simulation using NMR with 1-DOF setup. . . . .	205
E.1	Nodal displacement-time history plots for nonlinear PMC hybrid simulation using NME with 2-DOF setup. . . . .	207
E.2	Nodal force-time history plots for nonlinear PMC hybrid simulation using NME with 2-DOF setup. . . . .	208
E.3	Bottom actuator command and measured displacement and force plots from STS controller for nonlinear PMC:Broyden hybrid simulation using NME with 2-DOF setup. . . . .	209
E.4	Top actuator command and measured displacement and force plots from STS controller for nonlinear PMC:Broyden hybrid simulation using NME with 2-DOF setup.	210
E.5	Bottom actuator command and measured displacement and force plots from STS controller for nonlinear PMC:BFGS hybrid simulation using NME with 2-DOF setup. . . . .	211

E.6	Top actuator command and measured displacement and force plots from STS controller for nonlinear PMC:BFGS hybrid simulation using NME with 2-DOF setup.	212
E.7	Bottom actuator command and measured displacement and force plots from STS controller for nonlinear PMC:Intrinsic hybrid simulation using NME with 2-DOF setup.	213
E.8	Top actuator command and measured displacement and force plots from STS controller for nonlinear PMC:Intrinsic hybrid simulation using NME with 2-DOF setup.	214
E.9	Nodal displacement-time history plots for nonlinear PMC hybrid simulation using $\alpha$ OS with 2-DOF setup.	216
E.10	Nodal force-time history plots for nonlinear PMC hybrid simulation using $\alpha$ OS with 2-DOF setup.	217
E.11	Bottom actuator command and measured displacement and force plots from STS controller for nonlinear PMC:Broyden hybrid simulation using $\alpha$ OS with 2-DOF setup.	218
E.12	Top actuator command and measured displacement and force plots from STS controller for nonlinear PMC:Broyden hybrid simulation using $\alpha$ OS with 2-DOF setup.	219
E.13	Bottom actuator command and measured displacement and force plots from STS controller for nonlinear PMC:BFGS hybrid simulation using $\alpha$ OS with 2-DOF setup.	220
E.14	Top actuator command and measured displacement and force plots from STS controller for nonlinear PMC:BFGS hybrid simulation using $\alpha$ OS with 2-DOF setup.	221
E.15	Bottom actuator command and measured displacement and force plots from STS controller for nonlinear PMC:Intrinsic hybrid simulation using $\alpha$ OS with 2-DOF setup.	222
E.16	Top actuator command and measured displacement and force plots from STS controller for nonlinear PMC:Intrinsic hybrid simulation using $\alpha$ OS with 2-DOF setup.	223
E.17	Nodal displacement-time history plots for nonlinear PMC hybrid simulation using NMF with 2-DOF setup.	225
E.18	Nodal force-time history plots for nonlinear PMC hybrid simulation using NMF with 2-DOF setup.	226
E.19	Bottom actuator command and measured displacement and force plots from STS controller for nonlinear PMC:Broyden hybrid simulation using NMF with 2-DOF setup.	227
E.20	Top actuator command and measured displacement and force plots from STS controller for nonlinear PMC:Broyden hybrid simulation using NMF with 2-DOF setup.	228
E.21	Bottom actuator command and measured displacement and force plots from STS controller for nonlinear PMC:BFGS hybrid simulation using NMF with 2-DOF setup.	229
E.22	Top actuator command and measured displacement and force plots from STS controller for nonlinear PMC:BFGS hybrid simulation using NMF with 2-DOF setup.	230
E.23	Bottom actuator command and measured displacement and force plots from STS controller for nonlinear PMC:Intrinsic hybrid simulation using NMF with 2-DOF setup.	231
E.24	Top actuator command and measured displacement and force plots from STS controller for nonlinear PMC:Intrinsic hybrid simulation using NMF with 2-DOF setup.	232

E.25	Nodal displacement-time history plots for nonlinear PMC hybrid simulation using NMR with 2-DOF setup. . . . .	234
E.26	Nodal force-time history plots for nonlinear PMC hybrid simulation using NMR with 2-DOF setup. . . . .	235
E.27	Bottom actuator command and measured displacement and force plots from STS controller for nonlinear PMC:BFGS hybrid simulation using NMR with 2-DOF setup. . . . .	236
E.28	Top actuator command and measured displacement and force plots from STS controller for nonlinear PMC:BFGS hybrid simulation using NMR with 2-DOF setup.	237
E.29	Bottom actuator command and measured displacement and force plots from STS controller for nonlinear PMC:Intrinsic hybrid simulation using NMR with 2-DOF setup. . . . .	238
E.30	Top actuator command and measured displacement and force plots from STS controller for nonlinear PMC:Intrinsic hybrid simulation using NMR with 2-DOF setup.	239
E.31	Nodal displacement-time history plots for nonlinear SMC hybrid simulation using NME with 2-DOF setup. . . . .	241
E.32	Nodal force-time history plots for nonlinear SMC hybrid simulation using NME with 2-DOF setup. . . . .	242
E.33	Bottom actuator command and measured displacement and force plots from STS controller for nonlinear SMC:Broyden hybrid simulation using NME with 2-DOF setup. . . . .	243
E.34	Top actuator command and measured displacement and force plots from STS controller for nonlinear SMC:Broyden hybrid simulation using NME with 2-DOF setup.	244
E.35	Bottom actuator command and measured displacement and force plots from STS controller for nonlinear SMC:BFGS hybrid simulation using NME with 2-DOF setup. . . . .	245
E.36	Top actuator command and measured displacement and force plots from STS controller for nonlinear SMC:BFGS hybrid simulation using NME with 2-DOF setup.	246
E.37	Bottom actuator command and measured displacement and force plots from STS controller for nonlinear SMC:Intrinsic hybrid simulation using NME with 2-DOF setup. . . . .	247
E.38	Top actuator command and measured displacement and force plots from STS controller for nonlinear SMC:Intrinsic hybrid simulation using NME with 2-DOF setup.	248
E.39	Nodal displacement-time history plots for nonlinear SMC hybrid simulation using $\alpha$ OS with 2-DOF setup. . . . .	250
E.40	Nodal force-time history plots for nonlinear SMC hybrid simulation using $\alpha$ OS with 2-DOF setup. . . . .	251
E.41	Bottom actuator command and measured displacement and force plots from STS controller for nonlinear SMC:Broyden hybrid simulation using $\alpha$ OS with 2-DOF setup. . . . .	252
E.42	Top actuator command and measured displacement and force plots from STS controller for nonlinear SMC:Broyden hybrid simulation using $\alpha$ OS with 2-DOF setup.	253
E.43	Bottom actuator command and measured displacement and force plots from STS controller for nonlinear SMC:BFGS hybrid simulation using $\alpha$ OS with 2-DOF setup.	254

E.44	Top actuator command and measured displacement and force plots from STS controller for nonlinear SMC:BFGS hybrid simulation using $\alpha$ OS with 2-DOF setup.	255
E.45	Bottom actuator command and measured displacement and force plots from STS controller for nonlinear SMC:Intrinsic hybrid simulation using $\alpha$ OS with 2-DOF setup.	256
E.46	Top actuator command and measured displacement and force plots from STS controller for nonlinear SMC:Intrinsic hybrid simulation using $\alpha$ OS with 2-DOF setup.	257
E.47	Nodal displacement-time history plots for nonlinear SMC hybrid simulation using NMF with 2-DOF setup.	259
E.48	Nodal force-time history plots for nonlinear SMC hybrid simulation using NMF with 2-DOF setup.	260
E.49	Bottom actuator command and measured displacement and force plots from STS controller for nonlinear SMC:Broyden hybrid simulation using NMF with 2-DOF setup.	261
E.50	Top actuator command and measured displacement and force plots from STS controller for nonlinear SMC:Broyden hybrid simulation using NMF with 2-DOF setup.	262
E.51	Bottom actuator command and measured displacement and force plots from STS controller for nonlinear SMC:BFGS hybrid simulation using NMF with 2-DOF setup.	263
E.52	Top actuator command and measured displacement and force plots from STS controller for nonlinear SMC:BFGS hybrid simulation using NMF with 2-DOF setup.	264
E.53	Bottom actuator command and measured displacement and force plots from STS controller for nonlinear SMC:Intrinsic hybrid simulation using NMF with 2-DOF setup.	265
E.54	Top actuator command and measured displacement and force plots from STS controller for nonlinear SMC:Intrinsic hybrid simulation using NMF with 2-DOF setup.	266
E.55	Nodal displacement-time history plots for nonlinear SMC hybrid simulation using NMR with 2-DOF setup.	268
E.56	Nodal force-time history plots for nonlinear SMC hybrid simulation using NMR with 2-DOF setup.	269
E.57	Bottom actuator command and measured displacement and force plots from STS controller for nonlinear SMC:Broyden hybrid simulation using NMR with 2-DOF setup.	270
E.58	Top actuator command and measured displacement and force plots from STS controller for nonlinear SMC:Broyden hybrid simulation using NMR with 2-DOF setup.	271
E.59	Bottom actuator command and measured displacement and force plots from STS controller for nonlinear SMC:BFGS hybrid simulation using NMR with 2-DOF setup.	272
E.60	Top actuator command and measured displacement and force plots from STS controller for nonlinear SMC:BFGS hybrid simulation using NMR with 2-DOF setup.	273
E.61	Bottom actuator command and measured displacement and force plots from STS controller for nonlinear SMC:Intrinsic hybrid simulation using NMR with 2-DOF setup.	274

E.62 Top actuator command and measured displacement and force plots from STS controller for nonlinear SMC: Intrinsic hybrid simulation using NMR with 2-DOF setup. 275

# List of Algorithms

3.1	Implicit FC Algorithm . . . . .	25
3.2	BFGS Method . . . . .	28
4.1	Mode Switching Algorithm at Time Step $n$ by Elkhoraibi and Mosalam [7] . . . . .	77
4.2	Force limit switching strategy for softening specimen at time step $n$ . . . . .	79
4.3	Force limit switching strategy for softening specimen at time step $n$ . . . . .	79
4.4	Secant limit switching strategy at time step $n$ . . . . .	80

# List of Symbols and Acronyms

The following symbols and acronyms are used in this dissertation:

FEA	=	finite element analysis
DOF	=	degree-of-freedom
DOFs	=	degrees-of-freedom
DAQ	=	data acquisition system
PID controller	=	proportionalintegralderivative controller
LVDT	=	linear variable differential transformers
LC	=	load cell
NEES	=	George E. Brown, Jr. Network for Earthquake Engineering Simulation
nees@berkeley	=	NEES Structural Engineering lab at University of California, Berkeley
MTS-STS	=	MTS-Structural Testing Software
OpenFresco	=	Open-source Framework for Experimental Setup and Control
OpenSEES	=	Open System for Earthquake Engineering Simulation
API	=	application program interface
SCRAMNet	=	Shared Common Random Access Memory Network
UML	=	United Modeling Language
<b>M</b>	=	mass matrix
<b>C</b>	=	viscous damping matrix
<b>K</b>	=	stiffness matrix
<b>K<sub>t</sub></b>	=	tangent stiffness matrix
<b>K<sub>0</sub></b>	=	initial stiffness matrix
<i>k</i>	=	stiffness
<i>k<sub>t</sub></i>	=	tangent stiffness
<i>k<sub>0</sub></i>	=	initial stiffness
HHT	=	Hiber-Hughes-Talyor
<i>u<sub>i</sub></i>	=	nodal displacement at time step <i>i</i>
<b>u<sub>i</sub></b>	=	nodal displacement vector at time step <i>i</i>
<i>u̇<sub>i</sub></i>	=	nodal velocity at time step <i>i</i>
<b>u̇<sub>i</sub></b>	=	nodal velocity vector at time step <i>i</i>
<i>ü<sub>i</sub></i>	=	nodal acceleration at time step <i>i</i>
<b>ü<sub>i</sub></b>	=	nodal acceleration vector at time step <i>i</i>
<i>ü<sub>g</sub></i>	=	recorded ground accelerations
<b>f<sub>i</sub></b>	=	nodal load vector at time step <i>i</i>
<i>r(u<sub>i</sub>)</i> or <i>r</i>	=	nodal resisting force function of <i>u<sub>i</sub></i>



- $\mathbf{r}(\mathbf{u}_i)$  or  $\mathbf{r}$  = nodal resisting force vector function of  $\mathbf{u}_i$   
 $\hat{r}(\hat{u}_i)$  or  $\hat{r}$  = element resisting force function of  $\hat{u}_i$   
 $\hat{\mathbf{r}}(\hat{\mathbf{u}}_i)$  or  $\hat{\mathbf{r}}$  = element resisting force vector function of  $\hat{\mathbf{u}}_i$   
 $\tilde{u}_i$  = nodal predictor displacement at time step  $i$   
 $\tilde{\mathbf{u}}_i$  = nodal predictor displacement vector at time step  $i$   
 $\tilde{\dot{u}}_i$  = nodal predictor velocity at time step  $i$   
 $\tilde{\dot{\mathbf{u}}}_i$  = nodal predictor velocity vector at time step  $i$   
 $\hat{u}_i$  = trial displacement in experimental element coordinate system at time step  $i$   
 $\hat{\mathbf{u}}_i$  = trial displacement vector in experimental element coordinate system at time step  $i$   
 $\hat{f}_i$  = trial force in experimental element coordinate system at time step  $i$   
 $\hat{\mathbf{f}}_i$  = trial force vector in experimental element coordinate system at time step  $i$   
 $\check{u}_i$  = trial displacement in actuator coordinate system at time step  $i$   
 $\check{\mathbf{u}}_i$  = trial displacement vector in actuator coordinate system at time step  $i$   
 $\check{f}_i$  = trial force in actuator coordinate system at time step  $i$   
 $\check{\mathbf{f}}_i$  = trial force vector in actuator coordinate system at time step  $i$   
 $\bar{u}_i$  = measured displacement in actuator coordinate system at time step  $i$   
 $\bar{\mathbf{u}}_i$  = measured displacement vector in actuator coordinate system at time step  $i$   
 $\bar{\dot{u}}_i$  = measured displacement in actuator coordinate system at time step  $i$   
 $\bar{\dot{\mathbf{u}}}_i$  = measured displacement vector in actuator coordinate system at time step  $i$   
 $\bar{\ddot{u}}_i$  = measured displacement in actuator coordinate system at time step  $i$   
 $\bar{\ddot{\mathbf{u}}}_i$  = measured displacement vector in actuator coordinate system at time step  $i$   
 $\bar{f}_i$  = measured resisting force in actuator coordinate system at time step  $i$   
 $\bar{\mathbf{f}}_i$  = measured resisting force vector in actuator coordinate system at time step  $i$   
 $\check{\check{u}}_i$  = measured displacement in experimental element coordinate system at time step  $i$   
 $\check{\check{\mathbf{u}}}_i$  = measured displacement vector in experimental element coordinate system at time step  $i$   
 $\check{\check{f}}_i$  = measured resisting force in experimental element coordinate system at time step  $i$   
 $\check{\check{\mathbf{f}}}_i$  = measured resisting force vector in experimental element coordinate system at time step  $i$   
 $r(\tilde{u}_i)$  = nodal measured resisting force at time step  $i$   
 $\mathbf{r}(\tilde{\mathbf{u}}_i)$  = nodal measured resisting force vector at time step  $i$   
EFT = Effective Force Testing  
ODE = ordinary differential equation  
DC = Displacement control

- CFC = Compatibility force control
- EFC = Equilibrium force control
- NME = Newmark Explicit time integration scheme
- NMR = Newmark Reduced Increment time integration scheme
- NMF = Newmark Fixed Iteration time integration scheme
- $\alpha$ OS =  $\alpha$  Operator Splitting time integration scheme

## Acknowledgments

First and foremost, I am thankful to my Lord Jesus Christ, through whom all this is possible. I am truly a blessed man.

I am grateful for my wife, Tami Kim. She has supported me on this endeavor from start to finish. She has encouraged me to press on during the low times and rejoiced with me during the high times. Without her support and prayer, I could not have completed my studies. My children, Evie and Caleb, have encouraged me as well with their many smiles, hugs and kisses.

I want express my deepest gratitude to Professor Božidar Stojadinović, my advisor. He has given me inspiration and support to conduct my research. He possesses a rare balance of providing direction as well as freedom to his students. I am privileged to have worked with the very best.

Professor Stephen A. Mahin and Dean Gregory L. Fenves advised me early in my graduate studies to pursue this field of hybrid simulation. Both are first-rate academics. I am grateful for their direction and foresight. I want to thank Professors Jon Wilkening, Khalid M. Mosalam and Dr. Horst D. Simon for participating in my qualifying exam committee. They have provided me with invaluable guidance. Among these, a special note of appreciation is extended to Professors Stephen A. Mahin and Jon Wilkening for agreeing to serve on my dissertation committee.

I would not have completed my doctoral studies without the help and companionship of my fellow graduate students and friends. I am greatly indebted to Dr. Andreas Schellenberg for his infectious passion for structural engineering. He has aided me at every turn and extended his kind hand of friendship to me. Professor Tony Yang has inspired me with his academic diligence and enthusiasm. Dr. John-Michael Wong and Dr. Matthew Dryden have been great brothers in Christ. I am thankful for their prayers and advice throughout my study.

A special note of acknowledgment is due to the staff at *nees@berkeley* lab. Without their support, I could not have finished my experiments. Dr. Selim Gunay rearranged his scheduled countless number of times to help me finish my experiments. He has been a good friend to me as well. Last but not least, Dr. Shakhzod Takhirov and Donald Patterson were always there for me when I needed assistance with the lab equipment.

# Chapter 1

## Introduction

Predicting the behavior of structures such as buildings and bridges under dynamic loading especially earthquake loading still remains a challenge in structural engineering. Researchers deploy various methods for studying the behavior of structures under such loading condition. The most common method is through simulation using a computer software. Today's structural computer analysis applications are more capable than ever before, and the processing power of computers is continuing to increase. These factors are allowing engineers to analyze larger and more complicated models in a relatively short amount of time. However, the results of an analysis are only as good as the model. Computational simulations may not produce realistic results when structures contain highly nonlinear structural members or subassemblages. In these cases, testing structures experimentally using a dynamic testing method such as the shake table testing method may predict the behavior in a more accurate manner. Shake table tests are not as commonly performed as computational simulations because they are expensive and consume more time. The shake table models are usually scaled down to fit on the shaking table, so size, scale and rate effects become issues.

Hybrid simulation is increasing in popularity since it integrates computer simulations with experimental tests. A structure is partitioned into two different classes. One class includes the portions of the structure that are well understood or can be modeled well in structural analysis software packages. The other consists of the partitions that can not be simulated well using computer programs. Thus, these parts of the structure are tested experimentally. The computational partitions interact with the experimental parts to produce a more realistic simulation of the structure under dynamic loading. There are many benefits of using hybrid simulation besides the simulation results. It takes advantage of the recent advances in structural analysis software and computing power. Also, it is not as costly or does not require as much equipment and resources as shake table tests. Thus, it is easy to see why researchers prefer hybrid simulation.

Hybrid simulation was conceived by Takanashi, et al. [27] in 1975. Mahin and Nakashima further pioneered the method starting in the early 1980s. It was then called pseudodynamic testing since only the restoring forces from the specimen were considered while the other rate dependent effects were not. In the late 1980s, Shing and Mahin [23, 24] studied the effects error and the rate-of-loading of the specimen had on pseudodynamic test results. Nakashima, et al. [16] developed the operator-splitting (OS) time integration scheme for pseudodynamic testing to minimize the errors in 1990. This is most widely used time integration scheme for hybrid simulation today. All

the while, researchers have performed many successful pseudodynamic tests.

Hybrid simulation is still an active area of research in structural and earthquake engineering. Some researchers are investigating fast or “real-time” hybrid simulation [25] where not only does the restoring forces of the specimen have to be considered but also the inertial forces and the effects of the transfer system. Others are exploring ways of improving the OS time integration scheme [1, 6, 10] and developing new time integration algorithms altogether. Some are researching different methods of control such as force, switch [7] and mixed control [18]. The development of a systematic customizable software framework for performing hybrid simulation [26] has been another area of research interest. Indeed, there still remains many challenges in hybrid simulations that need to be addressed.

## **1.1 Components of hybrid simulation**

There are three major parts in hybrid simulation. The explanation of these parts provides a good introduction and primer to hybrid simulation. The three parts are the computational driver, the physical transfer system and the middleware.

### **1.1.1 Computational driver**

The analytical part of the structure is modeled in the computational driver during a hybrid simulation. This is the part of structure whose behavior is relatively well understood and can be simulated well using a computer software. This component drives the hybrid simulation in several ways. It first initiates the simulation. Then during the simulation, it sends command signals to the transfer system and may or may not use the feedback, depending on the integration scheme, to advance the simulation to the next integration time-step. After the simulation ends, it cleans up the processes and closes the communication.

The computational driver can actually take many forms. A commercial Finite Element Analysis (FEA) software is commonly used. A FEA software must be flexible enough where a new user-defined element can be programmed and that user-defined element must be able to communicate with an outside program. In this dissertation, OpenSees is used as the FEA computational driver. Computational drivers are not limited to just FEA programs. For example, Matlab and Matlab-based programs can be use as a computational driver too. UI-SimCor [14], a Matlab-based hybrid simulation program, has been successfully used as the computational driver [21]. A Matlab script that runs a simple forced based structural analysis is used as the computational driver in this dissertation as well. As long as a language is capable of performing or coordinating structural analysis and communicating with an outside program, it can be used as a computation driver in hybrid simulation. The computational driver is a key necessary component of hybrid simulation.

### **1.1.2 Physical transfer system**

The transfer system is used to impose the command signals from the computational driver to the physical portion of the structure. The command signals considered in this dissertation are displacements and forces. It then sends the measured feedback (displacements and forces) back to

the computational driver. A typical transfer system consists of a control system, data acquisition (DAQ) system and a physical experimental setup including the specimen being tested.

A control system usually has hardware and software components. The software component is a graphical-user-interface to the hardware component. Typically, a PID controller is used to controller the actuators for hybrid simulation. The DAQ system is used to collected data during the test. It uses various devices such as load cells, linear variable differential transformers (LVDT) and accelerometers. It also has software and hardware components much like the control system. The experimental setup usually includes the load boundary elements, actuators and DAQ devices.

The specimen is fabricated to represent a part of the structure that cannot be modeled well in a computational driver. It maybe a complicated nonlinear sub-structural element. It maybe a new innovative structural component where the computer model has never been verified. Whatever the case, a physical specimen is needed to provide a more realistic behavior of a structural component. Detailed information about the transfer system used in this dissertation is provided in Appendix B.

### **1.1.3 Middleware**

A component that mediates the interaction between the computational driver and the transfer system is required for hybrid simulation. This is called the middleware. This middleware is, often times, a computer program that is custom programmed for each hybrid simulation experiment and testing lab. This makes it difficult for researches to collaborate and port what others have already done. This is the impetus for the development of OpenFresco. OpenFresco is transparent extensible software framework for hybrid simulation. Appendix A provides more information on OpenFresco. OpenFresco is used in this dissertation as the middleware.

## **1.2 Motivation**

The motivation for this research is to extend hybrid simulation techniques, with the hopes that hybrid simulation will continue to be an effective method of structural testing. The ability to simulate structures with computer software is very import to structural engineers. Some structures and structural components can not be simulated well using purely numerical models. These components need to tested experimentally. Thus, hybrid simulation is used.

The current need in hybrid simulation stems from the limitations of control systems. It is common that during hybrid simulation, the controller is pushed to its limits. Typically hybrid simulation is conducted with the control system in displacement control mode. This, in many cases, produce good results and is quite effective. There are however specimens where displacement control mode does not suffice. The control system has difficulty imposing command displacements accurately. This results in tests that do not provide reliable results and waste much time, money and effort. As more structural component and assemblages become complex, this problem may become more common than ever. Therefore, other control modes are explored for hybrid simulation, not only to meet the current needs, but also to anticipate future needs and to conceptually complete the formation of hybrid simulation.

### 1.3 Organization and scope

This dissertation is organized into six major chapters. Chapter 1 provides a general introduction to the dissertation. It gives a general overview of hybrid simulation and the motivation for further research. Chapter 2 explains displacement control hybrid simulation. No new research is presented in this chapter. Chapter 2 provides a backdrop for the rest of the chapters. It explains how hybrid simulation is typically performed and how the other types of hybrid simulation in the subsequent chapters differ from displacement control hybrid simulation.

The next three chapters cover force control, switch control and mixed control hybrid simulations. A chapter is devoted to each type. These chapters clearly define the terms that are used. They then present the motivation, develop the methods, and verify these methods through hybrid simulation tests. OpenFresco, the  $\mu$ -NEES experimental setup, the MTS-STS controller and xPC-Target real time operating system are utilized for the testing. In implementing and testing of these methods, a new class is developed in OpenFresco. Finally, the last chapter concludes this dissertation and gives recommendations for future research.

# Chapter 2

## Displacement control hybrid simulation

Displacement control (DC) hybrid simulation is described and explained in this chapter as a backdrop to different types of hybrid simulation set forth in the latter chapters. It defines DC hybrid simulation. The necessary components of DC hybrid simulation are presented. The interaction of these components with one another is explained. This chapter shows the sequential order of processes that typically occur in DC hybrid simulation for one integration time step. A simple one-bay-frame example is used to illustrate the concepts presented in this chapter. It concludes by presenting some advantages and limitations of DC hybrid simulation.

### 2.1 Introduction

DC hybrid simulation is a type of hybrid simulation where the control system imposes displacements on a physical specimen via actuation means such as actuators during hybrid simulation. Thus, “DC” refers to the control mode of the control system. Hybrid simulation is typically run in this manner. This lends to a relatively transparent and stable way of running hybrid simulation for at least two reasons. First, most Finite Element Analysis (FEA) programs use the displacement-based formulation. As a result, the displacements applied on the specimen are easily acquired. Second, the control system is usually designed to run well in DC mode. So the displacements are imposed in a stable and sufficient manner

### 2.2 Computational driver

As stated in Chapter 1, the analytical portion of the structure is numerically modeled and analyzed in the computational driver during hybrid simulation. During DC hybrid simulation, the computational driver sends the trial displacements for the control system to impose on the specimen. The computational driver may or may not use the feedback values from the control system to advance to the next time step. This depends on the type of time integration scheme being deployed for the simulation. When feedback is used, force feedback is commonly used and not displacement feedback.

In the one-bay-frame example, the model consists of two columns, Element 1 and 2, connected by a spring, Element 3, as shown in Figure 2.1. A lumped mass is placed at the top of each



column. The two column bases are fixed. The columns are axially rigid, and the tops are free to rotate. The rotational DOFs are condensed out so that only DOF 1 and DOF 2 remain. Elements 2 and 3 are modeled in a FEA software along with the mass and damping. A time integration scheme and a ground motion are selected for the finite element analysis. Explicit and predictor-corrector time integration methods are preferred over implicit methods because of the iterative nature of implicit methods. Iterating on path-dependent physical specimens may produce unrealistic results in hybrid simulation. For this example, the  $\alpha$ OS (Operator Splitting) time integration scheme is selected.

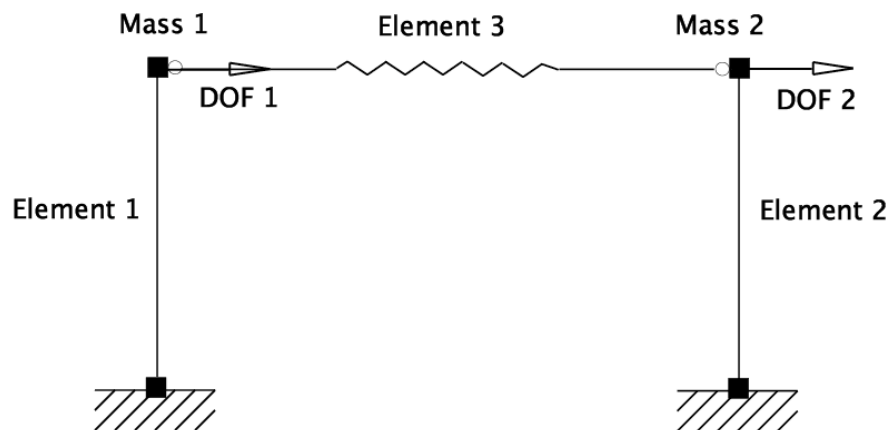


Figure 2.1: One-bay-frame model.

## 2.3 Physical transfer system

The role of the physical transfer system in DC hybrid simulation is to impose the trial displacements from the computational driver to the physical specimen. It then sends the measured feedback (displacements and forces) back to the computational driver. A typical physical transfer system consists of a control system, data acquisition (DAQ) system and a physical experimental setup including the specimen being tested. The test specimen represents a portion of the structure that cannot be modeled well in a computational driver.

A control system typically employs a PID controller to control the actuators for DC hybrid simulation as seen in Figure 2.2. It is important to note that the Integrator Loop is running at a different clock speed than the Servo-Control Loop. The Servo-Control Loop runs at a fixed clock speed while Integrator Loop runs at a varying clock speed depending the complexity of the analysis at a given time step. It is the role of the control system to bridge the disparity between the two clock speeds.

One common way of overcoming this difficulty is for the control system to ramp and hold the trial displacement until the next trial displacement is received from the computational driver. While this method solves the problem, it has some drawbacks. One main drawback occurs when

the the control system has to hold the trial displacement for a prolonged period of time while the computational driver is calculating the next command displacement. This adversely effects the result due to the rate-dependent property of the specimen and stress relaxation within the specimen. The ramp time also needs to be set properly so that the control system has an adequate time to achieve the trial displacement in a stable manner. At the same time, it cannot be too long such that the trial displacement is never reached before the next trial displacement is received.

To overcome the problems associated with the ramp and hold method, Mosequeda [15] developed and implemented a host of event-driven predictor-corrector methods. This is a two-stage method. It predicts the next trial displacement where the ramp and hold method holds the previous command displacement, and it corrects to reach the new trial displacements where the ramp and hold method ramps to the new command displacement. It predicts using an extrapolation polynomial from previous trial displacements and calculated velocities and accelerations. If the computational driver takes a relatively long time to send the next trial displacement, this methods eventually holds the last calculated predictor displacement to avoid unrealistic predictions.

When the next trial displacement is received from the computational driver, this method corrects to reach the new trial displacement using an interpolation polynomial formulated from previous trial displacements, calculated velocities and accelerations and the new trial displacement. Different predictor-correct methods are implemented using combinations of polynomials of degree 0, 1, 2 and 3 for the extrapolation and interpolation polynomials. Higher degree polynomials are avoided because they produce unnecessary oscillation and because they require too much history data. Upon unloading, that long a history is irrelevant. The predictor-corrector methods provide smooth and continuous actuator movement which is ideal for hybrid simulation. In the *nees@berkeley* lab, these predictor-corrector methods were implement using Mathworks Simulink and Stateflow models in combination with xPC real-time operating system. Additional information about the *nees@berkeley* lab can be found in Appendix B.

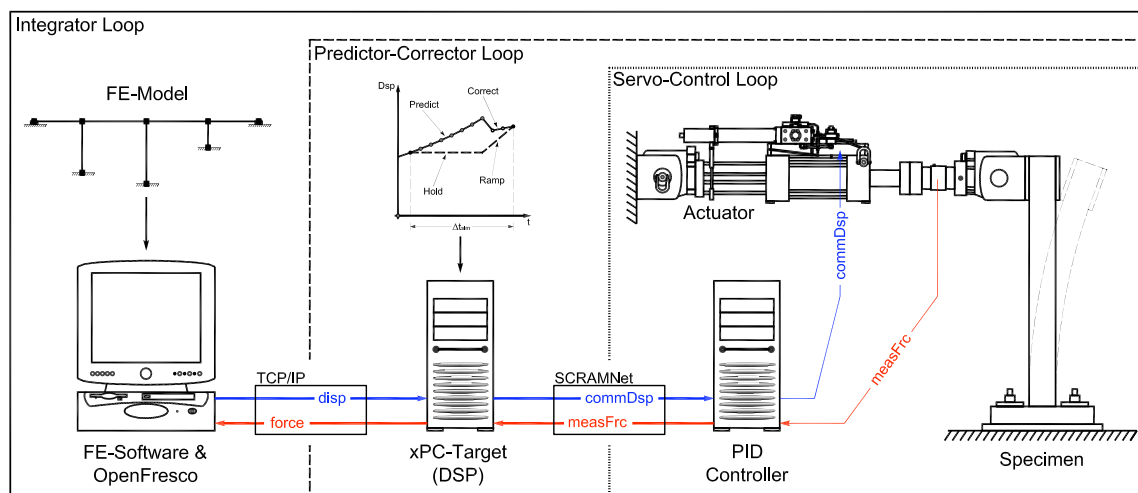


Figure 2.2: Control loop in the control system. for DC hybrid simulation

The DAQ system collects and records data during the test. Load cells (LC) and linear variable

differential transformers (LVDT) measure force and displacements respectively. These measure forces and displacements may be fed back to the computational driver. These measured values are in the actuator coordinate system and need to be transformed into element coordinate system before being fed back. Lastly, the experimental setup usually includes load boundary elements, actuators and DAQ devices.

In the one-bay-frame example, Element 1 is tested physically using a specimen. The  $\mu$ -NEES experimental setup at *nees@berkeley* (Figure 2.3) is used for this example. Additional information about the  $\mu$ -NEES experimental setup and the lab can be found in Appendix B. One of the columns with the clevis at the bottom and the actuator at the top is the physical specimen for this example. xPC real-time operating system with a predictor-correct method is also used in conjunction with the MTS-STC controller.

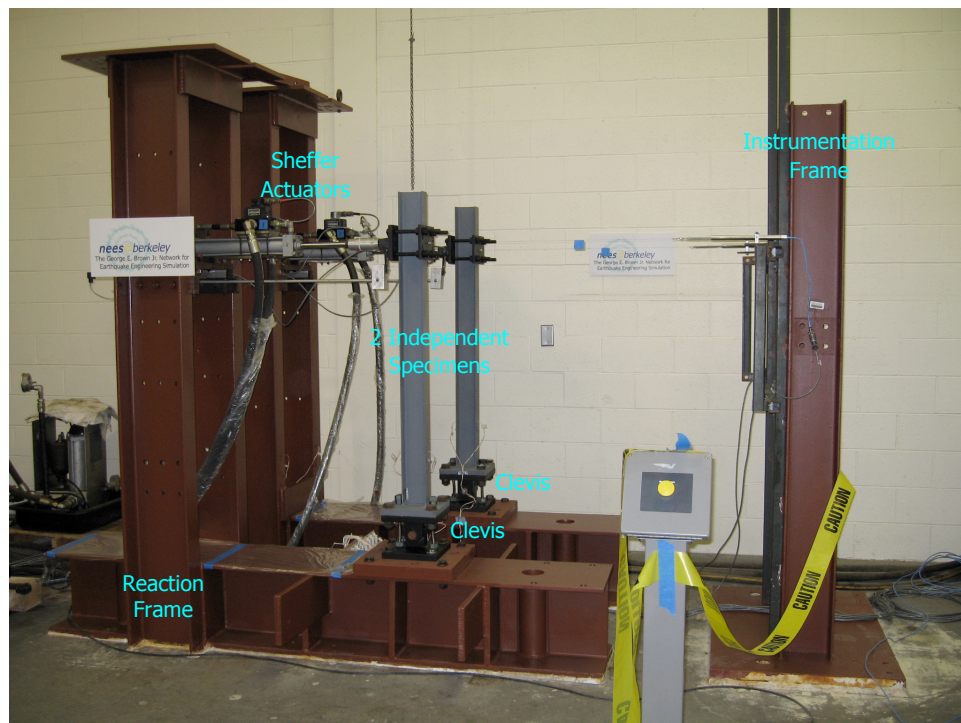


Figure 2.3:  $\mu$ -NEES experimental setup at *nees@berkeley*.

## 2.4 OpenFresco middleware hybrid simulation framework

The purpose of OpenFresco middleware is to mediate the transactions between the computational driver and the physical transfer system [20]. The interaction between the computational driver and the physical transfer system can become complicated. There are several factors to consider. First factor is the method of communication, deciding how one process will communicate with the other. It is also important to consider that the trial displacements from the computational driver may not be in the appropriate coordinate system for actuators to apply to the specimen.

Necessary transformations need to be performed to make sure that specimen is loaded correctly and the feedback is properly handled by the computational driver. Interfacing with the control and DAQ system can also become a challenge since each control and DAQ systems have unique interfaces. Therefore, there is a need for a third system that mediates the interactions between the computational driver and the physical transfer system.

In many previous instances, this third system was custom programmed for each hybrid simulation. Each laboratory or equipment site had their own unique way of conducting hybrid simulation. This made it difficult for sites to collaborate with one another especially in multi-site tests. It was also challenging for a lab to port and modify another lab's implementation even though the processes were essentially the same with different interfaces. OpenFresco solves this difficulty by providing a transparent and easily extendable software framework for conducting hybrid simulation. Figure 2.4 shows the OpenFresco abstract classes in green. An in-depth discussion of OpenFresco can be found in Appendix A. Table 2.1 shows how OpenFresco is configured for the one-bay-frame example. The subsequent subsections explain the settings in detail.

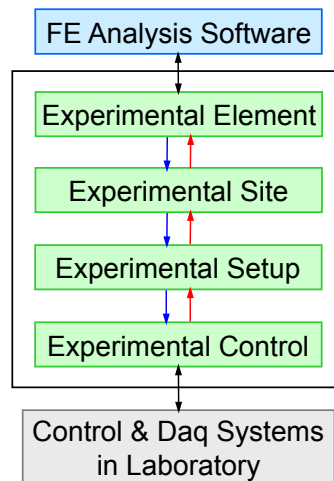


Figure 2.4: OpenFresco abstract classes and processes.

Table 2.1: OpenFresco configuration for the one-bay-frame example

<i>Experimental Element:</i>	Two-Node Link experimental element
<i>Experimental Site:</i>	Local experimental site
<i>Experimental Setup:</i>	One Actuator experimental setup
<i>Experimental Control:</i>	xPC-Target experimental control

### 2.4.1 Two-Node Link experimental element

The one-bay-frame example employs the Two-Node Link experimental element (Figure 2.5). This experimental element is imbedded in the FEA model in the computational driver and repre-

sents the physical specimen. It is the interface between the computational driver and OpenFresco. It provides the stiffness, mass and damping matrices to the computational driver. It also provides the resisting force vector for a given deformation state. This experimental element is responsible for transforming the trial deformations from the global coordinate system to the basic experimental element coordinate system and for reverse transforming the measured forces and/or displacements from the basic experimental element coordinate system to the global coordinate system. For this example, the 1-dimensional Two-Node Link element in the shear direction (2-direction) is used because the rotational and axial DOFs are condensed out at the nodes in this model.

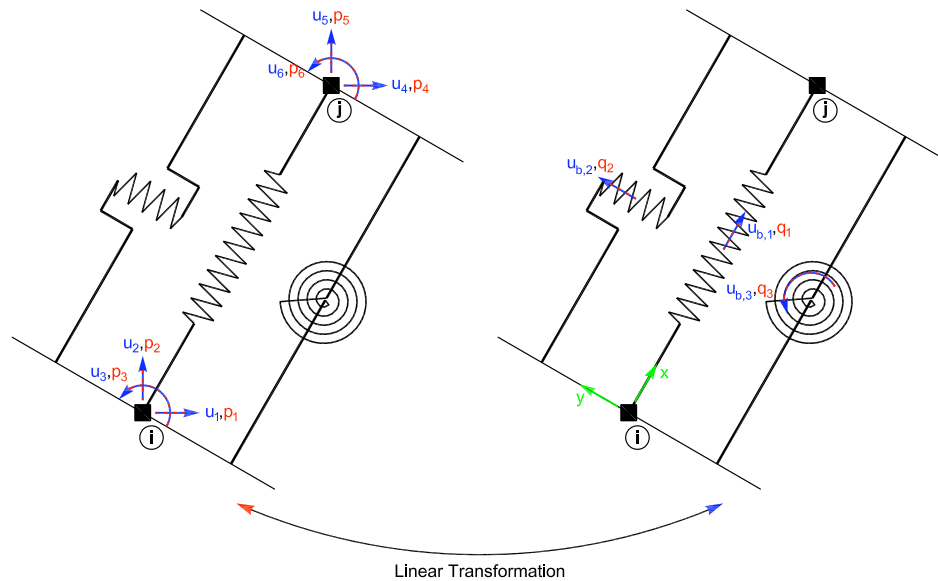


Figure 2.5: OpenFresco Two-Node Link experimental element in linear transformation.

## 2.4.2 Local experimental site

This example is a local test meaning that the computational driver, the OpenFresco middleware server and the physical transfer system are all running at the same location. Thus, the Local experimental site is deployed in OpenFresco. There are no network communications between the computation driver and OpenFresco and no network communication within OpenFresco. This is the simplest experimental site configuration.

## 2.4.3 One Actuator experimental setup

The One Actuator experimental setup (Figure 2.6) is used for this example. The One Actuator experimental setup represents the physical transfer system in the lab excluding the control and DAQ systems. Thus, the experimental setup in Figure 2.6 looks very similar to  $\mu$ -NEES setup in Figure 2.3. The One Actuator experimental setup transforms the trial deformation in the basic experimental element system (Figure 2.7) to the trial displacements in the actuator coordinate

system. It executes the reverse transformations for the measured forces and displacements as well. This experimental setup assembles the vector that is sent to the control system and extracts the vector that is received from the control system.

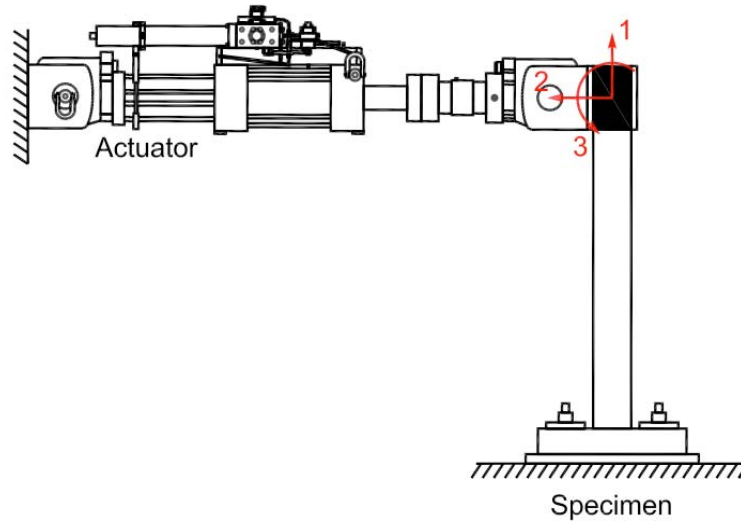


Figure 2.6: OpenFresco One Actuator experimental setup.

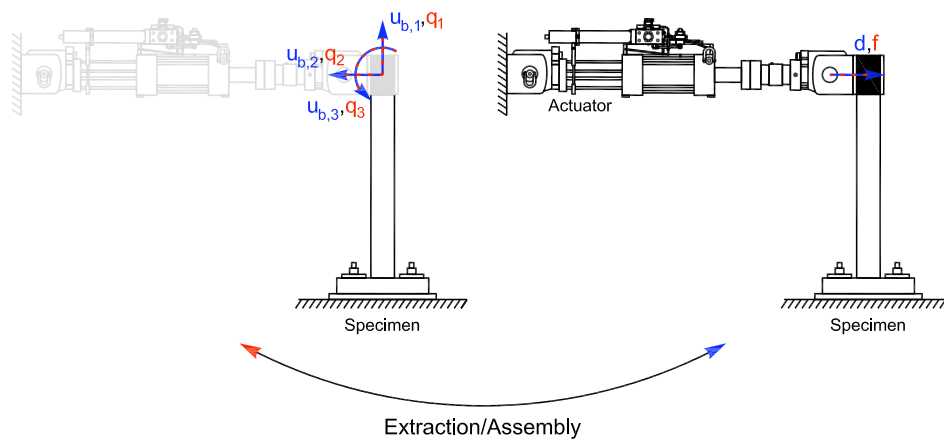


Figure 2.7: OpenFresco One Actuator experimental setup transformation.

#### 2.4.4 xPC-Target experimental control

The one-bay-frame example uses the xPC-Target experimental control to communicate with the control system. This experimental control uses the xPC-Target application program interface (API) to communicate with the xPC-Target computer running the Simulink predictor-corrector

model. As shown in Figure 2.2, OpenFresco and the xPC-Target computer communicate through a TCP/IP connection. The xPC-Target computer and the PID controller are on a SCRAMNet ring which is a type of a shared-memory network. The displacement being generated by the predictor-corrector algorithm running on xPC-Target can nearly instantaneously be seen by the PID controller through SCRAMNet. This allows instantaneous and deterministic communication between the two processes.

## 2.5 Sequential Processes of DC hybrid simulation

This section sequentially lays out the processes in DC hybrid simulation at integration time step,  $n$ . It explains each process in detail and shows how the computational driver, the OpenFresco middleware and the physical transfer system interact with each other. Each process is applied to the one-bay-frame example to aid in the explanation. This section does not cover the processes responsible for initializing and ending the simulation. Only the key methods in the OpenFresco concrete classes are highlighted. The UML (United Modeling Language) sequence diagram in Figure 2.8 shows the sequence of methods that are called in the OpenFresco classes when the computational driver or the simulation application sets the trial response. Figure 2.8 corresponds to Processes 1 to 5 and are labeled accordingly. Figure 2.9 shows the methods that are called when the simulation application tries to acquire the resisting force. Figure 2.9 corresponds to Processes 6 to 10 and are labeled accordingly

1. **The computational driver calculates the predictor velocity ( $\tilde{\mathbf{u}}_n$ ) and displacement ( $\tilde{\mathbf{u}}_n$ ) vectors.**

The one-bay-frame is modeled in the FEA software using the mass matrix ( $\mathbf{M}$ ), the viscous damping matrix ( $\mathbf{C}$ ) and the tangent stiffness matrix ( $\mathbf{K}$ ). This example uses the  $\alpha$ OS time integration scheme. The  $\alpha$ OS time integration scheme is based the Hiber-Hughes-Talyor (HHT) equation of motion (Equation 2.1) [9].  $\ddot{\mathbf{u}}_i$ ,  $\dot{\mathbf{u}}_i$ ,  $\mathbf{u}_i$  and  $\mathbf{f}_i$  are the nodal acceleration, velocity, displacement and load vectors at time step,  $i$ , respectively.  $\mathbf{r}(\mathbf{u}_i)$  is the nodal resisting force vector function with  $\mathbf{u}_i$  as its input. The current time step  $n$  and the previous time step  $n - 1$  are used in Equation 2.1.  $\alpha$  where  $\alpha \in [2/3, 1]$  is a parameter selected by the user that allows the tuning of numerical damping.

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{u}}_n + \alpha\mathbf{C}\dot{\mathbf{u}}_n + (\alpha - 1)\mathbf{C}\dot{\mathbf{u}}_{n-1} + \alpha\mathbf{r}(\mathbf{u}_n) + (\alpha - 1)\mathbf{r}(\mathbf{u}_{n-1}) \\ = \alpha\mathbf{f}_n + (\alpha - 1)\mathbf{f}_{n-1} \end{aligned} \quad (2.1)$$

The  $\alpha$ OS method uses the Newmark relationships between the displacements, velocities and accelerations at time steps  $n$  and  $n - 1$ . Equation 2.2 estimates the velocities at time step  $n$ . It can be organized as a sum of two parts in Equation 2.3. The first part is the predictor velocities where  $\tilde{\dot{\mathbf{u}}}_n = \dot{\mathbf{u}}_{n-1} + \Delta t(1 - \gamma)\ddot{\mathbf{u}}_{n-1}$ . The second part is the corrector.

$$\dot{\mathbf{u}}_n = \dot{\mathbf{u}}_{n-1} + \Delta t(1 - \gamma)\ddot{\mathbf{u}}_{n-1} + \Delta t\gamma\ddot{\mathbf{u}}_n \quad (2.2)$$

$$= \tilde{\dot{\mathbf{u}}}_n + \Delta t\gamma\ddot{\mathbf{u}}_n \quad (2.3)$$

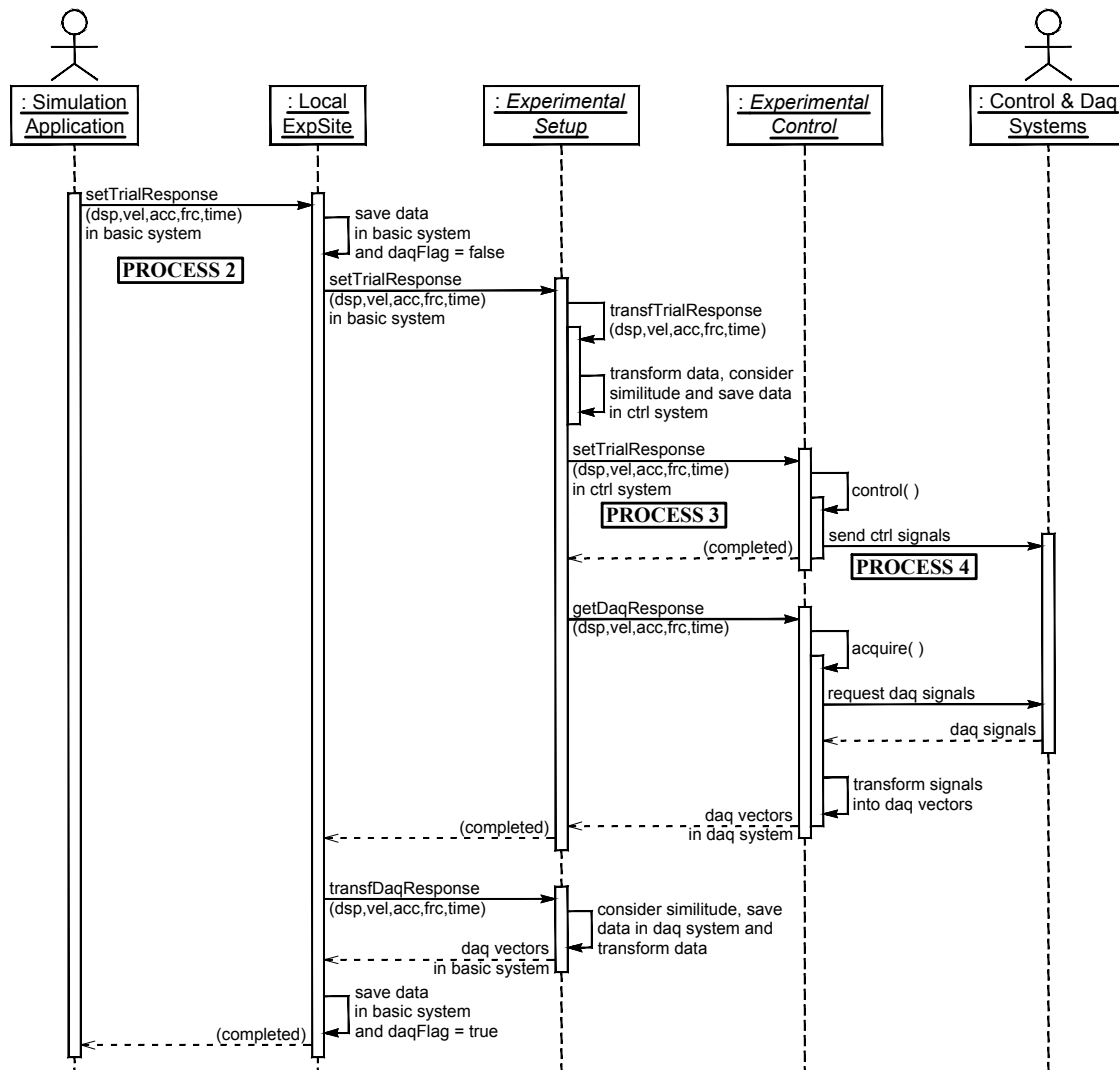


Figure 2.8: OpenFresco UML sequence diagram for setting trial response during a local test.



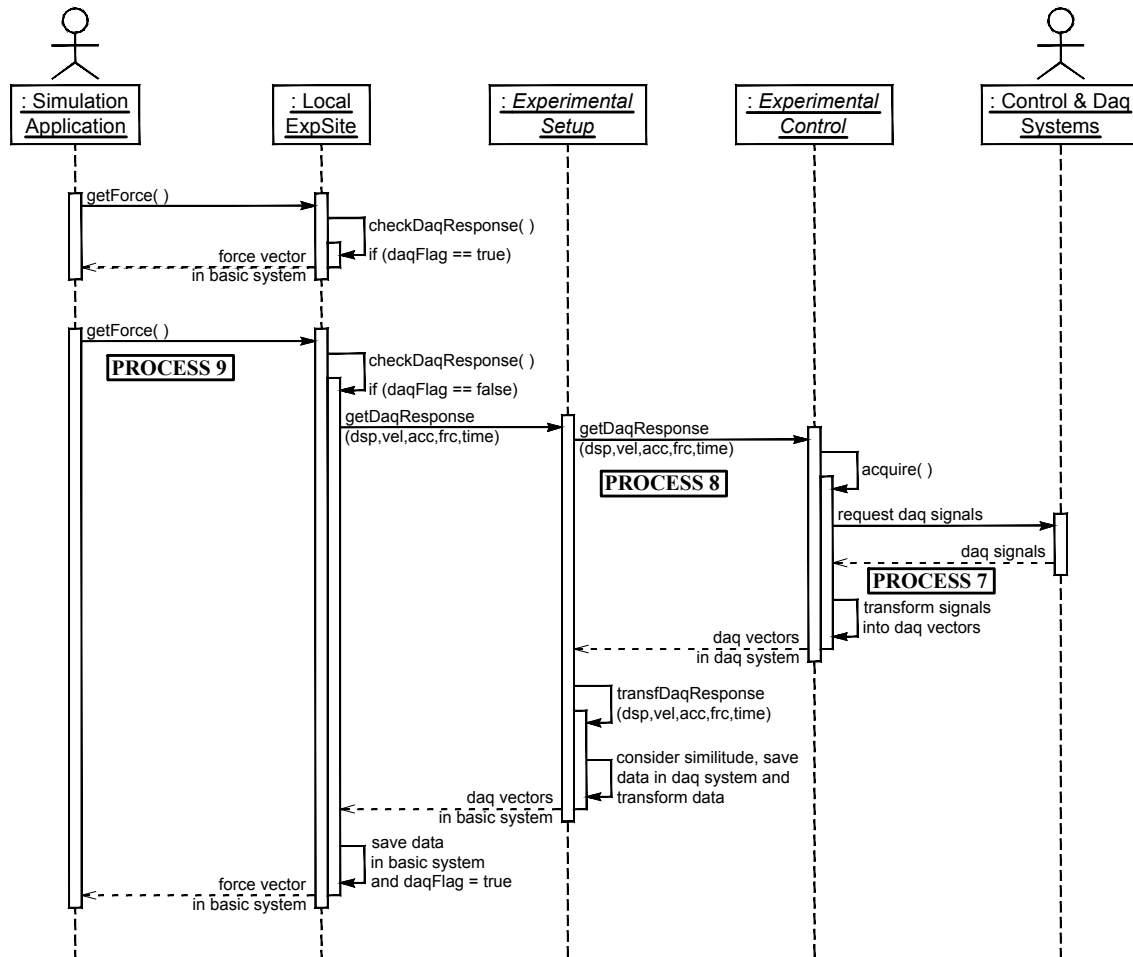


Figure 2.9: OpenFresco UML sequence diagram for obtaining the resisting force during a local test.

Equation 2.4 approximates the the displacements at time step  $n$ . It likewise can be organized into a sum of the predictor and corrector parts. The predictor displacements are  $\tilde{\mathbf{u}}_n = \mathbf{u}_{n-1} + \Delta t \dot{\mathbf{u}}_{n-1} + (\Delta t)^2(1/2 - \beta)\ddot{\mathbf{u}}_{n-1}$ .

$$\mathbf{u}_n = \mathbf{u}_{n-1} + \Delta t \dot{\mathbf{u}}_{n-1} + (\Delta t)^2(1/2 - \beta)\ddot{\mathbf{u}}_{n-1} + (\Delta t)^2\beta\ddot{\mathbf{u}}_n \quad (2.4)$$

$$= \tilde{\mathbf{u}}_n + (\Delta t)^2\beta\ddot{\mathbf{u}}_n \quad (2.5)$$

It is important to note that these predictor velocities and displacements are calculated from the known velocities and displacements from the previous time step,  $n - 1$ .  $\beta$  and  $\gamma$  are computed from Equations 2.6 given a selected  $\alpha$  value.

$$\beta = \frac{(2 - \alpha)^2}{4} \quad \gamma = \frac{(3 - 2\alpha)}{2} \quad (2.6)$$

The correction is performed in Process 10. In this step, the computational driver only calculates  $\tilde{\mathbf{u}}_n$  and  $\tilde{\mathbf{u}}_n$ .

**2. The experimental element transforms the predictor displacement vector ( $\tilde{\mathbf{u}}_n$ ) to the trial displacement vector ( $\hat{\mathbf{u}}_n$ ) and sends  $\hat{\mathbf{u}}_n$  to the experimental setup.**

The Two-Node Link experimental element in the FEA software running on the computational driver transforms the predictor displacement vector ( $\tilde{\mathbf{u}}_n$ ) calculated in Process 1 to the trial displacement vector ( $\hat{\mathbf{u}}_n$ ) in the basic experimental element coordinate system. In this example, it takes the nodal displacement at DOF 1 of the one-bay-frame model in Figure 2.1 and transforms it to the trial displacement in the 2-direction of the Two-Node Link element as seen in Figure 2.5. Since this is an one-dimensional element, the transformation is simple. The vector  $\hat{\mathbf{u}}_n$  has only one entry. After the transformation is complete, the Two-Node Link element sends  $\hat{\mathbf{u}}_n$  to the One Actuator experimental setup through the Local experimental site. The Two-Node Link object uses the update method to invoke the `setTrialResponse` public method in the Local Site object. This `setTrialResponse` method calls the `setTrialResponse` public method in the One Actuator object.

**3. The experimental setup transforms the trial displacement vector ( $\hat{\mathbf{u}}_n$ ) from the basic experimental element coordinate system to the actuator coordinate system and sends the transformed trial displacement vector ( $\check{\mathbf{u}}_n$ ) to the experimental control.**

The One Actuator experimental setup receives  $\hat{\mathbf{u}}_n$  from the Two-Node Link element. It transforms  $\hat{\mathbf{u}}_n$  to the actuator coordinate system and assembles the new trial displacement vector ( $\check{\mathbf{u}}_n$ ) as shown in Figure 2.7. This transformation is performed in the `setTrialResponse` method of the One Actuator object which calls the `transfTrialResponse` protected method. It then sends the  $\check{\mathbf{u}}_n$  to the xPC-Target experimental control by calling the `setTrialResponse` public method of the xPC-Target object. The transformations in the experimental element and the experimental setup for this example are very simple by design. However, these transformations can become complicated depending on the model and the setup. OpenFresco can perform both linear and non-linear transformations for both of these processes.

**4. The experimental control sends the transformed trial displacement vector ( $\check{\mathbf{u}}_n$ ) to the control system.**

The xPC-Target experimental control sends  $\check{\mathbf{u}}_n$  from the One Actuator experimental setup to the xPC-Target computer via a TCP/IP connection (Figure 2.2) by calling the `control` protected method of xPC-Target object. The xPC-Target `setTrialResponse` method calls the `control` method. Along with sending  $\check{\mathbf{u}}_n$ , the `control` method tells xPC-Target that a new command displacement has been sent and to switch the predictor-corrector algorithm from predicting displacements to start correcting to the new trial displacement.

**5. The control system imposes the trial displacement.**

The predictor-corrector model running on xPC-Target starts correcting to the new trial displacement  $\check{\mathbf{u}}_n$ . The MTS-STS controller sees the correcting displacements via SCRAMNet. The controller imposes these displacements through the actuator in the  $\mu$ -NEES experimental setup (Figure 2.3). The predictor-corrector algorithm switches from correcting to predicting once it reaches the new trial displacement  $\check{\mathbf{u}}_n$ . These displacements are again imposed on

the specimen by the MTS-STS controller and the  $\mu$ -NEES setup. The predictor-corrector algorithm continues to predict until the next trial displacement  $\check{\mathbf{u}}_{n+1}$  is received. In an event where  $\check{\mathbf{u}}_{n+1}$  does not arrive in a timely manner, the predictor-corrector algorithm will go into slow down mode. It will eventually hold the last calculated predictor displacement until  $\check{\mathbf{u}}_{n+1}$  arrives.

**6. The DAQ system measures the displacements and forces.**

The DAQ system measures the displacements and forces via the LVDT and LC respectively. The LVDT and LC are part of the  $\mu$ -NEES experimental setup. The PID controller uses these feedbacks as a part of its closed-loop control system. The controller also writes these on the SCRAMNet for the xPC-Target to record and feed them back to the xPC-Target experimental control.

**7. The experimental control reads measured displacement ( $\bar{\mathbf{u}}_n$ ) and force ( $\bar{\mathbf{f}}_n$ ) from the DAQ system and sends  $\bar{\mathbf{u}}_n$  and  $\bar{\mathbf{f}}_n$  to the experimental setup.**

The xPC-Target experimental control checks first to see if the trial displacement has been imposed by the control system, meaning it has finished correcting to the new trial displacement. If the trial displacement has been imposed, the xPC-Target experimental control reads the measured displacement ( $\bar{\mathbf{u}}_n$ ) and force ( $\bar{\mathbf{f}}_n$ ) at the point when the trial displacement was imposed. Both of these actions are performed in the `acquire()` protected method in xPC-Target object. The `acquire` method is called within the `getDaqResponse` public method of the xPC-Target object. This `getDaqResponse` method is invoked by the `getDaqResponse` public method of the One Actuator object.

**8. The experimental setup transforms the measured displacement ( $\bar{\mathbf{u}}_n$ ) and force ( $\bar{\mathbf{f}}_n$ ) from the actuator coordinate system to the basic element coordinate system and sends the transformed force ( $\check{\mathbf{f}}_n$ ) vectors to the experimental element.**

The One Actuator experimental setup receives  $\bar{\mathbf{u}}_n$  and  $\bar{\mathbf{f}}_n$  from the xPC-Target experimental control. It transforms  $\bar{\mathbf{u}}_n$  and  $\bar{\mathbf{f}}_n$  in the actuator coordinate system to  $\check{\mathbf{u}}_n$  and  $\check{\mathbf{f}}_n$  the basic experimental element coordinate system. This is the reverse transformation of that in Process 3. The One Actuator `getDaqResponse` method calls the `transfDaqResponse` private method to execute the transformation. The One Actuator `getDaqResponse` method is invoked by the `checkDaqResponse` public method of the Local Site object. The `checkDaqResponse` method checks to see if this is the first time this method has been called. If it is, it calls the One Actuator `getDaqResponse` method. The `checkDaqResponse` method is called by the `getForce` public method of the Local Site object.

**9. The experimental element transforms the measure force ( $\check{\mathbf{f}}_n$ ) from the basic experimental element coordinate system to the global coordinate system and feeds back the force  $\mathbf{r}(\check{\mathbf{u}}_n)$  to the computational driver.**

The Two-Node Link experimental element obtains  $\check{\mathbf{f}}_n$  from the One Actuator Setup. It transforms  $\check{\mathbf{f}}_n$  in the basic experimental element coordinate system to  $\mathbf{r}(\check{\mathbf{u}}_n)$  in the global coordinate system. This is the reverse transformation of Process 2. Then  $\mathbf{r}(\check{\mathbf{u}}_n)$  is fed back to the FEA program running on the computational driver.  $\check{\mathbf{f}}_n$  is obtained by the `getResistingForce`

public method of the Two-Node Link object which calls the getForce method of the Local Site object. Then the getResistingForce execute the reverse transformation to the global coordinate system.

10. **The computational driver uses the measured force ( $\mathbf{r}(\tilde{\mathbf{u}}_n)$ ) and moves to time step,  $n + 1$ .** The  $\alpha$ OS method uses Equation 2.7 to approximate  $\mathbf{r}(\mathbf{u}_{n-1})$  in Equation 2.1.

$$\mathbf{K}\mathbf{u}_{n-1} - \mathbf{r}(\mathbf{u}_{n-1}) \approx \mathbf{K}\tilde{\mathbf{u}}_{n-1} - \mathbf{r}(\tilde{\mathbf{u}}_{n-1}) \quad (2.7)$$

Figure 2.10 gives a graphical representation of this approximation where the deviation from linearity of the resisting force at predictor displacement  $\tilde{u}_{n-1}$  ( $K\tilde{u}_{n-1} - r(\tilde{u}_{n-1})$ ) approximates the unknown deviation at  $u_{n-1}$  ( $Ku_{n-1} - r(u_{n-1})$ ). Using Equations 2.2 and 2.4,  $\dot{\mathbf{u}}_n$

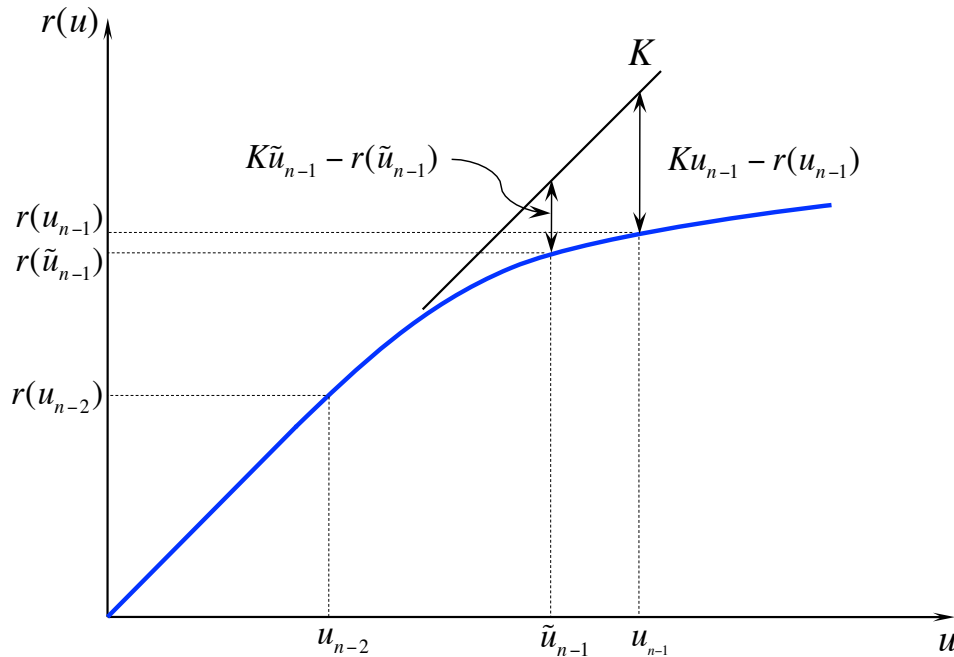


Figure 2.10: Force-deformation curve showing the  $\alpha$ OS approximation of  $r(u_{n-1})$  for 1 DOF system.

are  $\ddot{\mathbf{u}}_n$  are formulated in terms of  $\mathbf{u}_n$  and  $\tilde{\mathbf{u}}_n$ .

$$\dot{\mathbf{u}}_n = \frac{\gamma}{\beta\Delta t}(\mathbf{u}_n - \tilde{\mathbf{u}}_n) + \dot{\mathbf{u}}_{n-1} + \Delta t(1 - \gamma)\ddot{\mathbf{u}}_{n-1} \quad (2.8)$$

$$\ddot{\mathbf{u}}_n = \frac{1}{\beta(\Delta t)^2}(\mathbf{u}_n - \tilde{\mathbf{u}}_n) \quad (2.9)$$

By substituting Equations 2.8, 2.9 and 2.7 into Equation 2.1, Equation 2.10 can be formu-

lated where  $\Delta \mathbf{u} = \mathbf{u}_n - \tilde{\mathbf{u}}_n$  and  $\mathbf{r}(\tilde{\mathbf{u}}_n)$  is obtained from Process 9.

$$\begin{aligned} & \left[ \frac{1}{\beta(\Delta t)^2} \mathbf{M} + \alpha \frac{\gamma}{\beta \Delta t} \mathbf{C} + \alpha \mathbf{K} \right] \Delta \mathbf{u} \\ & = -\alpha [\mathbf{C}(\dot{\mathbf{u}}_{n-1} + \Delta t(1-\gamma)\ddot{\mathbf{u}}_{n-1}) + \mathbf{r}_n(\tilde{\mathbf{u}}_n) - \mathbf{f}_n] \\ & \quad - (1-\alpha) [\mathbf{C}\dot{\mathbf{u}}_{n-1} + (\mathbf{r}_{n-1}(\tilde{\mathbf{u}}_{n-1}) + \mathbf{K}\mathbf{u}_{n-1} - \mathbf{K}\tilde{\mathbf{u}}_{n-1}) - \mathbf{f}_{n-1}] \quad (2.10) \end{aligned}$$

The  $\alpha$ OS method does not require any iterations. The FEA program solves for  $\Delta \mathbf{u}$  in one step.  $\Delta \mathbf{u}$  is then substituted into Equation 2.9 to solve for the accelerations ( $\ddot{\mathbf{u}}_n$ ) at time step  $n$ .  $\ddot{\mathbf{u}}_n$  is used to correct the velocities and displacements in Equations 2.3 and 2.5 respectively. The analysis now advances to time step  $n+1$  and starts again from Process 1.

## 2.6 Advantages and limitations of DC hybrid simulation

There are advantages of performing hybrid simulation with the control system in DC mode as opposed to other modes of control. First, the DC hybrid simulation formulation is consistent with most FEA programs. Most FEA programs use the displacement-based method to solve both the static and dynamic problems. Having the computational driver send out displacements to the control system and use the measured forces as the feedback works well with how the FEA program proceeds from the global level to the element level and back to the global level again. Second, most control and loading systems are designed to run well in displacement or position control. Therefore, running these systems in force control for example becomes much more difficult. Lastly, most of the work done in hybrid simulation have been in DC hybrid simulation. Therefore, not much pioneering research has been performed using other modes of control, making other types of hybrid simulation much more difficult.

This does not, however, mean that DC hybrid simulation is not without shortcomings. There are some specimens that cannot be tested well in DC. If the specimen is very stiff in the direction of loading, DC will not suffice. This type of specimen is very common in structural engineering. It appears in shearwalls in the lateral direction or in beam-columns in the axial direction. Another case is rapid or real-time hybrid simulation. When running hybrid simulation at relatively high speeds, the inertial effects of both the specimen and the loading system must be considered. DC will not suffice in this situation. There are other testing situations where switching of the control mode during the simulation is necessary. Specimens that are initially very stiff and exhibit softening behavior later may require force control in the initial stage and switch to DC in the later stage. One experimental setup may require multiple control modes to be active simultaneously. As in the beam-column example, the loading in the axial direction may be in force control and the lateral direction may be in DC, both at the same time. Therefore, there exists a need to explore other control strategies in hybrid simulation.

# Chapter 3

## Force control hybrid simulation

Chapter 3 addresses the challenges of force control (FC) hybrid simulation. This chapter starts by explaining FC hybrid simulation and how it differs from DC hybrid simulation in Chapter 2. Then the difficulties of FC hybrid simulation are defined; the motivation is stated; and the previous research is surveyed. The methods for FC hybrid simulation are proposed. These methods are categorized into two types, displacement-based and force-based methods. Both types of methods are explained in detail, and their implementations using OpenFresco at *nees@berkeley* lab are described. It concludes with a presentation and a discussion of the results from these FC methods using both the 1-DOF and 2-DOF  $\mu$ -NEES experimental setups.

### 3.1 Introduction

#### 3.1.1 Problem definition

FC hybrid simulation is a type of hybrid simulation where the control system imposes forces on the physical specimen through the loading system during hybrid simulation as opposed to DC hybrid simulation where the loading system imposed displacements on the physical specimen as explained in Chapter 2. The loading system is typically composed of actuators and load-boundary elements. Thus, “FC” refers to the control mode of the control system. If the computation driver requires feedback, the measure displacements are used where as in DC hybrid simulation the measured forces are used. In general, FC hybrid simulation is more difficult to perform than DC hybrid simulation.

FC hybrid simulation poses two major challenges. The first challenge is acquiring the forces to apply to the specimen from the computational driver. This becomes difficult because most FEA programs in structural engineering use displacement-based formulation where trial deformations are imposed on each element and the corresponding resisting forces are calculated. Therefore, there are no straightforward ways of acquiring the trial forces from most existing FEA programs.

The second challenge lies within the control and loading systems. Most control and loading systems used in structural testing are designed primarily to perform well in displacement or position control. Consequently, these systems do not run as well in force or load control. One primary reason for this is that the stiffness of the loading system as well as the stiffness of the specimen

contribute to the proportional gain of the control system. When testing a very stiff specimen in FC, the proportional gain of the control system must be set relatively low. This results in poor tracking of the forces, meaning that the command forces do not coincide well with the feedback forces. However, if the proportional gain is increased to improve tracking, the control system becomes unstable. The change in the stiffness of the specimen during an experiment may also adversely affect the performance of the system. Stick and slip along with friction within the moving parts of the actuator cause jerky actuator motion in FC. Friction may also adversely affect the tracking of the command forces. As a result, stick and slip along with friction degrades the performance of the control system in FC.

### 3.1.2 Motivation

Even though FC hybrid simulation is difficult to perform well, there exists a need for it in structural testing. The primary motivation for it arises from the limitation of the control and loading systems in DC. There are certain situations where the control and loading systems do not perform well in DC. Testing a specimen that is very stiff in the direction of loading is a prime example. The one-bay-frame model (Figure 2.1) is used to demonstrate this point. Table 3.1 shows two one-bay-frame models and their properties. The main difference between Model 1 and 2 is that the elastic stiffness of Element 1 in Model 2 is 25 times more than that of Model 1.

Table 3.1: Properties of soft and stiff one-bay-frame models.

	<b>Model 1: Soft Model</b>	<b>Model 2: Stiff Model</b>
Elastic Stiffness (kip/in.): [Element 1   Element 2   Element 3]	[4   2   5]	[100   2   5]
Mass (kip/g): [ $Mass_1$ $Mass_2$ ]	[0.10   0.05]	[0.10   0.05]
Period (sec): [ $T_1$ $T_1$ ]	[0.88   0.51]	[0.54   0.20]
Mass Proportional Damping with $\xi = 5\%$ : [ $\alpha_M = 2\xi\omega_1$ ]	[0.71]	[1.17]

Both models are analyzed using the  $\alpha$ OS time integration scheme from Section 2.5 with  $\alpha = 2/3$ . The models are subjected to the first 350 points of the El Centro ground motion as shown in Figure 3.1. This ground motion is scaled by 15%. In the subsequent sections of this chapter, the one-bay-frame model will be used again to test the FC methods. Element 1 of the one-bay-frame model will be substituted with a specimen. The ground motion is scaled to 15% in these tests to keep the specimen within its elastic range. The ground motion is scaled here to be consistent with the rest of the chapter. The scaled peak ground acceleration is 0.048g.

Figures 3.2 shows the trial displacement-time history plots of Element 1 for Model 1 and 2 produced from this analysis. The trial displacement values for Element 1 of Model 1 range from -0.660 in. to 0.712 in. while for Model 2 it ranges from -0.066 in. to 0.065 in. The measured forces of Element 1 for both Models from these trial displacements are plotted in Figure 3.3. The range of measured forces for Model 1 is from -2.640 kips to 2.846 kips. For Model 2, the measured force ranged from -6.594 kips to 6.485 kips.

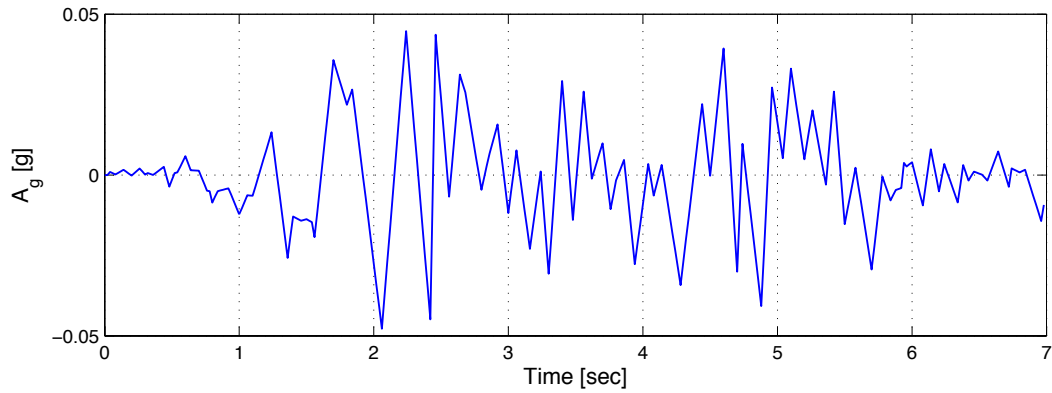


Figure 3.1: The first 350 points of the scaled horizontal North-South component of the ground motion recorded at a site in El Centro, California during the Imperial Valley earthquake of May 18, 1940 [5]. The ground acceleration data was recorded at every 0.02 second.

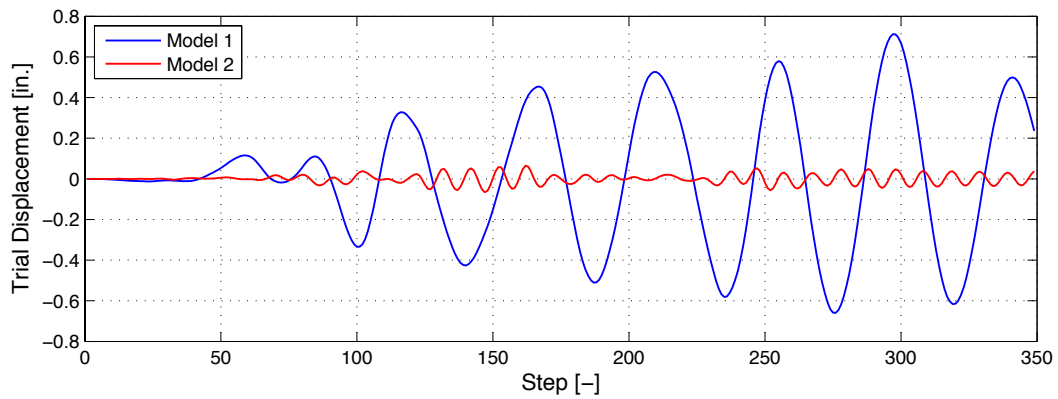


Figure 3.2: Element 1 trial displacement-time history plots for Model 1 and 2.

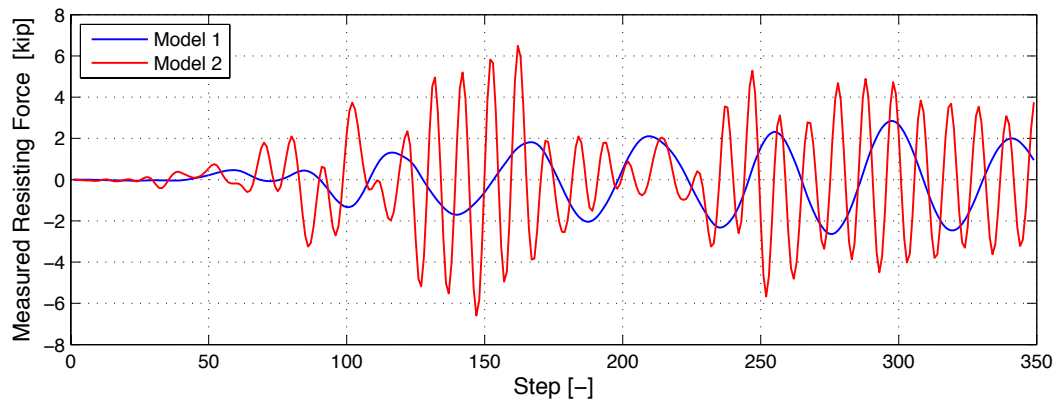


Figure 3.3: Element 1 measured force-time history plots for Model 1 and 2.



In general, as the elastic stiffness of one element increases, the global deformation of the structure decreases for a given loading condition. In these models, increasing elastic stiffness of Element 1 by a factor of 25 decreased the range of trial displacement approximately 10 times and increased the range of measured forces approximately 3 times. It is important to note that Element 1 in Model 2 is not only much stiffer than Element 1 in Model 1 but also much stiffer than Element 2 and 3. Element 1 is resisting more of the load as it is stiffer than the other elements. Yet because it is so stiff, it is not deforming as much. Numerically, the ranges of trial displacements for both models do not pose a problem. They are well within machine precision.

In hybrid simulation where Element 1 is replaced with a specimen in the lab, it is difficult to accurately impose these small Element 1 trial displacements produced by Model 2, depending on the control and loading systems. The same control and loading systems, however, are able to sufficiently impose the range of Element 1 measured forces from Model 2. In addition, small displacements are generally better measured than applied. In this case, FC hybrid simulation consisting of these trial force and measured displacement pairs produces more realistic results than DC hybrid simulation with the same trial displacement and measured force pairs. This is assuming that the range of trial forces produced by the FC algorithm is similar in range to that of measured forces from the DC algorithm.

Real-time and rapid hybrid simulations are other cases where FC is better than DC. In real-time hybrid simulation, the simulation time step is equal to the time interval of the recorded ground-motion. The simulation time step may be scaled relative to the time interval of the recorded ground-motion depending of the scale of the specimen. For example, the 1940 El Centro ground motion [5] contains the ground acceleration recorded every 0.02 second. In real-time hybrid simulation, the unscaled simulation time step is set at 0.02 second, meaning that the control system expects a new trial displacement or force every 0.02 second. Rapid hybrid simulation refers to hybrid simulation that runs relatively fast. The simulation time step is set relatively small.

The control and loading systems are running fast in both real-time and rapid hybrid simulation. When hybrid simulation runs fast, the dynamics effect of both the specimen and the loading system have to be considered. Only the resisting force is feed back to the computational driver during DC hybrid simulation. The mass and viscous damping characteristics of the specimen are modeled in the FEA program. The forces generated from the specimen's inertial and viscous damping properties during the simulation are ignored. During fast testing, the dynamics effects have a greater contribution to the overall feedback force. The loading system has its own dynamic effect from the movement of the loading elements like header beams, actuator rods and hydraulic fluid in the actuators. These must be accounted for during the simulation. It is impossible to apply a trial displacement, velocity and acceleration on the specimen simultaneously. It makes more sense to calculate the applied force containing all these dynamic effects [28].

FC hybrid simulation expands the capabilities of hybrid simulation in respect to the types of specimens that can be tested and the speed at which these tests are performed. One of the main benefits of hybrid simulation is that new and innovative structural systems can be tested experimentally. Testing of such structural systems requires equally new and innovative testing methods. Another main benefit is that it produces results comparable to shake table test without the cost and having to scale the specimen down to the table size. As hybrid simulation runs faster, it better able to capture the dynamic effects of the specimen, making more comparable to shake table tests.

### 3.1.3 Previous research

A few researchers experimented with FC hybrid simulation. One of the earliest attempts at FC was made by Thewalt [28] in 1987. He proposed the effective force testing (EFT) method as a way to perform real-time and rapid hybrid simulation. Given the discretized equation of motion at time step  $n$  (Eq. 3.1), the discretized forcing function is  $\mathbf{f}_n = -\mathbf{M}\ddot{\mathbf{u}}_g(t_n)$ , where  $\ddot{u}_g(t_n)$  is the recorded ground acceleration.  $\mathbf{f}_n$  can be calculated before the start of the simulation once the mass matrix ( $\mathbf{M}$ ) is formulated.  $\mathbf{f}_n$  becomes the trial force vector  $\hat{\mathbf{f}}_n$  in the EFT method.

$$\mathbf{M}\ddot{\mathbf{u}}_n + \mathbf{C}\dot{\mathbf{u}}_n + \mathbf{r}_n = \mathbf{f}_n \quad (3.1)$$

$\hat{\mathbf{f}}_n$  is transformed to the actuator coordinate system ( $\check{\mathbf{f}}_n$ ) and then applied to the specimen through the loading system. Once  $\check{\mathbf{f}}_n$  is applied, the restoring forces ( $\bar{\mathbf{f}}_n$ ), displacements ( $\bar{\mathbf{u}}_n$ ), velocities ( $\bar{\dot{\mathbf{u}}}_n$ ) and accelerations ( $\bar{\ddot{\mathbf{u}}}_n$ ) are measured and recorded at each actuator DOFs. The algorithm is then advanced to the next time step,  $n + 1$ .

The obvious advantage of the EFT method is that the applied forces  $\hat{\mathbf{f}}_n$  are easily computed as long as the structure is not partitioned into numerical and physical substructures. This means that the entire structure is tested physically while the mass is modeled numerically. Another advantage is that it tries to take into account the inertial and energy dissipating effects during real-time and rapid testing. These factors make the EFT method a good alternative to shaking table tests. However, the EFT method breaks down when the structural model is partitioned into numerical and experimental parts. The problem is that the computed effective forces are still in the global coordinate system at each discretized nodes. It is unknown how much of these global nodal forces should be applied to the physical specimen. It is also important to note that  $\hat{\mathbf{f}}_n$  is precomputed meaning that  $\hat{\mathbf{f}}_n$  does not depend on any measured feedback values.

In 2005, Pan, et al. [18] performed switch and mixed control hybrid simulation where the control modes used were displacement and force. Only the FC aspect is reviewed in this chapter. The switch and mixed control portion of their research are discussed in the subsequent chapters. Pan, et al. [18] performed hybrid simulation on High Damping Rubber Bearing (HDRB) base isolators. Their structural model of eight-story two-span steel moment frame was isolated by HDRBs. The eight-story moment frame was modeled analytically, and HDRBs were tested experimentally. The analytical portion of the structure was modeled with two DOFs per floor, one horizontal and one vertical. This totaled to 18 DOFs. The experimental part of the structure was setup with one horizontal actuator and one vertical actuator. The horizontal actuator was controlled in position control as the vertical actuator switched between position and FC. The horizontal and vertical DOFs were considered to be uncoupled. Hence the horizontal target displacement was computed independently of the vertical target displacement and force. They used both the horizontal and vertical components of the 1995 Hyogoken-Nanbu (Kobe) recorded ground motion simultaneously.

The predictor displacements for both actuators were calculated using the operator-splitting (OS) predictor-corrector scheme using Eq. 3.1 as the equation of motion at time step  $n$ . The predictor displacement was imposed on the specimen through the horizontal actuator. The measured force was fed-back to the scheme to correct the predictor displacement. The vertical trial force was computed as a product of the predictor vertical displacement and the vertical elastic stiffness of the bearings. The bearing was assumed to remain vertically elastic in FC. The measured displacements and forces were not fed back to the time integration scheme. The computed vertical trial force was

considered to be the resisting force ( $\mathbf{r}_n$ ) in Eq. 3.1. One disadvantage of Pan, et al.'s method for calculating the trial forces is that it is only applicable when the specimen is elastic. Therefore, the researcher must know the behavior of the specimen (force-deformation curve) beforehand in order to use their method.

Elkhoraibi and Mosalam [7] conducted research in switch control hybrid simulation in 2007 where they developed a method for performing FC. Only the FC portion of their research is discussed in this chapter. Their work on switch control hybrid simulation is presented in Chapter 4. Elkhoraibi and Mosalam used the secant stiffness estimation with the  $\alpha$ -OS time integration scheme to compute the trial forces. The DOFs were considered to be uncoupled meaning that the secant stiffness matrix only contained diagonal terms. This makes the calculation of the secant stiffness was rather straightforward. The secant stiffness term for each DOF or the diagonal term of the secant stiffness matrix,  $k_{n+1}^i$ , at iteration  $i$  and time step  $n+1$  was calculated using Eq. 3.2 where  $\check{f}_{n+1}^{i-1}$  and  $r_n$  are the trial force at iteration  $i-1$  and time step  $n+1$  and the converged resisting force at time step  $n$  respectively. Similarly,  $\check{u}_{n+1}^{i-1}$  and  $u_n$  are the measured displacement at iteration  $i-1$  and time step  $n+1$  and the converged displacement at time step  $n$  respectively.

$$k_{n+1}^i = (\check{f}_{n+1}^{i-1} - r_n) / (\check{u}_{n+1}^{i-1} - u_n) \quad (3.2)$$

The trial forces,  $\check{\mathbf{f}}_{n+1}^i$ , were obtained by using the predicted and correct displacement (Eq. 2.5) and velocity (Eq. 2.3) formulations and rearranging the HHT equation of motion (Eq. 2.1) into Equations 3.3, 3.4, 3.5 and 3.6. (The HHT equation of motion used by Elkhoraibi and Mosalam is a slight variation from Eq. 2.1. Nevertheless, both equations are numerically equivalent. The equations present by Elkhoraibi and Mosalam are reformulated in this Chapter to be consistent with Eq. 2.1.)

$$\mathbf{K}^* \check{\mathbf{u}}_{n+1}^i + \alpha \check{\mathbf{f}}_{n+1}^i = \mathbf{f}_{n+1}^* \quad (3.3)$$

$$\mathbf{K}^* = \frac{\mathbf{M}}{\Delta t^2 \beta} + \alpha \frac{\gamma \mathbf{C}}{\Delta t \beta} \quad (3.4)$$

$$\check{\mathbf{u}}_{n+1}^i = \mathbf{u}_n + \Delta \mathbf{u}_{n+1}^i = \mathbf{u}_n + (\mathbf{K}_{n+1}^i)^{-1} (\check{\mathbf{f}}_{n+1}^i - \mathbf{r}_n) \quad (3.5)$$

$$\mathbf{f}_{n+1}^* = \alpha \check{\mathbf{f}}_{n+1}^i + (1 - \alpha) \mathbf{f}_n - \alpha \mathbf{C} \check{\mathbf{u}}_{n+1}^i - (1 - \alpha) \mathbf{C} \mathbf{u}_n + \mathbf{K}^* \check{\mathbf{u}}_{n+1}^i - (1 - \alpha) \mathbf{r}_n \quad (3.6)$$

Using these equations, Eq. 3.7 was formulated and solved for  $\check{\mathbf{f}}_{n+1}^i$ .

$$[\mathbf{K}^* (\mathbf{K}_{n+1}^i)^{-1} + \alpha \mathbf{I}] \check{\mathbf{f}}_{n+1}^i = \mathbf{f}_{n+1}^* + \mathbf{K}^* (\mathbf{K}_{n+1}^i)^{-1} \mathbf{r}_n - \mathbf{K}^* \mathbf{u}_i \quad (3.7)$$

This results in the iterative implicit FC algorithm, Algorithm 3.1.

Iterative time integration schemes tend to produce unrealistic loading/unloading cycles on the specimen in hybrid simulation, while iterating to find the converged resisting forces,  $\mathbf{r}_{n+1}^{last}$ . Thus, a fraction of the incremental trial force,  $\lambda (\check{\mathbf{f}}_{n+1}^{i+1} - \check{\mathbf{f}}_{n+1}^i)$  with  $\lambda < 1$ , was applied to the specimen instead of the entire incremental force during each iteration.  $\lambda$  was a tuning parameter determined by the researchers. Elkhoraibi and Mosalam's method has a wider range of applications than the Pan, et al.'s method since Elkhoraibi and Mosalam's method calculates the trial forces even

---

**Algorithm 3.1** Implicit FC Algorithm
 

---

Define initial parameters  $\alpha, \beta, \gamma, \epsilon_r, i_{max}, \mathbf{K}_0$  and  $\mathbf{K}^*$  (Eq. 3.4)

**for**  $n = 0$  to (max # of steps-1) **do**

  Compute  $\tilde{\mathbf{u}}_{n+1}$  (Eq. 2.5),  $\tilde{\mathbf{u}}_{n+1}$  (Eq. 2.3) and  $\mathbf{f}_{n+1}^*$  (Eq. 3.6)

  Set  $\mathbf{K}_{n+1}^1 = \mathbf{K}_n$ ,  $\mathbf{u}_{n+1}^1 = \tilde{\mathbf{u}}_{n+1}$  and  $i = 0$

  Compute  $\check{\mathbf{f}}_{n+1}^{i+1}$  (Eq. 3.7)

**repeat**

    Apply  $\check{\mathbf{f}}_{n+1}^{i+1}$  in FC to the specimen and measure corresponding  $\check{\mathbf{u}}_{n+1}^{i+1}$

    Updated  $\mathbf{K}_{n+1}^{i+1}$  (Eq. 3.2)

    Compute  $\check{\mathbf{f}}_{n+1}^{i+2}$

$i = i + 1$

**until**  $i > i_{max}$  OR  $|\check{\mathbf{f}}_{n+1}^{i+1} - \check{\mathbf{f}}_{n+1}^i| \leq \epsilon_r$

  Set  $\mathbf{K}_{n+1} = \mathbf{K}_{n+1}^i$ ,  $\mathbf{u}_{n+1} = \check{\mathbf{u}}_{n+1}^i$  and  $\mathbf{r}_{n+1} = \check{\mathbf{f}}_{n+1}^{i+1}$

**end for**

---

when the specimen becomes inelastic. One limitation of their method is that it is only applicable for structural models that have uncoupled DOFs because the formulation is based on the secant stiffness matrix calculation of uncoupled DOFs (Eq. 3.2). In practice, structural models with uncoupled DOFs are difficult to find.

In 2006, Sivaselvan, et al. [25] have suggested an experimental setup where the force to be applied is first converted to a displacement. The actuator receives the displacement and imposes it on the elastic spring with a known elastic stiffness. Hence, now the spring is imposing a certain force to the specimen. The force is measured through a load cell that is placed between the actuator and the spring. This method attempts to work around the challenge of sending a force command directly to the actuator. Instead, the actuator is controlled by a displacement command which is a more stable way of controlling the actuator. Adding a spring makes the stiff system more compliant making it easier to control. One of disadvantages of this setup is that the spring must remain elastic across its range of motion. There arises an issue with properly attaching the spring to the actuator and the specimen as to avoid any slippage. Scaling this setup to impose a relatively large force may pose additional problems with the amount of energy being stored in the spring.

In summary, Thewalt, Pan, et al. and Elkhoraibi and Mosalam all developed methods for calculating the trial force in FC hybrid simulation. Sivaselvan, et al.'s researched focus on the control system aspect of FC hybrid simulation. While their research expanded the capabilities of hybrid simulation, their methods are limited in respect to their application. They were developed for specific control systems and experimental setups. They are not general enough for other researchers to easily for modify for their own purposes. There still exists a need for a more transparent, portable and robust way of performing FC hybrid simulation.

## 3.2 Methods for acquiring the trial force

One of the main challenges in FC hybrid simulation is calculating the trial forces, the forces to apply to the specimen. The methods for finding the trials forces are proposed in this section.

These methods are categorized into two types based on their formulation. The first type is based on the displacement-based formulation of solving the equation of motion (3.1) by enforcing the compatibility of deformations. Thus they are called compatibility force control (CFC) methods. The second type is the equilibrium force control (EFC) method. This type is based on solving for the forces in the equation of motion by enforcing the equilibrium of the forces. Both types of the methods have their advantages and disadvantages depending on their application. This section describes the formulation of these methods and discusses the pros and cons of such methods.

### 3.2.1 Compatibility methods

Most FEA software packages in structural engineering use the displacement-based formulation to solve the equation of motion. The compatibility methods use the trial displacements calculated by the FEA program for FC hybrid simulations. These displacements are the same ones calculated in Section 2.5 of Chapter 2. The compatibility methods transform or convert these displacements to forces for the control system to apply to the specimen. One way the transformation can be performed is by using the tangent stiffness matrix. The methods that use the tangent stiffness matrix are presented in Section 3.2.1.1. Another way of converting these displacements to forces is by using the Krylov subspaces. This method is explained in Section 3.2.1.2.

#### 3.2.1.1 Tangent-based compatibility methods

Provided that the trial force vector in the actuator coordinate system at time step  $n$ ,  $\check{\mathbf{f}}_n = \check{\mathbf{f}}(\check{\mathbf{u}}_n)$ , is a function of trial displacement vector in the same coordinate system at time step  $n$  ( $\check{\mathbf{f}}_n$ ), the Taylor expansion in Eq. 3.10 shows that  $\check{\mathbf{f}}_n$  can be approximated with the previous trial force  $\check{\mathbf{f}}_{n-1}$  and the current incremental trial displacement  $\Delta\check{\mathbf{u}}_n$ .

$$\check{\mathbf{f}}(\check{\mathbf{u}}_n) = \check{\mathbf{f}}(\check{\mathbf{u}}_{n-1} + \Delta\check{\mathbf{u}}_n) \quad (3.8)$$

$$= \check{\mathbf{f}}(\check{\mathbf{u}}_{n-1}) + \frac{\partial \check{\mathbf{f}}(\check{\mathbf{u}}_{n-1})}{\partial \mathbf{u}} (\Delta\check{\mathbf{u}}_n) + H.O.T. \quad (3.9)$$

$$\approx \check{\mathbf{f}}(\check{\mathbf{u}}_{n-1}) + \mathbf{K}_t(\check{\mathbf{u}}_{n-1}) \cdot (\Delta\check{\mathbf{u}}_n) \quad (3.10)$$

The tangent stiffness matrix ( $\mathbf{K}_t$ ) of the structure is the Jacobian of the resisting force vector function,  $\check{\mathbf{f}}(\check{\mathbf{u}}_{n-1})$ . The trial forces ( $\check{\mathbf{f}}_n$ ) in the actuator coordinate system at time step  $n$  can be calculated in a similar way as shown in Eq. 3.11.

$$\check{\mathbf{f}}_n = \mathbf{K}_{n-1}^{Act} \Delta\check{\mathbf{u}}_n + \bar{\mathbf{f}}_{n-1} \quad (3.11)$$

In Eq. 3.11,  $\mathbf{K}_{n-1}^{Act}$  is the tangent stiffness matrix of the specimen calculated from previously measured and calculated response quantities at time step  $n-1$ .  $\Delta\check{\mathbf{u}}_n$  is the incremental trial displacement vector where  $\Delta\check{\mathbf{u}}_n = \check{\mathbf{u}}_n - \check{\mathbf{u}}_{n-1}$ .  $\check{\mathbf{u}}_n$  and  $\check{\mathbf{u}}_{n-1}$  are the same displacement vectors that were calculated in Step 3 of Section 2.5 of Chapter 2.  $\bar{\mathbf{f}}_{n-1}$  is the measured forces in the actuator coordinates system at  $n-1$ .  $\mathbf{K}_{n-1}^{Act}$ ,  $\Delta\check{\mathbf{u}}_n$  and  $\bar{\mathbf{f}}_{n-1}$  are all in the actuator coordinate system.

The key to the tangent-based compatibility methods is in the formulation of the tangent stiffness matrix of the specimen in the actuator coordinate system ( $\mathbf{K}^{Act}$ ). In a DC hybrid simulation,

$\mathbf{K}_t$  for the experimental element is commonly not formulated at each time step. Instead, the initial tangent stiffness matrix which is the elastic stiffness matrix of the experimental element ( $\mathbf{K}_0$ ) is used throughout the dynamic analysis. The errors associated with using  $\mathbf{K}_0$  is either accepted or corrected in the time integration scheme.  $\mathbf{K}_0$  can be calculated analytically or experimentally [3]. There are benefits in hybrid simulation to updating the tangent stiffness matrix at each time step. It is not commonly done because of its inherent difficulty. However, there have been attempts to estimate the tangent stiffness matrix experimentally during a hybrid test. These methods for experimentally estimating the tangent stiffness matrix can be used to calculate the trial force. The remainder of this section covers various methods for calculating the tangent stiffness matrix of a specimen.

There is a subclass of Newton method called the secant or quasi-Newton methods. They are called such because no true Newton equation is ever formed throughout all iterations [12]. In other words, the Jacobian is not directly formulated at all. A secant approximation of the Jacobian is used instead. This approximation is executed through successive rank one updates to the current Jacobian matrix. One of the most well known of these is the Broyden method [2]. Carrion and Spencer [4] deployed the Broyden method to update the stiffness matrix for model-base time delay compensation.

$$\mathbf{K}_n = \mathbf{K}_{n-1} + \frac{(\Delta \mathbf{r}_n - \mathbf{K}_{n-1} \Delta \mathbf{u}_n) \Delta \mathbf{u}_n^T}{\Delta \mathbf{u}_n^T \Delta \mathbf{u}_n} \quad (3.12)$$

The update uses the incremental displacement ( $\Delta \mathbf{u}_n$ ) and force ( $\Delta \mathbf{r}_n$ ) vectors. Though their formulation is in the global coordination system, it can be easily adapted to the actuator coordinate system.

$$\mathbf{K}_n^{Act} = \mathbf{K}_{n-1}^{Act} + \frac{(\Delta \bar{\mathbf{f}}_n - \mathbf{K}_{n-1}^{Act} \Delta \bar{\mathbf{u}}_n) \Delta \bar{\mathbf{u}}_n^T}{\Delta \bar{\mathbf{u}}_n^T \Delta \bar{\mathbf{u}}_n} \quad (3.13)$$

$\Delta \bar{\mathbf{u}}_n$  and  $\Delta \bar{\mathbf{f}}_n$  are the incremental measured displacement and force vectors in the actuator coordinate system at time step  $n$ .

Taking a similar approach to Carrion and Spencer, Igarashi, et al. [11] used the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method to develop Algorithm 3.2 to update the tangent stiffness matrix during a hybrid simulation of a 5-story full scale building. They utilized the stiffness matrix for scaling the actuator movement. The algorithm uses already know pair of  $\Delta \bar{\mathbf{u}}_n$  and  $\Delta \bar{\mathbf{f}}_n$ . Eq. 3.14 can be formulated from Eq. 3.11.

$$\Delta \bar{\mathbf{f}}_n = \mathbf{K}_n^{Act} \Delta \bar{\mathbf{u}}_n \quad (3.14)$$

BFGS is a variation on the Broyden method. The BFGS method assumes the stiffness matrix to be symmetric and positive-definite, which is a valid assumption. The algorithm follows the minimal update approach. The incremental stiffness,  $\Delta \mathbf{K}_n^{Act}$ , with a smaller Forbenius norm updates the stiffness matrix. The  $\epsilon$  parameter, featured in Algorithm 3.2, depends on the fidelity of the measuring devices. This parameter is set so that the noise from the measuring devices do not spuriously trigger an update.

In recent years, researchers have formulated non-secant based methods for estimating the tangent stiffness matrix. Ahmadizadeh and Mosqueda [1] developed such a method to improve the performance of the OS time integration scheme. They update the stiffness matrix by using the ‘‘intrinsic’’ coordinate system.  $\mathbf{K}_n^{Act}$  results from the transformation of the stiffness matrix in the

---

**Algorithm 3.2** BFGS Method
 

---

Specify the value of  $\mathbf{K}_0^{Act}$  and set  $\hat{\mathbf{K}}_0^{Act} = \mathbf{K}_0^{Act}$   
 Set  $\varepsilon$   
**for**  $n = 0$  to (max # of steps-1) **do**  
 Measure  $\Delta\hat{\mathbf{x}}_n$  and  $\Delta\hat{\mathbf{f}}_n$   
**if**  $\Delta\hat{\mathbf{f}}_n^T \Delta\hat{\mathbf{x}}_n \geq \varepsilon \|\Delta\hat{\mathbf{f}}_n\| \|\Delta\hat{\mathbf{x}}_n\|$  **then**  
 $\hat{\mathbf{K}}_{n+1}^{Act} = \hat{\mathbf{K}}_n^{Act}$   
**else**  
 $\Delta\mathbf{K}1_n^{Act} = [1 + \frac{\Delta\hat{\mathbf{x}}_n^T \mathbf{K}_0^{Act} \Delta\hat{\mathbf{x}}_n}{\Delta\hat{\mathbf{f}}_n^T \Delta\hat{\mathbf{x}}_n}] \frac{\Delta\hat{\mathbf{f}}_n \Delta\hat{\mathbf{f}}_n^T}{\Delta\hat{\mathbf{f}}_n^T \Delta\hat{\mathbf{x}}_n} - \frac{\Delta\hat{\mathbf{f}}_n \Delta\hat{\mathbf{x}}_n^T \mathbf{K}_0^{Act}}{\Delta\hat{\mathbf{f}}_n^T \Delta\hat{\mathbf{x}}_n} - \frac{\mathbf{K}_0^{Act} \Delta\hat{\mathbf{x}}_n \Delta\hat{\mathbf{f}}_n^T}{\Delta\hat{\mathbf{f}}_n^T \Delta\hat{\mathbf{x}}_n}$   
 $\Delta\mathbf{K}2_n^{Act} = [1 + \frac{\Delta\hat{\mathbf{x}}_n^T \hat{\mathbf{K}}_n^{Act} \Delta\hat{\mathbf{x}}_n}{\Delta\hat{\mathbf{f}}_n^T \Delta\hat{\mathbf{x}}_n}] \frac{\Delta\hat{\mathbf{f}}_n \Delta\hat{\mathbf{f}}_n^T}{\Delta\hat{\mathbf{f}}_n^T \Delta\hat{\mathbf{x}}_n} - \frac{\Delta\hat{\mathbf{f}}_n \Delta\hat{\mathbf{x}}_n^T \hat{\mathbf{K}}_n^{Act}}{\Delta\hat{\mathbf{f}}_n^T \Delta\hat{\mathbf{x}}_n} - \frac{\hat{\mathbf{K}}_n^{Act} \Delta\hat{\mathbf{x}}_n \Delta\hat{\mathbf{f}}_n^T}{\Delta\hat{\mathbf{f}}_n^T \Delta\hat{\mathbf{x}}_n}$   
**if**  $\|\Delta\mathbf{K}1_n^{Act}\|_F \leq \|\Delta\mathbf{K}2_n^{Act}\|_F$  **then**  
 $\hat{\mathbf{K}}_{n+1}^{Act} = \mathbf{K}_0^{Act} + \Delta\mathbf{K}1_n^{Act}$   
**else**  
 $\hat{\mathbf{K}}_{n+1}^{Act} = \hat{\mathbf{K}}_n^{Act} + \Delta\mathbf{K}2_n^{Act}$   
**end if**  
**end if**  
**end for**

---

intrinsic coordination system  $\mathbf{P}_n$  via the transform matrix  $\mathbf{T}_p$ .

$$\mathbf{K}_n^{Act} = \mathbf{T}_p^T \mathbf{P}_n \mathbf{T}_p \quad (3.15)$$

$\mathbf{T}_p$  transforms the displacements from the actuator coordinate system to the intrinsic coordinate system. The important result of this transformation is that  $\mathbf{P}_n$  is now diagonal. The local measured incremental displacements and forces are transformed to the intrinsic coordinate system via Equations 3.16 and 3.17 respectively.

$$\Delta\mathbf{u}_n^P = \mathbf{T}_p \Delta\bar{\mathbf{u}}_n \quad (3.16)$$

$$\Delta\mathbf{f}_n^P = \mathbf{P}_{n-1} \mathbf{T}_p (\mathbf{K}_{n-1}^{Act})^{-1} \Delta\bar{\mathbf{f}}_n \quad (3.17)$$

$\mathbf{P}_n$  is easily updated using  $\Delta\mathbf{x}_n^P$  and  $\Delta\mathbf{f}_n^P$ .

$$\mathbf{P}_n = \text{diag}(\Delta\mathbf{u}_n^P)^{-1} \text{diag}(\Delta\mathbf{f}_n^P) \quad (3.18)$$

The intrinsic coordinate system needs to be determined for each experimental setup and specimen. Then,  $\mathbf{T}_p$  and  $\mathbf{P}_n$  are formulated from the intrinsic coordinate system. Ahmadizadeh and Mosqueda show an example using a simple 2-DOF experimental setup, but this process may become more difficult as the setup becomes more complex. This method is referred to as the Intrinsic method from here on.

Another non-secant base method was proposed by Hung and El-Tawi [10]. Their intent was the same as Ahmadizadeh and Mosqueda to improve the OS time integration scheme. They based their algorithm on Eq. 3.14 and on the assumption that the tangent stiffness matrix does not change

drastically between time steps. The matrices,  $\Delta\bar{\mathbf{F}}$  and  $\Delta\bar{\mathbf{U}}$  are formed from  $m$  vectors of  $\Delta\bar{\mathbf{f}}$  and  $\Delta\bar{\mathbf{u}}$  respectively from time step  $n+1$  to  $n+2-m$ .  $\Delta\bar{f}_j^{(i)}$  and  $\Delta\bar{u}_j^{(i)}$  are the measure incremental force and displacement at the  $i^{th}$  DOF at time step  $j$  or column  $j$  of  $\Delta\bar{\mathbf{F}}$  and  $\Delta\bar{\mathbf{U}}$  respectively. Now Eq. 3.19 is formed where  $k_{n+1}^{(i,j)}$  is the entry at  $i^{th}$  row and  $j^{th}$  column of  $\mathbf{K}_{n+1}^{Act}$  at time step  $n+1$  where  $dof$  is the total number of DOFs.

$$\begin{bmatrix} \Delta\bar{f}_{n+2-m}^{(1)} & \Delta\bar{f}_{n+3-m}^{(1)} & \cdots & \Delta\bar{f}_{n+1}^{(1)} \\ \Delta\bar{f}_{n+2-m}^{(2)} & \Delta\bar{f}_{n+3-m}^{(2)} & \cdots & \Delta\bar{f}_{n+1}^{(2)} \\ \vdots & \vdots & & \vdots \\ \Delta\bar{f}_{n+2-m}^{(dof)} & \Delta\bar{f}_{n+3-m}^{(dof)} & \cdots & \Delta\bar{f}_{n+1}^{(dof)} \end{bmatrix} = \begin{bmatrix} k_{n+1}^{(1,1)} & k_{n+1}^{(1,2)} & \cdots & k_{n+1}^{(1,dof)} \\ k_{n+1}^{(2,1)} & k_{n+1}^{(2,2)} & \cdots & k_{n+1}^{(2,dof)} \\ \vdots & \vdots & & \vdots \\ k_{n+1}^{(dof,1)} & k_{n+1}^{(dof,2)} & \cdots & k_{n+1}^{(dof,dof)} \end{bmatrix} \begin{bmatrix} \Delta\bar{u}_{n+2-m}^{(1)} & \Delta\bar{u}_{n+3-m}^{(1)} & \cdots & \Delta\bar{u}_{n+1}^{(1)} \\ \Delta\bar{u}_{n+2-m}^{(2)} & \Delta\bar{u}_{n+3-m}^{(2)} & \cdots & \Delta\bar{u}_{n+1}^{(2)} \\ \vdots & \vdots & & \vdots \\ \Delta\bar{u}_{n+2-m}^{(dof)} & \Delta\bar{u}_{n+3-m}^{(dof)} & \cdots & \Delta\bar{u}_{n+1}^{(dof)} \end{bmatrix} \quad (3.19)$$

When both sides of Eq. 3.19 are transposed, the following results.

$$\begin{bmatrix} \begin{pmatrix} \Delta\bar{f}_{n+2-m}^{(1)} \\ \Delta\bar{f}_{n+3-m}^{(1)} \\ \vdots \\ \Delta\bar{f}_{n+1}^{(1)} \end{pmatrix} & \begin{pmatrix} \Delta\bar{f}_{n+2-m}^{(2)} \\ \Delta\bar{f}_{n+3-m}^{(2)} \\ \vdots \\ \Delta\bar{f}_{n+1}^{(2)} \end{pmatrix} & \cdots & \begin{pmatrix} \Delta\bar{f}_{n+2-m}^{(dof)} \\ \Delta\bar{f}_{n+3-m}^{(dof)} \\ \vdots \\ \Delta\bar{f}_{n+1}^{(dof)} \end{pmatrix} \end{bmatrix} = \begin{bmatrix} \Delta\bar{u}_{n+2-m}^{(1)} & \Delta\bar{u}_{n+2-m}^{(2)} & \cdots & \Delta\bar{u}_{n+2-m}^{(dof)} \\ \Delta\bar{u}_{n+3-m}^{(1)} & \Delta\bar{u}_{n+3-m}^{(2)} & \cdots & \Delta\bar{u}_{n+3-m}^{(dof)} \\ \vdots & \vdots & & \vdots \\ \Delta\bar{u}_{n+1}^{(1)} & \Delta\bar{u}_{n+1}^{(2)} & \cdots & \Delta\bar{u}_{n+1}^{(dof)} \end{bmatrix} \begin{bmatrix} \begin{pmatrix} k_{n+1}^{(1,1)} \\ k_{n+1}^{(1,2)} \\ \vdots \\ k_{n+1}^{(1,dof)} \end{pmatrix} & \begin{pmatrix} k_{n+1}^{(2,1)} \\ k_{n+1}^{(2,2)} \\ \vdots \\ k_{n+1}^{(2,dof)} \end{pmatrix} & \cdots & \begin{pmatrix} k_{n+1}^{(dof,1)} \\ k_{n+1}^{(dof,2)} \\ \vdots \\ k_{n+1}^{(dof,dof)} \end{pmatrix} \end{bmatrix} \quad (3.20)$$

$m$  must be greater than or equal to  $dof$ . When  $m = dof$ , the rows of  $\mathbf{K}_{n+1}^{Act}$  are obtained by solving Eq. 3.21 using Gaussian elimination. When  $m > dof$ , the rows of  $\mathbf{K}_{n+1}^{Act}$  are found via the least-squares method.

$$\begin{pmatrix} \Delta\bar{f}_{n+2-m}^{(i)} \\ \Delta\bar{f}_{n+3-m}^{(i)} \\ \vdots \\ \Delta\bar{f}_{n+1}^{(i)} \end{pmatrix} = \begin{bmatrix} \Delta\bar{u}_{n+2-m}^{(1)} & \Delta\bar{u}_{n+2-m}^{(2)} & \cdots & \Delta\bar{u}_{n+2-m}^{(dof)} \\ \Delta\bar{u}_{n+3-m}^{(1)} & \Delta\bar{u}_{n+3-m}^{(2)} & \cdots & \Delta\bar{u}_{n+3-m}^{(dof)} \\ \vdots & \vdots & & \vdots \\ \Delta\bar{u}_{n+1}^{(1)} & \Delta\bar{u}_{n+1}^{(2)} & \cdots & \Delta\bar{u}_{n+1}^{(dof)} \end{bmatrix} \begin{pmatrix} k_{n+1}^{(n,1)} \\ k_{n+1}^{(n,2)} \\ \vdots \\ k_{n+1}^{(n,dof)} \end{pmatrix} \quad (3.21)$$

The algorithm uses the initial tangent matrix for the unloading/reloading tangent matrix. It also does not update the stiffness matrix when the incremental trial displacements are smaller than the user-defined threshold. Hung and El-Tawi recommend setting the threshold to two or three times the displacement resolution. This method is referred to as the Transpose method from here on.



In this subsection, four methods for estimating the tangent stiffness matrix were presented: Broyden, BFGS, Intrinsic and Transpose methods. These four methods were not originally formulated to be used for FC hybrid simulation. However, they can be used to estimate  $\mathbf{K}_{n-1}^{Act}$  in Eq. 3.11 which is the key element in calculating the trial forces. The implementation of these methods for FC hybrid simulation is presented in Section 3.3.1 of this chapter.

### 3.2.1.2 Krylov subspace-based compatibility method

Using the tangent stiffness matrix to acquire the target force is promising, but these methods do have an inherent difficulty of overcoming spurious updates caused by the noises from the measuring instruments. There are also some questions as to the accuracy of these estimates. A more ideal way would be to not use or bypass the tangent stiffness matrix to find the trial force. This subsection describes a subspace method for acquiring the trial forces directly from  $\Delta\check{\mathbf{u}}$  and  $\Delta\check{\mathbf{f}}$ .

Scott and Fenves [22] used a Krylov subspace to accelerate the Newton method. In a similar way, a Krylov subspace can become the bases for the trial force vector. Starting with Eq. 3.22 where the flexibility matrix in the actuator coordinate system ( $\mathbf{F}^{Act}$ ) is the inverse of the stiffness matrix ( $\mathbf{K}^{Act}$ )

$$0 = \Delta\check{\mathbf{u}}_{n+1} - \mathbf{F}^{Act} \Delta\check{\mathbf{f}}_{n+1} \quad (3.22)$$

where  $\Delta\check{\mathbf{u}}_{n+1}$  is the trial displacement vector in the actuator coordinate system from the FEA software.  $\Delta\check{\mathbf{f}}_{n+1}$  is written as an addition of two vectors  $\mathbf{p}_{n+1}$  and  $\mathbf{q}_{n+1}$ .

$$\Delta\check{\mathbf{f}}_{n+1} = \mathbf{p}_{n+1} + \mathbf{q}_{n+1} \quad (3.23)$$

Now Eq. 3.22 becomes

$$0 = \Delta\check{\mathbf{u}}_{n+1} - \mathbf{F}^{Act} \Delta\mathbf{p}_{n+1} - \mathbf{F}^{Act} \Delta\mathbf{q}_{n+1} \quad (3.24)$$

The key to this algorithm is to minimize the 2-norm of the difference of  $\Delta\check{\mathbf{x}}_{n+1}$  and  $\mathbf{F}^{Act} \mathbf{p}_{n+1}$ .

$$\min \|\Delta\check{\mathbf{u}}_{n+1} - \mathbf{F}^{Act} \mathbf{p}_{n+1}\|_2 \quad (3.25)$$

$\mathbf{p}_{n+1}$  is assumed to be a linear combination of the previous force increment vectors,  $\Delta\check{\mathbf{f}}_1, \Delta\check{\mathbf{f}}_2, \dots, \Delta\check{\mathbf{f}}_n$ .

$$\mathbf{p}_{n+1} = c_1 \Delta\check{\mathbf{f}}_1 + c_2 \Delta\check{\mathbf{f}}_2 + \dots + c_n \Delta\check{\mathbf{f}}_n \quad (3.26)$$

Now then, the vector  $\mathbf{F}^{Act} \mathbf{p}_{n+1}$  is

$$\mathbf{F}^{Act} \mathbf{p}_{n+1} = c_1 \mathbf{F}^{Act} \Delta\check{\mathbf{f}}_1 + c_2 \mathbf{F}^{Act} \Delta\check{\mathbf{f}}_2 + \dots + c_n \mathbf{F}^{Act} \Delta\check{\mathbf{f}}_n \quad (3.27)$$

From Eq. 3.22,  $\mathbf{F}^{Act} \Delta\check{\mathbf{f}}_1, \mathbf{F}^{Act} \Delta\check{\mathbf{f}}_2, \dots, \mathbf{F}^{Act} \Delta\check{\mathbf{f}}_n$  are estimated using the previous measured incremental displacements  $\Delta\check{\mathbf{u}}_1, \Delta\check{\mathbf{u}}_2, \dots, \Delta\check{\mathbf{u}}_n$ . Solving for the constants  $c_1, c_2, \dots, c_n$  using Eq. 3.25 becomes a least squares problem. Using  $c_1, c_2, \dots, c_n$  and substituting  $\mathbf{F}^{Act} \Delta\check{\mathbf{f}}_1, \mathbf{F}^{Act} \Delta\check{\mathbf{f}}_2, \dots, \mathbf{F}^{Act} \Delta\check{\mathbf{f}}_n$  with  $\Delta\check{\mathbf{u}}_1, \Delta\check{\mathbf{u}}_2, \dots, \Delta\check{\mathbf{u}}_n$ ,  $\mathbf{F}^{Act} \mathbf{p}_{n+1}$  is calculated through Eq. 3.27. The least square residual  $\mathbf{v}_{n+1}$  is then

$$\mathbf{v}_{n+1} = \Delta\check{\mathbf{u}}_{n+1} - \mathbf{F}^{Act} \mathbf{p}_{n+1} \quad (3.28)$$

Eq. 3.24 is rewritten in terms of  $\mathbf{v}_{n+1}$  as

$$0 = \mathbf{v}_{n+1} - \mathbf{F}^{Act} \mathbf{q}_{n+1} \quad (3.29)$$

by substituting  $\check{\mathbf{u}}_{n+1} - \mathbf{F}^{Act} \mathbf{p}_{n+1}$  for  $\mathbf{v}_{n+1}$ .  $\mathbf{q}_{n+1}$  is solved using

$$\mathbf{q}_{n+1} = \mathbf{A} \mathbf{v}_{n+1} \quad (3.30)$$

The initial stiffness matrix  $\mathbf{K}_0^{Act}$  is a reasonable substitute for  $\mathbf{A}$  since it aptly relates the deformation  $\mathbf{v}_{n+1}$  to the force  $\mathbf{q}_{n+1}$ . Finally, the incremental trial force  $\Delta \check{\mathbf{f}}_{n+1}$  is found via Eq. 3.23.

The only time the stiffness matrix is used is in Eq. 3.30. Since  $\mathbf{K}_0^{Act}$  is used, this bypasses the need for estimating  $\mathbf{K}_t^{Act}$ . However if a good estimate of  $\mathbf{K}_t^{Act}$  exists, it can be used as well.

### 3.2.2 Equilibrium method

In static structural analysis, the displacement method is commonly used for analysis where the nodal deformations are solved from the given stiffness matrix and nodal forces. This is a boundary value problem. The displacement method in static analysis also works well with the time integration schemes that are used for hybrid simulation to solve the equation of motion. There also exists a less commonly used method in static structural analysis called the force method. In the force method, the element forces are solved directly from the given flexibility matrix and nodal forces. Thus, the force method becomes a logical starting point for formulating a method to calculate the trial force for FC hybrid simulation.

The force method is used to determine the element forces of a statically indeterminate structure. This method starts with the static equilibrium equation

$$\mathbf{r} = \mathbf{B} \hat{\mathbf{r}} = \mathbf{f} - \mathbf{f}_w \quad (3.31)$$

where  $\mathbf{B}$  is the equilibrium matrix that relates the element resisting forces  $\hat{\mathbf{r}}$  to the nodal resisting forces  $\mathbf{r}$ .  $\mathbf{f}$  and  $\mathbf{f}_w$  are the applied nodal loads and equivalent element loads respectively. The equivalent element loads are typically caused by distributed loading, thermal loading and initial element deformations. The element resisting forces in Eq. 3.32 are written as a combination of the primary system  $\mathbf{B}_i(\mathbf{f} - \mathbf{f}_w)$  and the redundant system  $\mathbf{B}_x \hat{\mathbf{r}}_x$ .

$$\hat{\mathbf{r}} = \mathbf{B}_i(\mathbf{f} - \mathbf{f}_w) + \mathbf{B}_x \hat{\mathbf{r}}_x \quad (3.32)$$

The primary system is a stable statically determinate structure. It is created from the original structure by introducing cuts into certain elements. The resisting forces in the cut elements are the redundant forces  $\hat{\mathbf{r}}_x$ . In the case where the entire structure is statically determinate,  $\mathbf{B}_x \hat{\mathbf{r}}_x = \mathbf{0}$ .  $\mathbf{B}_i$  is the force influence matrix that determine the relationship between the element forces of the primary structure ( $\hat{\mathbf{r}}_p$ ) and the nodal loading forces,  $\hat{\mathbf{r}}_p = \mathbf{B}_i(\mathbf{f} - \mathbf{f}_w)$ .  $\mathbf{B}_x$  is the redundant force influence matrix where each column consists of the element forces from a loading of each unit redundant force.

$\hat{\mathbf{r}}_p$  creates a gap ( $\mathbf{d}_x$ ) in the cut element. The key to the force method is to close this gap by enforcing

$$\mathbf{0} = \mathbf{B}_x^T \mathbf{v} \quad (3.33)$$

Substituting Eq. 3.32 into the constitutive relationship

$$\mathbf{v} = f(\hat{\mathbf{r}}) + \mathbf{v}_0 \quad (3.34)$$

where  $\mathbf{v}$  is the element deformations as a function of the element forces  $\hat{\mathbf{f}}$  and the initial element deformations ( $\mathbf{v}_0$ ), Eq. 3.34 becomes

$$\mathbf{v} = f\left(\mathbf{B}_i(\mathbf{f} - \mathbf{f}_w) + \mathbf{B}_x \hat{\mathbf{f}}_x\right) + \mathbf{v}_0 \quad (3.35)$$

Eq. 3.35 is now substituted into Eq. 3.33 and becomes

$$\mathbf{0} = \mathbf{B}_x^T \left[ f\left(\mathbf{B}_i(\mathbf{f} - \mathbf{f}_w) + \mathbf{B}_x \hat{\mathbf{f}}_x\right) + \mathbf{v}_0 \right] \quad (3.36)$$

and in the linear case  $\mathbf{v} = \mathbf{F}_s \hat{\mathbf{f}} + \mathbf{v}_0$  where  $\mathbf{F}_s$  is the unconnected flexibility matrix of the elements, and Eq. 3.36 becomes

$$\mathbf{0} = \mathbf{B}_x^T \left[ \mathbf{F}_s \left( \mathbf{B}_i(\mathbf{f} - \mathbf{f}_w) + \mathbf{B}_x \hat{\mathbf{f}}_x \right) + \mathbf{v}_0 \right] \quad (3.37)$$

and

$$\mathbf{0} = \mathbf{B}_x^T [\mathbf{F}_s(\hat{\mathbf{f}}_p) + \mathbf{v}_0] + (\mathbf{B}_x^T \mathbf{F}_s \mathbf{B}_x) \hat{\mathbf{f}}_x \quad (3.38)$$

and

$$\mathbf{0} = \mathbf{d}_x + \mathbf{F}_x \hat{\mathbf{f}}_x \quad (3.39)$$

where  $\mathbf{d}_x = \mathbf{B}_x^T (\mathbf{F}_s(\hat{\mathbf{f}}_p) + \mathbf{v}_0)$  and  $\mathbf{F}_x = \mathbf{B}_x^T \mathbf{F}_s \mathbf{B}_x$ . The only unknown in Eq. 3.39 is  $\hat{\mathbf{f}}_x$ . Once  $\hat{\mathbf{f}}_x$  is found, the element forces  $\hat{\mathbf{f}}$  is computed from Eq. 3.32. Then using the constitutive relationship (Eq. 3.34), the element deformations ( $\mathbf{v}$ ) is calculated. The compatibility relationship in Eq. 3.40 gives the nodal deformations for given element deformations.

$$\mathbf{u} = \mathbf{B}_i^T \mathbf{v} \quad (3.40)$$

Patnaik [19] developed a variation of the force method called the integrated force method. The integrated force method treats the element forces as independent variables and does not divide the element forces into primary and redundant element forces. It solves for  $\hat{\mathbf{f}}$  directly. Patnaik's formulation is shown in Eq. 3.41. He uses Eq. 3.31 for the top rows. In Eq. 3.33,  $\mathbf{v}$  substituted with Eq. 3.34 to become the bottom rows.

$$\begin{bmatrix} \mathbf{B} \hat{\mathbf{f}} \\ \mathbf{B}_x^T [f(\hat{\mathbf{f}}) + \mathbf{v}_0] \end{bmatrix} = \begin{bmatrix} \mathbf{f} - \mathbf{f}_w \\ \mathbf{0} \end{bmatrix} \quad (3.41)$$

Eq. 3.41 can simplified to

$$F(\hat{\mathbf{f}}) = \mathbf{f}^* \quad (3.42)$$

where  $F(\hat{\mathbf{f}})$  is a vector function of  $\hat{\mathbf{f}}$  and  $\mathbf{f}^* = \begin{bmatrix} \mathbf{f} - \mathbf{f}_w \\ -\mathbf{B}_x^T \mathbf{v}_0 \end{bmatrix}$ . In the linear case,  $F(\hat{\mathbf{f}}) = \begin{bmatrix} \mathbf{B} \\ -\mathbf{B}_x^T \mathbf{F}_s \end{bmatrix} \hat{\mathbf{f}}$ .

$\begin{bmatrix} \mathbf{B} \\ -\mathbf{B}_x^T \mathbf{F}_s \end{bmatrix}$  is a square non-singular matrix provided that the structure is stable, and  $\hat{\mathbf{f}}$  can be solved directly. In the nonlinear case, the element forces can be solved by setting up the following residual function

$$R(\hat{\mathbf{f}}) = F(\hat{\mathbf{f}}) - \mathbf{f}^* \quad (3.43)$$

Newton's method can be utilized to solve for the zero of  $R(\hat{\mathbf{r}})$  iteratively.

$$\left[ \frac{\partial R(\hat{\mathbf{r}}^i)}{\partial \hat{\mathbf{r}}^i} \right] \Delta \hat{\mathbf{r}}^i = -R(\hat{\mathbf{r}}^i) \quad (3.44)$$

In Eq. 3.44,  $i$  designates the iteration number,  $\Delta \hat{\mathbf{r}}^i = \hat{\mathbf{r}}^{i+1} - \hat{\mathbf{r}}^i$ , and  $\frac{\partial R(\hat{\mathbf{r}}^i)}{\partial \hat{\mathbf{r}}^i} = \begin{bmatrix} \mathbf{B} \\ \mathbf{B}_x^T \frac{\partial f(\hat{\mathbf{r}}^i)}{\partial \hat{\mathbf{r}}^i} \end{bmatrix}$ .  $\Delta \hat{\mathbf{r}}^i$  is

solved for Eq. 3.44.  $\frac{\partial f(\hat{\mathbf{r}}^i)}{\partial \hat{\mathbf{r}}^i}$  in Eq. 3.44 is the unconnected tangent flexibility matrix. Subsequently,  $\hat{\mathbf{r}}^{i+1}$  is found for the next iteration,  $i + 1$ . This method continues to iterate until a certain tolerance criteria is reached.

The integrated force method provides a way to develop a force-based time integration scheme to solve the equation of motion. Starting with Eq. 3.1,  $\mathbf{r}_n$  can be substituted with  $F(\hat{\mathbf{r}}_n)$ , resulting in the following equation of motion

$$\begin{bmatrix} \mathbf{M} \\ \mathbf{0} \end{bmatrix} \ddot{\mathbf{u}}_n + \begin{bmatrix} \mathbf{C} \\ \mathbf{0} \end{bmatrix} \dot{\mathbf{u}}_n + \begin{bmatrix} \mathbf{B} \hat{\mathbf{r}}_n \\ \mathbf{B}_x^T [f(\hat{\mathbf{r}}_n) + \mathbf{v}_0] \end{bmatrix} = \begin{bmatrix} \mathbf{f}_n \\ \mathbf{0} \end{bmatrix} \quad (3.45)$$

With  $\mathbf{M}_b = \begin{bmatrix} \mathbf{M} \\ \mathbf{0} \end{bmatrix}$ ,  $\mathbf{C}_b = \begin{bmatrix} \mathbf{C} \\ \mathbf{0} \end{bmatrix}$ ,  $\mathbf{r}_n^* = \begin{bmatrix} \mathbf{B} \hat{\mathbf{r}}_n \\ \mathbf{B}_x^T [f(\hat{\mathbf{r}}_n) + \mathbf{v}_0] \end{bmatrix}$  and  $\mathbf{f}_n^* = \begin{bmatrix} \mathbf{f}_n \\ \mathbf{0} \end{bmatrix}$ , Eq. 3.45 is reformulated into the residual function

$$R(\hat{\mathbf{r}}_n) = \mathbf{M}_b \ddot{\mathbf{u}}_n + \mathbf{C}_b \dot{\mathbf{u}}_n + \mathbf{r}_n^* - \mathbf{f}_n^* \quad (3.46)$$

The Newmark approximations (Equations 2.2 and 2.4) are rearranged into equations for  $\ddot{\mathbf{u}}_n$  and  $\dot{\mathbf{u}}_n$ .

$$\ddot{\mathbf{u}}_n = \frac{1}{\beta(\Delta t)^2} (\mathbf{u}_n - \mathbf{u}_{n-1}) - \frac{1}{\beta \Delta t} \dot{\mathbf{u}}_{n-1} - \left( \frac{1}{2\beta} - 1 \right) \ddot{\mathbf{u}}_{n-1} \quad (3.47)$$

$$\dot{\mathbf{u}}_n = \frac{\gamma}{\beta \Delta t} (\mathbf{u}_n - \mathbf{u}_{n-1}) - \left( \frac{\gamma}{\beta} - 1 \right) \dot{\mathbf{u}}_{n-1} - \Delta t \left( \frac{\gamma}{2\beta} - 1 \right) \ddot{\mathbf{u}}_{n-1} \quad (3.48)$$

Substituting Eq. 3.34 into Eq. 3.40 for  $\mathbf{v}$ ,  $\mathbf{u}$  can be written in terms of  $\hat{\mathbf{r}}$ . This equation is then discretized at time step  $n$  and substituted into Equations 3.47 and 3.48 for  $\mathbf{u}_n$ . Then Equations 3.47 and 3.48 can be substituted into Eq. 3.46 for  $\ddot{\mathbf{u}}$  and  $\dot{\mathbf{u}}$  respectively, resulting in Eq. 3.49.

$$\begin{aligned} R(\hat{\mathbf{r}}_n) = \mathbf{M}_b & \left[ \frac{1}{\beta(\Delta t)^2} \left( \mathbf{B}_i^T (f(\hat{\mathbf{r}}_n) + \mathbf{v}_0) - \mathbf{u}_{n-1} \right) - \frac{1}{\beta \Delta t} \dot{\mathbf{u}}_{n-1} - \left( \frac{1}{2\beta} - 1 \right) \ddot{\mathbf{u}}_{n-1} \right] \\ & + \mathbf{C}_b \left[ \frac{\gamma}{\beta \Delta t} \left( \mathbf{B}_i^T (f(\hat{\mathbf{r}}_n) + \mathbf{v}_0) - \mathbf{u}_{n-1} \right) - \left( \frac{\gamma}{\beta} - 1 \right) \dot{\mathbf{u}}_{n-1} \right. \\ & \quad \left. - \Delta t \left( \frac{\gamma}{2\beta} - 1 \right) \ddot{\mathbf{u}}_{n-1} \right] + \begin{bmatrix} \mathbf{B} \hat{\mathbf{r}}_n \\ \mathbf{B}_x^T [f(\hat{\mathbf{r}}_n) + \mathbf{v}_0] \end{bmatrix} - \mathbf{f}_n^* \quad (3.49) \end{aligned}$$

The zero of Eq. 3.49 can be solved by the Newton's method using a similar equation to Eq. 3.44.

$$\left[ \frac{\partial R(\hat{\mathbf{r}}_n)}{\partial \hat{\mathbf{r}}_n} \right] \Delta \hat{\mathbf{r}}_n^i = -R(\hat{\mathbf{r}}_n^i) \quad (3.50)$$

Eq. 3.50 is formulated at time step  $n$  at iteration  $i$  with the Jacobian

$$\frac{\partial R(\hat{\mathbf{r}}_n^i)}{\partial \hat{\mathbf{r}}_n^i} = \mathbf{M}_b \frac{1}{\beta(\Delta t)^2} \left( \mathbf{B}_i^T \frac{\partial f(\hat{\mathbf{r}}_n^i)}{\partial \hat{\mathbf{r}}_n^i} \right) + \mathbf{C}_b \frac{\gamma}{\beta \Delta t} \left( \mathbf{B}_i^T \frac{\partial f(\hat{\mathbf{r}}_n^i)}{\partial \hat{\mathbf{r}}_n^i} \right) + \left[ \begin{array}{c} \mathbf{B} \\ \mathbf{B}_x^T \frac{\partial f(\hat{\mathbf{r}}_n^i)}{\partial \hat{\mathbf{r}}_n^i} \end{array} \right] \quad (3.51)$$

$\hat{\mathbf{r}}_n^i$  is the trial force vector that is transformed to the actuator coordinate system and applied to the specimen. The deformations are measured, transformed and fed back to this the integration scheme as  $f(\hat{\mathbf{r}}_n^i)$  in Eq. 3.49 to calculate the residual at iteration  $i$ . This method continues to iterate until a certain convergence criteria is met. Then it advances to time step  $n + 1$  and continues until it reaches  $N$ , the number of recorded ground acceleration points.

### 3.2.3 Summary of methods for acquiring the trial forces

The methods presented in this section for acquiring the trial forces were categorized into two types, compatibility and equilibrium methods, based on their formulation. The compatibility methods were further divided into two types: tangent-based (Broyden, BFGS, Intrinsic and Transpose) and Krylov subspace-based compatibilities methods. Both compatibility and equilibrium methods have their advantages and disadvantages. One type's advantage is consequently the other's disadvantage. The main advantage of the compatibilities methods is that they can be used with any computational driver that can run DC hybrid simulation. The conversion of the trial deformations to trial forces can be performed outside the computational driver. The next section shows the implementation of such methods in the OpenFresco middleware. The equilibrium method can not be used with most existing FEA software since most FEA programs used the displacement-method to solve the equation of motion. Therefore, a custom force-based FEA program is required, which limits complexity and size of the analytical portions of the simulation.

However, the advantage of the equilibrium method is that the trial forces are calculated directly, without the need of any type of conversions. The tangent stiffness matrix estimation, which is sensitive to noise from DAQ system, is not required. The initial flexibility matrix can be used in the Jacobian (Eq. 3.51). The method itself is more native to FC hybrid simulation since the entire solution algorithm is force-based. The remain sections describe the implementation and verification of these methods in the *nees@berkeley* lab.

## 3.3 Implementation

The implementations of the methods proposed in Section 3.2 are presented in this section. These implementation involve development in various software packages. The compatibility methods are developed in the OpenFresco software framework environment as a new abstract class. A small-scale custom force-based FEA software package is programmed in Matlab to implement the equilibrium method. Mathworks SimuLink and Stateflow predictor-corrector models are created to interface the MTS-STC control system with the OpenFresco middleware. A new experimental control concrete class is programmed in OpenFresco to act as the interface to the FC Mathworks models. All these components are necessary and interact with each other to run FC hybrid simulation.

### 3.3.1 Implementation of compatibility methods

The compatibility methods convert the trial deformations received from the computational driver to trial forces. Figure 3.4 shows this implementation. The computational driver sends the

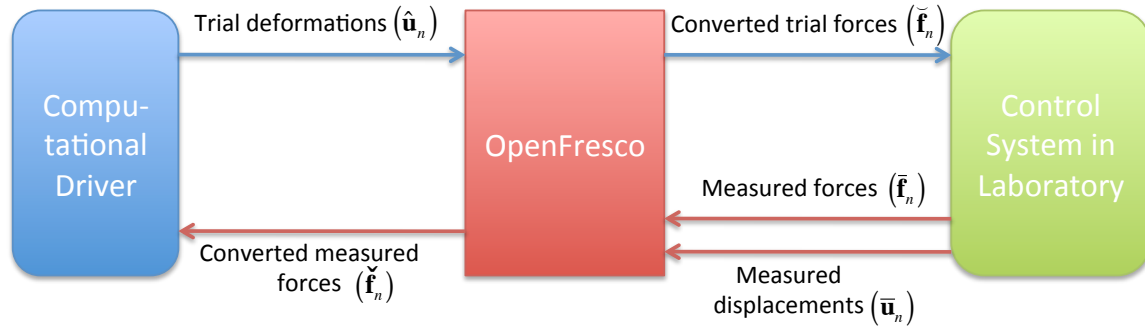


Figure 3.4: The implementation of compatibility method using OpenFresco.

trial deformations and receives the corresponding measured forces as in DC hybrid simulation. This is very similar to processes presented in Section 2.5. Steps 1 to 3 are the same as in Section 2.5. In Step 4, the experimental control object converts the trial displacements ( $\check{\mathbf{u}}_n$ ) to the converted trial forces ( $\check{\mathbf{f}}_n$ ) and sends  $\check{\mathbf{f}}_n$  to the control system. The control system imposes the  $\check{\mathbf{f}}_n$  and send  $\bar{\mathbf{u}}_n$  and  $\bar{\mathbf{f}}_n$  pair back to OpenFresco (Steps 5 and 6). In Step 7, the experimental control object reads the measured pair, converts  $\bar{\mathbf{u}}_n$  into the converted measure forces ( $\bar{\mathbf{f}}_n$ ) and sends it to the experimental setup. Steps 8 to 10 remain the same. There are two conversions that occur in this implementation: 1) trial deformations ( $\check{\mathbf{u}}_n$ ) to trial forces ( $\check{\mathbf{f}}_n$ ) and 2) measured deformations ( $\bar{\mathbf{u}}_n$ ) to measured forces ( $\bar{\mathbf{f}}_n$ ). All DC computational drivers that are supported by OpenFresco can be used with this implementation for FC hybrid simulation, since the computational driver is performing the same actions as in DC hybrid simulation.

In Figure 3.4, OpenFresco converts deformations to forces. The OpenFresco *ExperimentalSignalFilter* (*ESF*) class is extended to handle this conversion. The *ESF* class can modify signals received from the *Experimental Control* class. The relationship between the *ExperimentalControl* and *ESF* classes is shown in Figure A.2. The *ExperimentalControl* class can have zero to many *Experimental Signal Filters*.

Figure 3.5 shows the modified version of the *ESF* interface. converting public methods were added for the implementation of the compatibility methods on OpenFresco. Operator overloading is used in converting. `converting (Vector* td)` has the argument `td`, the trial displacement vector ( $\check{\mathbf{u}}_n$ ). It returns the address of the converted trial forces vector ( $\check{\mathbf{f}}_n$ ). `converting (Vector* dd, Vector* df)` has the arguments ( $\bar{\mathbf{u}}_n$ ) and ( $\bar{\mathbf{f}}_n$ ). It uses these arguments to make the necessary updates for the next time `converting (Vector* td)` is called. *Experimental Control* method `setTrialResponse` calls `converting (Vector* td)` method to convert the trial displacement vector to trial force vector. *Experimental Control* method `getDaqResponse` calls `converting (Vector* td, Vector* df)` method to convert the measured displacement vector to the converted force vector.

Two new concrete classes and one abstract class are implemented to extend the capabilities of the *ESF* class for FC hybrid simulation. Both inherent from the *ESF* abstract class as shown

```

class ExperimentalSignalFilter : public TaggedObject
{
public :
    // constructors
    ExperimentalSignalFilter(int tag);
    ExperimentalSignalFilter(const
        ExperimentalSignalFilter& esf);

    // method to get class type
    virtual const char *getClassType() const;

    // destructor
    virtual ~ExperimentalSignalFilter();

    virtual double filtering(double data) = 0;
    virtual Vector& converting(Vector* td) = 0;
    virtual Vector& converting(Vector* dd, Vector* df) = 0;
    virtual int setSize(const int sz) = 0;
    virtual void update() = 0;
    virtual Matrix& getTangStiffMat();

    virtual ExperimentalSignalFilter *getCopy() = 0;

    // public methods for experimental signal filter recorder
    virtual Response *setResponse(const char **argv, int argc,
        OPS_Stream &output);
    virtual int getResponse(int responseID, Information &info);
};

```

Figure 3.5: Interface for the OpenFresco *ExperimentalSignalFilter* abstract class.

in Figure 3.6. ESFTangForceConverter class converts displacements to forces using the tangent stiffness matrix. Figure A.4 shows ESFTangForceConverter class definition. ESFTangForceConverter constructor takes **int** tag (an unique tag), Matrix& initStif (a reference to the initial tangent stiffness matrix) and ExperimentalTangentStiff\* tangStif (a pointer to ExperimentalTangentStiff) as its arguments. **int** tag and Matrix& initStif are supplied by the user. It uses the *ExperimentalTangentStiff* class to estimate the tangent stiffness matrix of the specimen. The ESFTangForceConverter object must have one *ExperimentalTangentStiff* object. The ESFTangForceConverter converting(Vector\* td, Vector\* df) method calls updateTangentStiff method to update the tangent stiffness matrix using the measured displacement-force pair before transforming  $\bar{\mathbf{u}}_n$  to  $\bar{\mathbf{f}}_n$ . The ESFTangForceConverter class uses Eq. 3.11 to convert the deformations to forces.

The methods for estimating the tangent stiffness matrix from Section 3.2.1.1 are implemented as concrete classes of *ExperimentalTangentStiff* class. ETBroyden, ETBfgs, ETTranspose and ETIntrinsic classes deploy the Broyden, BFGS, Transpose and Intrinsic methods respectively in

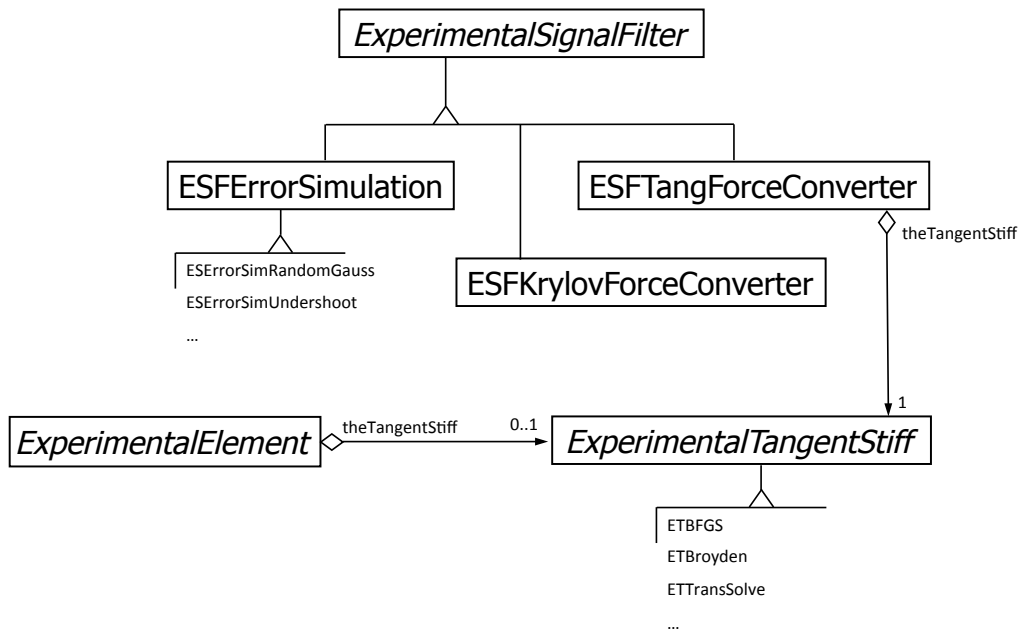


Figure 3.6: OpenFresco UML class diagram with ESFTangForceConverter, ESFKrylovForceConverter and ExperimentalTangentStiff classes.

OpenFresco. Making *ExperimentalTangentStiff* a separate class and not a subclass of *ESFTangForceConverter* allows *ExperimentalTangentStiff* to be more versatile. This decoupled architecture allows the tangent stiffness estimation to be used by the *ExperimentalElement* class to estimate the tangent stiffness matrix of the experimental element. In the original version of OpenFresco, the *ExperimentalElement* class always returns the initial stiffness matrix, which is supplied by the user, even when the method for obtaining tangent stiffness matrix is called. Using the tangent stiffness matrix over the initial stiffness matrix give better results. In fact, the Intrinsic and Transpose methods were developed for this purpose, to provide an estimation of the tangent stiffness matrix of the specimen to improve the results during DC hybrid simulation.

Figure 3.6 shows the relationship between *ExperimentalElement*, *ESFTangForceConverter* and *ExperimentalTangentStiff*. *ExperimentalElement* has 0 to 1 *ExperimentalTangentStiff* and *ESFTangForceConverter* must have one *ExperimentalTangentStiff*. The software design pattern that is used for the implementation of *ExperimentalTangentStiff* class is called Strategy or Policy pattern[8]. This pattern defines a family of algorithms that estimate the tangent stiffness matrix, encapsulates each algorithm (Broyden, BFGS, Transpose and Intrinsic) and makes them interchangeable. These are algorithms can vary independently of the clients that call them. In Figure 3.6, the *ExperimentalElement* and *ESFTangForceConverter* classes can call different tangent stiffness matrix estimating algorithm through the *ExperimentalTangentStiff* interface. *ExperimentalTangentStiff* does not depend on what is calling it.

The other class in Figure 3.6 is the *ESFKrylovForceConverter* concrete class. Figures A.5 show the *ESFKrylovForceConverter* class definition. This class deploys the Krylov subspace-based compatibility method presented in Section 3.2.1.2. The constructor takes tag (an unique



```

class ExperimentalTangentStiff : public TaggedObject
{
public :
    // constructors
    ExperimentalTangentStiff(int tag);
    ExperimentalTangentStiff(const
        ExperimentalTangentStiff& ets);

    // method to get class type
    virtual const char *getClassType() const;

    // destructor
    virtual ~ExperimentalTangentStiff();

    virtual Matrix& updateTangentStiff(const Vector* disp ,
                                        const Vector* vel ,
                                        const Vector* accel ,
                                        const Vector* force ,
                                        const Vector* time ,
                                        const Matrix* kInit ,
                                        const Matrix* kPrev) = 0;

    virtual ExperimentalTangentStiff *getCopy() = 0;

    // public methods for experimental signal filter recorder
    virtual Response *setResponse(const char **argv , int argc ,
        OPS_Stream &output);
    virtual int getResponse(int responseID , Information &info);
};

```

Figure 3.7: Interface for the OpenFresco *ExperimentalTangentStiff* abstract class.

tag), *ss* (the number of subspaces) and *initStif* (the initial stiffness matrix) as its arguments. These arguments are supplied by the user. The converting method is overloaded very much like the *ESFTangForceConverter* class. *converting*(*Vector\** *td*) uses the Krylov subspaces of measured incremental force vectors to calculate the next incremental trial force vector. When the *ss* defined by the user to be greater than or equal to the trial displacement vector size, the least square method is used to solve Eq. 3.25. When *ss* is less than the size of the force vector, Eq. 3.25 is solved by the method of Lagrange multipliers [13]. The *converting*(*Vector\** *dd*, *Vector\** *df*) method updates the subspaces with the incremental measured pair to be used next time.

*ESFTangForceConverter* and *ESFKrylovForceConverter* have filters that prevent noises from causing spurious updates. The incremental measured displacement and force vectors are filtered in the *updateIncreMat* method of *ESFTangConverter* and prevent unnecessary updates to the tangent

stiffness matrix. In `ESFKrylovForceConverter`, the filtering of incremental measured displacement and force vectors ensures that noises from the control system do not populate the Krylov subspace. These filters take the 1 – Norm or 2 – Norm of the measured pair vectors. If these norms are both greater than the user-defined limit, the updates are made.

### 3.3.2 Implementation of equilibrium method

The implementation of the Equilibrium method proposed in Section 3.2.2 are presented in this section. The Equilibrium method requires a computational driver that has a force-based time integration scheme. Not many commercial FEA software packages use force-based time integration schemes. Therefore, a basic FEA software package is programmed using Matlab. The purpose of this software package is to demonstrate the validity of the equilibrium method.

This software package consists of three main Matlab functions. The first function is the `createModel()` function. The hybrid model is hard coded into this function. The following model properties are defined in this function: mass matrix ( $\mathbf{M}$ ), viscous damping matrix ( $\mathbf{C}$ ) stiffness matrix ( $\mathbf{K}$ ), compatibility matrix ( $\mathbf{A}$ ), equilibrium matrices ( $\mathbf{B}$ ,  $\mathbf{B}_i$  and  $\mathbf{B}_f$ ) and material types. Only the 1-D truss element is available. There are however multiple 1-D material types. These material types can also double as elements if the geometric effects are factor into the material properties. All materials are both displacement- and force-based materials. The materials return a force value for given a trial deformation and a deformation value for a given trial force. This allows this software package to perform switch control hybrid simulation using the equilibrium method. This is discussed in Chapter 4. The following materials are available:

- Elastic material - This is a simple linearly elastic material. Its input is the elastic modulus ( $E$ ).
- Bilinear elastic material - This material exhibits bilinear behavior. It is linearly elastic until it reaches a predefined yield point. Then the linear stiffness changes. It has three inputs:  $E$  - elastic modulus,  $f_y$  - yield force and  $b$  - post-yield modulus ratio where the post-yield modulus  $E_y = b * E$ .
- Bilinear elastic RDM material - This element is the same as the bilinear elastic material, but it adds noise to the return values. This is to simulate the noise in the control system. It has the same input arguments as the bilinear elastic material.
- Bilinear hysteretic material - This element is similar to the bilinear elastic element, but it exhibits hysteretic behavior. It has the same input arguments as the bilinear elastic material.
- Experimental material/element - This element interfaces with OpenFresco. It has three inputs:  $E$  - initial stiffness,  $ipAddr$  - IP address of the machine running the OpenFresco Simulation Application Server and  $ipPort$  - IP port number of the machine running the OpenFresco Simulation Application Server. It uses TCP sockets to communicate with OpenFresco server.

The second function is the `initializeAnalysis(deltaT)` function. This function has an input argument of `deltaT`, which is the integration time step defined by the user. The FC methods

are defined in this function. It has various force-based time integration schemes available that use the equilibrium method. The following are the time integration scheme available for FC hybrid simulation:

- `FM_NLDynamicNR` - This time integration scheme implements the equilibrium method from Section 3.2.2 using the Newton's method with a predefined tolerance on the residual and maximum number of iterations. If the residual is not within in the tolerance limit after reaching the maximum number of iterations, the algorithm stops the simulation.
- `FM_NLDynamicNRLimit` - This is similar to `FM_NLDynamicNR`. The only difference is that it scales the trial forces using the increment limit which is a predefined variable. The purpose of scaling the trial force to produce a smooth trial force curve with the Newton's method. Without scaling, the Newton's method tends to produces trial force curves that oscillate because of the Newton's method to converge quadratically. The quadratic convergence attribute is desirable in most purely numerical simulations but not in hybrid simulation.
- `FM_NLDynamicNRwFixIter` - This is the similar to `FM_NLDynamicNR`. The differences is that it does only does fixed number of iterations. At each iteration, it scales the trial forces according the following scaling factor

$$sf = \frac{1}{maxIter - iter + 1} \quad (3.52)$$

where the *maxiter* is the maximum number of iteration and *iter* is the current iteration. The time integration scheme does not check to see if the residual is within the tolerance limit.

- `FM_NMDynamicExplicit` - This is the Newmark explicit time integration algorithm with  $\beta = 0$ . It was programmed into this software package to test the single-conversion implementation of the compatibility methods as presented in the previous section.
- `FM_aOSDynamicPC` - This is the  $\alpha$ OS predictor-corrector algorithm. It was programmed into this software package to test the single-conversion implementation of the compatibility methods as presented in the previous section.

The time-integrations parameters along with convergence criteria including the tolerance, maximum number of iterations and increment limit are set in this function as well.

The last function is the `main` function. This is the gateway function into the software package. It creates the model by calling the `createModel()` function and initializes the simulation by calling the `initializeAnalysis(deltaT)` function. The recorded ground acceleration is also loaded in `main` function. It starts the simulation using the time-integration method defined in `initializeAnalysis(deltaT)` function. It records all the data from the simulation. After the completion of the simulation, it plots the figures using the recorded data. The figures include such things as the time-history plots, the element hysteresis plots, the trial force plots.

### 3.3.3 Simulink/Stateflow model

A FC predictor-corrector Simulink model is required to bridge the clock-speeds between the integrator loop and the servo-control loop. A FC predictor-corrector Simulink model is similar to the DC Simulink models described in Section: 2.3. The main difference is that it corrects and predicts forces instead of displacements. As Figure 3.8 shows the signal coming from the xPC Hybrid Controller (F1F1) is written as command forces in SCRAMNet memory. xPC Target Hybrid

#### Hybrid Controller

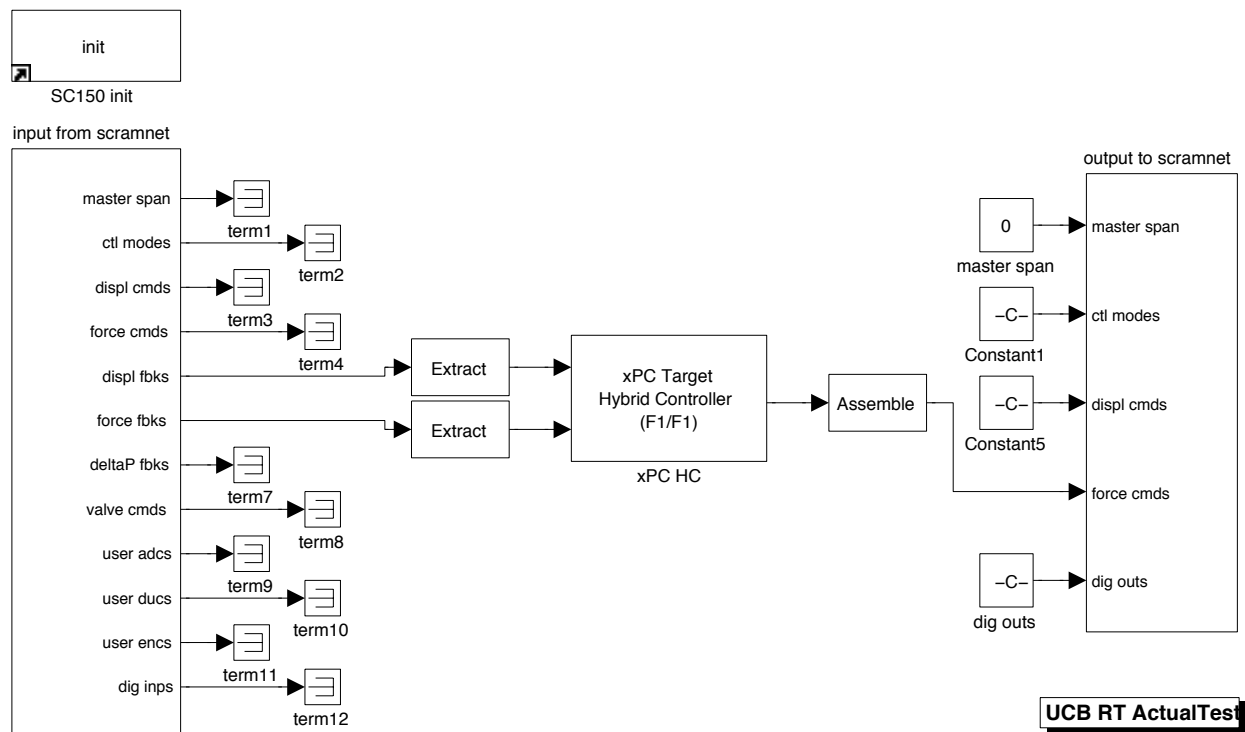


Figure 3.8: SimuLink model for FC hybrid simulation that uses xPC Target real-time workshop with SCRAMNet at *nees@berkeley* lab.

Controller (F1F1) serves many of the same functions as its displacement counterpart. Shellengberg's dissertation [20] provides more detailed information on xPC Target Hybrid Controller for DC. There are two important differences to note in Figure 3.9. First, xPC Target Hybrid Controller receives target forces (*targFrc*) from OpenFresco and not target displacements. Second, the feedback displacement (*dspIn*) and force (*frcIn*) are being offset in the Signal Offset box and being averaged in the Moving Average Filter box before being relayed back to OpenFresco. This is done to compensate for the noise in the load cell and position transducers. The offset or the number of measurements to average is determined by the user.

The StateFlow model in Figure 3.10 is also similar to its displacement counterpart except that it now predicts and corrects for forces and not displacements.

The following C functions are deployed for this purpose:



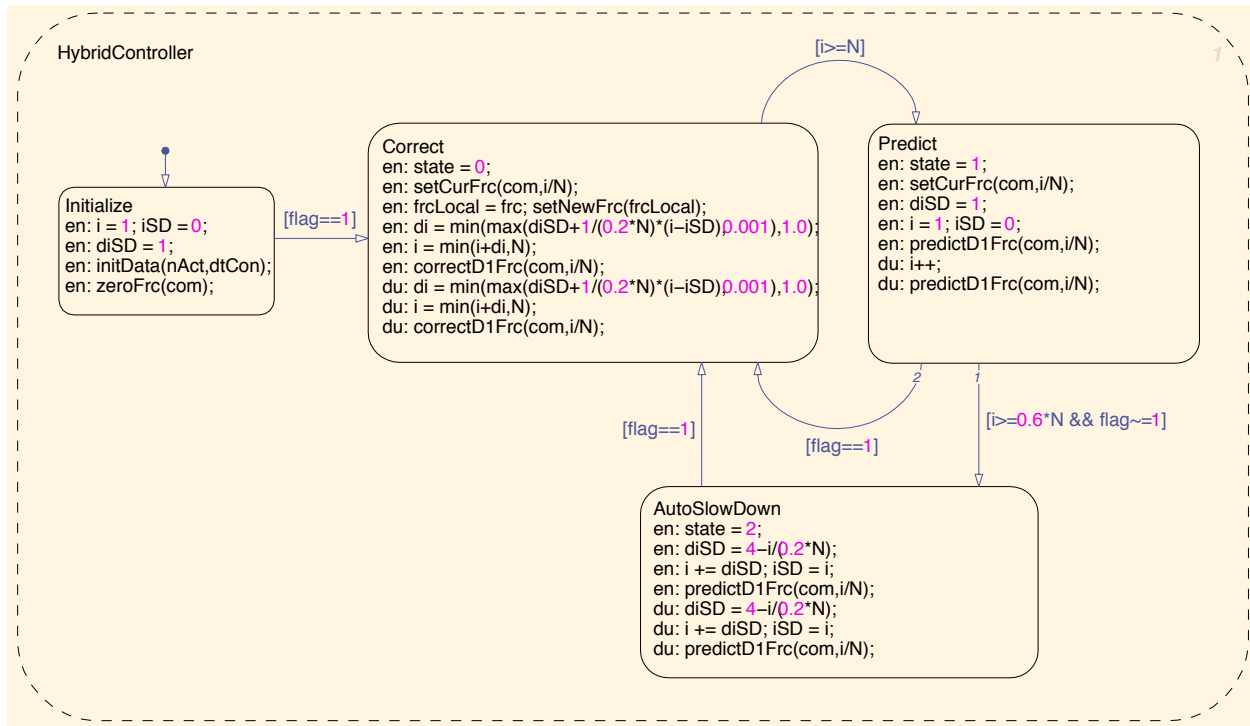


Figure 3.10: The StateFlow predictor-corrector model is represented by the Predictor-Corrector box in Figure 3.9. It predicts and corrects for forces. It has an auto slow down mode to prevent the model from over predicting.

receive the next target force before 60% of the simulation time step has elapsed. In the AutoSlow-Down state, the command force are predicted at a progressively slower rate until eventually it just holds a command force until the next target force comes in.

The simulation time step is set to 0.10 second for all FC hybrid simulation tests. This means that the Simulink model is running at 0.10 second for each integration time step. The control system runs at 1024 Hz. Therefore, there are 102 cycles in a simulation time step. 102 is the variable N in Figure 3.10. 102 cycles give 0.0996 second. Thus, the simulation time step is more accurately 0.0996 second. This is quite slow compared to the integration time step. As discussed later in this chapter, most time integration schemes used for FC hybrid simulation have 20 substeps, which results in an integration time step of 0.001 second. The simulation time step is 100 times greater than the integration time step. Thus, it is considered a slow test.

### 3.3.4 OpenFresco force experimental control

The target forces required by the Simulink and Stateflow models are generated by the Open-Fresco experimental control class. This class interfaces with the control system through the xPC Hybrid Controller (F1F1) model (Figure 3.9). The ECxPCtargetForce experimental control class is programmed for this purpose. The class definition is shown in Figure A.6. The target forces are sent to the Simulink model in the control private method. The acquire private method read the

measured displacement and force pair from the MTS-STC control system when the target force has been reached. `ECxPCtargetForce` can have a 1 or more `ExperimentalSignalFilter` as shown in class diagram in Figure A.3. Both the compatibility and equilibrium methods use this experimental control to communicate with the control system. If a CFC method is deployed, it uses `TangentForceConverter` to convert displacements to forces. If the EFC method is used, it just passes the forces to the control system.

## 3.4 Experimental results and verification

This section contains the experimental results of the FC hybrid simulation methods presented in this chapter. The  $\mu$ -NEES experimental setup at *nees@berkeley* lab is used. The  $\mu$ -NEES setup is configured in two ways, 1-DOF and 2-DOF configurations. With each configuration, two types of hybrid simulation are conducted. The first type is the linear hybrid simulation where the input ground motion is scaled to keep the specimen in the linear range. The second is the nonlinear hybrid simulation where the ground motion is scaled to yield the specimen to the nonlinear region. Different time integration schemes are deployed for both type of hybrid simulations.

The results are analyzed to determine the performance of the FC methods. The FC experimental results are compared to the DC experiment results. The control system errors between command and feedback signals are compared. The trial values sent to the control system are examined for excessive oscillation. The displacement and force simulation results recorded by the FEA program are compared to the numerical results.

### 3.4.1 1-DOF $\mu$ -NEES experimental setup and results

The 1-DOF  $\mu$ -NEES experimental setup is shown in Figures 3.11 and 3.12. The actuator is positioned at the lowest possible point in the setup to create a stiff setup, which is the ideal setup for FC hybrid simulation. Stiff systems are difficult to setup. Typically, these systems experience higher forces for relatively small displacements. This means that any slippage in the system can alter the results adversely. Slippage occurs in the connections. For this setup, slippage occurs at the clevis pin, actuator pin connecting the actuator to the specimen and the actuator connection to the reaction frame. These connections were tightened with thin metal shims to limit excessive slippage.

The coupons are placed into the clevis (Figure 3.13) in such a way to prevent unnecessary lateral movement. In Figure 3.13, the right coupon is tightened at the bottom and the top using nuts. With the left couple, only the bottom is tightened while the top is loose. Nuts are placed at the top of the left coupon leaving a gap as seen in Figure 3.13. This is to prevent damage to the actuator and the setup when the actuator is in FC mode. If the actuator receives a command force of 10 kips, and the coupons break while the actuator is trying to impose 10 kips to the specimen, the actuator may keep moving indefinitely, trying to reach 10 kips. With this coupon configuration, once the right coupon breaks during the hybrid simulation, the left coupon will catch the specimen. These two coupons are mirrored on the other side of the clevis.

The proportional gain of the actuator in FC mode for this setup is about 10 times less than the proportional gain of setup in DC mode. The tuning results of the setup are provided in Appendix

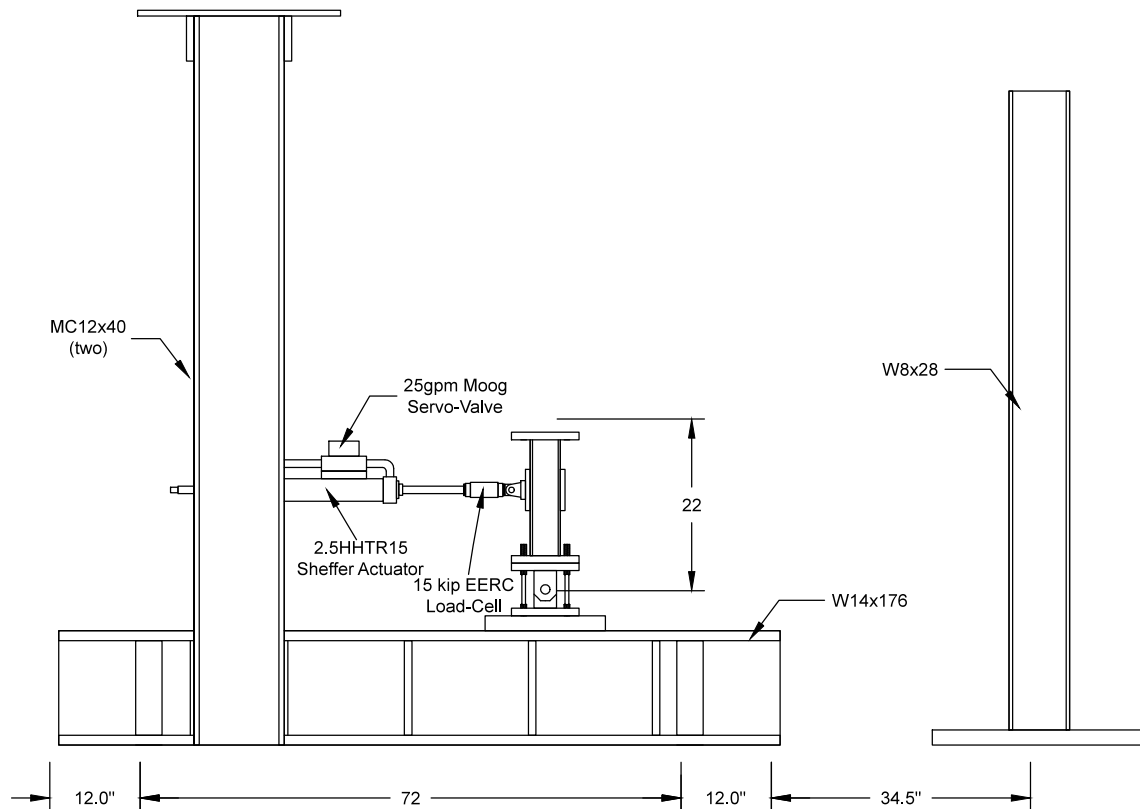


Figure 3.11: 1-DOF  $\mu$ -NEES experimental setup schematic at *nees@berkeley* lab. This setup is used for the 1-DOF DC, FC and SC hybrid simulation.

B. In FC mode, the proportional gain is dependent on the stiffness of the system. Higher stiffness usually drives the proportional gain down to keep the control system stable. If the stiffness of the setup changes during the experiment, for example if the specimen yields, the control system may become unstable in FC mode. However if the proportional gain is too low, it degrades the tracking performance. Tracking is how well the command and measured forces match. Thus, tuning the actuator in FC mode is difficult. The proportional gain has to be low enough to keep the control system stable in FC but not too low such that the tracking is unreliable. Friction and stick-and-slip can also cause problems in FC mode, but this actuator is not affected by them.

The FC hybrid simulation results are compared to the analytical results and the DC hybrid simulation results for each time integration scheme. Table 3.2 contains the four time integration schemes used and their parameters. The Newmark Explicit (NME) and  $\alpha$ OS from Section 2.2 methods are commonly used for DC hybrid simulation. The Newmark time integration method using a fixed number of iterations (NMF) and the Newmark method using a reduced increment (NMR) are less commonly used [20].



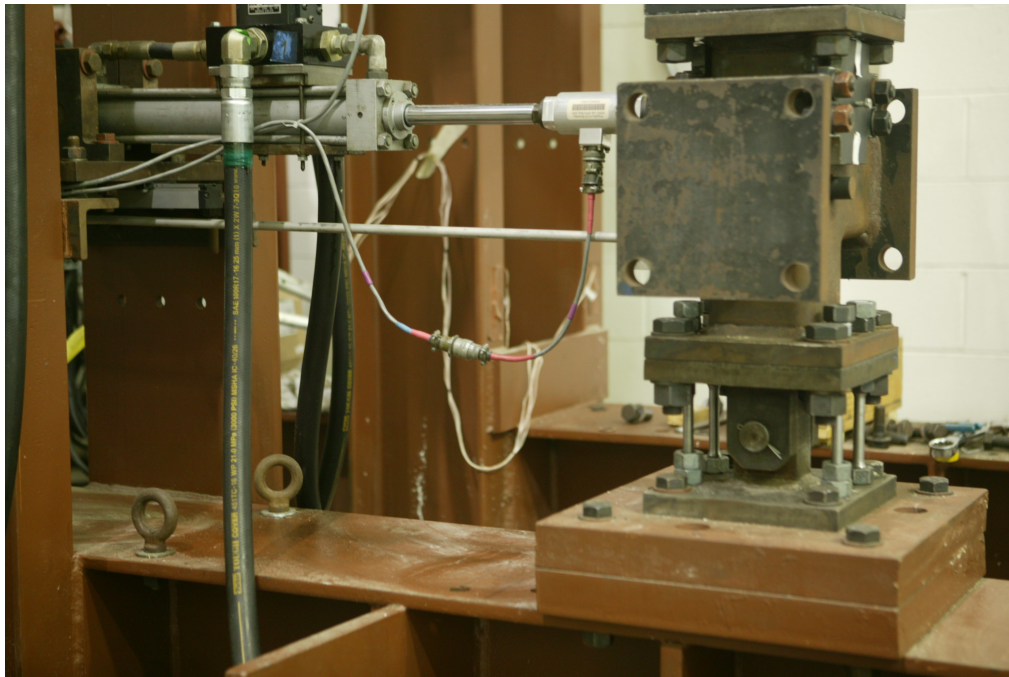


Figure 3.12: 1-DOF  $\mu$ -NEES experimental setup at *nees@berkeley* lab.



Figure 3.13: 1-DOF  $\mu$ -NEES experimental setup clevis and coupon configuration. Total of 4 steel coupons are used, two on each side.

Table 3.2: Time integration schemes and their parameters.

Time integration scheme	Parameters
Newmark Explicit (NME)	$\gamma = 1/2, substep = 20$
$\alpha$ Operator Splitting ( $\alpha$ OS)	$\alpha = 1.0, substep = 20$
Newmark Fixed Iteration (NMF)	$\gamma = 1/2, \beta = 1/4, iter_{fix} = 20$
Newmark Reduced Increment (NMR)	$\gamma = 1/2, \beta = 1/4, iter_{max} = 20, tol_{energy} = 1 \times 10^{-6}$

### 3.4.1.1 Hybrid model and OpenFresco configuration

Figure 2.1 shows the hybrid model used in conjunction with 1-DOF  $\mu$ -NEES experimental setup. Table 3.3 shows the properties of the hybrid model. The linear model is stiffer than the nonlinear model. Four coupons are engaged in the clevis (3.13) during the linear hybrid simulation while two are engaged in the nonlinear hybrid simulation. When all four coupons are engaged, the coupons will not yield even when the force levels reach the actuator force capacity of 12 kips. Therefore, only two coupons are used for the nonlinear simulation to yield the coupons. The four coupons configuration is only used for the 1-DOF linear hybrid simulation setup. The rest of the experiments for force control, switch control and mixed control hybrid simulation use the two coupon configuration.

Table 3.3: :Properties of one-bay-frame model for 1-DOF  $\mu$ -NEES setup

	Linear hybrid model	Nonlinear hybrid model
Elastic Stiffness (kip/in.): [Element 1   Element 2   Element 3]	[45   2   5]	[32   2   5]
Initial Tangent Stiffness Matrix (kip/in.)	$\begin{bmatrix} 47 & -2 \\ -2 & 7 \end{bmatrix}$	$\begin{bmatrix} 34 & -2 \\ -2 & 7 \end{bmatrix}$
Mass (kip/g): [ $Mass_1$ $Mass_2$ ]	[0.10   0.05]	[0.10   0.05]
Period (sec): [ $T_1$ $T_1$ ]	[0.54   0.20]	[0.54   0.34]
Mass Proportional Damping with $\xi = 5\%$ : [ $\alpha_M = 2\xi\omega_1$ ]	[1.17]	[1.17]

Element 1 in the hybrid model is the TwoNodeLink experimental element that communicates with OpenFresco. OpenSees is the FEA software. For DC hybrid simulations, OpenFresco configuration from Table 2.1 is deployed. The same OpenFresco configuration is used for FC hybrid simulation with one exception. The only difference is that xPC-Target Force experimental control from Section 3.3.4 is used instead of xPC-Target experimental control.

### 3.4.1.2 Linear FC hybrid simulation

OpenSees analysis is perform on the hybrid model to assess the quality of the experimental results. This numerical analysis uses the properties for the linear hybrid model shown in Table 3.3. It uses a linear numerical element for Element 1 to model the behavior of the 1-DOF  $\mu$ -NEES

setup. A separate OpenSees analysis is run for each of the following time integration schemes in Table 3.2 and compared against the results using the Newmark Implicit (NMI) scheme with the same parameters as the NMR scheme without the increment limit. The analyses use the first 400 recorded points of the El Centro ground motion scaled to 15% (Figure 3.1).

Figures 3.14 and 3.15 show the displacement-time history and the force-time history of Element 1 from the analyses of the model. Table 3.4 shows the minimum, mean and maximum values of the absolute displacement and force errors. These errors are calculated using the NMI result as the baseline. It can be seen from Figures 3.14 and 3.15 and Table 3.4 that all the schemes do not deviate much from NMI except NME. The displacement and force errors for  $\alpha$ OS, NMF and NMR are comparable while NME errors are approximately an order of magnitude larger.

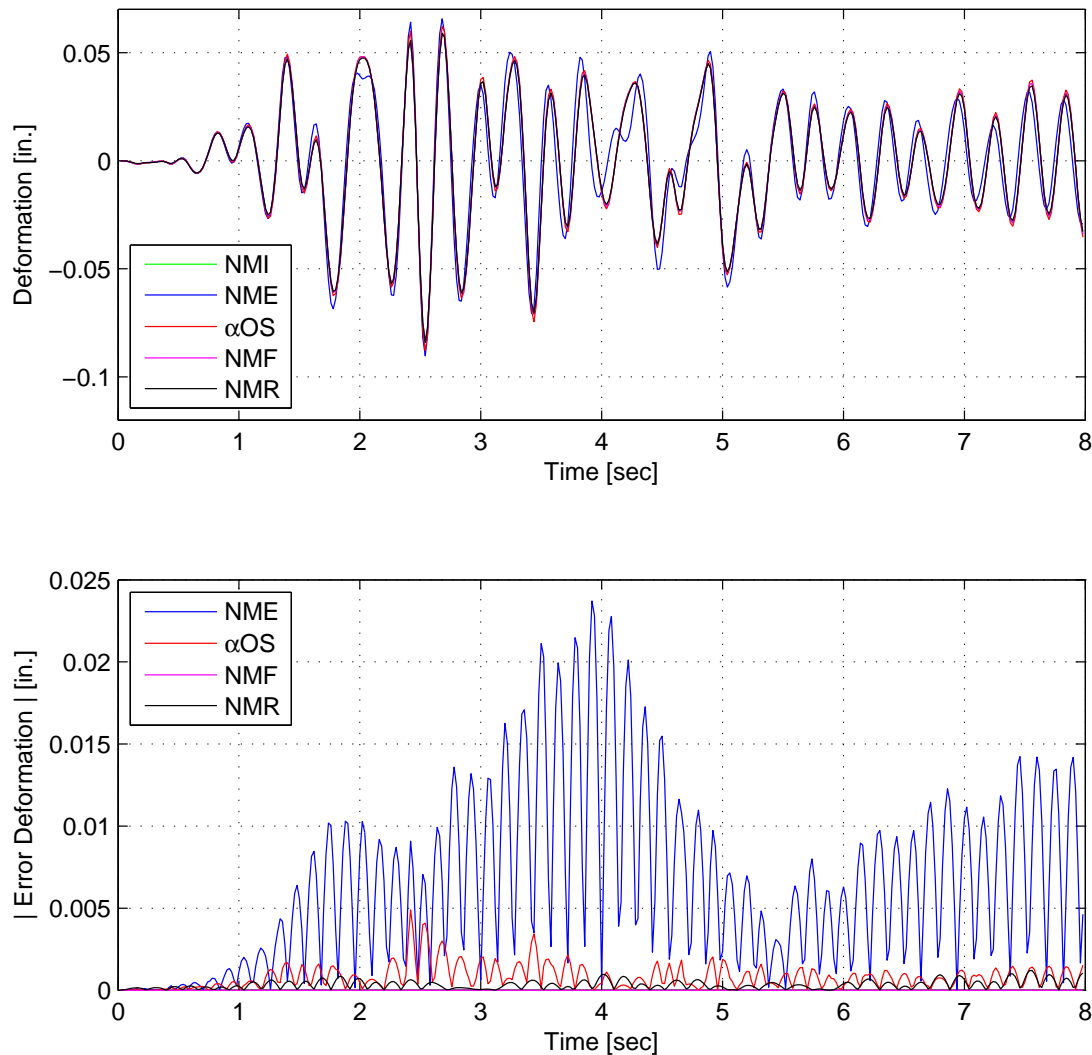


Figure 3.14: Linear numerical simulation results using NMI, NME,  $\alpha$ OS, NMF and NMI for one-bay-frame OpenSees model. Top: Element 1 deformation-time history plot. Bottom: Absolute displacement error from NMI.

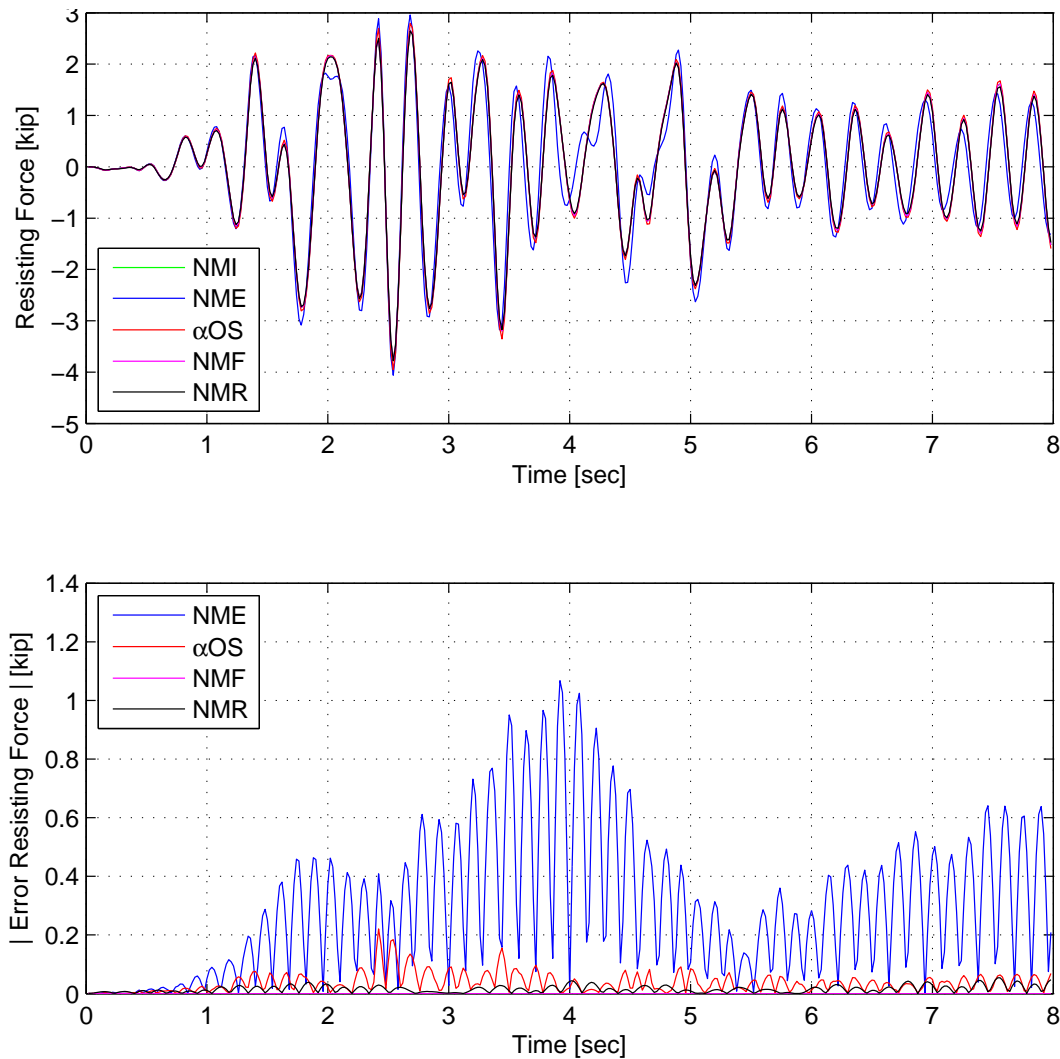


Figure 3.15: Linear numerical simulation results using NMI, NME,  $\alpha$ OS, NMF and NMI for one-bay-frame OpenSees model. Top: Element 1 force-time history plot. Bottom: Absolute force error from NMI.

Tables 3.5 and 3.6 show the absolute errors of the experimental results from DC and FC hybrid simulations. These errors are calculate from the OpenSees recorded values. The time-history plots and their errors are in Section C.1 of Appendix C. For each time integration scheme, the errors are calculated between the experimental results and their numerical counterpart from Figures 3.14 and 3.15, except for the NME method. The NME experimental results are compared to the NMI numerical results since the NME numerical results are not comparable to the other numerical results. Two type of FC methods are tested, CFC and EFC methods. EFC method is only tested for the NMR scheme because only NMR is consistently stable with EFC method. The other time integration schemes with EFC method are unstable or do not give reliable results.

The results from Tables 3.5 and 3.6 agree well with one another and the numerical results.

Table 3.4: Absolute errors of displacement (Figure 3.14) and force (Figure 3.15) results from linear numerical simulation using one-bay-frame OpenSees model.

<b>Control method</b>	<b>NME</b>	<b><math>\alpha</math>OS</b>	<b>NMF</b>	<b>NMR</b>
$ ErrD _{Min}$ (in.)	$1.95 \times 10^{-6}$	$9.10 \times 10^{-7}$	0	$1.00 \times 10^{-8}$
$ ErrD _{Mean}$ (in.)	0.0065	$8.14 \times 10^{-4}$	0	$3.30 \times 10^{-4}$
$ ErrD _{Max}$ (in.)	0.0237	0.0049	0	0.00120
$ ErrF _{Min}$ (kip)	$8.77 \times 10^{-5}$	$4.08 \times 10^{-5}$	0	0
$ ErrF _{Mean}$ (kip)	0.291	0.0366	0	0.0149
$ ErrF _{Max}$ (kip)	1.07	0.221	0	0.0539

The force errors are approximately equaled to the displacement errors multiplied by the linear stiffness of Element 1. This is expected since the specimen remained within the linear range. The main reason for running linear hybrid simulations is to check that all the equipment, specimen and software are functioning properly. The linear tests play a crucial role in detecting any software bugs and loose connections that are causing slippage in the system.

Table 3.5: Absolute errors of displacement results from linear hybrid simulations for the 1-DOF setup for various time integration schemes at the FEA level.

<b>Control method</b>	<b>NME (in.)</b>	<b><math>\alpha</math>OS (in.)</b>	<b>NMF (in.)</b>	<b>NMR (in.)</b>
DC-Min	$1.61 \times 10^{-5}$	0	$6.43 \times 10^{-6}$	$7.43 \times 10^{-6}$
DC-Mean	0.00632	0.00761	0.00583	0.00613
DC-Max	0.0226	0.0299	0.0248	0.0217
CFC-Min	$1.16 \times 10^{-5}$	0	$1.79 \times 10^{-5}$	$3.30 \times 10^{-6}$
CFC-Mean	0.00709	0.00893	0.00785	0.00823
CFC-Max	0.0227	0.0291	0.0292	0.0352
EFC-Min	N/A	N/A	N/A	$1.64 \times 10^{-5}$
EFC-Mean	N/A	N/A	N/A	0.00599
EFC-Max	N/A	N/A	N/A	0.0201

Table 3.7 shows the error at the control system level. The control system errors in DC and FC modes are similar to one another. The force errors are expected to be 45 times larger than the displacement errors. 45 is the value of the linear stiffness of the specimen. In that respect, the force errors are slightly less than the displacement errors for all time integration schemes. It is suspected that if the specimen is stiffer, then the force errors will be noticeably less than the displacement errors. It is however difficult to build a repeatable test setup, like the  $\mu$ -NEES setup that reaches that high level of stiffness. Figure C.13 contains the plot of the command forces of the EFC method using NMR. It shows an uneven and jagged command force curve, which is not desirable in hybrid simulation.

Table 3.6: Experimental absolute errors of force results from linear hybrid simulations for the 1-DOF setup for various time integration scheme at the FEA level.

Control method	NME (kip)	$\alpha$ OS (kip)	NMF (kip)	NMR (kip)
DC-Min	$5.03 \times 10^{-5}$	$5.32 \times 10^{-5}$	$4.88 \times 10^{-5}$	$8.90 \times 10^{-5}$
DC-Mean	0.295	0.342	0.265	0.2934
DC-Max	1.21	1.30	1.08	1.02
CFC-Min	$1.50 \times 10^{-4}$	0.00227	0.00116	$3.55 \times 10^{-4}$
CFC-Mean	0.340	0.406	0.361	0.358
CFC-Max	1.12	1.31	1.30	1.41
EFC-Min	N/A	N/A	N/A	$3.14 \times 10^{-4}$
EFC-Mean	N/A	N/A	N/A	0.256
EFC-Max	N/A	N/A	N/A	0.937

Table 3.7: Absolute errors of linear hybrid simulations for the 1-DOF setup for various time integration schemes at the control system level.

Control method	NME	$\alpha$ OS	NMF	NMR
DC-Min (in.)	$1.03 \times 10^{-10}$	$3.74 \times 10^{-10}$	$2.43 \times 10^{-9}$	$2.61 \times 10^{-10}$
DC-Mean (in.)	$4.14 \times 10^{-4}$	$3.81 \times 10^{-4}$	$5.07 \times 10^{-4}$	$4.88 \times 10^{-4}$
DC-Max (in.)	0.00254	0.00285	0.00303	0.00363
CFC-Min (kip)	$1.49 \times 10^{-8}$	$5.96 \times 10^{-8}$	$1.12 \times 10^{-8}$	0
CFC-Mean (kip)	0.0164	0.0456	0.0180	0.0160
CFC-Max (kip)	0.128	0.129	0.216	0.106
EFC-Min (kip)	N/A	N/A	N/A	$2.08 \times 10^{-7}$
EFC-Mean (kip)	N/A	N/A	N/A	0.0186
EFC-Max (kip)	N/A	N/A	N/A	0.0726

### 3.4.1.3 Nonlinear FC hybrid simulation

As with the linear hybrid simulation, OpenSees analyses are performed on the nonlinear hybrid model. The model properties of the nonlinear are shown in Table 3.3. Instead of using a simple linear element for Element 1, a more complicated model (Figure 3.16) is built in OpenSees to numerically simulate the 1-DOF  $\mu$ -NEES setup. All the elements of the OpenSees  $\mu$ -NEES model are elastic beam-column elements except for the red elements. The red elements simulate the coupons and are nonlinear force beam-column elements with a circular fiber section using the Steel02 OpenSees material. The red nodes are offset to simulate damage of the coupons. The top most node of this model is connected to Node 1 of the hybrid model. The El Centro ground motion is scaled to 40% to push the specimen into the nonlinear region but at the same time not exceeding the actuator load capacity of 12 kips.

The numerical results from OpenSees are shown in Figures 3.17 and 3.18 for NMF, NMR and NMI. The nonlinear OpenSees model of the 1-DOF  $\mu$ -NEES setup does not converge when using the NME and  $\alpha$ OS time integration schemes. Table 3.8 shows that the NMF numerical results

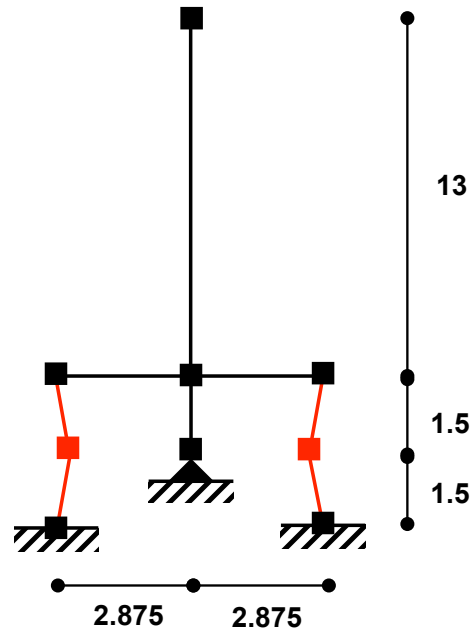


Figure 3.16: OpenSees model of the 1-DOF  $\mu$ -NEES experimental setup for numerical simulation.

better match the NMI results than the NMR numerical results match the NMI results. The NMR errors are two orders of magnitude higher than the NMF results.

Table 3.8: Absolute errors of displacement (Figure 3.17) and force (Figure 3.18) results from nonlinear numerical simulation using one-bay-frame OpenSees model.

Control method	NMF	NMR
$ ErrD _{Min}$ (in.)	0	$01.40 \times 10^{-6}$
$ ErrD _{Mean}$ (in.)	$5.88 \times 10^{-5}$	0.00442
$ ErrD _{Max}$ (in.)	$2.65 \times 10^{-4}$	0.0223
$ ErrF _{Min}$ (kip)	0	$4.38 \times 10^{-4}$
$ ErrF _{Mean}$ (kip)	0.00269	0.145
$ ErrF _{Max}$ (kip)	0.0105	0.487

The displacement errors (Table 3.9) and force errors (Table 3.10) show that the hybrid simulation results in general match the numerical results well. Only the NMF results are presented for the CFC method because NME,  $\alpha$ OS and NMR for the CFC method tripped the controller's force error interlock of 0.75 kips even after multiple attempts. The plots of the hybrid simulation results are given in Section C.2 of Appendix C. The errors are calculated from taking the difference for the hybrid simulation results from the numerical results for each time integration scheme and then taking the absolute value of that error. The error mean is approximately 6% to 8% of the span of the displacement and force levels for both DC and FC hybrid simulation tests. One method is not significantly better than the other, except when comparing just the NMF results. In this case,

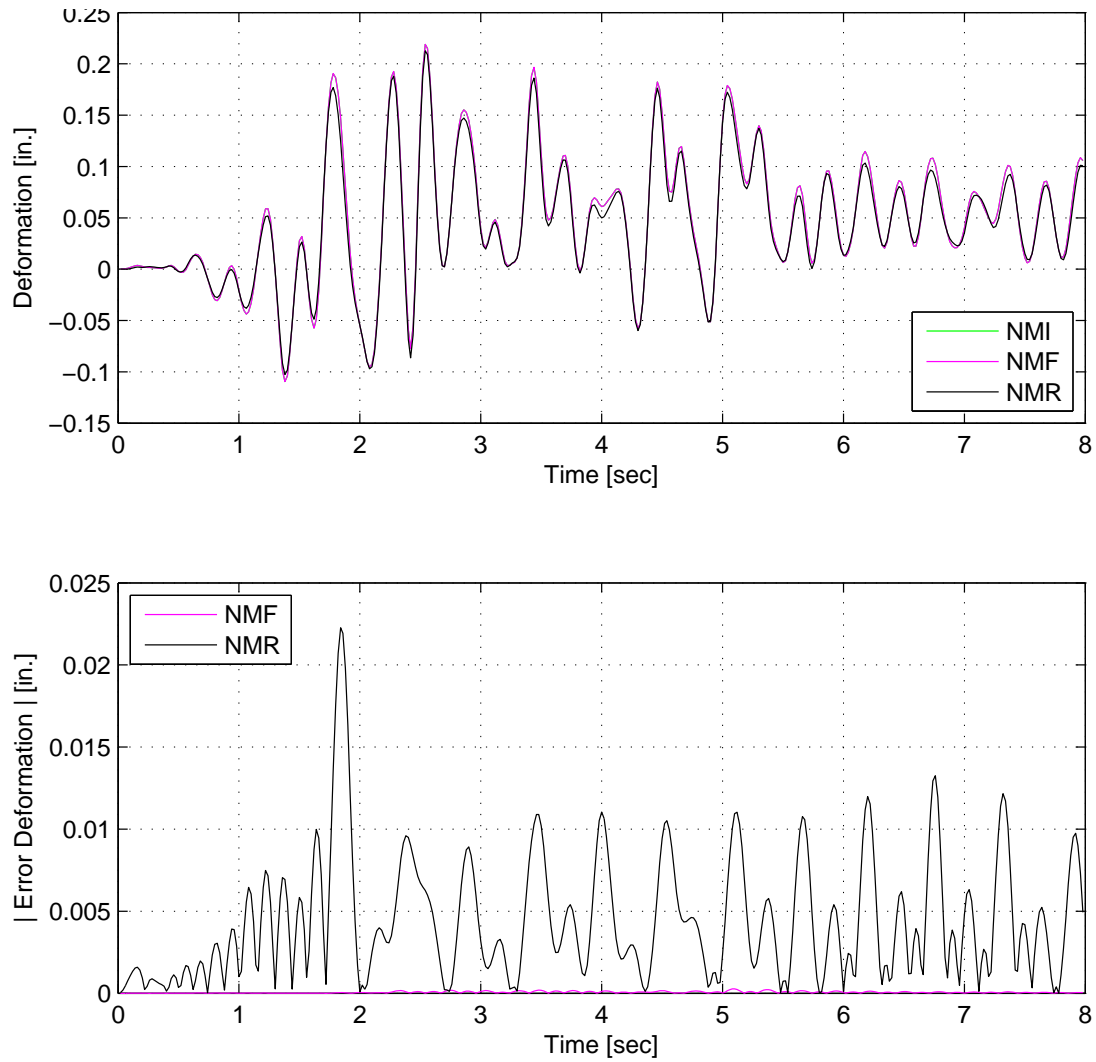


Figure 3.17: Nonlinear numerical simulation results using NMI, NMF and NMR for one-bay-frame OpenSees model. Top: Element 1 deformation-time history plot. Bottom: Absolute displacement error from NMI.



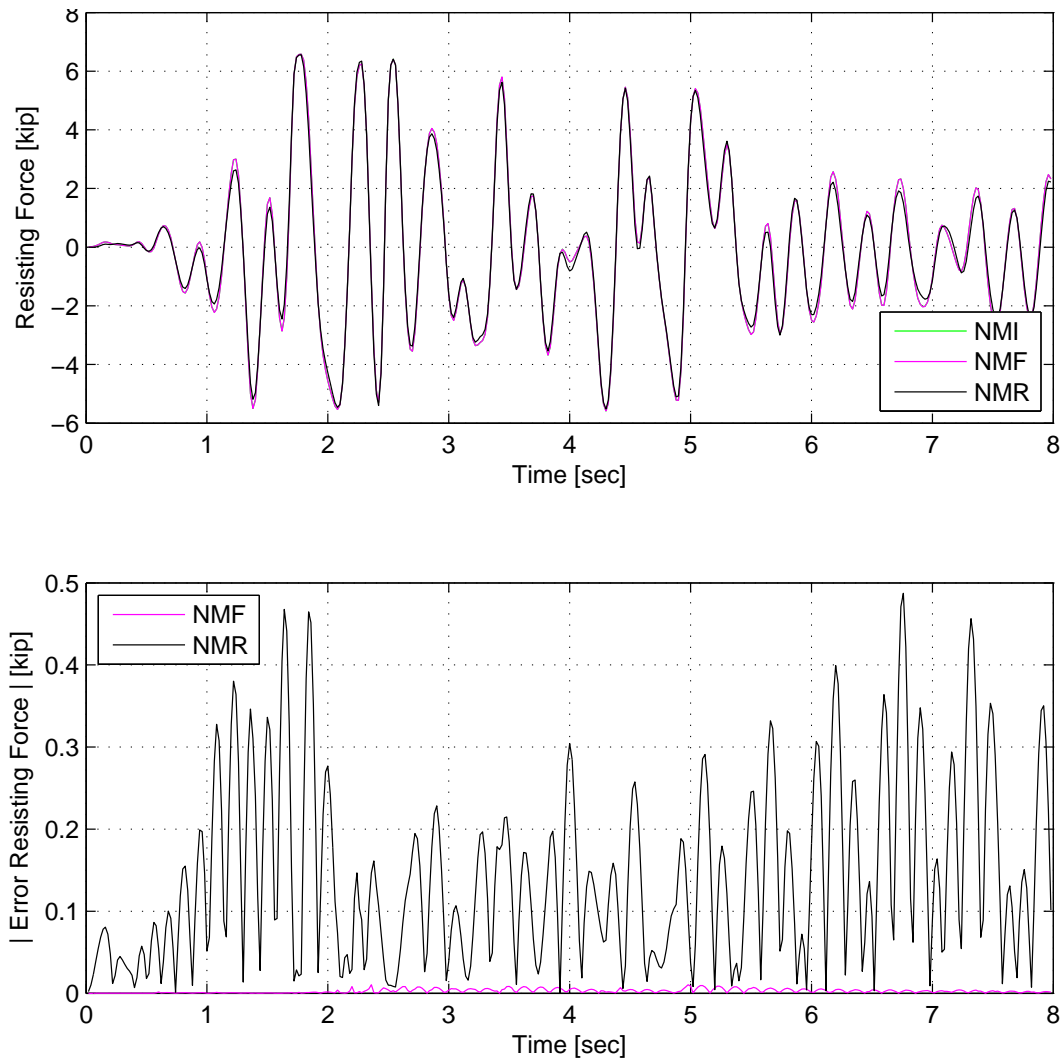


Figure 3.18: Nonlinear numerical simulation results using NMI, NMF and NMR for one-bay-frame OpenSees model. Top: Element 1 force-time history plot. Bottom: Absolute force error from NMI.

the CFC result is much better than DC results. It is worth noting that from the force-time history plots (Figures C.14 and C.17) that the second coupon in the clevis (Figure 3.13) was engaged. The force-time history curve shows the specimen softening and then later becoming stiffer. This effect is not modeled in the OpenSees  $\mu$ -NEES 1-DOF model.

Table 3.9: Absolute errors of displacement results from nonlinear hybrid simulations for the 1-DOF setup for various time integration method at the FEA level.

Control method	NME (in.)	$\alpha$ OS (in.)	NMF (in.)	NMR (in.)
DC-Min	$2.84 \times 10^{-5}$	$7.30 \times 10^{-7}$	$1.64 \times 10^{-5}$	$1.68 \times 10^{-5}$
DC-Mean	0.0266	0.0288	0.114	0.0711
DC-Max	0.116	0.118	0.357	0.253
CFC-Min	N/A	N/A	$1.14 \times 10^{-5}$	N/A
CFC-Mean	N/A	N/A	0.0409	N/A
CFC-Max	N/A	N/A	0.143	N/A
EFC-Min	N/A	N/A	N/A	$1.28 \times 10^{-4}$
EFC-Mean	N/A	N/A	N/A	0.0922
EFC-Max	N/A	N/A	N/A	0.280

Table 3.10: Absolute errors of force results from nonlinear hybrid simulations for the 1-DOF setup for various time integration method at the FEA level.

Control method	NME (kip)	$\alpha$ OS (kip)	NMF (kip)	NMR (kip)
DC-Min	0.00150	$9.60 \times 10^{-4}$	$5.20 \times 10^{-4}$	0.00972
DC-Mean	0.394	0.435	0.804	1.28
DC-Max	2.03	2.08	4.05	5.42
CFC-Min	N/A	N/A	$6.65 \times 10^{-4}$	N/A
CFC-Mean	N/A	N/A	0.593	N/A
CFC-Max	N/A	N/A	2.34	N/A
EFC-Min	N/A	N/A	N/A	0.00700
EFC-Mean	N/A	N/A	N/A	1.26
EFC-Max	N/A	N/A	N/A	5.77

Table 3.11 shows the error at the control system level. The errors are calculated from the command and measure values. It shows how well the MTS-STS control system is tracking the command values. The controller is tracking equally well in both control modes. Force errors are about 34 (initial stiffness of the specimen) times the displacement errors. Figure C.19 shows that the EFC method produces an undesirable command force curve for hybrid simulation.

Figure 3.19 displays the changes in the tangent stiffness calculation during the nonlinear hybrid simulation using NMF with the CFC method. The tangent stiffness values are calculated at each time step and not at each iteration between time steps. For the result in Figure 3.19, a filter of 0.005 is used for both displacement and force. This means that if the difference between the measured displacement and force values from the current time step and previous time step is less

Table 3.11: Absolute errors of nonlinear hybrid simulations for the 1-DOF setup for various time integration schemes at the control system level.

Control method	NME	$\alpha$ OS	NMF	NMR
DC-Min (in.)	$1.38 \times 10^{-9}$	$7.86 \times 10^{-9}$	$7.94 \times 10^{-9}$	$3.14 \times 10^{-9}$
DC-Mean (in.)	$8.53 \times 10^{-4}$	$8.20 \times 10^{-4}$	0.00167	$8.66 \times 10^{-4}$
DC-Max (in.)	0.00513	0.00526	0.0129	0.0181
CFC-Min (kip)	N/A	N/A	0	N/A
CFC-Mean (kip)	N/A	N/A	0.0328	N/A
CFC-Max (kip)	N/A	N/A	0.285	N/A
EFC-Min (kip)	N/A	N/A	N/A	0
EFC-Mean (kip)	N/A	N/A	N/A	0.0282
EFC-Max (kip)	N/A	N/A	N/A	0.640

than 0.005, the tangent stiffness value is not updated. The purpose of the filter is to prevent spurious updates to the tangent stiffness value. Figure 3.19 shows that the tangent updates are quite stable except for at the 2000 iteration mark when the specimen first enters the nonlinear region.

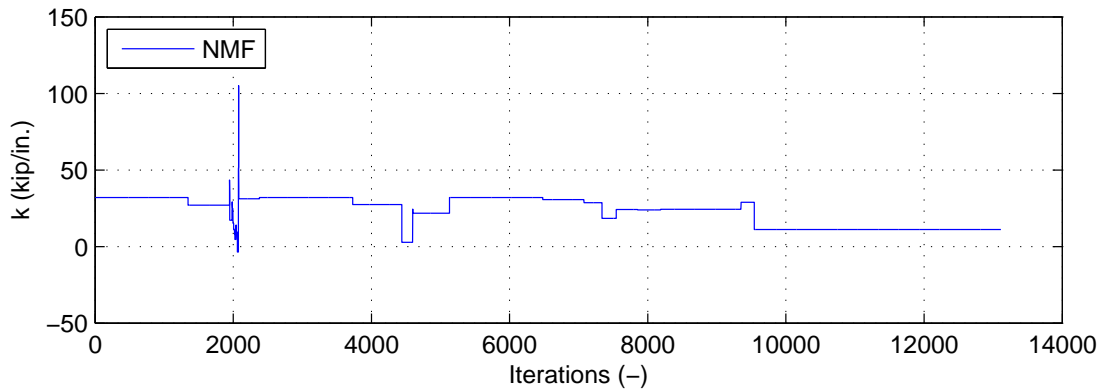


Figure 3.19: Estimated tangent stiffness values vs. number of iterations for the 1-DOF setup during the nonlinear hybrid simulation using NMF with the CFC method.

### 3.4.2 2-DOF $\mu$ -NEES experimental setup and results

The 2-DOF  $\mu$ -NEES experimental setup is configured to test the CFC methods. In the 1-DOF setup all CFC tangent methods degenerate to using the secant. The 2-DOF  $\mu$ -NEES experimental setup schematic is detailed in Figure 3.20. The photograph of the 2-DOF setup is shown in Figure 3.21. The 1-DOF setup (Figure 3.11) is modified by stacking another clevis and a S4x7.7 steel member on top. A second actuator is added to the reaction frame and attached to the s4x7.7 member. Two steel coupons are placed in the top clevis as shown in Figure 3.22. The bottom coupon configuration remains the same as the 1-DOF setup.

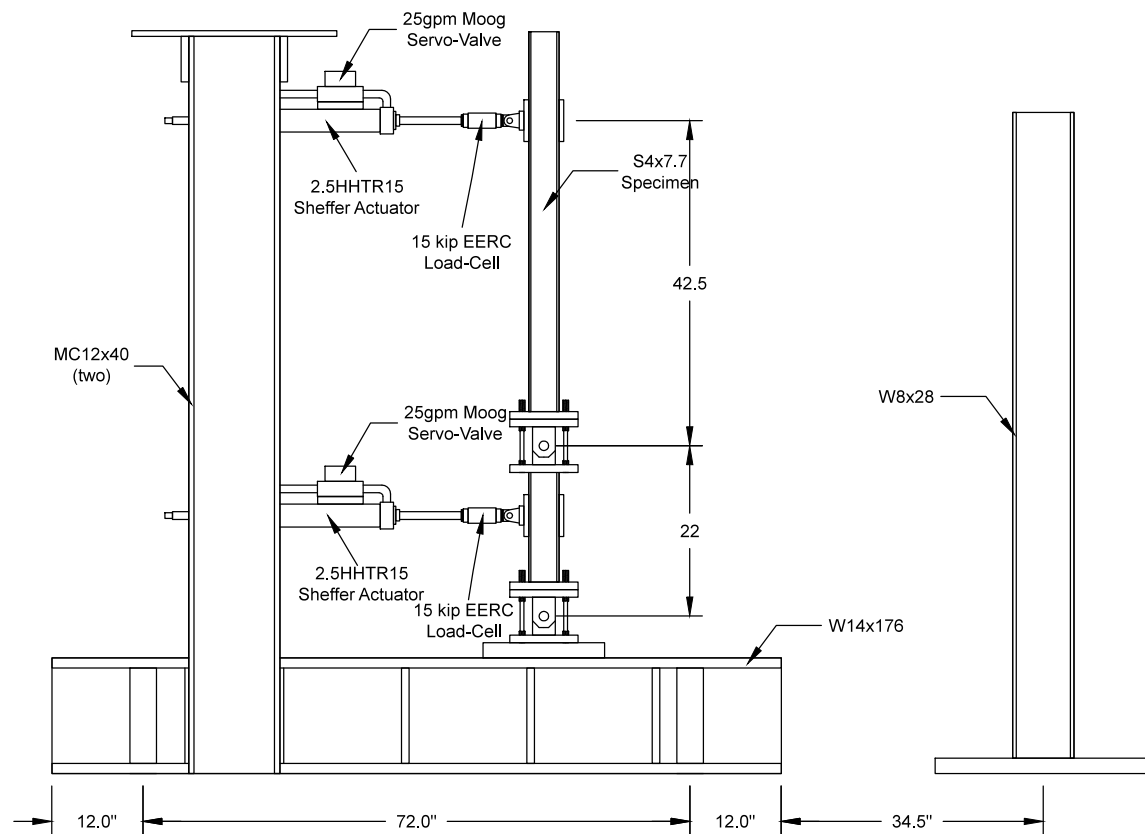


Figure 3.20: 2-DOF  $\mu$ -NEES experimental setup schematic at *nees@berkeley* lab. This setup is used for the 2-DOF DC, FC and MC hybrid simulation.

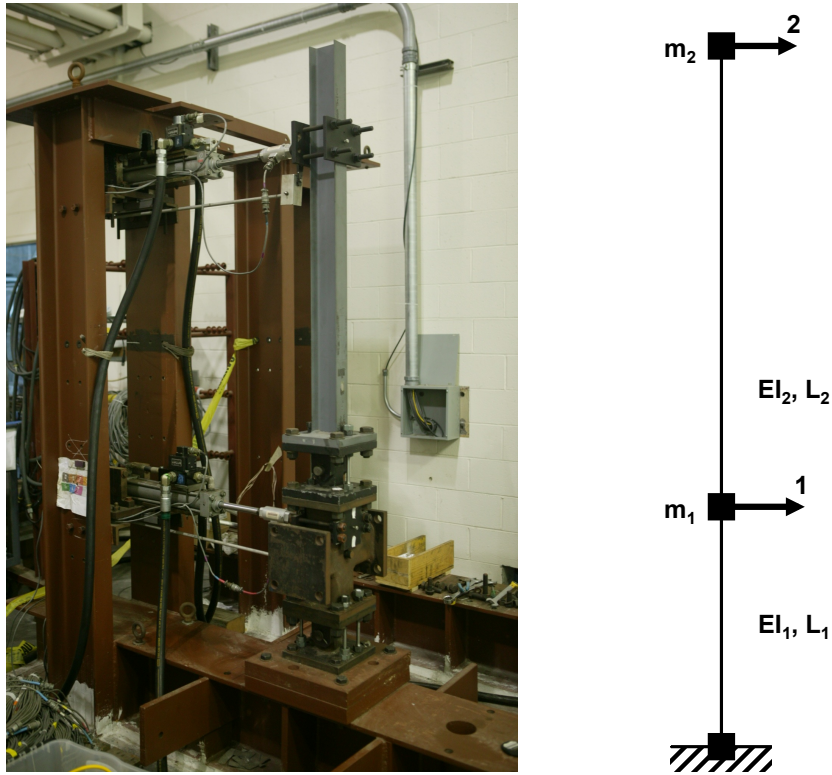


Figure 3.21: 2-DOF  $\mu$ -NEES experimental setup at *nees@berkeley* lab (left) and 2-DOF hybrid model (right).



Figure 3.22: 2-DOF  $\mu$ -NEES experimental setup clevis and coupon configuration. Total of 2 steel coupons are used, one on each side of the pin.

### 3.4.2.1 Hybrid model and OpenFresco configuration

The 2-DOF hybrid model in Figure 3.21 (right) is deployed for testing the FC methods. The lumped masses are placed on the top and bottom nodes. Only the mass and damping are modeled numerically. Mass proportional damping is used. The model properties are provided in Table 3.12. After the vertical and rotational DOFs are condensed out, only the two horizontal DOFs remain. The bottom support is fixed. Lumped mass values are selected such that the periods are similar to the 1-DOF hybrid model. The same time integration schemes and parameters from Table 3.2 are used.

Table 3.12: Properties of 2-DOF hybrid model for 2-DOF  $\mu$ -NEES setup.

	<b>2-DOF hybrid model</b>
Initial Tangent Stiffness Matrix (kip/in.)	$\begin{bmatrix} 81 & -12 \\ -12 & 3.4 \end{bmatrix}$
Mass (kip/g): $[Mass_1 \quad Mass_2]$	$[0.10 \quad 0.01]$
Period (sec): $[T_1 \quad T_1]$	$[0.55 \quad 0.20]$
Mass Proportional Damping with $\xi = 5\%$ : $[\alpha_M = 2\xi\omega_1]$	$[1.17]$

Table 3.13 contains the OpenFresco configuration for the 2-DOF setup. Generic experimental element allows the user to define arbitrary nodes and DOFs for each node. Two nodes are defined for 2-DOF setup, one node for each of nodes shown in Figure 3.21 (right) and 1 DOF in the horizontal direction for each node. The initial stiffness matrix is defined from Table 3.12. No Transformation experimental setup is deployed since no transformation is required between the 2-DOF hybrid model and the 2-DOF “ $\mu$ -NEES setup. All hybrid simulations are local simulations meaning that there is no network communication between OpenSees and OpenFresco.

Table 3.13: OpenFresco configuration for 2-DOF setup.

<i>Experimental Element:</i>	Generic experimental element
<i>Experimental Site:</i>	Local experimental site
<i>Experimental Setup:</i>	No Transformation experimental setup
<i>Experimental Control:</i>	xPC-Target Force experimental control

Numerical analyses are done in OpenSees using the 2-DOF OpenSees model (Figure 3.23). Both the linear and nonlinear results are compared to the FC hybrid simulation experimental results. It uses the similar elements as the 1-DOF OpenSees model in Figure 3.16. The red elements are identical to the nonlinear beam-column elements used in the 1-DOF OpenSees model and simulate the steel coupons. The rest of the model is made of elastic beam-column elements. The red nodes are offset to simulate coupon damage. Lumped masses are placed at the top most node and at the node between the two clevises, 16 units from the bottom.

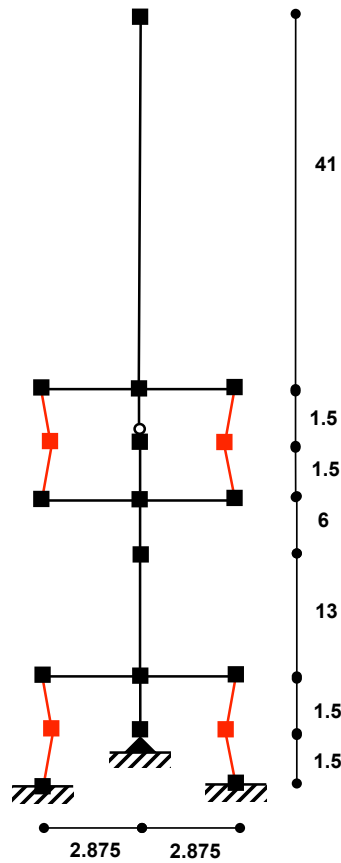


Figure 3.23: 2-DOF OpenSees numerical model for 2-DOF  $\mu$ -NEES experimental setup for numerical simulation.

### 3.4.2.2 Linear FC hybrid simulation

Figures 3.24 and 3.25 show the displacement and force-time histories respectively of the numerical simulation of the 2-DOF OpenSees model. Table 3.14 contains the absolute errors from these Figures. The El Centro ground motion is scaled to 10% to keep the specimen within its linear range. As with the one-bay-frame linear analyses, the NMF result matches the NMI result exactly. This is because NMI is able to converge to the tolerance before 20 iterations. The  $\alpha$ OS result is slightly better than the NME and NMR results.

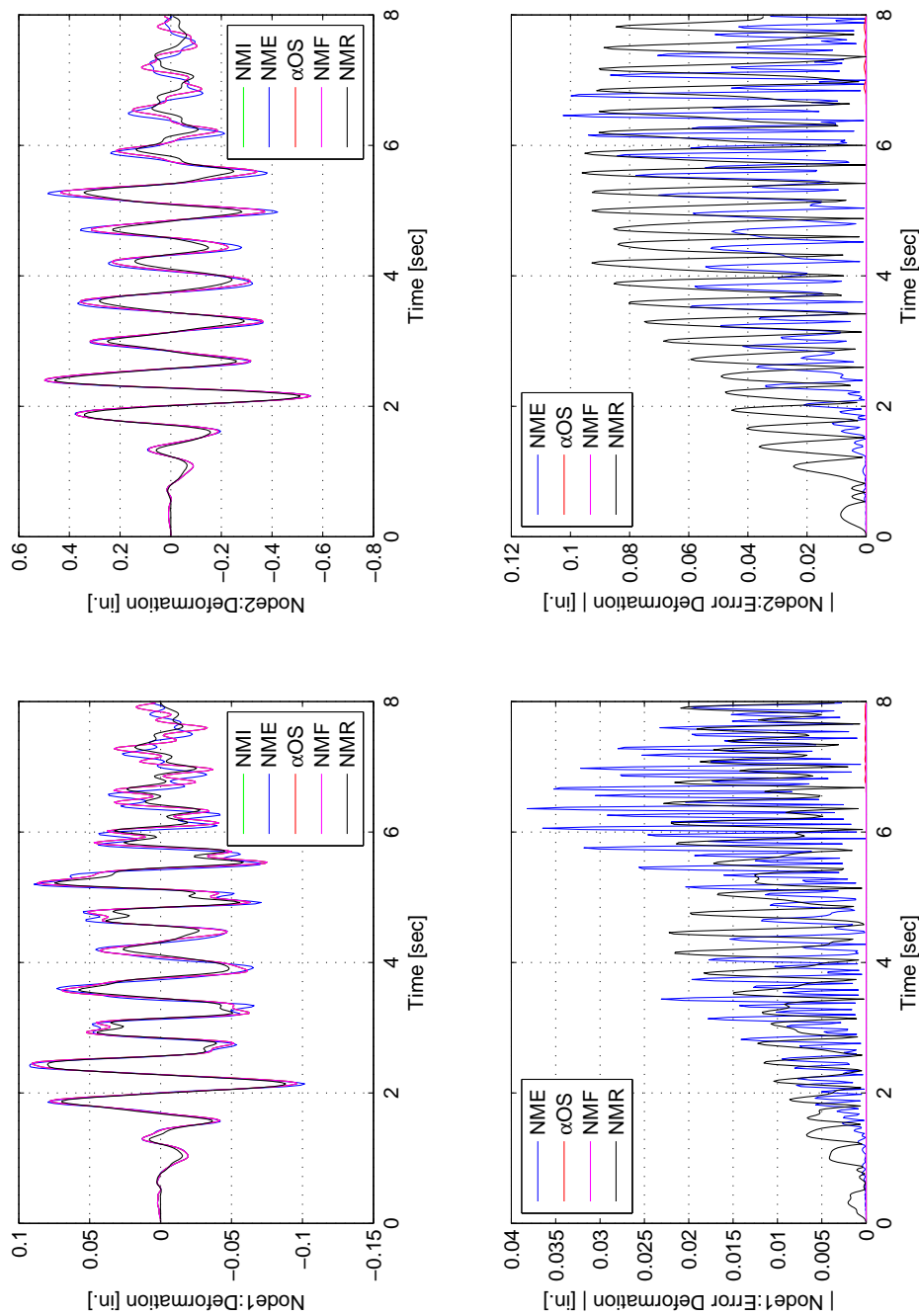


Figure 3.24: Linear numerical simulation results of using NMI, NME,  $\alpha$ OS, NMF and NMI for 2-DOF OpenSees model. Top-Left: Node 1 displacement-time history. Top-Right: Node 2 displacement-time history. Bottom-Left: Node 1 absolute displacement error from NMI. Bottom-Right: Node 2 absolute displacement error from NMI.



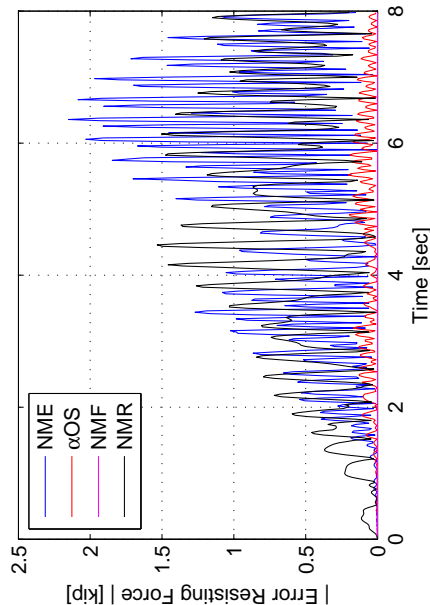
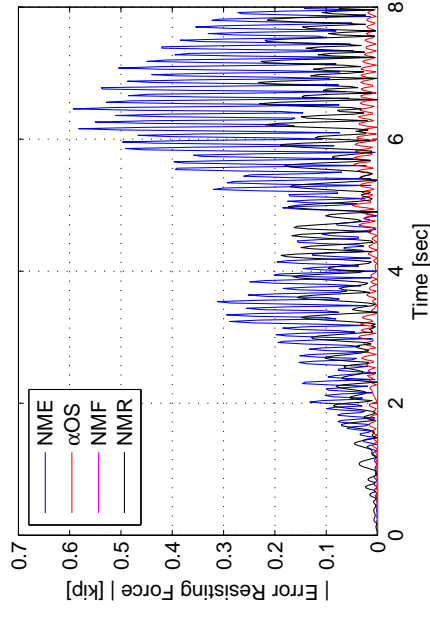
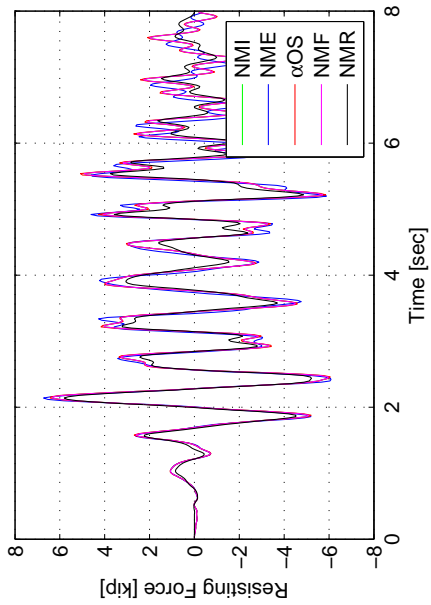
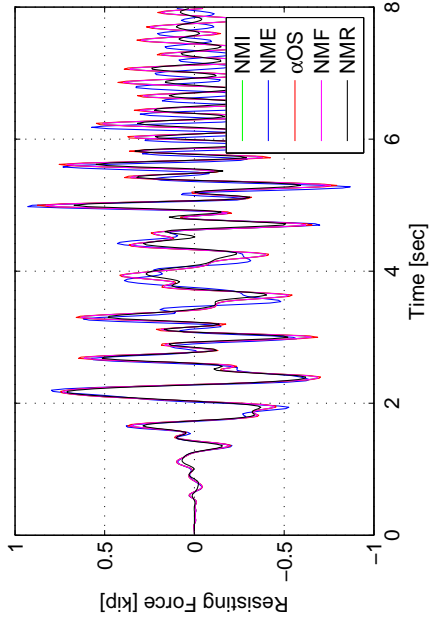


Figure 3.25: Linear numerical simulation results using NMI, NME,  $\alpha$ OS, NMF and NMI for 2-DOF OpenSees model. Top-Left: Node 1 force-time history. Top-Right: Node 2 force-time history. Bottom-Left: Node 1 absolute force error from NMI. Bottom-Right: Node 2 absolute force error from NMI.

Table 3.14: Absolute errors of displacement (Figure 3.24) and force (Figure 3.25) results from NMI of linear numerical simulation using 2-DOF OpenSees model.

Control method	NME	$\alpha$ OS	NMF	NMR
$ ErrD_1 _{Min}$ (in.)	$2.05 \times 10^{-6}$	0	0	$1.19 \times 10^{-5}$
$ ErrD_1 _{Mean}$ (in.)	0.00837	$1.17 \times 10^{-5}$	0	0.00798
$ ErrD_1 _{Max}$ (in.)	0.0382	$1.33 \times 10^{-4}$	0	0.0228
$ ErrF_1 _{Min}$ (kip)	$1.13 \times 10^{-4}$	$2.608 \times 10^{-5}$	0	$6.53 \times 10^{-4}$
$ ErrF_1 _{Mean}$ (kip)	0.524	0.0524	0	0.506
$ ErrF_1 _{Max}$ (kip)	2.15	0.203	0	1.53
$ ErrD_2 _{Min}$ (in.)	$3.36 \times 10^{-6}$	0	0	$1.35 \times 10^{-5}$
$ ErrD_2 _{Mean}$ (in.)	0.0235	$5.01 \times 10^{-5}$	0	0.0405
$ ErrD_2 _{Max}$ (in.)	0.103	$6.29 \times 10^{-4}$	0	0.00120
$ ErrF_2 _{Min}$ (kip)	$1.90 \times 10^{-5}$	$1.68 \times 10^{-5}$	0	$9.68 \times 10^{-5}$
$ ErrF_2 _{Mean}$ (kip)	0.144	0.0136	0	0.0650
$ ErrF_2 _{Max}$ (kip)	0.593	0.0487	0	0.233

Tables 3.15 and 3.16 contain the absolute displacement errors calculated from their numerical counterpart for Nodes 1 and 2 respectively. Tables 3.17 and 3.18 shows the absolute force errors for Nodes 1 and 2 respectively. The plots of the hybrid simulation results are given in Section C.3 of Appendix C. All four tables contain errors calculated at the FEA level, meaning that the errors are from the OpenSees recorded values. For Node 1 and 2, the best result is the CFC NMR result. This is the only one comparable to the numerical result. The FC methods are either slightly better than or equivalent to their DC method counterparts for all time integration schemes for these linear tests.

Table 3.15: Node 1 absolute errors of displacement results from linear hybrid simulations for the 2-DOF setup for various time integration method at the FEA level.

Control method	NME (in.)	$\alpha$ OS (in.)	NMF (in.)	NMR (in.)
DC-Min	$5.95 \times 10^{-8}$	$5.95 \times 10^{-8}$	$6.06 \times 10^{-6}$	$1.06 \times 10^{-6}$
DC-Mean	0.0186	0.0184	0.0224	0.00989
DC-Max	0.0578	0.0569	0.0703	0.0281
CFC-Min	$5.95 \times 10^{-8}$	$5.95 \times 10^{-8}$	$4.29 \times 10^{-6}$	$1.40 \times 10^{-7}$
CFC-Mean	0.0187	0.0189	0.00904	0.00586
CFC-Max	0.0601	0.0591	0.0356	0.0162
EFC-Min	N/A	N/A	N/A	$3.25 \times 10^{-5}$
EFC-Mean	N/A	N/A	N/A	0.109
EFC-Max	N/A	N/A	N/A	0.380

Tables 3.19 and 3.20 show the absolute errors from the MTS-STC controller recorded values. These results show how well the controller is tracking in each control mode. Tracking of NMF hybrid simulation is exceptionally poor in both DC and FC. The span of the bottom actuator in DC

Table 3.16: Node 2 absolute errors of displacement results from linear hybrid simulations for the 2-DOF setup for various time integration method at the FEA level.

Control method	NME (in.)	$\alpha$ OS (in.)	NMF (in.)	NMR (in.)
DC-Min	$1.84 \times 10^{-8}$	$5.95 \times 10^{-8}$	$1.33 \times 10^{-5}$	$8.702 \times 10^{-7}$
DC-Mean	0.0835	0.0834	0.0571	0.0483
DC-Max	0.276	0.275	0.191	0.1435
CFC-Min	$5.95 \times 10^{-08}$	$5.95 \times 10^{-8}$	$1.33 \times 10^{-5}$	$2.34 \times 10^{-7}$
CFC-Mean	0.0850	0.0861	0.0390	0.0307
CFC-Max	0.275	0.277	0.120	0.0867
EFC-Min	N/A	N/A	N/A	$1.31 \times 10^{-5}$ ]
EFC-Mean	N/A	N/A	N/A	0.482
EFC-Max	N/A	N/A	N/A	1.79

Table 3.17: Node 1 absolute errors of force results from linear hybrid simulations for the 2-DOF setup for various time integration method at the FEA level.

Control method	NME (kip)	$\alpha$ OS (kip)	NMF (kip)	NMR (kip)
DC-Min	$1.12 \times 10^{-4}$	$6.49 \times 10^{-4}$	$6.28 \times 10^{-4}$	$4.32 \times 10^{-4}$
DC-Mean	0.700	0.671	2.13	0.495
DC-Max	2.72	2.61	6.49	1.66
CFC-Min	$9.37 \times 10^{-5}$	$1.99 \times 10^{-4}$	$1.99 \times 10^{-4}$	$5.36 \times 10^{-4}$
CFC-Mean	0.597	0.598	0.456	0.188
CFC-Max	2.25	2.39	1.53	0.865
EFC-Min	N/A	N/A	N/A	$7.05 \times 10^{-5}$
EFC-Mean	N/A	N/A	N/A	0.971
EFC-Max	N/A	N/A	N/A	3.51

Table 3.18: Node 2 absolute errors of force results from linear hybrid simulations for the 2-DOF setup for various time integration method at the FEA level.

Control method	NME (kip)	$\alpha$ OS (kip)	NMF (kip)	NMR (kip)
DC-Min	$1.69 \times 10^{-05}$	$1.50 \times 10^{-5}$	$3.97 \times 10^{-5}$	$1.73 \times 10^{-5}$
DC-Mean	0.163	0.158	0.0385	0.106
DC-Max	0.632	0.618	1.16	0.390
CFC-Min	$3.30 \times 10^{-6}$	$1.57 \times 10^{-5}$	$2.08 \times 10^{-4}$	$1.08 \times 10^{-4}$
CFC-Mean	0.147	0.148	0.0953	0.0501
CFC-Max	0.544	0.556	0.323	0.201
EFC-Min	N/A	N/A	N/A	$1.08 \times 10^{-5}$
EFC-Mean	N/A	N/A	N/A	0.3341
EFC-Max	N/A	N/A	N/A	1.23

Table 3.19: Bottom actuator absolute errors of linear hybrid simulations for the 2-DOF setup for various time integration schemes at the control system level.

Control method	NME	$\alpha$ OS	NMF	NMR
DC-Min (in.)	$1.16 \times 10^{-9}$	$8.64 \times 10^{-10}$	$6.92 \times 10^{-9}$	$2.15 \times 10^{-10}$
DC-Mean (in.)	$3.16 \times 10^{-4}$	$2.96 \times 10^{-4}$	0.00178	0.00102
DC-Max (in.)	0.00211	0.00186	0.00676	0.00473
CFC-Min (kip)	$6.98 \times 10^{-9}$	$1.12 \times 10^{-8}$	$5.96 \times 10^{-8}$	0
CFC-Mean (kip)	0.00783	0.00959	0.0389	0.0238
CFC-Max (kip)	0.0534	0.05207	0.181	0.287
EFC-Min (kip)	N/A	N/A	N/A	$2.98 \times 10^{-7}$
EFC-Mean (kip)	N/A	N/A	N/A	0.0341
EFC-Max (kip)	N/A	N/A	N/A	0.196

and FC are 0.05 in. and 1 kip respectively. The mean of the force error for the bottom actuator ranges from 0.8% to 2.4% of the force span while the displacement error mean ranges from 0.6% to 3.6%. It is difficult to say which control mode is significantly better for the bottom actuator. The span of the top actuator in DC and FC are 0.30 in. and 0.5 kips respectively. The mean of the force error for the top actuator ranges from 0.3% to 11% while only 0.15% to 0.9% for DC. DC is clearly better in tracking for the top actuator. It is worth noting that the top DOF of the setup is about 30 times softer than the bottom DOF. This supports the concept that a softer system is better controlled in DC and a stiffer system is better in FC. Figures C.34 and C.35 show the command displacement and force curves of both nodes for the CFC method using NMR. Figure C.36 shows the command displacement and force curves of both nodes for the EFC method using NMR. These methods do not produce desirable command value curves when using the NMR time integration scheme.

Table 3.20: Top actuator absolute errors of linear hybrid simulations for the 2-DOF setup for various time integration schemes at the control system level.

Control method	NME	$\alpha$ OS	NMF	NMR
DC-Min (in.)	$1.09 \times 10^{-9}$	$1.38 \times 10^{-9}$	$3.01 \times 10^{-8}$	$1.64 \times 10^{-9}$
DC-Mean (in.)	$4.60 \times 10^{-4}$	$4.5 \times 10^{-4}$	0.00262	$8.31 \times 10^{-4}$
DC-Max (in.)	0.00417	0.00344	0.00788	0.00535
CFC-Min (kip)	0.00227	$2.79 \times 10^{-9}$	0.0355	$5.77 \times 10^{-8}$
CFC-Mean (kip)	0.00859	0.00150	0.0569	0.00429
CFC-Max (kip)	0.0132	0.00875	0.0752	0.0403
EFC-Min (kip)	N/A	N/A	N/A	0
EFC-Mean (kip)	N/A	N/A	N/A	0.00529
EFC-Max (kip)	N/A	N/A	N/A	0.0682

### 3.4.2.3 Nonlinear FC hybrid simulation

Figures 3.26 and 3.27 show the displacement and force-time histories respectively of the numerical simulation of the 2-DOF OpenSees model. Table 3.21 contains the absolute errors from these Figures. The El Centro ground motion is scaled to 50% to push the specimen into its nonlinear range. Similar to the nonlinear one-bay-frame analysis,  $\alpha$ OS and NME time integration schemes do not converge for the nonlinear numerical analysis of the 2-DOF OpenSees model. Table 3.21 shows the absolute displacement and force errors for NMF and NMR compared to NMI. NMF numerical errors are two orders of magnitude smaller than the NMR results.

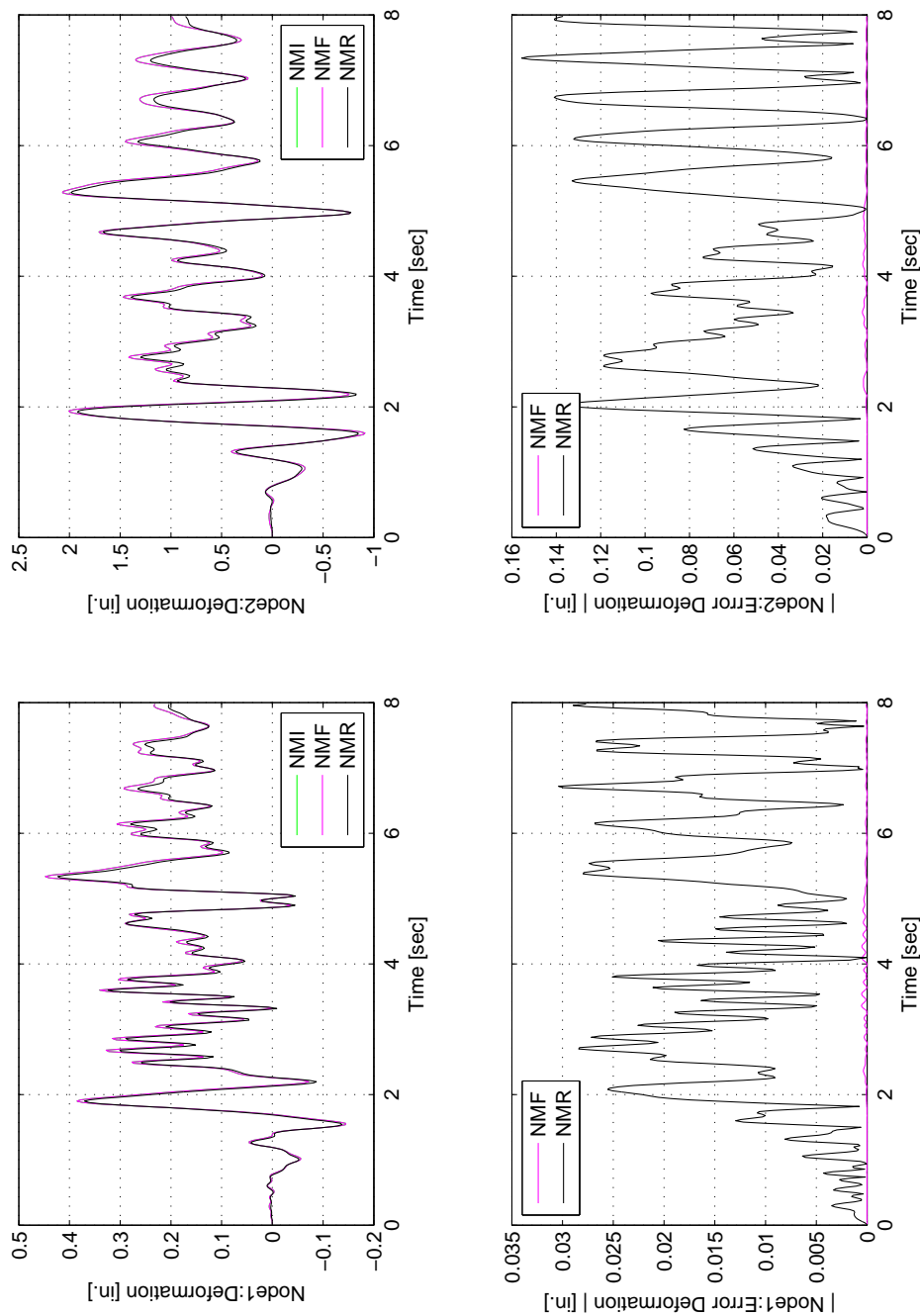


Figure 3.26: Nonlinear numerical simulation results of using NMI, NME,  $\alpha$ OS, NMF and NMI for 2-DOF OpenSees model. Top-Left: Node 1 displacement-time history. Top-Right: Node 2 displacement-time history. Bottom-Left: Node 1 absolute displacement error from NMI. Bottom-Right: Node 2 absolute displacement error from NMI.

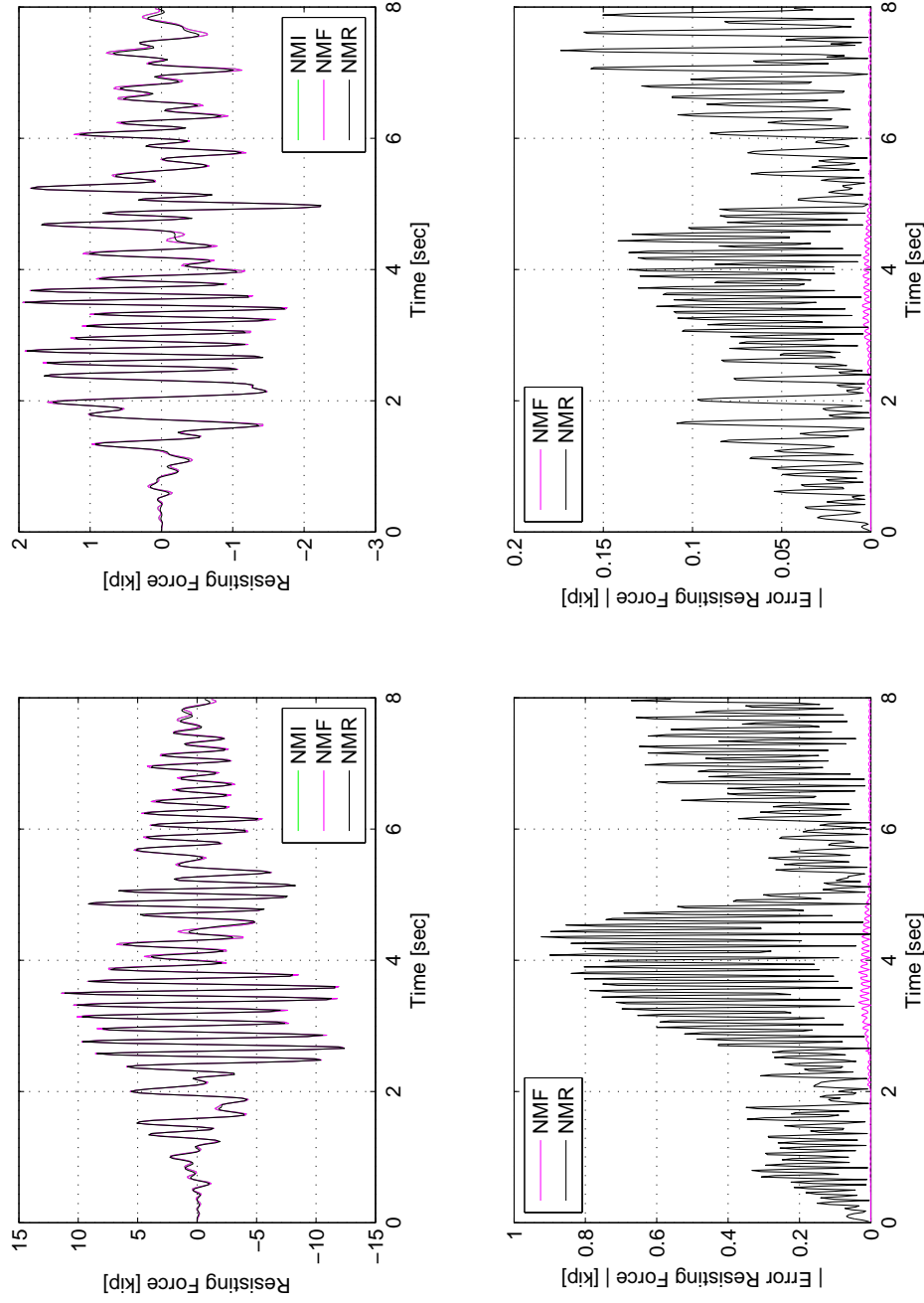


Figure 3.27: Nonlinear numerical simulation results using NMI, NME,  $\alpha$ OS, NMF and NMI for 2-DOF OpenSees model. Top-Left: Node 1 force-time history. Top-Right: Node 2 force-time history. Bottom-Left: Node 1 absolute force error from NMI. Bottom-Right: Node 2 absolute force error from NMI.

Table 3.21: Absolute errors of displacement (Figure 3.26) and force (Figure 3.27) results from nonlinear numerical simulation using 2-DOF OpenSees model.

Control method	NMF	NMR
$ ErrD_1 _{Min}$ (in.)	0	$4.49 \times 10^{-5}$
$ ErrD_1 _{Mean}$ (in.)	$1.05 \times 10^{-4}$	0.0122
$ ErrD_1 _{Max}$ (in.)	$5.52 \times 10^{-4}$	0.0304
$ ErrF_1 _{Min}$ (kip)	0	0.00101
$ ErrF_1 _{Mean}$ (kip)	0.00583	0.272
$ ErrF_1 _{Max}$ (kip)	0.0342	0.9235
$ ErrD_2 _{Min}$ (in.)	0	$6.15 \times 10^{-5}$
$ ErrD_2 _{Mean}$ (in.)	$5.00 \times 10^{-4}$	0.0570
$ ErrD_2 _{Max}$ (in.)	0.0021	0.156
$ ErrF_2 _{Min}$ (kip)	0	$1.60 \times 10^{-4}$
$ ErrF_2 _{Mean}$ (kip)	$8.90 \times 10^{-4}$	0.0493
$ ErrF_2 _{Max}$ (kip)	0.00457	0.1739

CFC:Broyden, CFC:Transpose, CFC:Krylov and EFC methods fail during FC hybrid simulation either due to the force error in the control system tripping the interlock or the control system going unstable during the test. Therefore, only the results for DC, CFC:BFGS and CFC:Intrinsic methods are provided. For CFC:Intrinsic method, it also fails for some certain time integration schemes. CFC:BFGS is the most robust FC method. It works with all the time integration schemes. The plots of the hybrid simulation results from OpenSees are given in Section C.4 of Appendix C. From these figures, the drift overall is best captured by the FC methods. From Tables 3.22 to 3.23, the FC methods for each time integration scheme overall give better results than the DC methods for both nodes with the exception of NMF. The force errors for FC methods are significantly better than DC method across the board.

Table 3.22: Node 1 absolute errors of displacement results from nonlinear hybrid simulations for the 2-DOF setup for various time integration method at the FEA level.

Control method	NME (in.)	$\alpha$ OS (in.)	NMF (in.)	NMR (in.)
DC-Min	$2.97 \times 10^{-7}$	$1.32 \times 10^{-8}$	$3.90 \times 10^{-6}$	$2.45 \times 10^{-5}$
DC-Mean	0.163	0.147	0.0307	0.379
DC-Max	0.419	0.337	0.153	0.664
CFC:BFGS-Min	$3.75 \times 10^{-8}$	1.15e-08	$4.19 \times 10^{-5}$	$1.81 \times 10^{-5}$
CFC:BFGS-Mean	0.0594	0.0319	0.0519	0.110
CFC:BFGS-Max	0.251	0.166	0.182	0.241
CFC:Intrinsic-Min	N/A	2.92e-08	N/A	$8.23 \times 10^{-6}$
CFC:Intrinsic-Mean	N/A	0.0527	N/A	0.0750
CFC:Intrinsic-Max	N/A	0.229	N/A	0.191

It is difficult to determine which control mode is better at tracking for the bottom actuator. However the ratio of the mean force error to the mean displacement errors increase from 20 to



Table 3.23: Node 2 absolute errors of displacement results from nonlinear hybrid simulations for the 2-DOF setup for various time integration method at the FEA level.

Control method	NME (in.)	$\alpha$ OS (in.)	NMF (in.)	NMR (in.)
DC-Min	$2.97 \times 10^{-7}$	$4.07 \times 10^{-9}$	$6.36 \times 10^{-6}$	$2.39 \times 10^{-5}$
DC-Mean	1.40	1.54	0.277	1.83
DC-Max	3.02	2.70	1.10	3.25
CFC:BFGS-Min	$1.20 \times 10^{-7}$	$1.54 \times 10^{-8}$	$4.77 \times 10^{-5}$	$2.02 \times 10^{-5}$
CFC:BFGS-Mean	0.271	0.438	0.327	0.522
CFC:BFGS-Max	1.12	1.13	0.9251	1.21
CFC:Intrinsic-Min	N/A	$4.67 \times 10^{-8}$	N/A	$1.54 \times 10^{-5}$
CFC:Intrinsic-Mean	N/A	0.300	N/A	0.348
CFC:Intrinsic-Max	N/A	1.35	N/A	0.876

Table 3.24: Node 1 absolute errors of force results from nonlinear hybrid simulations for the 2-DOF setup for various time integration method at the FEA level.

Control method	NME (kip)	$\alpha$ OS (kip)	NMF (kip)	NMR (kip)
DC-Min	0.000364	$7.94 \times 10^{-5}$	0.00213	0.00111
DC-Mean	1.16	1.40	1.46	0.699
DC-Max	4.47	5.62	5.70	3.13
CFC:BFGS-Min	$4.6 \times 10^{-5}$	$9.10 \times 10^{-5}$	0.00304	$2.74 \times 10^{-4}$
CFC:BFGS-Mean	1.19	1.32	1.06	0.723
CFC:BFGS-Max	4.94	5.47	3.68	3.23
CFC:Intrinsic-Min	N/A	$1.44 \times 10^{-4}$	N/A	$4.84 \times 10^{-5}$
CFC:Intrinsic-Mean	N/A	2.34	N/A	0.679
CFC:Intrinsic-Max	N/A	9.93	N/A	3.94

Table 3.25: Node 2 absolute errors of force results from nonlinear hybrid simulations for the 2-DOF setup for various time integration method at the FEA level.

Control method	NME (kip)	$\alpha$ OS (kip)	NMF (kip)	NMR (kip)
DC-Min	$4.57 \times 10^{-6}$	$4.20 \times 10^{-6}$	0.000273	$4.20 \times 10^{-5}$
DC-Mean	0.203	0.231	0.247	0.195
DC-Max	0.741	0.714	0.959	0.988
CFC:BFGS-Min	$5.30 \times 10^{-7}$	$3.94 \times 10^{-5}$	$5.32 \times 10^{-5}$	$3.31 \times 10^{-4}$
CFC:BFGS-Mean	0.193	0.216	0.182	0.197
CFC:BFGS-Max	0.658	0.689	0.620	0.714
CFC:Intrinsic-Min	N/A	$2.98 \times 10^{-5}$	N/A	$5.22 \times 10^{-5}$
CFC:Intrinsic-Mean	N/A	0.388	N/A	0.184
CFC:Intrinsic-Max	N/A	1.74	N/A	0.590

40 when comparing the linear results to nonlinear results. It is clear that the top actuator is better at tracking in DC than in FC. Figures C.53 and C.54 show the command force curve for the CFC:BFGS and CFC:Intrinsic respectively using NMR. Both do not produce smooth command curves.

Table 3.26: Bottom actuator absolute errors of nonlinear hybrid simulations for the 2-DOF setup for various time integration schemes at the control system level.

<b>Control method</b>	<b>NME</b>	<b><math>\alpha</math>OS</b>	<b>NMF</b>	<b>NMR</b>
DC-Min (in.)	$2.84 \times 10^{-9}$	$1.96 \times 10^{-9}$	$6.01 \times 10^{-10}$	$1.50 \times 10^{-9}$
DC-Mean (in.)	$7.30 \times 10^{-4}$	$6.69 \times 10^{-4}$	0.00108	$8.11 \times 10^{-4}$
DC-Max (in.)	0.00397	0.00392	0.00505	0.00789
CFC:BFGS-Min (kip)	0	$4.47 \times 10^{-8}$	0	0
CFC:BFGS-Mean (kip)	0.0125	0.0277	0.0391	0.0297
CFC:BFGS-Max (kip)	0.162	0.176	0.174	0.318
CFC:Intrinsic-Min (kip)	N/A	0	N/A	0
CFC:Intrinsic-Mean (kip)	N/A	0.0426	N/A	0.0801
CFC:Intrinsic-Max (kip)	N/A	0.172	N/A	0.287

Table 3.27: Bottom actuator absolute errors of nonlinear hybrid simulations for the 2-DOF setup for various time integration schemes at the control system level.

<b>Control method</b>	<b>NME</b>	<b><math>\alpha</math>OS</b>	<b>NMF</b>	<b>NMR</b>
DC-Min (in.)	$220 \times 10^{-9}$	$6.44 \times 10^{-9}$	$2.66 \times 10^{-9}$	$8.48 \times 10^{-9}$
DC-Mean (in.)	$4.84 \times 10^{-4}$	$5.74 \times 10^{-4}$	$6.58 \times 10^{-4}$	$6.49 \times 10^{-4}$
DC-Max (in.)	0.00472	0.00463	0.00562	0.00946
CFC:BFGS-Min (kip)	0.00125	0	0.00262	$2.74 \times 10^{-6}$
CFC:BFGS-Mean (kip)	0.0215	0.00697	0.0451	0.0105
CFC:BFGS-Max (kip)	0.0465	0.0288	0.0788	0.0418
CFC:Intrinsic-Min (kip)	N/A	$1.61 \times 10^{-6}$	N/A	0
CFC:Intrinsic-Mean (kip)	N/A	0.0224	N/A	0.00213
CFC:Intrinsic-Max (kip)	N/A	0.0487	N/A	0.0352

Figures 3.28 and 3.29 show the calculated entries of the tangent stiffness matrix for the CFC:BFGS and CFC:Intrinsic methods respectively. The filter of 0.01 is used for both displacement and force limits. This means that if the 2-norm of the incremental measured displacement or force vector is not greater than 0.01, the tangent stiffness matrix is not updated. These figures show that the CFC:Intrinsic method does not update the tangent stiffness matrix as frequently as the CFC:BFGS method. At the same time, the CFC:Intrinsic method does not produce spikes in the estimation as the CFC:BFGS method does. The CFC:BFGS method here is using  $\epsilon = 0.8$  from Algorithm 3.2.

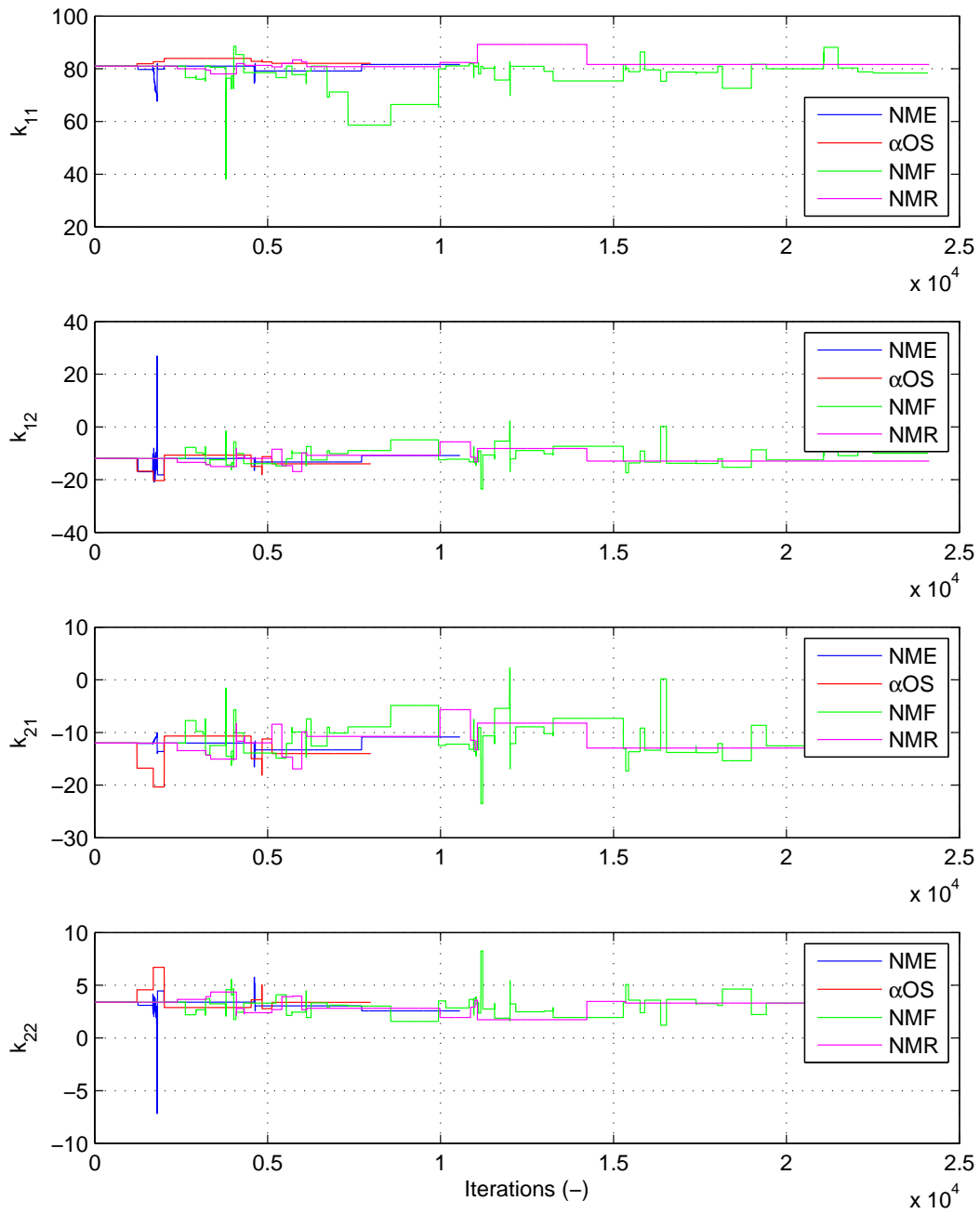


Figure 3.28: Tangent stiffness values for nonlinear 2-DOF hybrid simulation using CFC:BFGS for NME,  $\alpha$ OS, NMF and NMR

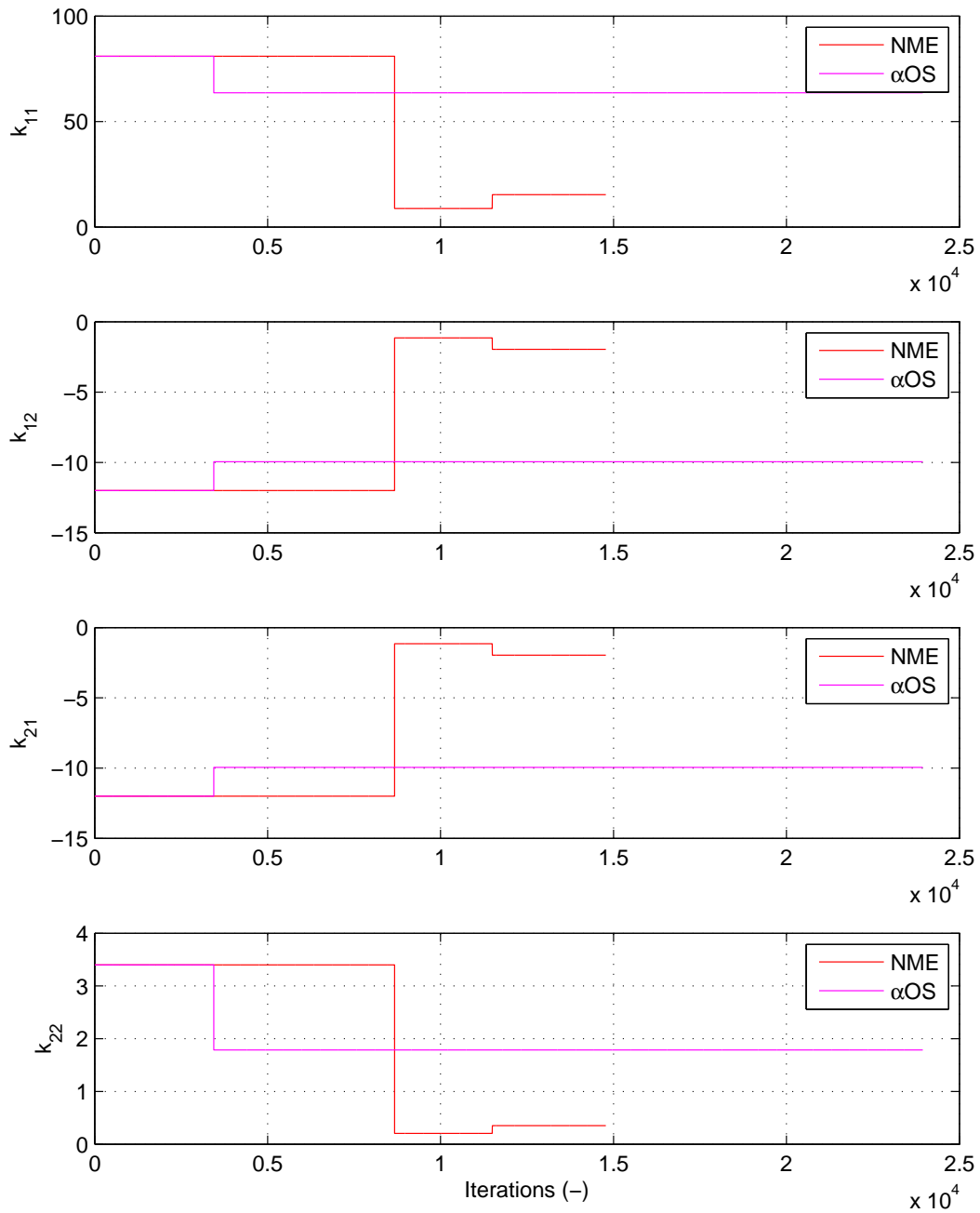


Figure 3.29: Tangent stiffness values for nonlinear 2-DOF hybrid simulation using CFC:Intrinsic.for  $\alpha$ OS and NMR.

### 3.4.3 Summary of experimental results and verification

The results from this section show that FC hybrid simulation is marginally better for testing with the stiff  $\mu$ -NEES setups than DC hybrid simulation. The CFC methods are better than the EFC even though EFC method does not depend on the tangent stiffness matrix. The MTS-STS control system tracks displacements and forces equally well. The BFGS method is the most robust method for estimating the tangent stiffness matrix, and it works with all four time integration schemes. Methods using NMR produces undesirable command force curves for the control system.

# Chapter 4

## Switch control hybrid simulation

This chapter discusses switch control (SC) hybrid simulation. It first defines SC hybrid simulation and lays out its challenges. The motivation for SC hybrid simulation is presented, and the previous research is surveyed. The methods for SC hybrid simulation are proposed. Then the implementation of these methods using OpenFresco and Mathworks Simulink/Stateflow is described in detail. The SC methods are tested with the 1-DOF  $\mu$ -NEES experimental setup at the *nees@berkeley* lab. The chapter concludes with a presentation and discussion of the experimental results from SC hybrid simulation tests.

### 4.1 Introduction

#### 4.1.1 Problem definition

SC hybrid simulation is a type of hybrid simulation where the control mode of an actuator changes during hybrid simulation. For example, an actuator may be in displacement control mode at the start of the simulation. During the simulation, it may be necessary for the actuator to change or switch to force control mode, for various reasons discussed later in this chapter. The control mode may change many times during a simulation depending on the specimen being tested and the actuator being used. While there are several control modes that may be considered for SC hybrid simulation, this chapter deals only with displacement and force control modes.

SC hybrid simulation has the same challenges of FC hybrid simulation as presented in Section 3.1.1, since part of the simulation may be in force control mode. In addition to the FC hybrid simulation challenges, there exists a challenge of determining the criteria for switching actuator control modes during a simulation. For a given criterion, the problem of when to initiate the switching of the control mode arises. The unnecessary switching of the control modes can occur when the state of the specimen oscillates around the switching point. This can produce unreliable results and possibly cause the control system to become unstable in the middle of a test. The control system must be capable of switching control modes on the fly during a test and be able to impose both displacements and forces on the specimen adequately.

### 4.1.2 Motivation

The switching of control modes during hybrid simulation is necessary when there is a change in the response of the specimen that renders the current control mode inadequate. Figure 4.1 shows a hysteresis curve of High Damping Rubber Bearing (HDBR). Performing hybrid simulation with this type of a specimen requires switching of control modes during the simulation. The specimen

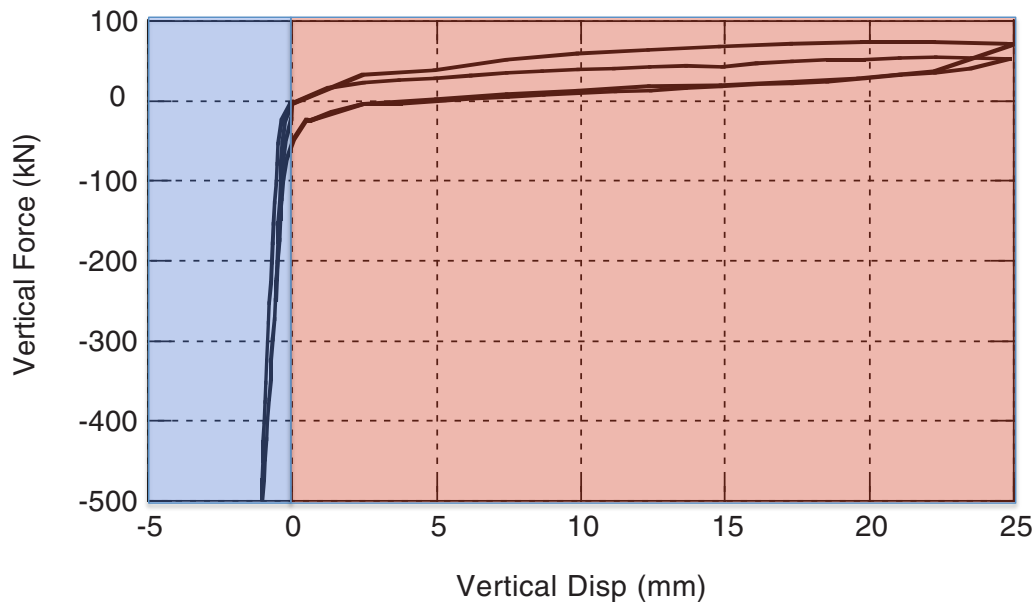


Figure 4.1: Typical hysteresis curve of High Damping Rubber Bearing (HDBR). HDBR is loaded vertically in compression and tension.

is very stiff in the blue region of the hysteresis when HDBR is in compression. The specimen becomes much softer in the red region when it is in tension. For such a specimen, using one control throughout a hybrid simulation test does not suffice. If the specimen is tested in DC mode only, the control system can not adequately impose the small displacements as the specimen enters the blue region. On the other hand if the specimen is in FC mode only, the control system can not impose the small forces as the specimen enters the red region. Both control modes must be used during a single hybrid simulation test. A mechanism for switching between the control mode is required. The type of behavior exhibited by HDBR is commonly found in other specimens in structural testing. Many specimens yield during testing and become soft. In most cases testing with actuators in DC modes is satisfactory. However, there are cases where the specimen is so stiff initially that FC mode is better suited for the task. In hybrid simulation where innovative structural components are often tested experimentally, this situation of having a very stiff specimen may be somewhat common.

### 4.1.3 Previous research

There have been some but not many researchers who have performed SC hybrid simulation. Pan, et al. [18] as a part of their research presented in Section 3.1.3 performed SC hybrid simulation on HDBR in the vertical loading direction. They used the sign of the trial force as a criterion for switching between force and displacement control modes. Since they had a prior knowledge of HDBR's hysteric behavior from running quasi-static cyclic tests, they knew that the bearing was very stiff in compression and very soft in tension. When the trial force was negative or in compression, they used force control. When the trial force was positive or in tension, they used displacement control. A loading limit was enforce to prevent large trial forces and displacements from being applied to the specimen.

Elkhoraibi and Mosalam [7] also performed switch control hybrid simulation as a part of their research presented in Section 3.1.3. Algorithm 4.1 shows the algorithm for switching control modes at each time step  $n$ . Elkhoraibi and Mosalam used two criteria for switching between displacement and force control modes, the secant stiffness  $k$  and the restoring force  $f$ . The thresholds for switching were defined before hand. The measured stiffness  $k_i$  and force  $f_i$  were compared against the thresholds for switching.  $k_d$  and  $k_f$  were the stiffness thresholds for the displacement and the force respectively, where  $k_d < k_f$ . The restoring force thresholds,  $f_d$  and  $f_f$ , were setup the same way. A buffer zone prevented the back and forth switching caused by small fluctuations in  $k$ .

---

#### Algorithm 4.1 Mode Switching Algorithm at Time Step $n$ by Elkhoraibi and Mosalam [7]

---

Define limits  $k_d, K_f, f_d$  and  $f_f$  where  $k_d < k_f$  and  $f_d < f_f$

Compute  $k_n$  and  $f_n$

**if**  $ctrl_{n-1} = DC$  **then**

**if**  $k_n > k_f$  AND  $|f_n| > f_f$  **then**

        Set  $ctrl_n = FC$

**else**

        Set  $ctrl_n = ctrl_{n-1}$

**end if**

**else if**  $ctrl_{n-1} = FC$  **then**

**if**  $k_n < k_d$  OR  $|f_n| < f_d$  **then**

        Set  $ctrl_n = DC$

**else**

        Set  $ctrl_n = ctrl_{n-1}$

**end if**

**end if**

---

## 4.2 Methods for SC hybrid simulation

Methods of SC hybrid simulation are link to methods for FC hybrid simulation discussed in Section 3.2 since a portion of the simulation is in FC mode. Each type of switching strategy presented in this section utilizes all the different FC methods. Two types of switching strategies



are developed in this section. The first type is switching based on the trial force. The second is switching based on the secant stiffness of the specimen. A buffer zone must be incorporated into both strategies to avoid unnecessary switching or chattering of the control modes. Each strategy has its advantages and disadvantages.

### 4.2.1 Trial force based switching strategy

The switching of the actuator control modes can be based on the calculated trial force. This indeed is what Pan, et al. [18] did. A predefined switching limit is determined before the start of the simulation. The control mode switches when this limit or threshold is reached. The switching limit can stem from any response quantity such as displacement, velocity, acceleration and force. Velocity and acceleration are not good candidates because they are difficult to control. It is more feasible to use a displacement value as a limit for switching. Displacement values are relatively easy to control and read. However, switching based on displacement values is not the best way in structural testing. The specimens often undergo permanent deformation causing a displacement offset. In this case, the displacement limit at which the specimen deviates from its linearity changes. The predefined displacement switching limit can no longer accurately detect the change in the specimen's behavior.

Switching control modes based on the calculated trial force,  $\check{f}$ , is better than switching based on trial displacement,  $\check{u}$ . It is true that, as some specimens experience more and more damage during a test, their ability to withstand applied forces degrades. Thus the predefined force limit for switching is no longer valid. However, the degradation of the resisting force of the specimen is in general more gradual and predictable than the permanent deformation that occurs during a hybrid simulation test. This degradation can be anticipated by scaling the force limit. The trial forces are calculated using the methods presented in Section 3.2.

The algorithm for determining the control mode for a softening and hardening specimens are shown in Algorithm 4.2 and 4.3 respectively. These are modifications of Algorithm 4.1 developed by Elkhoraibi and Mosalam [7]. Buffering is implemented in this algorithm to avoid control mode chattering when successive trial forces are near the force limit.  $f_d$  and  $f_f$  are the force limit for switching from FC mode to DC mode and the force limit for switching from DC mode to FC mode respectively.  $f_d$  and  $f_f$  are predefined before the start of a simulation and must be determined carefully. If the span between  $f_d$  and  $f_f$  is too large, the control mode may not switch correctly. On the other hand, if the span is too small, the control mode chattering is not avoided.

An advantage of force limit switching is that it is easy to implement and the algorithm is straightforward. It works well for certain types of specimen. A major disadvantage is that some foreknowledge of the specimen's behavior is required to effectively set the force limit and the buffer parameters. Essentially, the researcher must know when the specimen's behavior will change. Therefore, force limit switching is not appropriate when the specimen's behavior is unknown or a reason prediction cannot be made.

---

**Algorithm 4.2** Force limit switching strategy for softening specimen at time step  $n$ 


---

Define limits  $f_d$  and  $f_f$  where and  $f_d > f_f$   
 Compute  $\check{f}_n$   
**if**  $ctrlMode_{n-1} = DC$  **then**  
   **if**  $|\check{f}_n| < f_f$  **then**  
     Set  $ctrlMode_n = FC$   
   **else**  
     Set  $ctrlMode_n = ctrlMode_{n-1}$   
   **end if**  
**else if**  $ctrlMode_{n-1} = FC$  **then**  
   **if**  $|\check{f}_n| > f_d$  **then**  
     Set  $ctrlMode_n = DC$   
   **else**  
     Set  $ctrlMode_n = ctrlMode_{n-1}$   
   **end if**  
**end if**

---



---

**Algorithm 4.3** Force limit switching strategy for softening specimen at time step  $n$ 


---

Define limits  $f_d$  and  $f_f$  where and  $f_d < f_f$   
 Compute  $\check{f}_n$   
**if**  $ctrlMode_{n-1} = DC$  **then**  
   **if**  $|\check{f}_n| > f_f$  **then**  
     Set  $ctrlMode_n = FC$   
   **else**  
     Set  $ctrlMode_n = ctrlMode_{n-1}$   
   **end if**  
**else if**  $ctrlMode_{n-1} = fFC$  **then**  
   **if**  $|\check{f}_n| < f_d$  **then**  
     Set  $ctrlMode_n = DC$   
   **else**  
     Set  $ctrlMode_n = ctrlMode_{n-1}$   
   **end if**  
**end if**

---

### 4.2.2 Secant stiffness based switching strategy

Secant stiffness value can be used as a criterion for switching control modes. The secant stiffness value gives an estimation of the tangent stiffness value. It can be used to detect the softening or hardening behavior of a specimen. The secant stiffness values at step  $n$  is calculated as  $sec_n = (\bar{f}_{n-1} - \bar{f}_{n-2}) / (\bar{u}_{n-1} - \bar{u}_{n-2})$ . This value is calculated from the measured force and displacement values from the previous two time steps and is not calculated using values from any substeps. Once the secant limit for switching is determined, it works the same as force limit switching for hardening specimens. This one algorithm however works for both hardening and softening specimens. Algorithm 4.4 shows the strategy for determining the control mode in secant limit switching.  $k_d$  and  $k_f$  are secant stiffness limits for switching from FC to DC and for switching from DC to FC respectively. The absolute value of the secant stiffness,  $sec_n$ , is not used in determining the control mode. This is because DC is used for specimens that start to exhibit negative stiffness values regardless of the magnitude of those values.

---

#### Algorithm 4.4 Secant limit switching strategy at time step $n$

---

Define limits  $k_d$  and  $k_f$  where and  $k_d < k_f$

Compute  $sec_n$

**if**  $ctrlMode_{n-1} = DC$  **then**

**if**  $sec_n > k_f$  **then**

        Set  $ctrlMode_n = FC$

**else**

        Set  $ctrlMode_n = ctrlMode_{n-1}$

**end if**

**else if**  $ctrlMode_{n-1} = FC$  **then**

**if**  $sec_n < k_d$  **then**

        Set  $ctrlMode_n = DC$

**else**

        Set  $ctrlMode_n = ctrlMode_{n-1}$

**end if**

**end if**

---

Unlike force limit switching, a foreknowledge of the specimen's behavior is not required for the implementation of this method. The algorithm detects when the specimen is changing its behavior by calculating its secant stiffness value. This value can be determined from the tuning parameters of the control system. For a given a set of control system tuning parameters, a secant stiffness value limit can be determined for a control system at which it goes unstable in force control or the tracking performance degrades. This is because the stiffness of the specimen contributes the proportional gain of the control system. The secant limit for switching is not specimen dependent but control system dependent. Provided that the control system is likely to be reused more frequently than a specimen, this is a more effective strategy for switching than force limit switching. One disadvantage of secant limit switching is that it is susceptible to spurious updates as discusses in the Chapter 3. This can cause unnecessary switching of the control modes during a hybrid simulation test. Therefore a filter must be used to prevent control system noises from

causing the secant stiffness values to update.

## 4.3 Implementation

As mentioned before, SC hybrid simulation uses the methods for FC hybrid simulation. As such, each SC strategy is implemented using all the FC methods. Force limit switching strategy used both CFC and EFC methods. Force limit switching strategy that employs the CFC method is called Point Switch Control (PSC) since the switching occurs at a predefined point. Force limit switching strategies that use the EFC method is called Equilibrium Point Switch Control (EPSC). Secant limit switching strategies that use CFC and EFC methods are named Secant Switch Control (SSC) and Equilibrium Secant Switch Control (ESSC) respectively.

### 4.3.1 Compatibility methods for switching

PSC and SSC strategies are implemented in OpenFresco in the ECxPtargetSwitchCM experimental control class. The class definite is very similar to the class definition of ECxPtargetForce given in Figure A.6. OpenFresco receives the trial deformations from the computational drive as seen in Figure 4.2. Within OpenFresco, ECxPtargetSwitchCM receives the trial displacements from the experimental setup class. It then calculates  $\check{f}_n$  using the CFC method.  $\check{f}_n$  is passed through Algorithm 4.2 or 4.3 to set the control mode. It sends the control mode and the appropriate trial value to the control system.

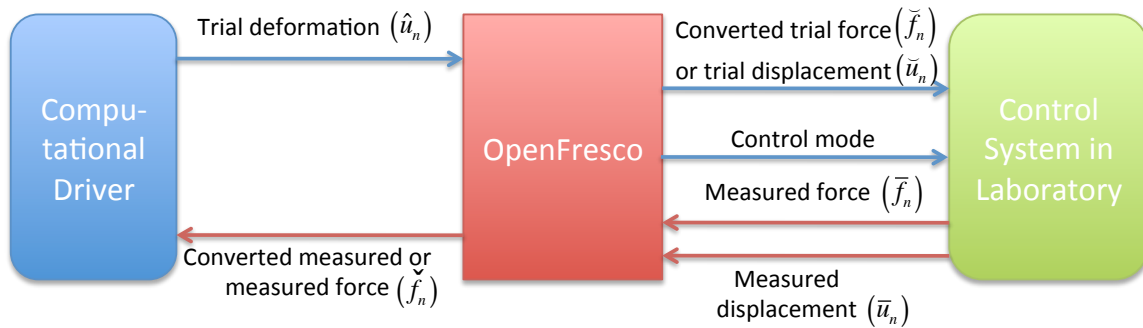


Figure 4.2: SC hybrid simulation components and their processes.

If SSC is used, ECxPtargetSwitchCM passes  $sec_n$  through Algorithm 4.4 to determine the control mode.  $sec_n$  is calculated in the acquire method at step  $n - 1$  from the measured displacement and force pairs from time steps  $n - 1$  and  $n - 2$ . Measured values are used instead of trial values because the measured values give a more accurate state of the specimen. Since the trial force is calculated from the trial displacement and only one is impose on the specimen, the trial displacement and force pair does not give an accurate state of the specimen. At least one of the trial values that is not impose on the specimen may not equal its measured counterpart. ECxPtargetSwitchCM stores the measured pair after it calculates the secant stiffness value for the next

secant stiffness calculation. If the incremental measure displacement and force are not greater than then noise limit of the control system, then the secant stiffness is not updated.

The MTS-STS controller sets the feedback value as the new zero point whenever it switches into another control mode. This means that if the controller is in DC mode and the force feedback is 1 kip, the controller offsets all the command forces by 1 kip when it switches into FC mode. For example if the ECxPtargetSwitchCM calculates the trial force of 2 kips, the MTS-STS controller will try to impose 3 kips to the specimen. Therefore, ECxPtargetSwitchCM compensates for this behavior. It keeps in memory the feedback value when the control mode was switched. It then offsets the trial value with feedback value in memory before sending it to the controller. Continuing with the previous example, when ECxPtargetSwitchCM calculates a trial force of 3 kips, it will offset the 3 kips by 2 kips and send 1 kip to the controller. The controller will then impose 3 kips on the specimen. ECxPtargetSwitchCM does this for both DC and FC modes. The MTS-STS does not offset measured values in any way, just the command values are offset.

### 4.3.2 Equilibrium methods for switching

The majority of the EPSC and ESSC strategies are programmed in Matlab. It is an extension of what was discussed in Section 3.3.2. The following are the additional functions required for SC hybrid simulation:

- `simpleYield` - This function uses Algorithm 4.2 or 4.3 to set the control mode during SC hybrid simulation with one exception. The force from the previous step is used to determine the control mode instead of the calculated trial force at the current time step. The displacement time integration and force time integration schemes from Section 3.3.2 are the arguments of this function. The displacements and forces at each time step are stored as global variables in the `main` function. The displacement and force pair from `main` is pass to whichever time integration scheme is chosen for the next time step. This function is only used with numerical models.
- `simpleYieldExperimental` - This is the same as `simpleYield` but it is used with a hybrid model. A different function is needed with the Experimental element from Section 3.3.2 because of the interaction with the MTS-STS control system.
- `secantUpdate` - This function uses Algorithm 4.4 to determine the control mode. Otherwise it operates the same as the `simpleYield` function. It has arguments of displacement time integration and force time integration schemes. This function is only used with numerical models.
- `secantUpdateExperimental` - This function is the same as `secantUpdate` but is used with a hybrid model and the Experimental element.

The numerical functions, `simpleYield` and `secantUpdate`, are needed to obtain a good idea of how the actual simulation will go, to scaled the ground motion to an appropriate level and not to exceed the control system limits. It is also useful for debugging the Matlab scripts.

The ECxPtargetSwitchEM experimental control class in OpenFresco handles all the communication from the Matlab functions to the Simulink/Stateflow model. ECxPtargetSwitchEM

receives the control mode and the appropriate trial values from the Matlab function. ECxPC-targetSwitchEM does not determine the control mode. The control mode is set by the Matlab function, and then the trial value is calculated. A separate experimental control is necessary for the equilibrium switching strategies because the trial displacement and force pair is not available as with PSC and SSC. The trial forces are calculated directly from the EFC method and not from the trial displacements. So only one of the trial values are compute at each time step. Therefore, ECxPCTargetSwitchEM cannot be used with EPSC and ESSC. No filters are necessary with ECxPCTargetSwitchEM. The offsetting discussed in the previous section is here performed by `simpleYieldExperimental` and `secantUpdatedExperimental` functions. This is why separate functions are required for the numerical model and the hybrid model. ECxPCTargetSwitchEM does not execute the offsetting of the trial values.

### 4.3.3 Simulink/Stateflow model

The predictor-corrector Simulink/Stateflow model for SC hybrid simulation is fairly complex because it has to predict and correct for both control modes. Figure 4.3 shows the Simulink model for SC hybrid simulation. It differs from the the FC Simulink model (Figure 4.3) in that SC Simulink model writes to three different SCRAMNet memory locations, `ctrl modes`, `displ cmds` and `force cmds`. The MTS-STC controller reads 0 in the `ctrl modes` memory location as DC mode and 1 as FC mode.

Figure 4.4 displays what is represented by the xPC Target Hybrid Controller (S1S1) block in Figure 4.3. `ctrlMode` (control mode) and `targVal` (target value) are received from ECxPCTargetSwitchCM or ECxPCTargetSwitchEM. All control strategies use the same Simulink model for SC hybrid simulation. A moving average of the measured displacements and forces are sent back to the OpenFresco experimental control object. Switch and Switch1 blocks feed either `targVal` or the feedback values to the Stateflow model depending on the control mode. If FC mode is used, Switch1 block sets `frc` (force) as `targVal`, and Switch block sets `dsp` (displacement) as feedback displacement. In DC mode, Switch1 block sets the feedback force as `frc` and Switch block sets `dsp` as the `targVal`. If Switch and Switch1 blocks are not sending `targVal` it does not matter what they are sending to the Stateflow model because the Stateflow model eventually zeros out these values.

## Hybrid Controller

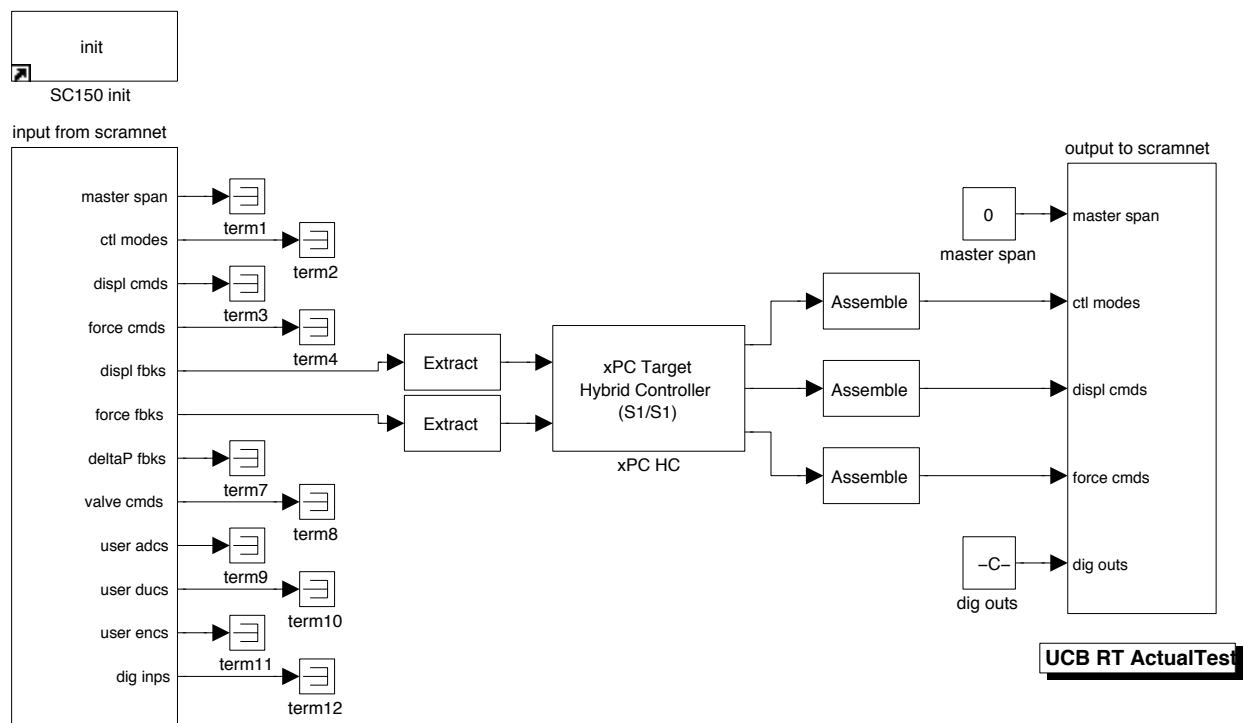


Figure 4.3: SimuLink model for SC hybrid simulation that uses xPC Target real-time workshop with SCRAMNet at *nees@berkeley* lab.

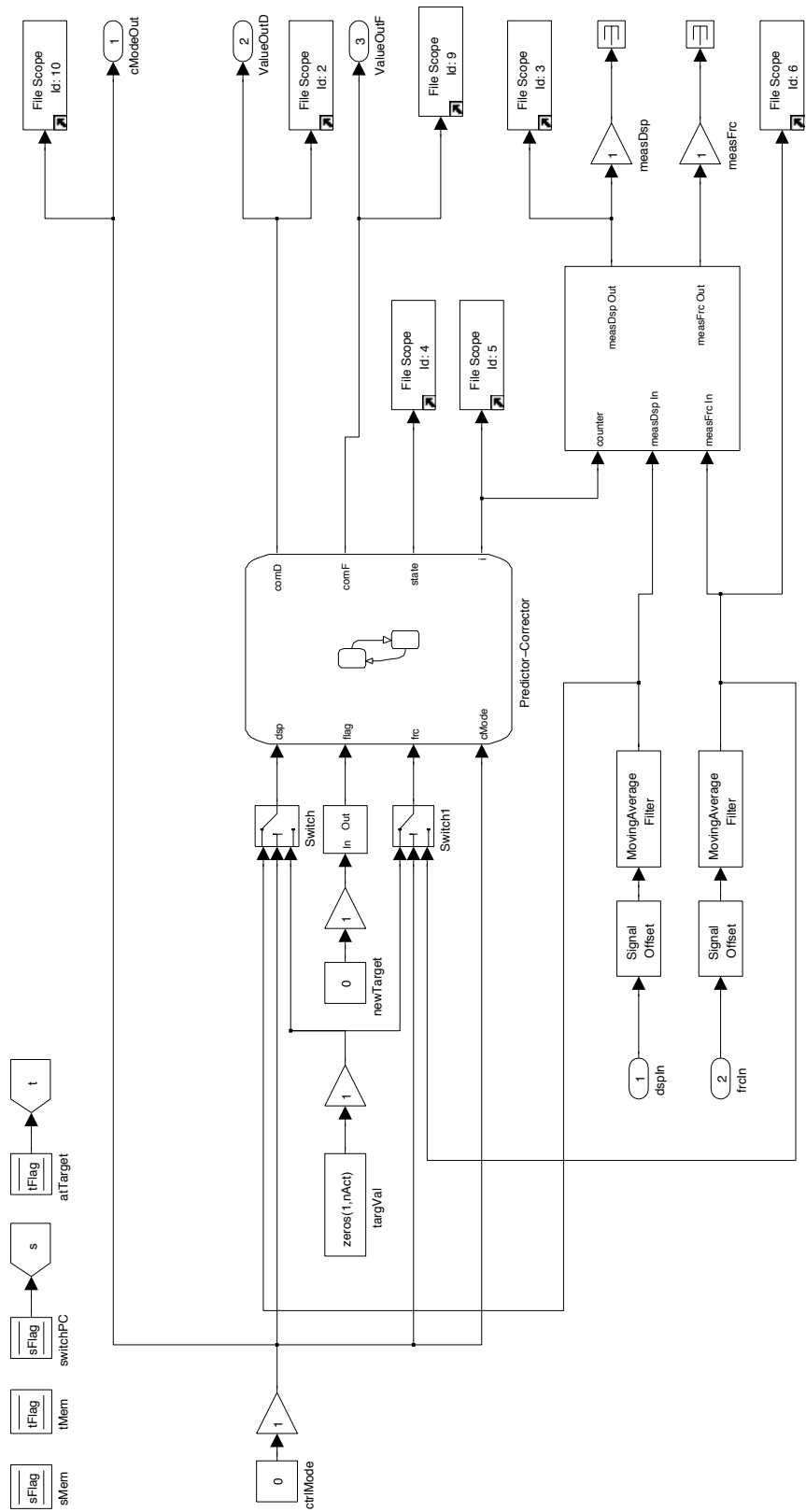


Figure 4.4: This is represented as the xPC Target Hybrid Controller (S1S1) box in Figure 4.3. It contains the Stateflow model and interfaces with OpenFresco.



The Stateflow model (Figure 4.5) does the actual predicting and correcting of the displacements and forces. This Stateflow model essentially links the FC Stateflow model in Figure 3.10 and the DC Stateflow model together. The top portion of the diagram contains CorrectDisp, PredictDisp and AutoSlowDownDisp states for DC mode. The bottom portion contains same three states for FC mode. Along with *flag* variable from its FC counterpart, it has another variable, *cMode* that specifies when to leave its current state and to which state to it should move to next. *flag* lets the Stateflow model know when a new *targVal* is set. *cMode* is the control mode received from OpenFresco.

From the Correct state, it can move to Predict state of the same control mode or of the other control mode. From the Predict state, it can shift to the Correct state of one of the two control modes or to the AutoSlowDown state of the same control mode. From the AutoSlowDown state it can move to the Correct state of the same control mode or the Correct state of the other control mode. These movements are controlled by the *flag* and *cMode* variables.

Each time the Stateflow model enters one of the Correct states, it stores *dsp* and *frc* values using the C functions, *setNewDsp* and *setNewFrc* respectively. These stored values are used for correcting, predicting and auto slowdown. S1S1 hybrid controller uses linear interpolation for correcting and linear extrapolation for predicting and auto slowdown. The Correct states zero out the stored value of the other control mode at the moment when the Stateflow model enters that state. This aids in the smooth switching of the control modes. Because the interpolation and extrapolation use the previous stored value and because the controller offsets with a feedback value when it switches modes, the transition of the control mode does not switch smoothly unless the previous recorded values used for the interpolation and extrapolation are zeroed out. Otherwise, it will use the stored value from before it switched and the new target value to interpolation and extrapolate. This causes Correct and Predict states to function incorrectly since the zero point has changed from the control mode switching. The simulation time step is set to 0.10 second for all SC hybrid simulation tests.

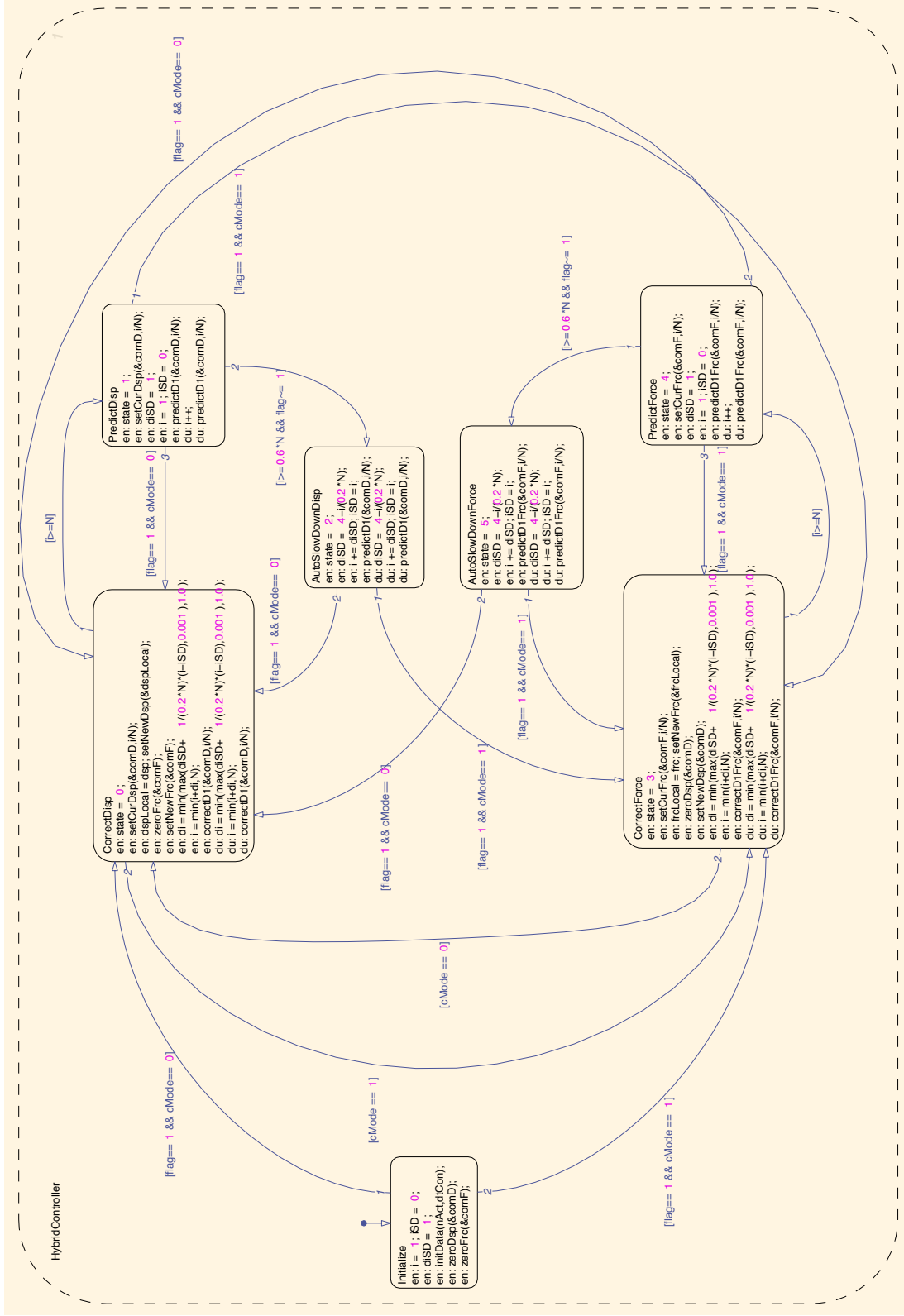


Figure 4.5: The StateFlow predictor-corrector model is represented by the Predictor-Corrector box in Figure 4.4. It predicts and corrects for displacements or forces depending on the control mode. It has an auto slow down to prevent over predicting.

## 4.4 Experimental results and verification

The same one-bay-frame hybrid model from Chapter 3 is deployed for SC hybrid simulation. This model is presented in Figure 2.1, and its properties are provided in Table 3.3. The same OpenFresco configuration from Table 2.1 is utilized as well with the exception of the experimental control class. Instead of `xPCtarget` and `xPCtargetForce`, `xPCtargetSwitchCM` and `xPCtargetSwitchEM` are used. The 1-DOF  $\mu$ -NEES experimental setup (Figure 3.11) with two coupons in the clevis (Figure 3.13) is deployed for these tests.

SC hybrid simulation is only conducted with the one-bay-frame model and the 1-DOF  $\mu$ -NEES setup and not with the 2-DOF model and the 2-DOF  $\mu$ -NEES setup. The control mode switching should be considered for each control DOF separately. It is not intelligent to assume that all the control DOF will switch to the same control at the same time throughout the simulation. When there are more than one control DOF, it is better to use mixed control hybrid simulation. This topic is discussed in the next chapter. Only nonlinear hybrid simulations are conducted to test SC hybrid simulation methods, since SC hybrid simulation is only necessary when the 1-DOF setup starts to soften. Thus, the ground motion is scaled to 40% as with FC nonlinear hybrid simulation results for comparison purposes.

SC hybrid simulation experiments use NME,  $\alpha$ OS, NMF and NMR time integration schemes with the parameters from Table 3.2. These results are compared to the OpenSees numerical results from Figures 3.17 and 3.18 and Table 3.8. As mentioned in the previous chapter, NME and  $\alpha$ OS do not converge with the nonlinear 1-DOF OpenSees numerical model (Figure 3.16). Therefore NME and  $\alpha$ OS experimental results are compared to the NMR numerical results. NME and  $\alpha$ OS time integration schemes use 20 substeps to keep the control system from going unstable. Tables 4.1 and 4.2 show the displacement and force errors from their numerical counterparts respectively. These tables contain the minimum, mean and maximum values of the absolute errors. The figures supporting these error values are provided in Appendix D. The DC and FC error values from Chapter 3 are present in these table for comparison purposes.

For PSC and EPSC hybrid simulations, 4 kips is the switching point and  $f_d$  and  $f_f$  are set at 10% from the predefined switching point.  $f_d = 4.4kips$  and  $f_f = 3.6kips$ . These values are determined from the DC and FC hybrid simulation results. 4 kips is where the specimen just starts to yield For SSC and ESSC hybrid simulation, the secant stiffness value for switching is 25 kip/in. The  $k_d$  and  $k_f$  are set 10% from 25 kip/in. The measured displacements and forces are filtered by 0.005, meaning that if the incremental measured displacement and force at a given time step is not greater than 0.005, it is not used to update the secant stiffness value.

At first inspection of the results in Tables 4.1 and 4.2, it can be seen that the PCS and SSC strategy worked for all the time integration schemes as opposed to the CFC method which only worked for NMF. The CFC method goes unstable for the other three time integration schemes in FC hybrid simulation. EPSC and ESSC strategies still only work with NMF.

The DC errors are the lowest and is slightly better than the SSC error for NME and  $\alpha$ OS but not for NMF and NMR. SSC results are more consistent across the board. PPS The EPSC and ESSC give comparable results to SSC for NMR. The PSC results are also very comparable to the others except for  $\alpha$ OS results. The force errors give very similar results as the displacement errors. The DC errors with NME is the lowest. Figures 4.6 plots the trial force and the control mode for PSC using NME. When the control mode is zero, it is in DC. When the control is greater than zero,

it is in FC. It is in FC mode for the majority of the simulation. There seems to be jerking of trial forces every time the control mode switches from FC to DC. Figure 4.7 shows the control modes with the secant stiffness values. The control is in DC mode majority of the time unlike the previous plot. The secant stiffness value is still somewhat noisy even after filtering the control system noise. The trial forces do not jerk when switching control modes with SSC. This can be verified from the command and measure force plot from the MTS-STC controller in Appendix D. Figure 4.6 and 4.7 are similar to the PSC and SSC results of the other time integration schemes.

Table 4.1: The absolute errors of displacement results from nonlinear SC hybrid simulations for the 1-DOF setup for various time integration method at the FEA level. DC, CFC and EFC results are included for comparison.

Control method	NME (in.)	$\alpha$ OS (in.)	NMF (in.)	NMR (in.)
DC-Min	$2.84 \times 10^{-5}$	$7.30 \times 10^{-7}$	$1.64 \times 10^{-5}$	$1.68 \times 10^{-5}$
DC-Mean	0.0266	0.0288	0.114	0.0711
DC-Max	0.116	0.118	0.357	0.253
CFC-Min	N/A	N/A	$1.14 \times 10^{-5}$	N/A
CFC-Mean	N/A	N/A	0.0409	N/A
CFC-Max	N/A	N/A	0.143	N/A
PSC-Min	$2.12 \times 10^{-5}$	$6.34 \times 10^{-6}$	$2.11 \times 10^{-5}$	$2.49 \times 10^{-5}$
PSC-Mean	0.0410	0.0816	0.0433	0.0422
PSC-Max	0.104	0.300	0.204	0.147
SSC-Min	$8.00 \times 10^{-6}$	$1.40 \times 10^{-5}$	$6.67 \times 10^{-5}$	$6.94 \times 10^{-6}$
SSC-Mean	0.0341	0.0315	0.0397	0.0419
SSC-Max	0.112	0.122	0.177	0.150
EFC-Min	N/A	N/A	N/A	$1.28 \times 10^{-4}$
EFC-Mean	N/A	N/A	N/A	0.0922
EFC-Max	N/A	N/A	N/A	0.280
EPSC-Min	N/A	N/A	N/A	$6.02 \times 10^{-5}$
EPSC-Mean	N/A	N/A	N/A	0.0339
EPSC-Max	N/A	N/A	N/A	0.161
ESSC-Min	N/A	N/A	N/A	$9.82 \times 10^{-6}$
ESSC-Mean	N/A	N/A	N/A	0.0322
ESSC-Max	N/A	N/A	N/A	0.0989

Figure 4.8 plots the control mode and trial force vs. time. for EPSC with NMR. There is no jerking of trial forces when switching control modes with this strategies. Like PSC, the majority of the control mode is in FC. mode The secant stiffness for ESSC with NMR is shown in Figure 4.9. The secant stiffness values fluctuate more than SSC. This is because no filters are used for secant updates. ESSC however does not use the secant to calculate the trial forces. Unlike the SSC counterpart, the control mode is in FC mode for the majority of the simulation.

The control system errors in Table 4.3 illustrates that the control system is tracks better for SC hybrid simulation in FC mode than FC hybrid simulation. SC hybrid simulation has error results

Table 4.2: The absolute errors of force results from nonlinear SC hybrid simulations for the 1-DOF setup for various time integration method at the FEA level. DC, CFC and EFC results are included for comparison.

Control method	NME (kip)	$\alpha$ OS (kip)	NMF (kip)	NMR (kip)
DC-Min	0.00150	$9.60 \times 10^{-4}$	$5.20 \times 10^{-4}$	0.00972
DC-Mean	0.394	0.435	0.804	1.28
DC-Max	2.03	2.08	4.05	5.42
CFC-Min	N/A	N/A	$6.65 \times 10^{-4}$	N/A
CFC-Mean	N/A	N/A	0.593	N/A
CFC-Max	N/A	N/A	2.34	N/A
PSC-Min	$1.14 \times 10^{-4}$	$4.83 \times 10^{-4}$	$6.45 \times 10^{-4}$	$9.37 \times 10^{-4}$
PSC-Mean	0.486	0.794	0.561	0.586
PSC-Max	3.26	6.24	2.31	2.19
SSC-Min	$2.46 \times 10^{-4}$	$3.00 \times 10^{-5}$	$2.53 \times 10^{-4}$	$1.27 \times 10^{-4}$
SSC-Mean	0.407	0.447	0.582	0.601
SSC-Max	2.13	2.17	2.55	2.23
EFC-Min	N/A	N/A	N/A	0.00700
EFC-Mean	N/A	N/A	N/A	1.26
EFC-Max	N/A	N/A	N/A	5.77
EPSC-Min	N/A	N/A	N/A	$7.06 \times 10^{-4}$
EPSC-Mean	N/A	N/A	N/A	0.771
EPSC-Max	N/A	N/A	N/A	3.03
ESSC-Min	N/A	N/A	N/A	0.0012
ESSC-Mean	N/A	N/A	N/A	0.702
ESSC-Max	N/A	N/A	N/A	2.79

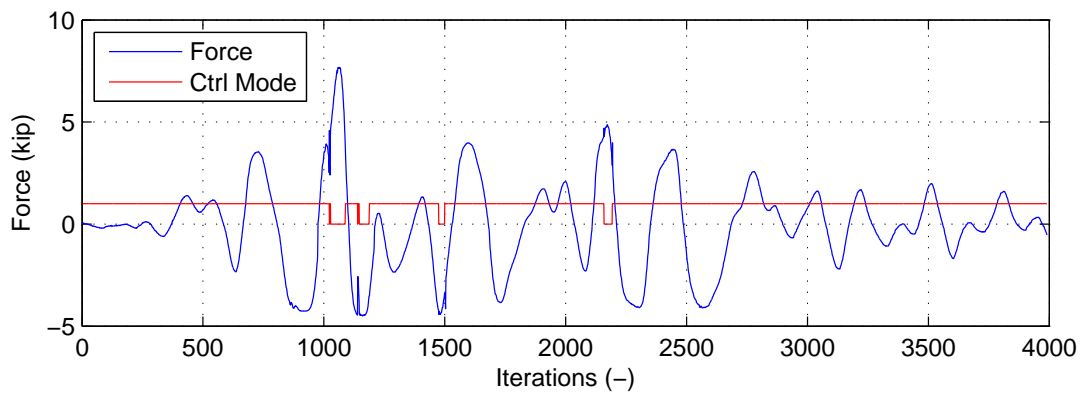


Figure 4.6: Plot of control mode and trial forces vs. time for PSC with NME.

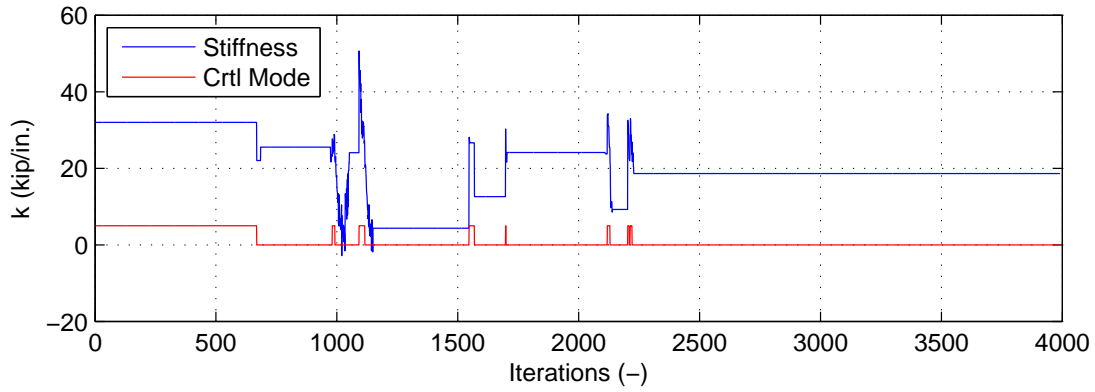


Figure 4.7: Plot of control mode and secant stiffness value vs. time for PSC with NME.

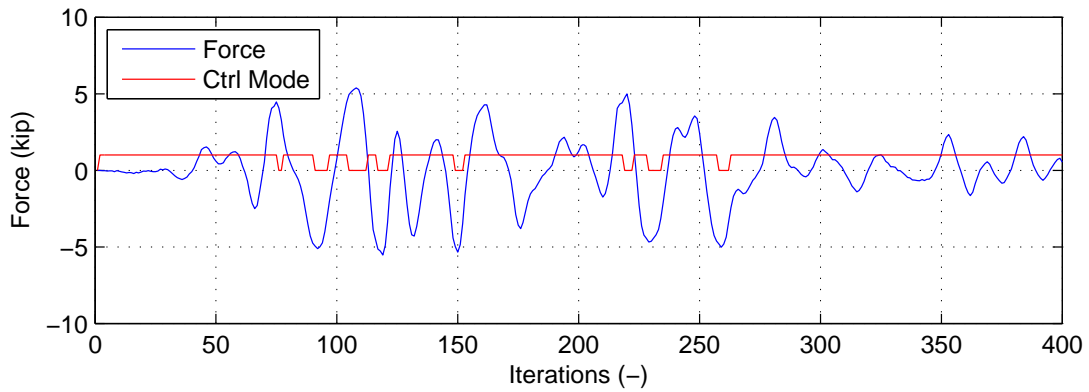


Figure 4.8: Plot of control mode and trial forces vs. time for EPSC with NMR.

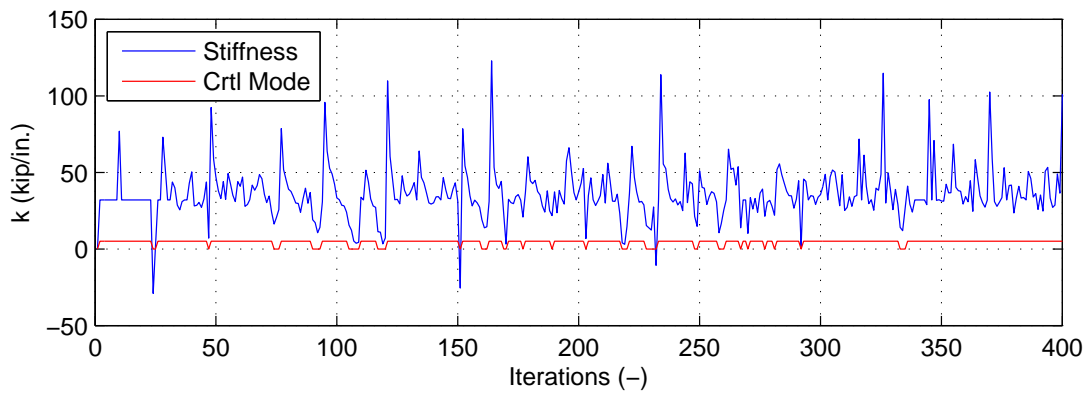


Figure 4.9: Plot of control mode and secant stiffness value vs. time for ESSC with NMR.

for both DC and FC modes. The tracking performance of SC hybrid simulation in DC mode is comparable to DC hybrid simulation. The error values in both DC and FC modes for DC, FC and SC hybrid simulation results are similar to one another. The FC mode errors are about 34 times the DC errors. 34 kip/in. is the linear stiffness of the specimen. All the command displacement and force plots for NME,  $\alpha$ OS, NMF in Appendix D are ideal for hybrid simulation. The command curves are fairly smooth. The command displacement and force plots for NMR for all types of hybrid simulation are not ideal because the curves are very jagged. This is a common behavior of NMI and NMR since they use Newton's method to solve the equation of motion that converges quadratically. The first iteration typically produces large command values and the subsequent values are proportionally smaller. It is worth noting that all the MTS-STS displacement and force error plots exhibit a spike in the measured values whenever the controller switches control modes.

In summary, the experimental results first show that SC hybrid simulation is achievable and that the switching of the control modes does not adversely affect the results. PSC, SSC, EPSC and ESSC results are comparable to the DC results and better than the FC results. The secant stiffness based switching strategies produce marginally better results than the trial force based switching strategy at the FEA level. PSC and SSC hybrid simulation methods are more robust in that it does not go unstable for NME,  $\alpha$ OS and NMR as CFC hybrid simulation method do even though both are using the identical method for computing the trial forces. At the control system level, the tracking overall, in both control modes for all types of hybrid simulation, performs equally well with the 1-DOF  $\mu$ -NEES experimental setup. However, the command values generated by the the NMR time integration scheme make it not ideal for hybrid simulation.

Table 4.3: The absolute errors of nonlinear SC hybrid simulations for the 1-DOF setup for various time integration schemes at the control system level. DC, CFC and EFC results are included for comparison.

<b>Control method</b>	<b>NME</b>	<b><math>\alpha</math>OS</b>	<b>NMF</b>	<b>NMR</b>
DC-Min (in.)	$1.38 \times 10^{-9}$	$7.86 \times 10^{-9}$	$7.94 \times 10^{-9}$	$3.14 \times 10^{-9}$
DC-Mean (in.)	$8.53 \times 10^{-4}$	$8.20 \times 10^{-4}$	0.00167	$8.66 \times 10^{-4}$
DC-Max (in.)	0.00513	0.00526	0.0129	0.0181
CFC-Min (kip)	N/A	N/A	0	N/A
CFC-Mean (kip)	N/A	N/A	0.0328	N/A
CFC-Max (kip)	N/A	N/A	0.285	N/A
PSC-Min (in.)	$1.33 \times 10^{-7}$	$5.38 \times 10^{-7}$	$8.62 \times 10^{-9}$	$4.93 \times 10^{-9}$
PSC-Mean (in.)	0.00144	0.00277	$9.25 \times 10^{-4}$	$7.83 \times 10^{-4}$
PSC-Max (in.)	0.0480	0.0788	0.0219	0.0403
PSC-Min (kip)	$2.38 \times 10^{-7}$	$2.38 \times 10^{-7}$	$1.79 \times 10^{-7}$	0
PSC-Mean (kip)	0.0452	0.0415	0.0278	0.0276
PSC-Max (kip)	0.565	1.29	0.371	0.770
SSC-Min (in.)	$1.24 \times 10^{-9}$	$6.57 \times 10^{-9}$	$1.19 \times 10^{-8}$	$1.90 \times 10^{-7}$
SSC-Mean (in.)	0.00110	0.00124	0.00108	0.00248
SSC-Max (in.)	0.0126	0.00532	0.00957	0.0152
SSC-Min (kip)	$3.73 \times 10^{-8}$	$2.24 \times 10^{-8}$	$5.96 \times 10^{-8}$	0
SSC-Mean (kip)	0.0393	0.0423	0.0272	0.0250
SSC-Max (kip)	0.183	0.141	0.358	0.494
EFC-Min (kip)	N/A	N/A	N/A	0
EFC-Mean (kip)	N/A	N/A	N/A	0.0282
EFC-Max (kip)	N/A	N/A	N/A	0.640
EPSC-Min (in.)	N/A	N/A	N/A	$9.00 \times 10^{-8}$
EPSC-Mean (in.)	N/A	N/A	N/A	0.00152
EPSC-Max (in.)	N/A	N/A	N/A	0.0162
EPSC-Min (kip)	N/A	N/A	N/A	0
EPSC-Mean (kip)	N/A	N/A	N/A	0.0337
EPSC-Max (kip)	N/A	N/A	N/A	0.141
ESSC-Min (in.)	N/A	N/A	N/A	$3.34 \times 10^{-7}$
ESSC-Mean (in.)	N/A	N/A	N/A	0.00159
ESSC-Max (in.)	N/A	N/A	N/A	0.01833
ESSC-Min (kip)	N/A	N/A	N/A	0
ESSC-Mean (kip)	N/A	N/A	N/A	0.0336
ESSC-Max (kip)	N/A	N/A	N/A	0.181



# Chapter 5

## Mixed control hybrid simulation

This chapter discusses and explores mixed control (MC) hybrid simulation. The term MC hybrid simulation is clearly defined. The challenges of MC hybrid simulation are explained. The motivation for it is present along with the previous research. Then the methods for MC hybrid simulation are developed. The implementation of these methods using OpenSees, OpenFresco and Simulink/Stateflow model is described. This chapter concludes with a discussion of experimental results from MC hybrid simulations tests using the 2-DOF  $\mu$ -NEES setup at *nees@berkeley* lab.

### 5.1 Introduction

#### 5.1.1 Problem definition

The term mixed control hybrid simulation refers to hybrid simulation that is conducted with the control system in multiple control modes simultaneously. It is analogous to switch control hybrid simulation from Chapter 4 in that mixed control hybrid simulation is switch control hybrid simulation with more than one control system DOF. For example, if an experimental setup has two actuator and both are switching independently of each other between DC and FC modes, then this is referred to as mixed control hybrid simulation. At any given time, one actuator maybe in DC mode and the other actuator maybe in FC mode. In the case of switch control hybrid simulation, there is only 1 control system DOF. That DOF is either in DC or FC mode. There can not be mixing of control modes for this setup.

The two terms, mixed control and switch control, are often times used interchangeably. However, a strict distinction is made between the two in this chapter and the previous chapter. Switch control hybrid simulation refers only to 1-DOF experimental setups switching between control modes. Mixed control hybrid simulation refer multiple control system DOF switching independently of each other. There may also be a hybrid simulation test where certain actuators are in DC and others are in FC, and these actuator never switch control modes. This is also considered mixed control hybrid simulation.

### 5.1.2 Motivation

The need for MC hybrid simulation arises when the specimen behaves very differently, depending on the direction it is being loaded. These types of specimens are common in structural testing. A beam-column is a very common structural component in buildings and bridges. A beam-column is stiffer in the axial direction than in the direction orthogonal to the axial direction. When seismic isolators are tested experimentally, they are typically loaded vertically in FC and horizontally in DC. Often times when these structural components are tested using hybrid simulation, the vertical or the axial force is kept constant throughout the test, and displacements are imposed on the specimen horizontally.

Keeping the vertical force constant throughout the test is not realistic. This force should vary during the test. The vertical force is kept constant because the vertical load is difficult to calculate and control. The vertical force is difficult to calculate because the horizontal displacements depend on the vertical force and vice versa. When the vertical force is varied in hybrid simulation, the calculation of this force is uncoupled from the horizontal displacement calculation. The difficulties of controlling the force is explained in Section 3.1.1.

### 5.1.3 Previous research

Few researchers have conducted MC hybrid simulation. Two works are surveyed here. The first one is the one by Pan, et al. [18]. The same experiment on HDBR mentioned in the previous two chapters. They conducted MC hybrid simulation with the vertical DOF switching between DC and FC and the horizontal DOF in DC. The vertical trial force was uncoupled from the horizontal trial displacement and calculated independently of each other. The initial stiffness value was used to calculate the vertical trial force at each time step while the horizontal trial displacement at each time step was calculated using DC hybrid simulation method. Both trial values then were applied to HDBR simultaneously.

The second one is by Nakata, et al. [17] who performed a sort of a MC hybrid simulation to study the P- $\Delta$  effect on an axially loaded beam-column. They used the Load and Boundary Condition Box at the University of Illinois at Urbana-Champaign as a part of their experimental setup. The target axial force was applied to the specimen in an iterative fashion through the corresponding target displacement. The target displacement was calculated using the axial stiffness of the beam-column which was updated at each iteration using the Broyden update [2] from the measured displacements and forces. This is not a true MC simulation in a sense that the force DOF was not controlled in FC but in DC to achieve the target force. The trial force was converted to trial displacement and then the trial displacements for all controller DOF was imposed on the specimen. This was performed iteratively until the measured force equaled the trial force calculated at the beginning of the time step.

## 5.2 Methods for MC hybrid simulation

The methods for MC hybrid simulation are a combination of both CFC and SC hybrid simulation methods. MC hybrid simulation uses one of the CFC methods from Section 3.2.1 to convert

the trial displacements ( $\check{\mathbf{u}}$ ) vector to trial forces ( $\check{\mathbf{f}}$ ) vector. It then uses the one of the switching strategies from Section 4.2.1 to determine the control mode of each control DOF independently. A switching strategy for each control DOF is determined by the researcher before the start of a test. This means that a different strategy can be assigned to each of the control DOFs. Thus, it is possible to have multiple switching strategies running simultaneously during a test.

The EFC methods is not used for obtaining the trial force for MC hybrid simulation. It is possible to run both the displacement based time integration scheme and the force based time integration scheme simultaneously to calculate  $\check{\mathbf{u}}$  and  $\check{\mathbf{f}}$  and the control modes. There is problem with how measured values  $\bar{\mathbf{u}}$  and  $\bar{\mathbf{f}}$  should be fed back to the NMR time integration scheme. The 2-DOF  $\mu$ -NEES experimental setup (Figure 3.20) is used to illustrate this obstacle. After the time integration schemes calculate  $\check{\mathbf{u}}$  and  $\check{\mathbf{f}}$ , a switching strategy determines that the top actuator is in DC mode and the bottom actuator is in FC mode. The control system imposes  $\check{u}_{bottom}$  and  $\check{f}_{top}$ . It then feeds back the corresponding measured values  $\bar{\mathbf{u}}$  and  $\bar{\mathbf{f}}$  to the computational drive. The computational driver must now decide what measured values to feed back to the time integration schemes. If it feeds back  $\bar{\mathbf{f}}$  vector to the displacement base time integration scheme, then  $\bar{f}_{bottom}$  may not correspond well to the trial displacement for the bottom actuator it has calculated. If  $\bar{f}_{bottom}$  is modified somehow to be more consistent with  $\check{u}_{bottom}$ , then the incorrect state of the specimen is passed to the displacement base time integration scheme. The same problem exist for  $\bar{\mathbf{u}}$  and the force base time integration scheme. Therefore, the EFC is not implemented as a part of MC hybrid simulation.

## 5.3 Implementation

The implementation of MC hybrid simulation is very similar to the implementation of PSC and SSC hybrid simulation from Sections 4.3.1. There are two major components necessary for MC hybrid simulation. The first one is the OpenFresco, especially the experimental control component. The second component is the Simulink/Stateflow model for MC hybrid simulation. These interact with one another and the computational drive to make MC hybrid simulation properly function.

### 5.3.1 OpenFresco MC experimental control

OpenFresco's role in MC hybrid simulation is illustrated in Figure 5.1. Like FC and SC hybrid simulation, Openfresco receives the trial deformations ( $\hat{\mathbf{u}}_n$ ) from the computational drive. OpenFresco converts  $\hat{\mathbf{u}}_n$  to  $\check{\mathbf{f}}_n$  in the experimental control class. The experiment control used for the MC hybrid simulation is the xPCTargetMixedCM experimental control. xPCTargetMixedCM is a combination of xPCTargetForce and xPCTargetSwitchCM. Like xPCTargetForce, it passes  $\check{\mathbf{u}}_n$  to the *ExperimentalSignalFilter* class. *ExperimentalSignalFilter* uses the *ExperimentalTangentStiff* class to convert  $\check{\mathbf{u}}_n$  to  $\check{\mathbf{f}}_n$ . xPCTargetMixedCM uses the specified switching strategies to determine the control mode of each of the control system DOFs and assembles the control mode vector.

As Figure 5.1 shows, it sends these three vector the control system. The control system imposes displacements or forces as specified by the control mode vector. After the control system successfully imposes the command values, It reads the measured displacements ( $\bar{\mathbf{u}}_n$ ) and forces

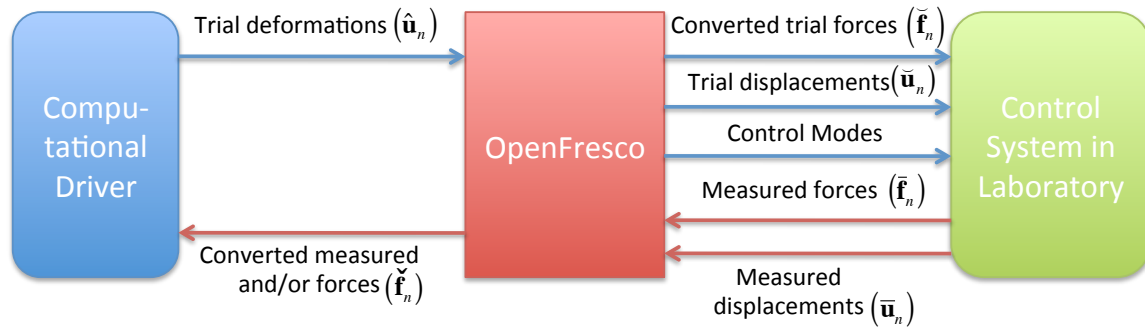


Figure 5.1: MC hybrid simulation components and their processes.

$(\tilde{\mathbf{f}}_n)$  and sends them back to `xPXtargetMixedCM`. `xPXtargetMixedCM` converts  $\tilde{\mathbf{u}}_n$  to forces. It then assembles  $\check{\mathbf{f}}_n$  and performs a coordinate transformation. It is sent back to the computational driver. In the assembling of  $\check{\mathbf{f}}_n$ , `xPXtargetMixedCM` places the actual measured forces for the control system DOFs in DC mode and the converted measured forces for the control system DOFs in FC mode. Once the computational driver receives the  $\check{\mathbf{f}}_n$ , it moves to the next time  $n + 1$ . This process repeats until the simulation is complete.

### 5.3.2 Simulink/Stateflow model

The Simulink/Stateflow model presented in this section is for MC hybrid simulation where the top actuator is in DC mode and the bottom actuator switches between DC and FC modes. It is straightforward to extend this Simulink/Stateflow model to have both actuators switch between DC and FC modes. It would add to the complexity of the Stateflow model in Figure 5.4. The 2-DOF  $\mu$ -NEES experimental setup for which this Simulink/Stateflow model is designed does not require that the top actuator be in FC mode because it is relatively soft and perform much better in DC mode.

The Simulink model in Figure 5.2 is nearly identical to the Simulink model for SC hybrid simulation (Figure 4.3). Only visible difference it is that it uses `xPC Target Hybrid Controller (M1/M1)` instead of the hybrid controller for SC hybrid simulation. The other difference that is not apparent in Figure 5.2 is that the Extract and Assemble blocks are extracting and assembling 2 dimensional vectors from and to the the SCRAMNet memory location instead of just a scalar value as in the SC model.

`xPC Target Hybrid Controller (M1M1)` in Figure 5.3 receives both the target displacements (*targDsp*), the target forces (*targFrc*) and the control modes (*ctrlMode*) from the `xPCtargetMixedCM` experimental control. These values along with the *newTarget* flag are passed into the Predictor-Corrector Stateflow model. As with hybrid controllers for FC and SC hybrid simulation, the raw measured displacements and forces are passed through a moving average filter before being fed back to OpenFresco.

## Hybrid Controller

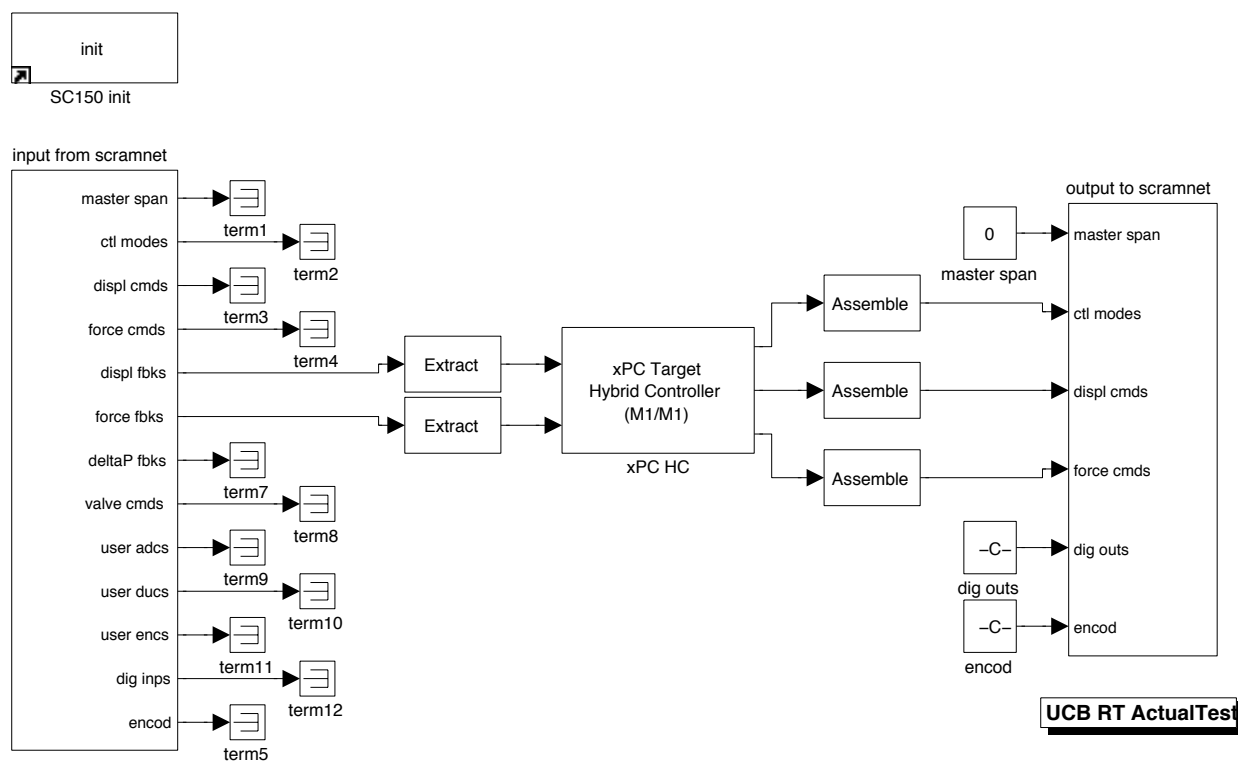


Figure 5.2: SimuLink model for MC hybrid simulation that uses xPC Target real-time workshop with SCRAMNet at *nees@berkeley* lab.

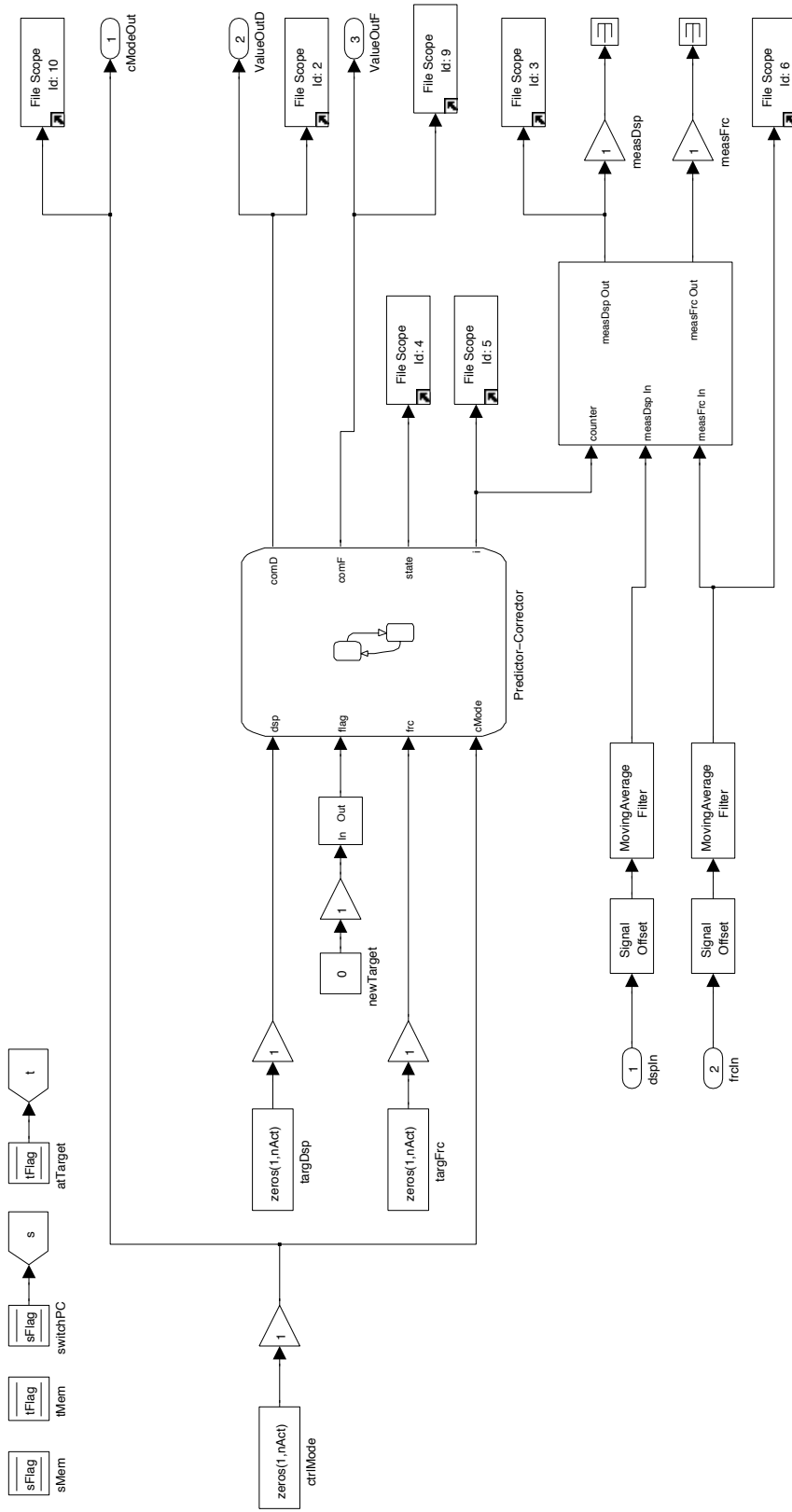


Figure 5.3: This is represented as the xPC Target Hybrid Controller (M1M1) box in Figure 4.3. It contains the Stateflow model and interfaces with OpenFresco.

The top three states (CorrectDisp, PredictDisp and AutoSlowDownDisp) in the Stateflow model (Figure 5.4) perform the correcting and predicting of the displacement commands that get written to the SCRAMNet memory for the MTS-STC controller. These states are in essence like the three displacement states in SC Stateflow model. In these states, all DOF are in DC mode.

The CorrectForce, PredictForce and AutoSlowDown states at the bottom portion of Figure 5.4 are more complicated. They are responsible for predicting and correcting for the displacements of the top actuator and for the forces of the bottom actuator. When the Stateflow model enters the CorrectForce and PredictForce States from any one of the displacement states, the MTS-STC controller uses the last measured displacement for the top actuator and the last measure force for the bottom actuator as the new zero points. This is the same controller behavior that is discussed in Section 4.3.3 for the SC Stateflow model, but it is more complicated because now it involves two separate control DOFs in different control modes.

xPCtargetMixedCM keeps these last measured values in memory and offsets the subsequent command values with this offset. The CorrectForce state sets the displacement of the bottom actuator to 0 and stores that as the value to use for correcting and predicting when the Stateflow model moves to the CorrectDisp and PredictDisp states. Otherwise, the interpolation and the extrapolation do not behave as intended since the displacement commands for the bottom actuator when it switches to from FC mode to DC mode are offset values. The *flag* variable which indicates when a new target values have arrived and *cMode*[0], the control mode of the bottom actuator, dictate the switching of the states. The simulation time step is set to 0.10 second for all MC hybrid simulation tests.

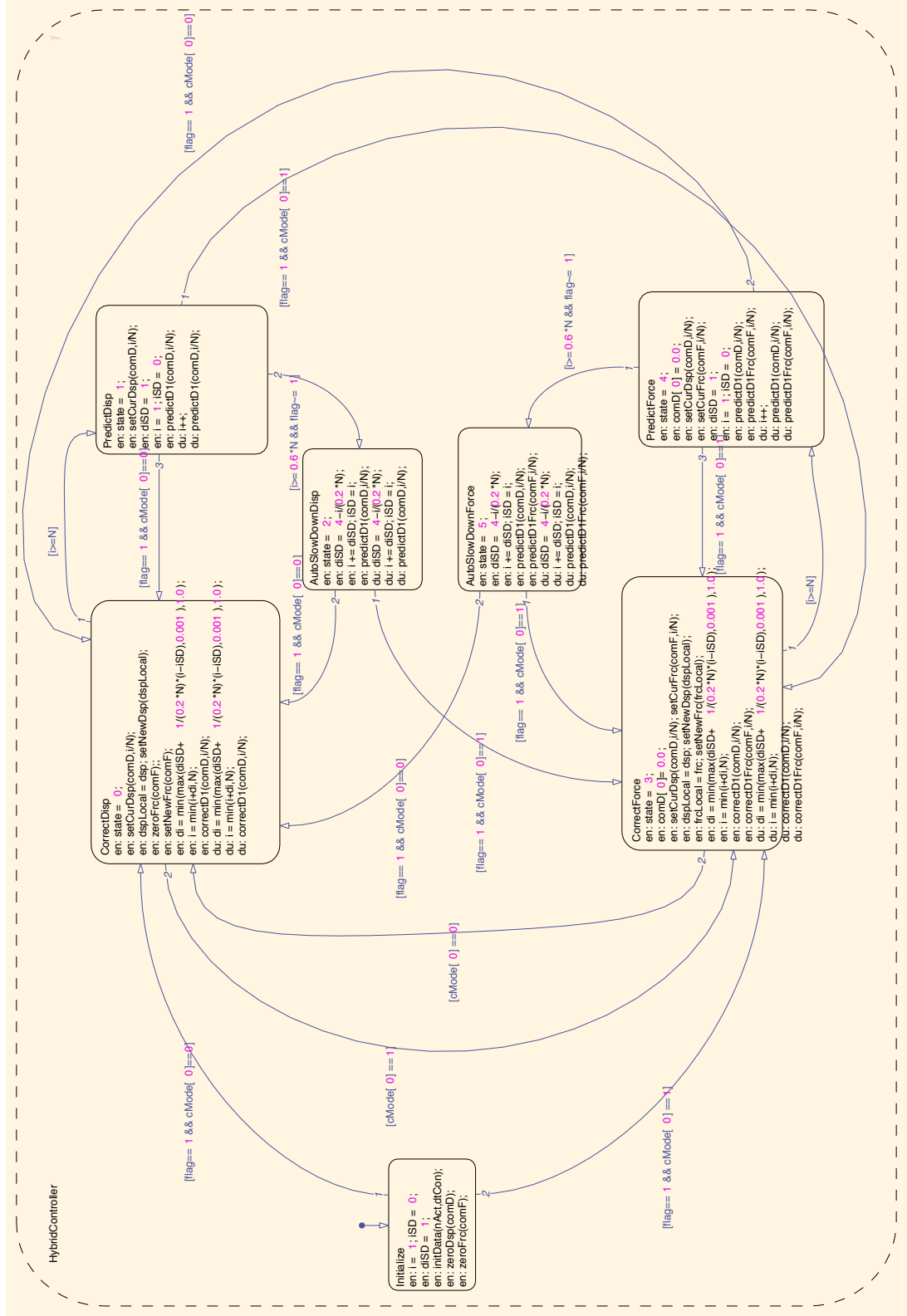


Figure 5.4: The StateFlow predictor-corrector model is represented by the Predictor-Corrector box in Figure 5.3. It predicts and corrects for displacements or forces depending on the control mode. It has an auto slow down to prevent over predicting.



## 5.4 Experimental results and verification

MC hybrid simulation is conducted using the 2-DOF model (Figure 3.21 (right)) and the 2-DOF  $\mu$ -NEES experimental setup (Figure 3.20) from Chapter 3. The first 400 recorded points of the El Centro ground motion is scaled to 50%. The OpenFresco configuration is the same as the ones in Table 3.13 except that xPCtargetMixedCM is deployed instead of xPCtargetForce. The top actuator in the 2-DOF  $\mu$ -NEES setup is always in DC mode, and the bottom switches between DC mode and FC mode using the either one of two switching strategies in Algorithm 4.2 and 4.4. Thus two types of MC hybrid simulations are tested.

NME,  $\alpha$ OS, NMF and NMR time integration schemes are used with the parameters from Table 3.2. For each time integration scheme, the MC results are compared to the OpenSees numerical results from Section 3.4.2.3. As mentioned before the NME and  $\alpha$ OS time integration schemes do not converge numerically with the 2-DOF OpenSees model. Therefore the experimental results of these two time integration schemes are compared to the NMI numerical results. The MC methods that use CFC:Krylov and CFC:Transpose methods for FC cause the MTS-STS control to become unstable. Thus, those results are not presented in the subsequent sections.

The incremental measured displacement and force vectors do not update the tangent stiffness matrix if the 2-norms of these vectors are not both greater than 0.01. This filtering value is used after testing 0.001, 0.005 and 0.02. 0.01 seems to update the tangent frequently without being overtly effected by the noise from the control system. The 1-norm and the  $\infty$ -norm do not perform well during tests.

### 5.4.1 MC hybrid simulation trial force based switching

The results in this section are from MC hybrid simulation using the force limit switching strategy for the bottom actuator only (PMC). The force is limit is set at 5 kips with  $f_d = 5.5kip$  and  $f_f = 4.5kip$ . These results are compare to the DC and CFC hybrid simulation results from Section 3.4.2.3 in Chapter 3. Tables 5.1 and 5.2 contain the absolute value of the displacement errors between the experimental results for a given time integration scheme and it's numerical counterpart of the time history analysis. The plots from which these tables are calculated is on Section E.1 of Appendix E. In Chapter 3, the nodal displacement results show that in general FC method are better than the DC method. The FC methods capture the drift better than the DC method. Both give pretty similarly shaped curves. The PMC displacement error overall are worse than even the DC results. Even though the shape of the displace-time history curves are like the DC and CFC curves, it drifts in the opposite direction. The only one that is comparable to the DC and CFC results are the PMC:Broyden displacement errors.

Table 5.3 and 5.4 show the absolute value of the force errors for Nodes 1 and 2 respectively. PMC give better results than the DC and CFC methods for NME and  $\alpha$ OS but not for NMF and NMR. Even though PMC method does not capture the drift correctly, it produces forces that match the numerical results relatively well.

The tracking of DC and FC hybrid simulation without switching control modes are marginally better than the tracking of the bottom actuator during PMC hybrid simulation as seen in Tables 5.5 and 5.6. One reason is the tracking is not quite as good for PMC is that when the MTS-STS controller switches control modes, the measured values spike momentarily. The tracking of top

Table 5.1: Node 1 absolute errors of displacement results from nonlinear PMC hybrid simulations for the 2-DOF setup for various time integration method at the FEA level.

Control method	NME (in.)	$\alpha$ OS (in.)	NMF (in.)	NMR (in.)
PMC:Broyden-Min	$2.97 \times 10^{-7}$	$7.79 \times 10^{-8}$	$1.05 \times 10^{-5}$	N/A
PMC:Broyden-Mean	0.0882	0.165	0.125	N/A
PMC:Broyden-Max	0.247	0.326	0.355	N/A
PMC:BFGS-Min	$2.97 \times 10^{-7}$	$1.20 \times 10^{-7}$	$3.2 \times 10^{-5}$	$1.96 \times 10^{-5}$
PMC:BFGS-Mean	0.301	0.411	0.319	0.263
PMC:BFGS-Max	0.593	0.710	0.618	0.585
PMC:Intrinsic-Min	$4.52 \times 10^{-8}$	$6.56 \times 10^{-8}$	$7.64 \times 10^{-6}$	$1.23 \times 10^{-5}$
PMC:Intrinsic-Mean	0.229	0.207	0.167	0.156
PMC:Intrinsic-Max	0.463	0.399	0.415	0.330

Table 5.2: Node 2 absolute errors of displacement results from PMC nonlinear hybrid simulations for the 2-DOF setup for various time integration method at the FEA level.

Control method	NME (in.)	$\alpha$ OS (in.)	NMF (in.)	NMR (in.)
PMC:Broyden-Min	$2.98 \times 10^{-7}$	$1.45 \times 10^{-7}$	$2.49 \times 10^{-6}$	N/A
PMC:Broyden-Mean	0.688	0.926	0.827	N/A
PMC:Broyden-Max	1.63	1.78	2.04	N/A
PMC:BFGS-Min	$2.98 \times 10^{-7}$	$3.23 \times 10^{-7}$	$7.23 \times 10^{-5}$	$3.75 \times 10^{-5}$
PMC:BFGS-Mean	1.49	2.14	1.92	1.73
PMC:BFGS-Max	2.80	3.78	3.43	3.44
PMC:Intrinsic-Min	$6.36 \times 10^{-8}$	$2.80 \times 10^{-7}$	$2.65 \times 10^{-6}$	$3.32 \times 10^{-5}$
PMC:Intrinsic-Mean	1.30	1.10	0.715	1.04
PMC:Intrinsic-Max	2.39	2.04	1.86	1.97

Table 5.3: Node 1 absolute errors of force results from nonlinear PMC hybrid simulations for the 2-DOF setup for various time integration method at the FEA level.

Control method	NME (kip)	$\alpha$ OS (kip)	NMF (kip)	NMR (kip)
PMC:Broyden-Min	$1.37 \times 10^{-4}$	$7.10 \times 10^{-6}$	0.00152	N/A
PMC:Broyden-Mean	0.769	0.764	1.79	N/A
PMC:Broyden-Max	3.53	3.85	6.97	N/A
PMC:BFGS-Min	$5.30 \times 10^{-5}$	$6.00 \times 10^{-6}$	0.00248	$3.44 \times 10^{-4}$
PMC:BFGS-Mean	0.803	0.708	1.47	1.12
PMC:BFGS-Max	3.34	3.07	5.02	3.62
PMC:Intrinsic-Min	$1.14 \times 10^{-4}$	$6.06 \times 10^{-5}$	0.00153	0.00302
PMC:Intrinsic-Mean	0.865	0.769	1.56	1.15
PMC:Intrinsic-Max	3.82	3.70	5.20	4.22

Table 5.4: Node 2 absolute errors of force results from nonlinear PMC hybrid simulations for the 2-DOF setup for various time integration method at the FEA level.

Control method	NME (kip)	$\alpha$ OS (kip)	NMF (kip)	NMR (kip)
PMC:Broyden-Min	$1.00 \times 10^{-6}$	$7.10 \times 10^{-6}$	$4.02 \times 10^{-4}$	N/A
PMC:Broyden-Mean	0.175	0.202	0.300	N/A
PMC:Broyden-Max	0.555	0.781	1.08	N/A
PMC:BFGS-Min	$3.00 \times 10^{-7}$	$1.05 \times 10^{-5}$	$8.82 \times 10^{-4}$	$6.00 \times 10^{-6}$
PMC:BFGS-Mean	0.167	0.189	0.239	0.196
PMC:BFGS-Max	0.671	0.871	0.745	0.618
PMC:Intrinsic-Min	$6.86 \times 10^{-6}$	$2.00 \times 10^{-5}$	$3.00 \times 10^{-5}$	$9.98 \times 10^{-5}$
PMC:Intrinsic-Mean	0.170	0.182	0.254	0.196
PMC:Intrinsic-Max	0.626	0.679	0.867	0.611

actuator in DC mode during PMC hybrid simulation performs about the same as the DC hybrid simulation results.

Table 5.5: Bottom actuator absolute errors of nonlinear PMC hybrid simulations for the 2-DOF setup for various time integration schemes at the control system level.

Control method	NME	$\alpha$ OS	NMF	NMR
PMC:Broyden-Min (in.)	$9.37 \times 10^{-8}$	$1.22 \times 10^{-8}$	$1.30 \times 10^{-8}$	N/A
PMC:Broyden-Mean (in.)	$7.65 \times 10^{-4}$	$7.70 \times 10^{-4}$	$8.04 \times 10^{-4}$	N/A
PMC:Broyden-Max (in.)	0.00333	0.00378	0.00900	N/A
PMC:Broyden-Min (kip)	0	0	0	N/A
PMC:Broyden-Mean (kip)	0.0330	0.138	0.0317	N/A
PMC:Broyden-Max (kip)	0.185	0.837	0.211	N/A
PMC:BFGS-Min (in.)	$4.99 \times 10^{-9}$	$6.79 \times 10^{-9}$	$1.51 \times 10^{-7}$	$6.52 \times 10^{-6}$
PMC:BFGS-Mean (in.)	$6.65 \times 10^{-4}$	$6.55 \times 10^{-4}$	$9.03 \times 10^{-4}$	0.00216
PMC:BFGS-Max (in.)	0.00386	0.00325	0.00332	0.00515
PMC:BFGS-Min (kip)	0	0	0	0
PMC:BFGS-Mean (kip)	0.0286	0.0764	0.0507	0.0368
PMC:BFGS-Max (kip)	0.253	0.306	0.219	0.277
PMC:Intrinsic-Min (in.)	$4.82 \times 10^{-8}$	$5.02 \times 10^{-8}$	$2.50 \times 10^{-8}$	$1.66 \times 10^{-8}$
PMC:Intrinsic-Mean (in.)	$8.84 \times 10^{-4}$	$8.38 \times 10^{-4}$	0.00140	$7.06 \times 10^{-4}$
PMC:Intrinsic-Max (in.)	0.00375	0.00354	0.00557	0.00324
PMC:Intrinsic-Min (kip)	0	0	0	0
PMC:Intrinsic-Mean (kip)	0.0281	0.0472	0.0563	0.0419
PMC:Intrinsic-Max (kip)	0.178	0.227	0.398	0.323

Figure 5.5 shows the force-time history curve of the bottom actuator for NME using the PMC:Broyden and the control mode during the hybrid simulation test. When the control mode is zero, the MTS-STC controller is in DC mode and when it is not zero it is in FC mode. This

Table 5.6: Top actuator absolute errors of nonlinear PMC hybrid simulations for the 2-DOF setup for various time integration schemes at the control system level.

Control method	NME	$\alpha$ OS	NMF	NMR
PMC:Broyden-Min (in.)	$1.41 \times 10^{-8}$	$1.25 \times 10^{-9}$	$8.67 \times 10^{-11}$	N/A
PMC:Broyden-Mean (in.)	$7.38 \times 10^{-4}$	$9.33 \times 10^{-4}$	$6.20 \times 10^{-4}$	N/A
PMC:Broyden-Max (in.)	0.00587	0.0475	0.00570	N/A
PMC:BFGS-Min (in.)	$1.68 \times 10^{-9}$	$4.84 \times 10^{-9}$	$1.06 \times 10^{-8}$	$1.98 \times 10^{-11}$
PMC:BFGS-Mean (in.)	$5.53 \times 10^{-4}$	$5.67 \times 10^{-4}$	$5.40 \times 10^{-4}$	$5.57 \times 10^{-4}$
PMC:BFGS-Max (in.)	0.00642	0.00497	0.00528	0.00684
PMC:Intrinsic-Min (in.)	$9.74 \times 10^{-9}$	$5.14 \times 10^{-9}$	$2.72 \times 10^{-9}$	$3.32 \times 10^{-9}$
PMC:Intrinsic-Mean (in.)	$7.47 \times 10^{-4}$	$6.12 \times 10^{-4}$	0.00144	$8.13 \times 10^{-4}$
PMC:Intrinsic-Max (in.)	0.00579	0.00469	0.00565	0.00564

best matches the numerical result. In contrast, the plot in Figure 5.6 shows the worst result, which uses  $\alpha$ OS with BFGS. From Figure E.10, the BFGS method starts to drift in the opposite direction compared to the numerical result, between the 1.5 and 2 second mark. This is the first time that the specimen enters the nonlinear region. By examining Figures 5.5 and 5.6, PMC:Broyden using NME is in DC mode and the PMC:BFGS using  $\alpha$ OS is in FC mode during this time period. Otherwise, the two produce almost identical control modes.

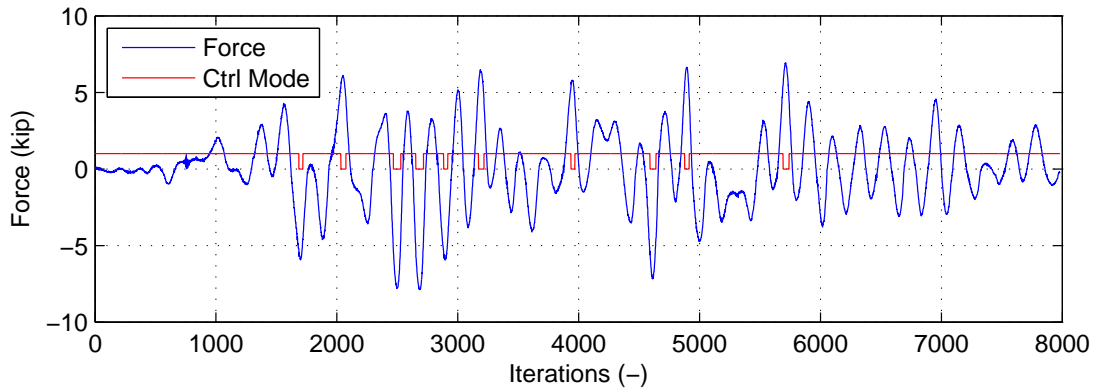


Figure 5.5: Plot of control mode and trial forces vs. time of the bottom actuator for PMC:Broyden with NME.

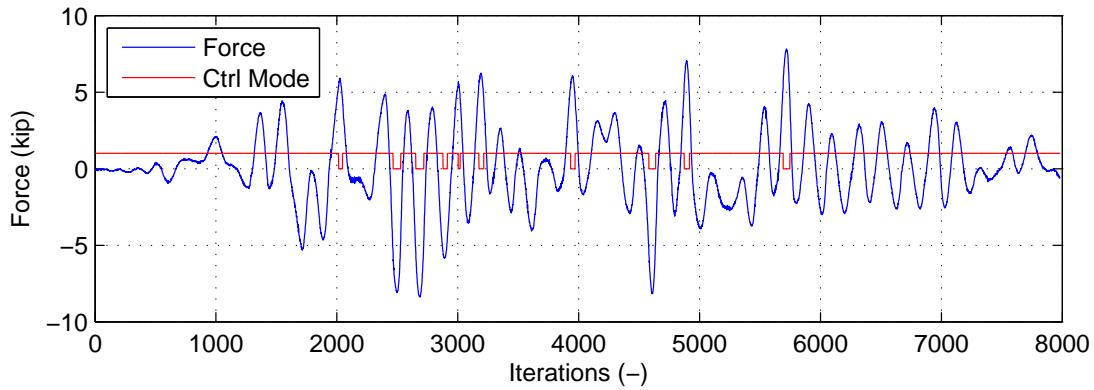


Figure 5.6: Plot of control mode and trial forces vs. time of the bottom actuator for PMC:BFGS with  $\alpha$ OS.

#### 5.4.2 MC hybrid simulation with secant stiffness based switching

The results of MC hybrid simulation using secant stiffness based switching (SMC) is presented in this section. The secant stiffness switching limit is set at 50 kip/in. with  $k_d = 45\text{kip/in.}$  and  $k_f = 55\text{kip/in.}$   $k_{11}$  of the initial stiffness matrix is 81 kip/in. Tables 5.7 and 5.8 contain the absolute displacement errors of Nodes 1 and 2 respectively. The NMF and NMR results capture the drift correctly except for the SMC:BFGS method. NME and  $\alpha$ OS do not match the numerical result well. Tables 5.9 and 5.10 show the force errors to be about the same level as DC, FC and PMC hybrid simulation results. No method or time integration scheme really stands out when examining the force errors. Tables 5.11 and 5.12 show the tracking performance of the MTS-STC control system. The tracking performance is similar to the others.

Table 5.7: Node 1 absolute errors of displacement results from nonlinear SMC hybrid simulations for the 2-DOF setup for various time integration method at the FEA level.

Control method	NME (in.)	$\alpha$ OS (in.)	NMF (in.)	NMR (in.)
SMC:Broyden-Min	0	$3.45 \times 10^{-7}$	$9.73 \times 10^{-6}$	$1.68 \times 10^{-5}$
SMC:Broyden-Mean	0.422	0.262	0.0529	0.0677
SMC:Broyden-Max	0.738	0.505	0.271	0.221
SMC:BFGS-Min	$5.63 \times 10^{-8}$	$6.28 \times 10^{-8}$	$6.39 \times 10^{-5}$	$5.59 \times 10^{-5}$
SMC:BFGS-Mean	0.309	0.188	0.0731	0.400
SMC:BFGS-Max	0.617	0.436	0.316	0.687
SMC:Intrinsic-Min	$2.97 \times 10^{-7}$	$3.20 \times 10^{-8}$	$2.46 \times 10^{-5}$	$2.19 \times 10^{-5}$
SMC:Intrinsic-Mean	0.356	0.137	0.162	0.0962
SMC:Intrinsic-Max	0.675	0.341	0.403	0.305

The best time integrations scheme for SMC hybrid simulation is NMF. The tangent stiffness estimation is examined in the following figures. The Broyden method (Figure 5.7) is the most active followed by BFGS (Figure 5.8) and then Intrinsic (Figure 5.9). The control mode switches

Table 5.8: Node 2 absolute errors of displacement results from SMC nonlinear hybrid simulations for the 2-DOF setup for various time integration method at the FEA level.

Control method	NME (in.)	$\alpha$ OS (in.)	NMF (in.)	NMR (in.)
SMC:Broyden-Min	$2.98 \times 10^{-7}$	$6.10 \times 10^{-7}$	$5.09 \times 10^{-5}$	$5.05 \times 10^{-5}$
SMC:Broyden-Mean	2.07	1.91	0.279	0.416
SMC:Broyden-Max	3.59	3.62	1.19	1.16
SMC:BFGS-Min	$2.98 \times 10^{-7}$	$3.33 \times 10^{-7}$	$7.32 \times 10^{-5}$	$4.71 \times 10^{-5}$
SMC:BFGS-Mean	1.52	0.864	0.355	1.84
SMC:BFGS-Max	3.02	2.08	1.37	3.29
SMC:Intrinsic-Min	$2.98 \times 10^{-7}$	$1.51 \times 10^{-7}$	$9.40 \times 10^{-6}$	$6.46 \times 10^{-5}$
SMC:Intrinsic-Mean	1.60	0.542	0.717	0.693
SMC:Intrinsic-Max	3.05	1.40	1.82	1.85

Table 5.9: Node 1 absolute errors of force results from nonlinear SMC hybrid simulations for the 2-DOF setup for various time integration method at the FEA level.

Control method	NME (kip)	$\alpha$ OS (kip)	NMF (kip)	NMR (kip)
SMC:Broyden-Min	$5.87 \times 10^{-6}$	$1.74 \times 10^{-4}$	$9.05 \times 10^{-4}$	0.00237
SMC:Broyden-Mean	0.776	1.19	1.46	1.04
SMC:Broyden-Max	3.55	4.27	5.02	4.87
SMC:BFGS-Min	$4.03 \times 10^{-4}$	$2.85 \times 10^{-4}$	$6.51 \times 10^{-4}$	0.00336
SMC:BFGS-Mean	0.814	1.28	1.42	1.12
SMC:BFGS-Max	3.68	4.82	5.13	3.87
SMC:Intrinsic-Min	$9.52 \times 10^{-5}$	$9.04 \times 10^{-5}$	$2.38 \times 10^{-4}$	$2.56 \times 10^{-4}$
SMC:Intrinsic-Mean	0.798	0.984	1.38	0.894
SMC:Intrinsic-Max	3.35	4.45	5.03	3.40

Table 5.10: Node 2 absolute errors of force results from nonlinear SMC hybrid simulations for the 2-DOF setup for various time integration method at the FEA level.

Control method	NME (kip)	$\alpha$ OS (kip)	NMF (kip)	NMR (kip)
SMC:Broyden-Min	$7.40 \times 10^{-6}$	$9.00 \times 10^{-6}$	$7.14 \times 10^{-4}$	$6.94 \times 10^{-5}$
SMC:Broyden-Mean	0.175	0.208	0.268	0.221
SMC:Broyden-Max	0.865	0.704	0.887	0.870
SMC:BFGS-Min	$5.83 \times 10^{-5}$	$1.30 \times 10^{-5}$	$2.93 \times 10^{-4}$	$3.37 \times 10^{-4}$
SMC:BFGS-Mean	0.175	0.299	0.263	0.219
SMC:BFGS-Max	0.671	1.06	0.848	0.987
SMC:Intrinsic-Min	$1.49 \times 10^{-5}$	$2.90 \times 10^{-5}$	$1.21 \times 10^{-4}$	$2.86 \times 10^{-4}$
SMC:Intrinsic-Mean	0.171	0.240	0.245	0.176
SMC:Intrinsic-Max	0.610	0.962	0.881	0.695

Table 5.11: Bottom actuator absolute errors of nonlinear SMC hybrid simulations for the 2-DOF setup for various time integration schemes at the control system level.

Control method	NME	$\alpha$ OS	NMF	NMR
SMC:Broyden-Min (in.)	$5.24 \times 10^{-10}$	$2.02 \times 10^{-8}$	$4.31 \times 10^{-9}$	$3.04 \times 10^{-9}$
SMC:Broyden-Mean (in.)	$9.96 \times 10^{-4}$	$8.40 \times 10^{-4}$	$7.85 \times 10^{-4}$	$9.18 \times 10^{-4}$
SMC:Broyden-Max (in.)	0.00694	0.00493	0.00431	0.00671
SMC:Broyden-Min (kip)	0	0	0	0
SMC:Broyden-Mean (kip)	0.0167	0.0362	0.0130	0.0260
SMC:Broyden-Max (kip)	0.163	0.172	0.0877	0.213
SMC:BFGS-Min (in.)	$2.47 \times 10^{-9}$	$6.75 \times 10^{-9}$	$1.60 \times 10^{-9}$	$1.08 \times 10^{-6}$
SMC:BFGS-Mean (in.)	0.00127	0.00130	0.00139	0.00159
SMC:BFGS-Max (in.)	0.00515	0.00676	0.00498	0.00329
SMC:BFGS-Min (kip)	0	0	0	0
SMC:BFGS-Mean (kip)	0.0318	0.0471	0.0306	0.0448
SMC:BFGS-Max (kip)	0.165	0.220	0.138	0.250
SMC:Intrinsic-Min (in.)	$3.68 \times 10^{-9}$	$1.29 \times 10^{-9}$	$3.61 \times 10^{-9}$	$2.79 \times 10^{-10}$
SMC:Intrinsic-Mean (in.)	0.00104	0.00119	$7.80 \times 10^{-4}$	$9.88 \times 10^{-4}$
SMC:Intrinsic-Max (in.)	0.00671	0.00524	0.00409	0.00536
SMC:Intrinsic-Min (kip)	0	0	0	0
SMC:Intrinsic-Mean (kip)	0.0178	0.0194	0.290	0.0295
SMC:Intrinsic-Max (kip)	0.117	0.171	0.178	0.240

Table 5.12: Top actuator absolute errors of nonlinear SMC hybrid simulations for the 2-DOF setup for various time integration schemes at the control system level.

Control method	NME	$\alpha$ OS	NMF	NMR
SMC:Broyden-Min (in.)	$4.69 \times 10^{-9}$	$2.19 \times 10^{-8}$	$5.99 \times 10^{-9}$	$4.15 \times 10^{-9}$
SMC:Broyden-Mean (in.)	$7.75 \times 10^{-4}$	$5.92 \times 10^{-4}$	$6.88 \times 10^{-4}$	0.00134
SMC:Broyden-Max (in.)	0.00532	0.00609	0.00531	0.00603
SMC:BFGS-Min (in.)	$8.53 \times 10^{-10}$	$6.34 \times 10^{-9}$	$6.88 \times 10^{-9}$	$1.98 \times 10^{-09}$
SMC:BFGS-Mean (in.)	$6.57 \times 10^{-4}$	$5.09 \times 10^{-4}$	$6.35 \times 10^{-4}$	$6.03 \times 10^{-4}$
SMC:BFGS-Max (in.)	0.00487	0.00538	0.00552	0.00546
SMC:Intrinsic-Min (in.)	$6.97 \times 10^{-10}$	$1.42 \times 10^{-8}$	$4.71 \times 10^{-10}$	$6.81 \times 10^{-9}$
SMC:Intrinsic-Mean (in.)	$5.53 \times 10^{-4}$	$5.42 \times 10^{-4}$	$5.53 \times 10^{-4}$	$6.22 \times 10^{-4}$
SMC:Intrinsic-Max (in.)	0.00459	0.00415	0.00485	0.00501

most frequently when using the BFGS method. The others only switch control modes one time. Out of these three the Broyden method produces the best results at the FEA level, but it is not much better than the BFGS method. The Intrinsic method has errors that are double the errors of BFGS.



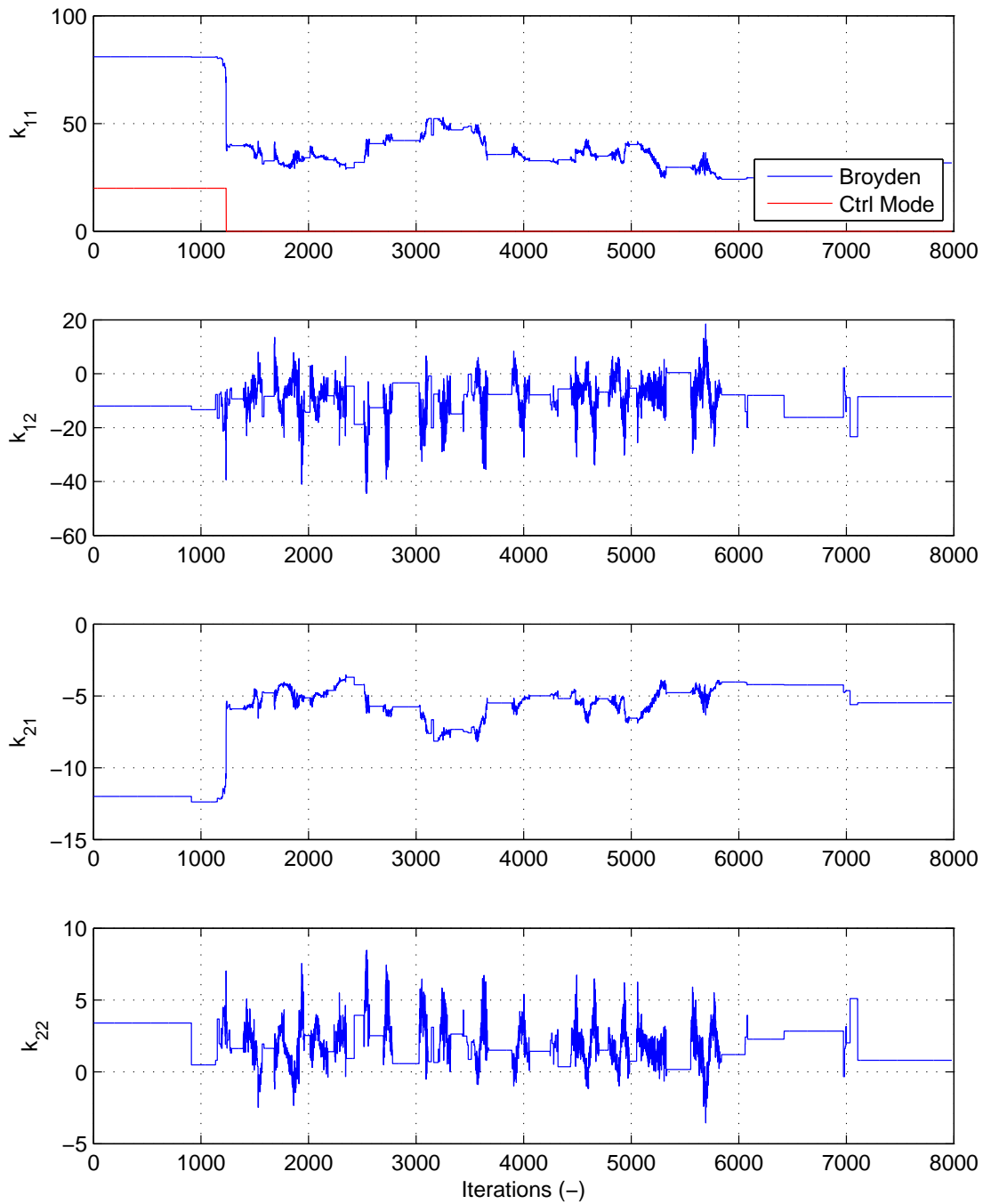


Figure 5.7: Plot of control mode and trial forces vs. time of the bottom actuator for PMC:Broyden with NME.

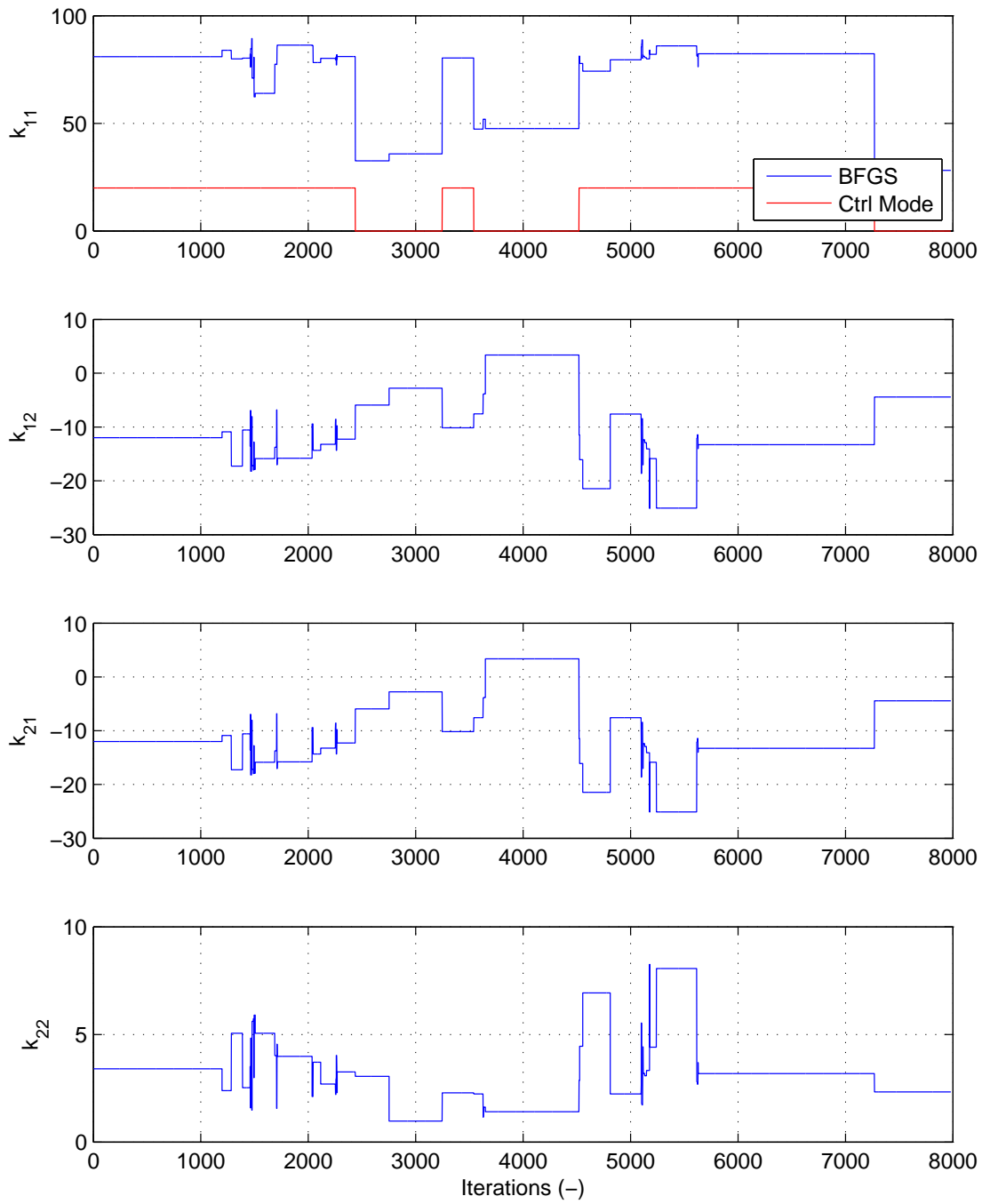


Figure 5.8: Plot of control mode and trial forces vs. time of the bottom actuator for PMC:BFGS with  $\alpha$ OS.

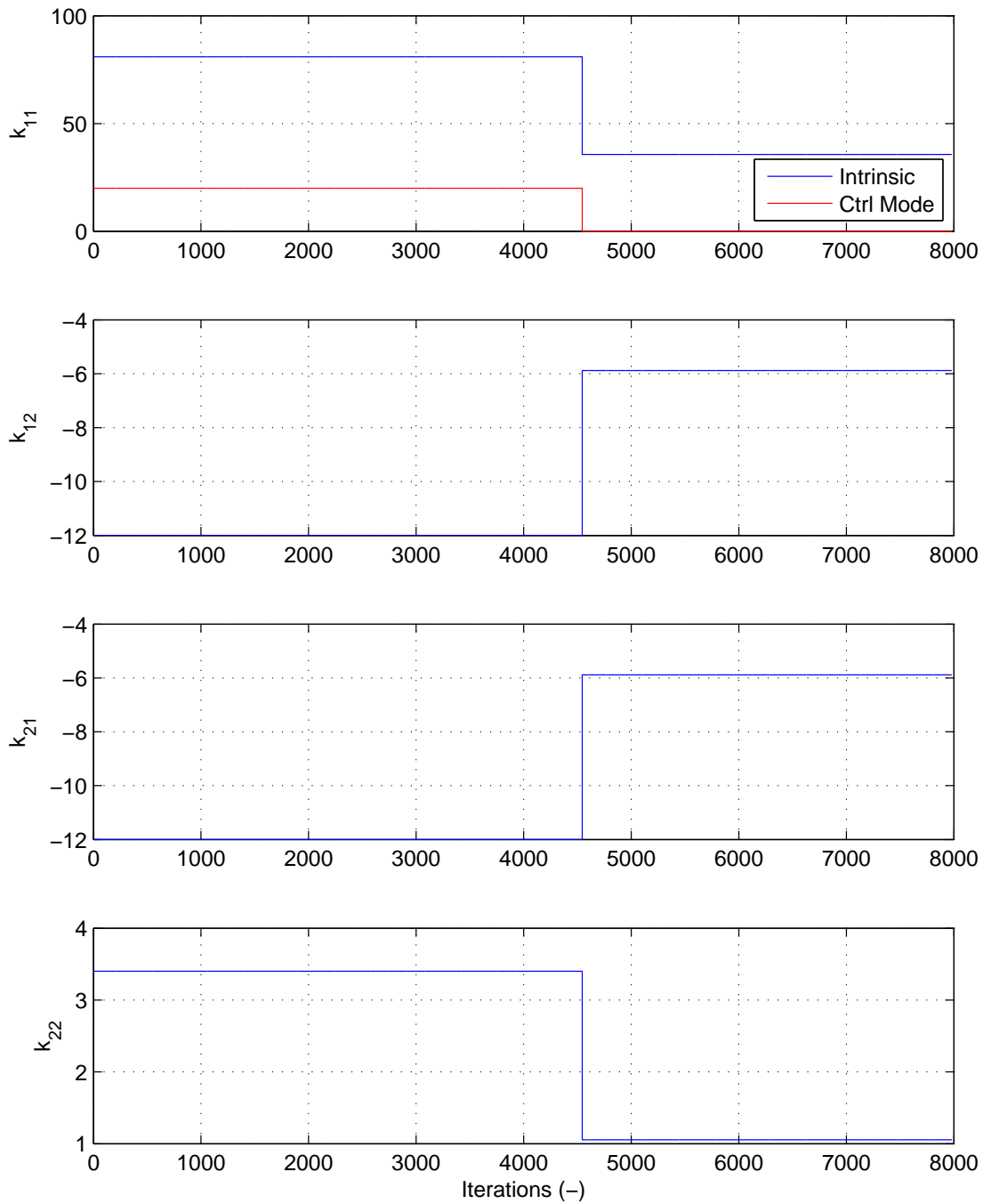


Figure 5.9: Plot of control mode and trial forces vs. time of the bottom actuator for PMC:BFGS with  $\alpha$ OS.

### 5.4.3 Summary of experimental results

These experimental results show that MC hybrid simulation is a viable type of hybrid simulation, even if the results are not the best. Particularly, the PMC hybrid simulation results overall are slightly better than the SMC hybrid simulation results. PMC seems to capture the change in the specimen's behavior better than the SMC. This may be caused by couple reasons. The secant stiffness value limit is too high and the tangent stiffness estimation is inaccurate. It may be just one of these reasons or both together. MC hybrid simulation does not provide better tracking performance of the control system then DC and FC hybrid simulation from Chapter 3.

# Chapter 6

## Conclusion

This dissertation develops and explores alternative control methods in hybrid simulation, namely force control (FC), switch control (SC) and mixed control (MC). These control methods are needed in hybrid simulation because of the control system's limitations. Most hybrid simulation tests are conducted with the control system in displacement control (DC) mode. In many of these cases, displacement control is the best control mode and produces good results. This is however not to say that the need does not exist for alternative control methods in hybrid simulation.

Structural engineers and researchers are always pushing the envelope and inventing new innovative structural components and assemblages. Civil structures are expected to provide high confidence in low probability of a life-loosing catastrophic failure during their expected lifetime. Thus, safety and reliability are very important in structural engineering. As structural engineers design and develop new advanced structural components, equally advanced experimental techniques and setups are required to test the performance of these components.

Hybrid simulation is a perfect platform for implementing these advanced experimental techniques and setups. Hybrid simulation is game-changing because it provides a method to proof-test structural systems by physically testing their components. Before researchers had to test entire systems, or settle for testing stand-alone components using qualification, not proof, test methods. Hybrid simulation produces realistic loading patterns, and the effects these components have on the entirety of the structure can be captured. In order for hybrid simulation to continue to be a relevant and effective systems engineering method in structural engineering, it must offer the means to test the entire portfolio of structural systems, components and assemblages under any realistic combination of long-duration and extreme challenges. This is the big picture motivation for this dissertation, to advance the capability of hybrid simulation so that it will continue to be an effective method of structural testing. Another motivational factor is the need to complete the conceptual formulation of hybrid simulation by providing the means to treat the duality between displacements and forces in structural mechanics.

### 6.1 Force Control

Chapter 3 defines, develops and tests hybrid simulation method where the control system is in FC mode. The methods for obtaining the trial force, the force applied to the specimen, is

devised. These methods are grouped into two different categories. One is based on enforcing the compatibility of structural deformations. The other is formulated from preserving the force equilibrium in a structure. These methods are implemented in Matlab and OpenFresco.

Then the research moves to the control aspect of FC hybrid simulation with the extension of the Simulink/Stateflow model for FC hybrid simulation. Chapter 3 presents results of FC hybrid simulation tests using the FC Simulink/Stateflow model and the methods for computing the trial forces. The  $\mu$ -NEES experimental setup is configured for FC hybrid simulation testing. Both DC and FC hybrid simulation tests are conducted and compared. FC results are overall marginally better than the DC results. If the specimen is even stiffer than the setup that was test, it is suspected that the results would be noticeably better for FC than DC. It is however very difficult to construct a repeatable test platform like the  $\mu$ -NEES that is very stiff. These results demonstrated that FC hybrid simulation is achievable and produces results that are at least as good as DC hybrid simulation if not better.

## 6.2 Switch Control

Switch control hybrid simulation is presented in Chapter 4. Switch control hybrid simulation is a logical extension of FC hybrid simulation. Throughout the course of a hybrid simulation test, the specimen may undergo deformations and forces that require the control mode of the control system to be change. Two switching strategies are implemented in this chapter, namely force limit and secant stiffness limit switching. SC hybrid simulation tests are conducted using these switching strategies with the FC hybrid simulation methods from Chapter 3.

The 1-DOF  $\mu$ -NEES setup and SC Simulink/Stateflow model are deployed for SC hybrid simulation tests. The test results indicate that the DC results are slightly better than the SC results. The SC results are better than the FC results. All three results practically speaking are equally good. It is also concluded, as with FC hybrid simulation results, that a much stiffer specimen would have produced better results for the SC methods. The NNE and  $\alpha$ OS time integrations schemes performed better than the NMF and NMR time integration schemes for SC hybrid simulation tests.

## 6.3 Mixed Control

Mixed control hybrid simulation is the next natural progression of SC hybrid simulation and multi-DOF FC hybrid simulation. Chapter 5 discusses MC hybrid simulation. It combines the FC hybrid simulation methods for acquiring the trial forces with the switch strategies from Chapter 4. MC Simulink/Stateflow model is implement for MC hybrid simulation where the top actuator of 2-DOF  $\mu$ -NEES setup is always in DC mode and the bottom actuator switches between DC and FC modes.

The MC hybrid simulation test results proved not to match the numerical results as well as the DC and FC results. In fact most of the MC results had trouble capturing the drift correctly. The force results however are in good agreement with the numerical results. The interaction of the actuators in different control modes may have contributed to some of the displacements being off. It is, nevertheless, encouraging that there are a few tests where the responses in MC were

comparable to those in DC and FC. At the very least, successful MC hybrid simulations can be conducted and the software framework and the predictor-corrector Simulink/Stateflow models are in place to continue the research.

## 6.4 Direction for future work

Even though this dissertation made some headway on these topics discussed above, there still remains more research and experimentation that need to be conducted for these types of hybrid simulation to come to full fruition. This section lay out the future research of alternative control methods in hybrid simulation beyond what is presented in this dissertation.

### 6.4.1 Further verification

Further verification of FC, SC and MC hybrid simulation methods are required. As discussed before, these methods do not always produce better results than the DC hybrid simulation methods. One possible reason for this is that the  $\mu$ -NEES experimental setup is not stiff enough. A stiff system is required where the tracking performance in FC mode is clearly better than the tracking performance in DC mode. The FC, SC and MC methods are required for setups and specimens where the control system does not function adequately in DC mode throughout the test. Only with such a setup can these methods be fully verified. Along with a stiff setup, the hybrid models in this dissertation are relatively simple and small in the number of DOFs. Testing these methods with a larger model is necessary to show that these methods are extensible.

An in-depth error analysis is required for all control modes (FC, SC, MC and DC). There are several possible sources of error that should be examined. First is the estimation of the tangent stiffness matrix and deployment of it to compute the trial forces. Adding this extra layer of computation to the hybrid simulation process changes the previous error analysis. Another source is in the control system loop implementation: it manifests itself through spikes in the measured values of the control system when it switches control modes. This may or may not effect the results. It depends whether this is an artifact particular to the MTS-STS control system and the  $\mu$ -NEES experimental setup. This spike may also adversely affect the tracking performance of the control system. On the other hand, the spike may occur at the instant the control system changes modes and may not play a big factor on its performance. It is not known for sure and thus is worth investigating. Error analysis for the equilibrium base force control (ECF) method is needed. This is the first time a force based time integration scheme is used for hybrid simulation. Therefore, no error analysis of this method exists.

### 6.4.2 Using power for switching

The performance and result of SC and MC hybrid simulation is mainly dependent on its switching strategy. Force and secant stiffness are the two criteria for switching that are examine in this dissertation. Other criteria should be researched as well. It is evident from the experimental results that force and secant stiffness based switching give comparable results. However, they do have their disadvantages as present in Chapter 4.

One criterion worth researching is the mechanical power generated by the experimental setup. The power can be calculated from the the measured values or be measure directly through instrumentation. The mechanical power is simple to calculate,  $P(t) = F(t) * V(t)$ .  $F(t)$  is the measured force and  $V(t)$  is the calculated velocity from the measure displacements. Figure 6.1 shows the calculated power,  $P(t)$ , plot for one of the SC hybrid hybrid simulation tests presented in Chapter 4. From a cursory examination of Figure 6.1, positive spikes in power may be a good switching point during a test. The positive power spikes occurs right before the specimen starts to enter the nonlinear region.

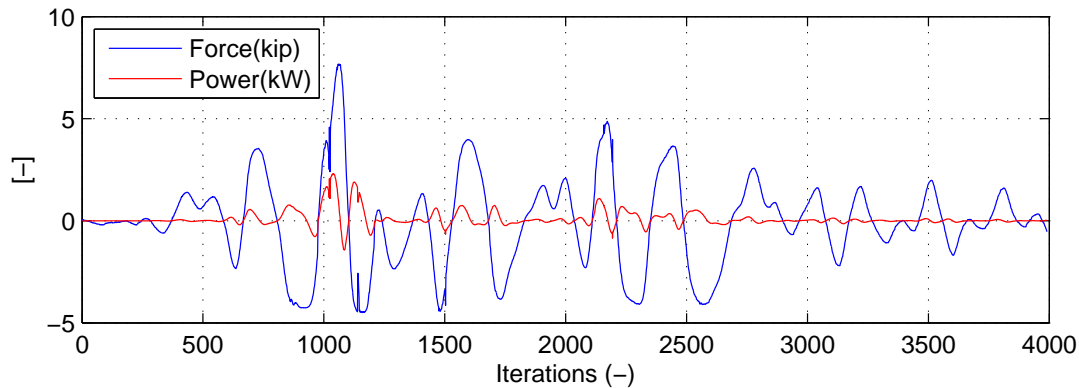


Figure 6.1: Plot of calculated power and trial forces vs. time from PSC with NME.

### 6.4.3 Further investigation of MC hybrid simulation

Out of the three hybrid simulation control methods, MC hybrid simulation methods are the most underdeveloped. It also does not perform as well as the other two. The definitive cause is unknown. It is speculated that the interaction of the two actuators running in different control modes may be one of the causes. The reason for this speculation is that both DC and FC results performed better. A new setup is required to investigate this possible cause. This new setup must first be stiffer for one of the control DOFs, and its actuator configuration needs to be changed. The actuators are parallel to each other in the 2-DOF  $\mu$ -NEES setup. The new configuration should have the actuators in an orthogonal configuration. Another reason for the poor performance may be linked to the switching strategies. The parameters for switching may not have been set properly. A parametric study is required to rule this out as a cause.

A new method for acquiring the trial force is worth investigating. Figure 6.2 illustrates a proposed method for acquiring the trial force when the two loading directions are orthogonal to each other. The red element is the beam-column element being tested. The horizontal DOF is in DC mode and vertical DOF is in FC mode. The horizontal trial displacements are calculated in the same way as DC hybrid simulation. The state of the numerical model is obtained for these given trial displacements. Then the unbalance vertical force ( $r_s + r_a - P_2$ ) is used as trial force. The implementation of this method will not work with most commercial FEA programs because it requires a highly customizable FEA software package.



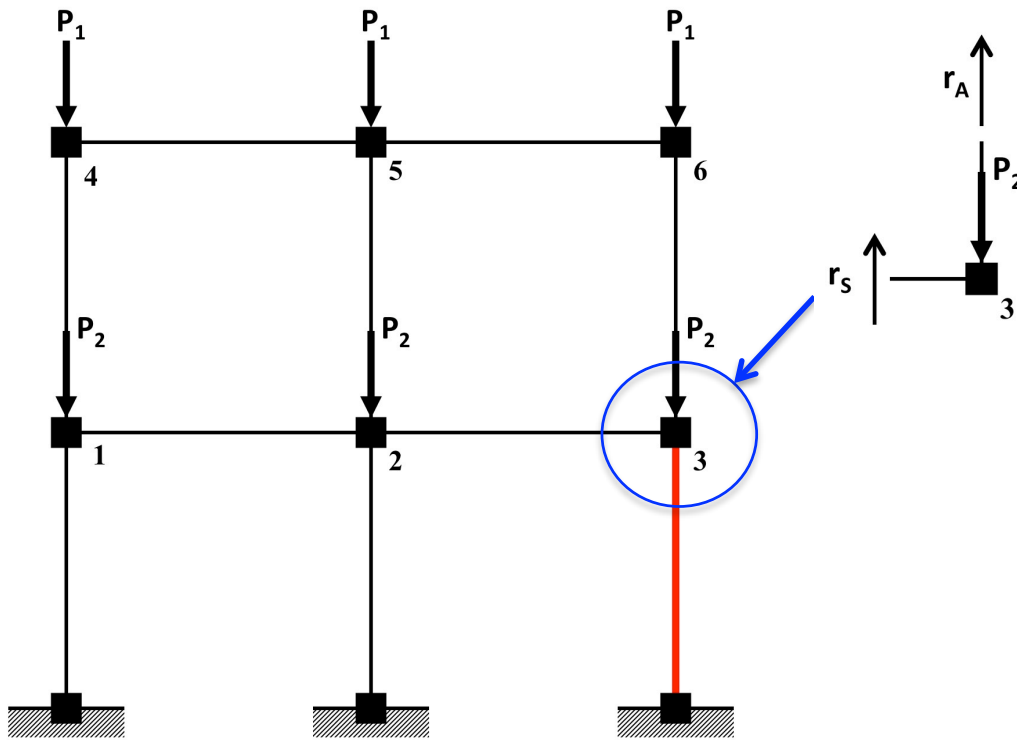


Figure 6.2: Proposed method for MC hybrid simulation using the unbalance force at the node as the trial force, in general, for multi-DOF multi-actuator systems.

#### 6.4.4 Simulink/Stateflow models

The Simulink/Stateflow models for FC, SC and MC hybrid simulation tests use linear interpolation and extrapolation for correcting and predicting respectively. It would be interesting to see how other types of interpolation and extrapolation perform. There are DC Simulink/Stateflow models that use  $2^{nd}$  and  $3^{rd}$  order polynomials. These models can be easily extended for FC, SC and MC hybrid simulation tests. Higher order polynomials are avoided initially because it is suspected that these would cause oscillations that may cause the control system to go unstable. This suspicion is not verified.

The simulation time step of 0.10 second is used for all the tests in this dissertation. These tests are relatively slow. The integration time step for NME,  $\alpha$ OS, NMF and NMR is 0.02 second but 20 substeps are used for NME,  $\alpha$ OS and NMF. This makes the actual integration time step to be 0.001 second. The simulation time step is 100 times larger. The reason for such a large simulation time step is to keep the control system stable throughout the test. Research should be conducted to see if these tests can be run faster. The investigation of different integration methods used by the computational driver and the interactions between these and the predictor/corrector strategies is a good starting point.

### **6.4.5 EFC method with other time integration schemes**

The EFC method uses the NMR time integration scheme because the other time integration schemes cause the simulation driven using this method to become unstable. A disadvantage of using NMR is that it does not produce good command value curves for the controller. To overcome this shortcoming, it may be possible to use NME,  $\alpha$ OS and NMF with this method. Some preliminary formulations have been started, but they are not complete. If this can be accomplished, it may make the EFC method more effective than the compatibility force control methods since it does not use an estimation of the tangent stiffness matrix.

### **6.4.6 Summary**

There is a need for further research in this field of structural engineering. These topics have been both challenging and rewarding. It is the hope of this author that the research presented in this dissertation will continue if not by the author himself, then by other researcher who find this topic equally fascinating.

# Bibliography

- [1] M. Ahmadizadeh and G. Mosqueda. Hybrid simulation with improved operator-splitting integration using experimental tangent stiffness matrix estimation. *J. Struct. Eng.-ASCE*, 134(12):1829–1838, Jan 2008.
- [2] C.G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, 19(92):577–593, Oct 1965.
- [3] P. Caravani, M.L. Watson, and W.T. Thomson. Recursive least-squares time domain identification of structural parameters. *J. Appl. Mech.-T. Asme*, 44(1):135–140, MAR 1977.
- [4] J.E. Carrion and B.F. Spencer. Model-based strategies for real-time hybrid testing. In *ASCE Structure Congress, Seattle, WA*, 2003.
- [5] Anil K. Chopra. *Dynamics of Structures*. Prentice Hall, 2001.
- [6] D. Combescure and P. Pegon.  $\alpha$ -operator splitting time integration technique for pseudodynamic testing - error propagation analysis. *Soil Dyn. Earthq. Eng.*, 16:427–443, Jan 1997.
- [7] T. Elkhoraibi and K.M. Mosalam. Towards error-free hybrid simulation using mixed variables. *Earthquake Engineering Structural Dynamics*, 36(11):1497–1522, 2007.
- [8] Erich Gamma, Richard Helm, Ralph Johnson, and John Vliissides. *Mathematical Handbook for Scientists and Engineers*. McGraw- Hill, New York, 1968.
- [9] H.M. Hilber, Hughes T.J.R., and R.L. Taylor. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering Structural Dynamics*, 5:283–292, Jan 1977.
- [10] C. Hung and S. El-Tawi. A method for estimating specimen tangent stiffness for hybrid simulation. *Earthquake Engineering Structural Dynamics*, 38(1):115–134, 2008.
- [11] A. Igarashi, F. Seible, and G. Hegemier. Development of the pseudodynamic technique for testing a full scale 5-story shear wall structure. In *Development and Future Dimensions of Structural Testing Techniques*, 1993.
- [12] H. Klie and M. Wheeler. Nonlinear krylov-secant solvers. Technical report, The Center for Subsurface Modeling, The Institute for Computational Engineering and Sciences, Univ. of Texas, Austin, Jan 2006.

- [13] G. Korn and publisher = McGraw- Hill, New York year = 1968 Korn, T., title= Mathematical Handbook for Scientists and Engineers.
- [14] O. Kwon, N. Nakata, K. Park, A. Elnashai, and B. Spencer. *User Manual and Examples for UI-SIMCOR v2.6 and NEES-SAM v2.0*.
- [15] G. Mosqueda. *Continuous hybrid simulation with geographically distributed substructures*. PhD thesis, University of California, Berkeley, 2003.
- [16] M. Nakashima, T. Kaminosono, M. Ishida, and K. Ando. Integration techniques for substructure pseudo dynamic test. In *Proceedings of the US National Conference on Earthquake Engineering*, 1990.
- [17] N. Nakata, B.F. Spencer, and A.S. Elnashai. Mixed load/displacement control strategy for hybrid simulation. In *Proc., 4th Int. Conf. on Earthquake Engineering, Taipei, Taiwan*, 2006.
- [18] M. Pan, P.and Nakashima and H. Tomofuji. Online test using displacement-force mixed control. *Earthquake Engineering Structural Dynamics*, 34(8):869–888, 2005.
- [19] S. Patnaik. Hybrid simulation of a zipper-braced steel frame under earthquake excitation. *International J. for Num. Meth. in Eng.*, 6:237–251, 1973.
- [20] A. Schellenberg. *Advanced Implementation of Hybrid Simulation*. PhD thesis, University of California, Berkeley, 2008.
- [21] A. Schellenberg, H.K. Kim, Y. Takahashi, Fenves G.L., and Mahin S.A. Openfresco framework for hybrid simulation: Ui-simcor v2.6 example. Technical report, Department of Civil and Environmental Engineering, University of California, Berkeley, Aug 2009.
- [22] M.H. Scott and G.L. Fenves. A krylov subspace accelerator newton algorithm. In *Proc., 4th Int. Conf. on Earthquake Engineering, Taipei, Taiwan*, 2006.
- [23] P.S.B. Shing and S.A. Mahin. Rate-of-loading effects on pseudodynamic tests. *J. Struct. Eng.-ASCE*, 114(11):2403–2420, Jan 1988.
- [24] P.S.B. Shing and S.A. Mahin. Experimental error effects in pseudodynamic testing. *J. Eng. Mech.-ASCE*, 116(4):805–821, Jan 1990.
- [25] M. Sivaselvan, A. Reinhorn, Z. Liang, and X. Shao. Real-time dynamic hybrid testing of structural systems. In *Proc. of the 13th World Conf. Earthquake Engineering*, 2004.
- [26] Y. Takahashi and G.L. Fenves. Software framework for distributed experimental-computational simulation of structural systems. *Earthquake Engineering Structural Dynamics*, 35(2):267–291, 2006.
- [27] K. Takanashi, K. Udagawa, M. Seki, T. Okada, and H. Tanaka. Non-linear earthquake response analysis of structures by a computer actuator on-line system (part 1 details of the system). *Trans. Architectural Inst. of Japan*, (229):77–83, 1975.

- [28] C.R. Thewalt and S.A. Mahin. Hybrid solution techniques for generalized pseudodynamic testing. Technical Report UCB/EERC-87/09, Earthquake Engineering Research Center, University of California, Berkeley, July 1987.

# Appendix A

## OpenFresco for FC, SC and MC hybrid simulation

OpenFresco was originally developed by Takahashi and Fenves [26]. Schellenberg et al. [20] made further refinements to the original concept. It has gone through several refinements and currently it is released under version 2.6. OpenFresco provides a standard way of mediating transactions between the computational driver and the transfer system. This software framework is programmed in C++ and takes advantage of object-orientated programming paradigm. Thus, it is flexible extensible and adaptable in that many different types of computational drivers transfer system are supported by OpenFresco. OpenFresco also allows users from different site collaborate more easily through its multi-tier software architecture. This aids in distributed hybrid simulation where multi-sites are involved.

### A.1 OpenFresco software architecture

OpenFresco is based on three- and multi-tier software architectures. These software architectures are variations of the well-known client-server architecture. The three-tier architecture consists of a client, a middle-tier server and a backend server. The computational driver represents the client. This is the top tier of the architecture where the structure is modeled numerically and the analysis is performed. The middle-tier server encapsulates the OpenFresco processes. In all practical purposes, the middle-tier server is OpenFresco. The middle-tier server mediates the interaction between the client and the backend server. It handles the communication and transformation of data from the client to the backend server and vice-versa. The backend server, the last tier, represents the laboratory including control and DAQ systems. This is the experimental portion of hybrid simulation.

This architecture provides flexibility such that a variety of computational drivers and laboratory equipment can be used for hybrid simulation. Currently OpenFresco support OpenSees, LS-Syna, Abacus and Matlab as computational drivers. It also supports various laboratory equipment as well.

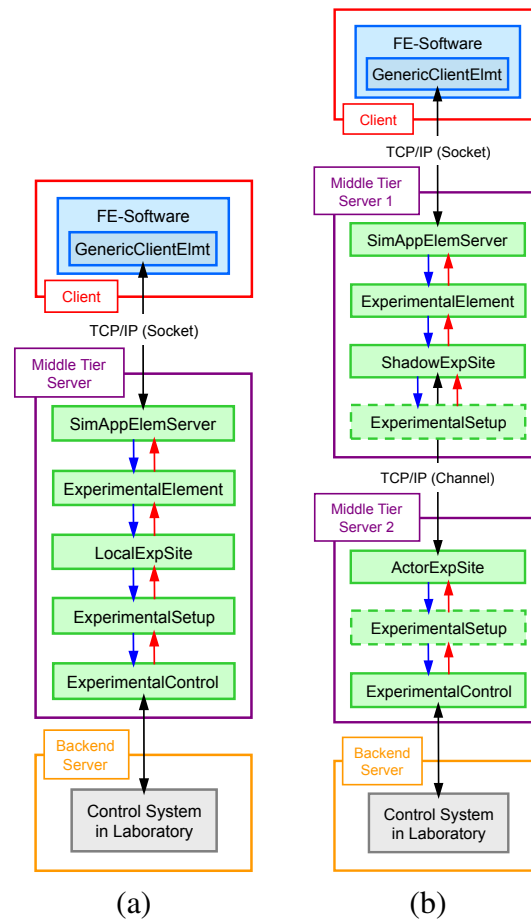


Figure A.1: OpenFresco architecture for (a) local deployment and (b) distributed deployment [20].

### A.1.1 OpenFresco UML classes diagram

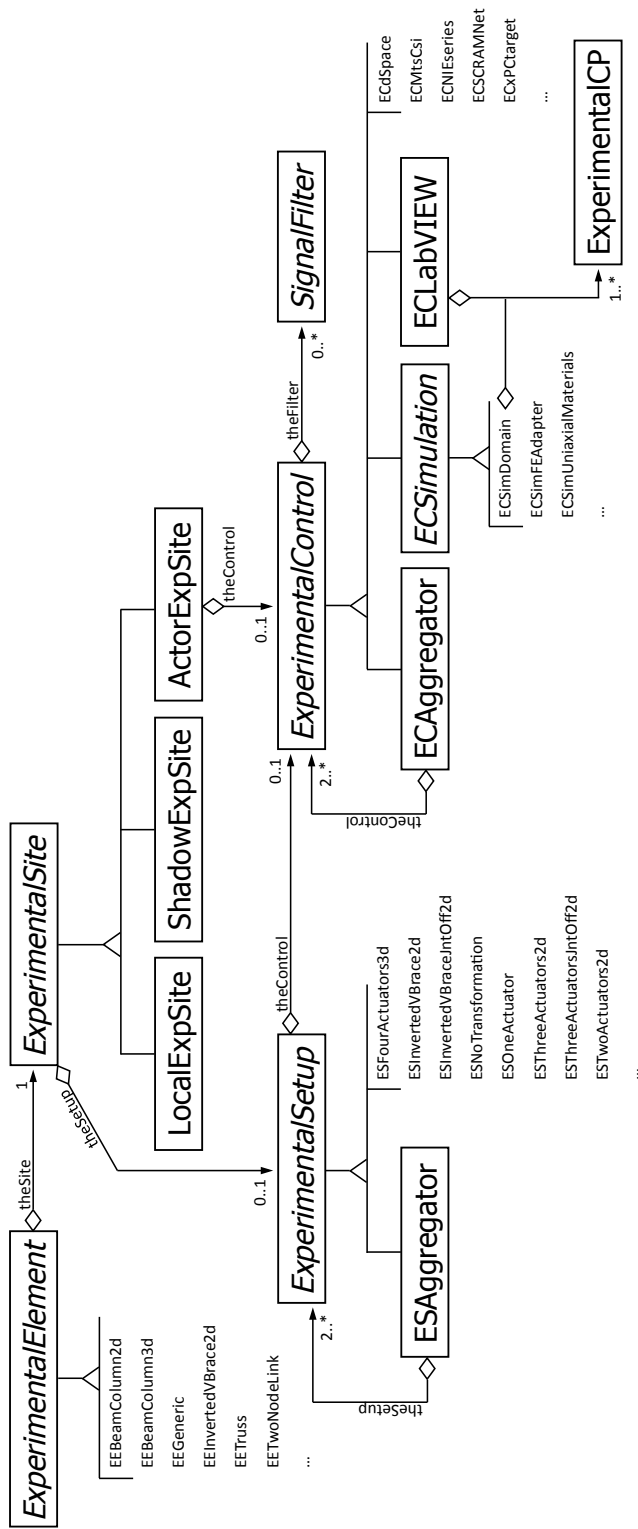


Figure A.2: OpenFresco class diagram for DC hybrid simulation.



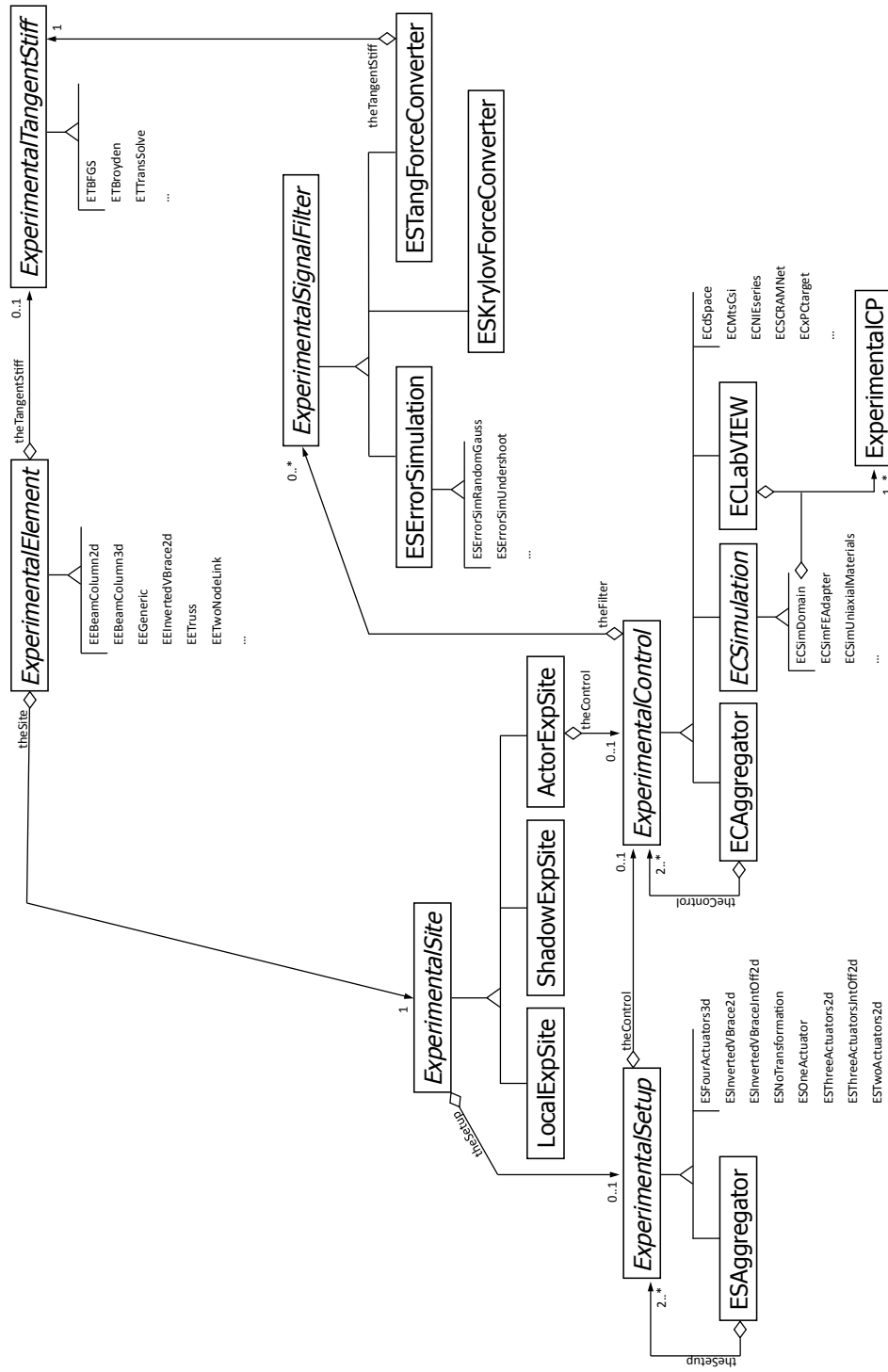


Figure A.3: OpenFresco class diagram for FC, SC, MC hybrid simulation.

## A.2 OpenFresco classes

### A.2.0.1 Experimental Element

The Experimental Element represent the physical specimen in the FEA program. It has many similarities to an analytical element. It provides mass, damping and stiffness matrix; it also returns the force vector for the a given deformation state. Consequently, the Experimental Element abstract class in OpenFresco inherits from the OpenSees Element class. The Experimental Element class performs the transformation between the global coordinate system of the analytical model to the local or basic coordinate system for testing purposes. It also executes the inverse of this transformation. This abstract class transforms the following response quantities: displacement, velocity, acceleration and force. The experimental element always returns the initial stiffness matrix because it is difficult to determine the tangent stiffness matrix of a physical specimen in an accurate way during a test.

The following is a list of Experimental Element concrete sub-classes:

- Truss (1D, 2D and 3D)
- Beam-Column (2D and 3D)
- Two-Node Link (1D, 2D and 3D)
- Generic (1D, 2D and 3D)
- Inverted-V Brace

### A.2.0.2 Experimental Sites

The Experimental Site represents the laboratories and the location where the computational driver is running. These sites do not necessarily have to be in the same location. To generalize the Experimental Site abstract class, communication methods were implemented. This class also provides secure communication between the sites.

The follow is a list of Experimental Site concrete sub-classes:

- Local
- Shadow and Actor

### A.2.0.3 Experimental Setups

The Experimental Setup represents the transfer system in the lab excluding the control and the data acquisition systems. The Experimental Setup abstract class performs transformations between the local or basic system to the actuator coordinate system. It also executes the reverse transformation.

-represents the transfer system in the lab -transformation The follow is a list of Experimental Setup concrete sub-classes:

- No Transformation

- One Actuator
- Two Actuators
- Three Actuators
- Inverted-V Brace
- Aggregator

#### **A.2.0.4 Experimental Controls**

The Experimental Control abstract class provides an interface to the control system. It allows OpenFresco to communicate with the control system. It is separated from the experimental setup class so that the lab technician working on the loading setup does not have to be concerned about the IT aspect of the control and data acquisition systems.

The follow is a list of Experimental Controls concrete sub-classes:

- dSpace, xPC Target and SCRAMNet+
- MTS-CSI and LabVIEW
- Uniaxial Materials and Domain Simulation

### **A.3 OpenFresco code for force control hybrid simulation**

```

class ESFTangForceConverter : public ExperimentalSignalFilter
{
public :
    // constructors
    ESFTangForceConverter(int tag , Matrix& initStif ,
                          ExperimentalTangentStiff* tangStif = 0);
    ESFTangForceConverter(const ESFTangForceConverter& esf);

    // destructor
    virtual ~ESFTangForceConverter();

    // method to get class type
    const char *getClassType() const {return "ESFTangForceConverter";};

    virtual int setSize(const int sz);
    virtual double filtering(double data);
    virtual Vector& converting(Vector* trialDisp);
    virtual Vector& converting(Vector* daqDisp , Vector* daqForce);
    virtual void update();

    virtual ExperimentalSignalFilter *getCopy();
    virtual Matrix& getTangStiffMat();

    // public methods for output
    void Print(OPS_Stream &s , int flag = 0);

private :
    ExperimentalTangentStiff *theTangStiff;
    bool firstWarning; // Flag for updating the stiffness matrix
    int size; // Size of ctrl and daq vectors
    Vector dispPast , forcePast , convertFrc , incrDisp , incrForce;
    Matrix kInit , kPrev , thetangStiffMat; // tangent stiff matrix

    int setInitialStiff();
    int updateMatrix(const Vector* daqDisp , const Vector* daqForce);
};

```

Figure A.4: ESFTangForceConverter concrete class definition that inherits from the *ExperimentalSignalFilter* class.

```

class ESFKrylovForceConverter : public ExperimentalSignalFilter
{
public :
    // constructors
    ESFKrylovForceConverter(int tag , int ss , Matrix& initStif);
    ESFKrylovForceConverter(const ESFKrylovForceConverter& esf);

    // destructor
    virtual ~ESFKrylovForceConverter ();

    // method to get class type
    const char *getClassType() const
        {return "ESFKrylovForceConverter";};

    virtual int setSize(const int sz);
    virtual double filtering(double data);
    virtual Vector& converting(Vector* td);
    virtual Vector& converting(Vector* daqDisp ,
                               Vector* daqForce);

    virtual void update ();

    virtual ExperimentalSignalFilter *getCopy ();

    // public methods for output
    void Print(OPS_Stream &s , int flag = 0);

private :
    bool firstWarning; // Flag for updating the stiffness matrix
    int size;           // Size of ctrl and daq vectors
    int szSubspace;    // number of spaces vectors to use
    Vector dispPast , forcePast , convertFrc , incrDisp , incrForce;
    Vector cn;
    Matrix kInit;      // tangent stiffness matrix
    Matrix iDMatrix;   // the incremental displacement Matrix
    Matrix iFMatrix;   // the incremental force Matrix

    int setInitialStiff ();
    int updateIncrMat(const Vector* daqDisp ,
                     const Vector* daqForce);
    int matTranspose(Matrix* kT , const Matrix* k);
};

```

Figure A.5: ESFKylovForceConverter concrete class definition that inherits from the *ExperimentalSignalFilter* class.

```

class ECxPtargetForce : public ExperimentalControl
{
public:
    // constructors
    ECxPtargetForce(int tag, char *ipAddress, char *ipPort,
        char *appName, char *appPath = 0);
    ECxPtargetForce(const ECxPtargetForce &ec);
    // destructor
    virtual ~ECxPtargetForce();
    // method to get class type
    const char *getClassType() const {return "ECxPtargetForce";};
    // public methods to set and to get response
    virtual int setup();
    virtual int setSize(ID sizeT, ID sizeO);
    virtual int setTrialResponse(const Vector* disp,
        const Vector* vel, const Vector* accel, const Vector* force,
        const Vector* time);
    virtual int getDaqResponse(Vector* disp,
        Vector* vel, Vector* accel, Vector* force, Vector* time);
    virtual int commitState();
    virtual ExperimentalControl *getCopy();
    // public methods for experimental control recorder
    virtual Response *setResponse(const char **argv, int argc,
        OPS_Stream &output);
    virtual int getResponse(int responseID, Information &info);
    // public methods for output
    void Print(OPS_Stream &s, int flag = 0);
protected:
    // protected methods to set and to get response
    virtual int control();
    virtual int acquire();
private:
    void sleep(const clock_t wait);
    int port;
    char *ipAddress, *ipPort, *appName, *appPath, char errMsg[256];
    double newTarget, switchPC, atTarget;
        bool frcFeedbackMode; // 0-converted frc or 1-daq frc feedback
    double *ctrlDisp, *ctrlForce, *daqDisp, *daqForce;
    int newTargetId, switchPCId, atTargetId;
    int ctrlForceId, *daqDispId, *daqForceId;
};

```

Figure A.6: ECxPtargetForce concrete class definition.

## Appendix B

### Tuning of $\mu$ -NEES experimental setup

The  $\mu$ -NEES experimental setup is an experimental setup that can be used to verify hybrid simulation methods. The clevis shown in Figure 3.13 gives repeatable behavior with replaceable steel coupons. It can be configured to provide different setup options. Two experimental setup configuration are used for the testing of force, switch and mixed control methods. This appendix shows the tuning of the  $\mu$ -NEES experimental setup.

#### B.1 Tuning of MTS-STS Channel 3: Bottom actuator

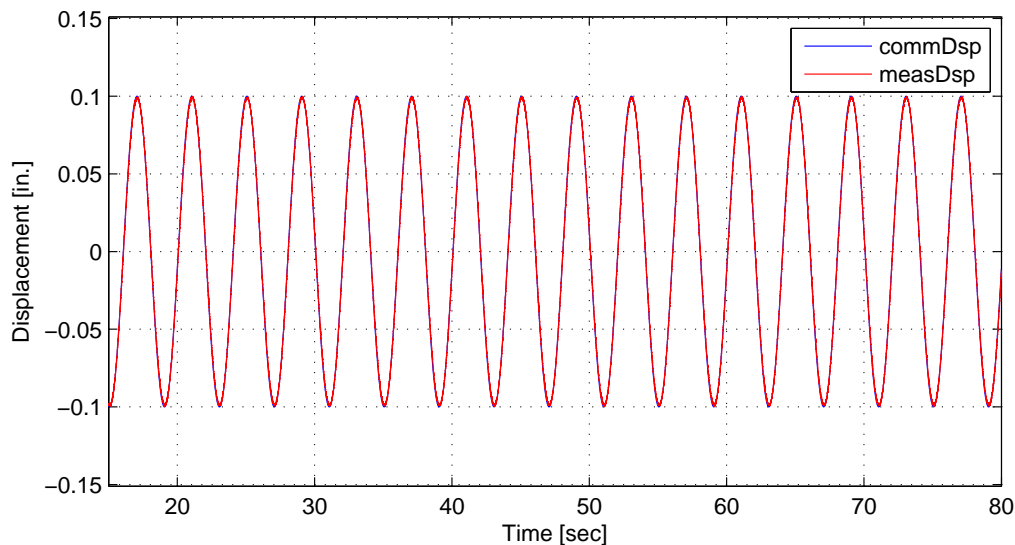


Figure B.1: Displacement control tuning plot using a sine wave with amplitude of 0.1 in. and frequency of 0.25 Hz of 1-DOF  $\mu$ -NEES experimental setup. Channel 3 of the MTS-STS controller is tuned for displacement control.

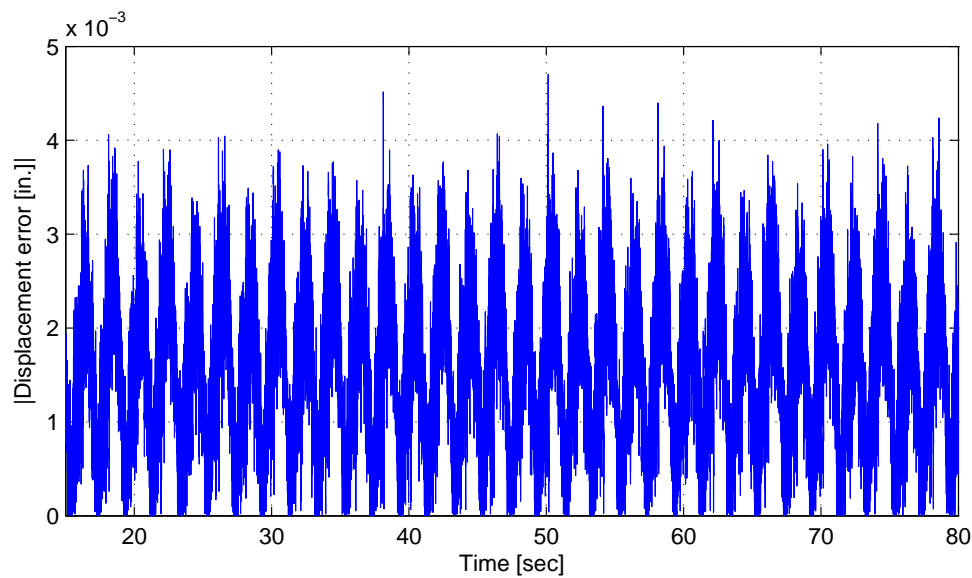


Figure B.2: Absolute value of the displacement error of the plot on Figure B.1. The mean error is 0.00154 in. The minimum error is  $4.04e - 6$  in. The maximum error is 0.00470 in.

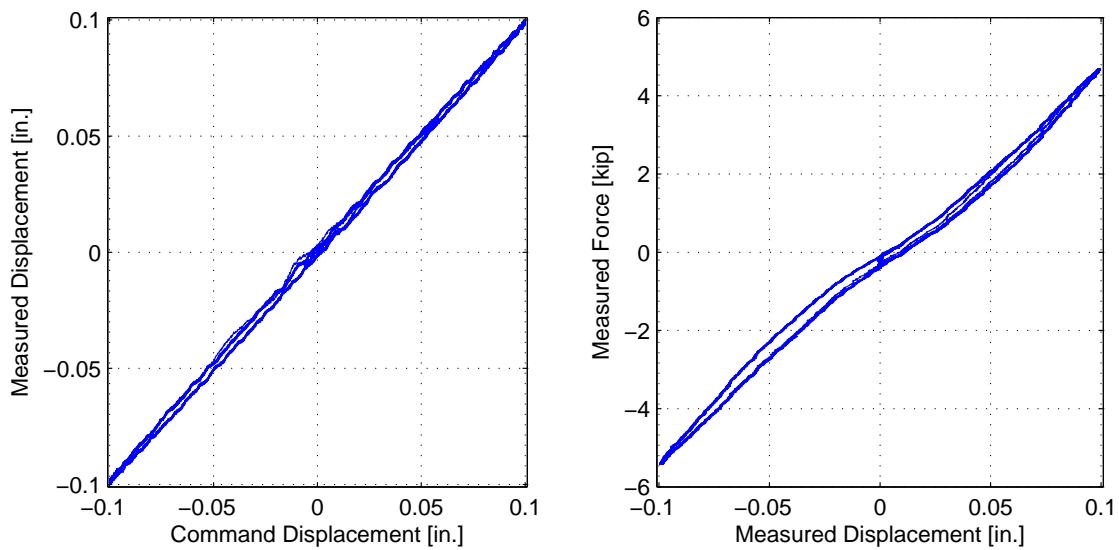


Figure B.3: Left - Tracking plot showing of 1-DOF  $\mu$ -NEES experimental setup using Channel 3 of the MTS-STC controller in displacement control. Right - Hysteresis plot of 1-DOF  $\mu$ -NEES experimental setup using Channel 3 of the MTS-STC controller in displacement control.



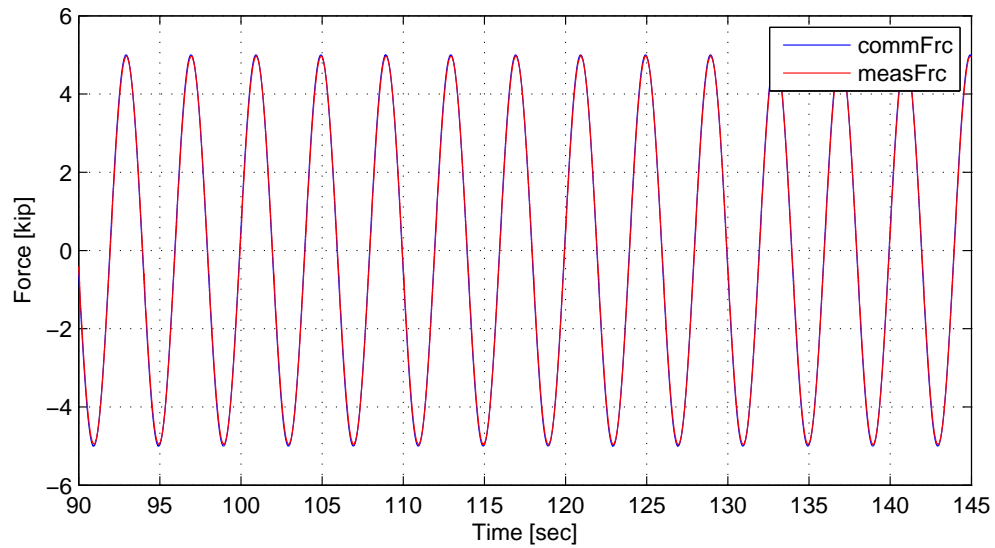


Figure B.4: Force control tuning plot using a sine wave with amplitude of 5 kips and frequency of 0.25 Hz of 1-DOF  $\mu$ -NEES experimental setup. Channel 3 of the MTS-STC controller is tuned for force control.

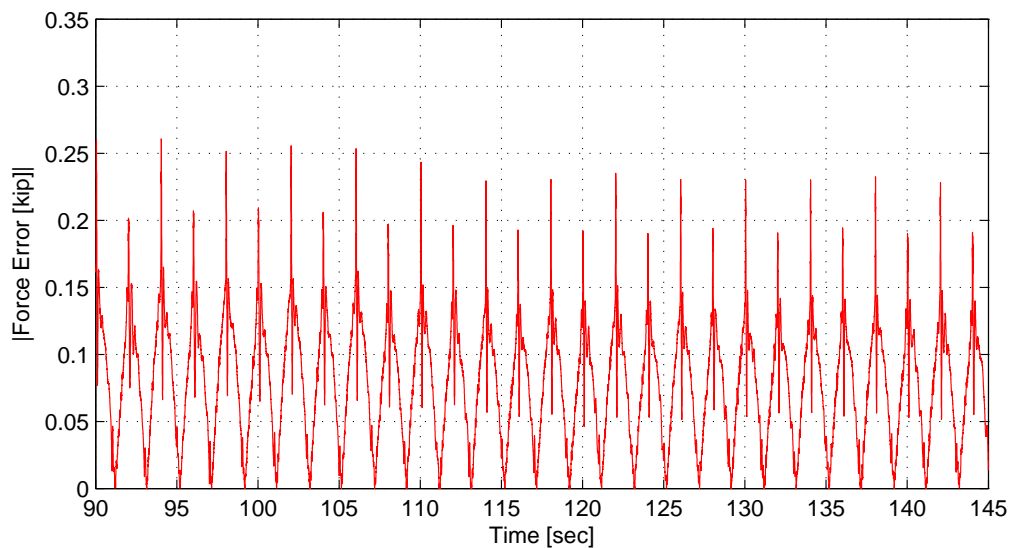


Figure B.5: Absolute value of the force error of the plot on Figure B.4. The mean error is 0.0796 kip. The minimum error is  $2.38e - 6$  kip. The maximum error is 0.261 kip.

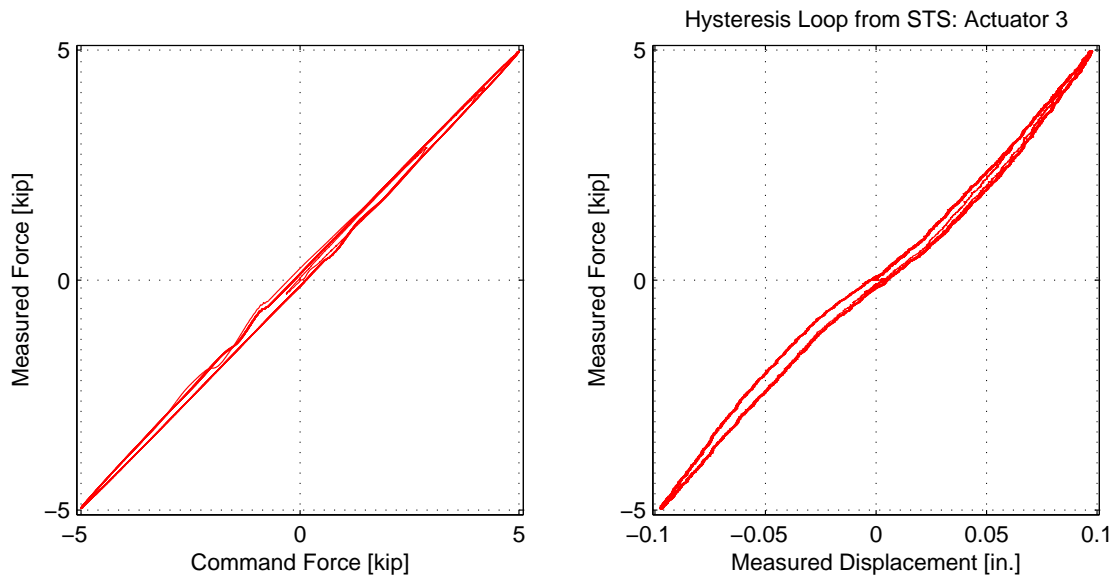


Figure B.6: Left - Tracking plot showing of 1-DOF  $\mu$ -NEES experimental setup using Channel 3 of the MTS-STS controller in displacement control. Right - Hysteresis plot of 1-DOF  $\mu$ -NEES experimental setup using Channel 3 of the MTS-STS controller in displacement control.

<b>Control mode</b>	$ error $	$ error $ (% span)
DC-Min	$4.04 \times 10^{-6}$ in.	0.00404%
DC-Mean	0.00154 in.	1.54%
DC-Max	0.00470 in.	4.60%
FC-Min	$2.38 \times 10^{-6}$ kip	$4.76 \times 10^{-5}$ %
FC-Mean	0.0796 kip	1.59%
FC-Max	0.261 kip	5.22%

Table B.1: The absolute errors of nonlinear hybrid simulations for the 1-DOF setup for various time integration schemes at the control system level. The minimums, means are maximums are shown in value and percent of span.

## B.2 Tuning of MTS-STS Channel 4: Top actuator

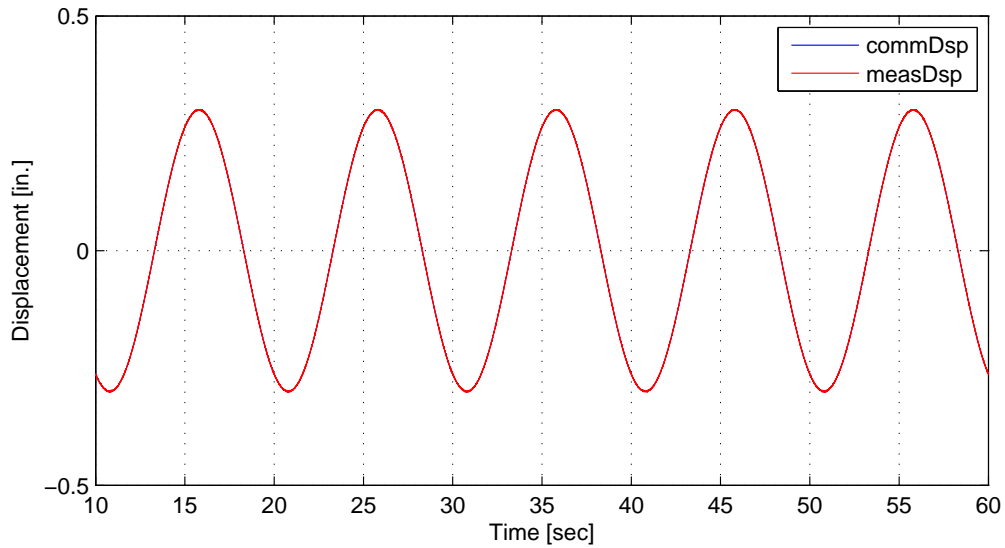


Figure B.7: Displacement control tuning plot using a sine wave with amplitude of 0.3 in. and frequency of 0.25 Hz of 2-DOF  $\mu$ -NEES experimental setup. Channel 4 of the MTS-STS controller is tuned for displacement control.

Control mode	$ error $	$ error $ (% span)
DC-Min	$1.10 \times 10^{-8}$ in.	$3.67 \times 10^{-6}\%$
DC-Mean	$5.47 \times 10^{-5}$ in.	0.0182%
DC-Max	0.00357 in.	1.19%

Table B.2: The absolute errors of nonlinear hybrid simulations for the 1-DOF setup for various time integration schemes at the control system level. The minimums, means and maximums are shown in value and percent of span.

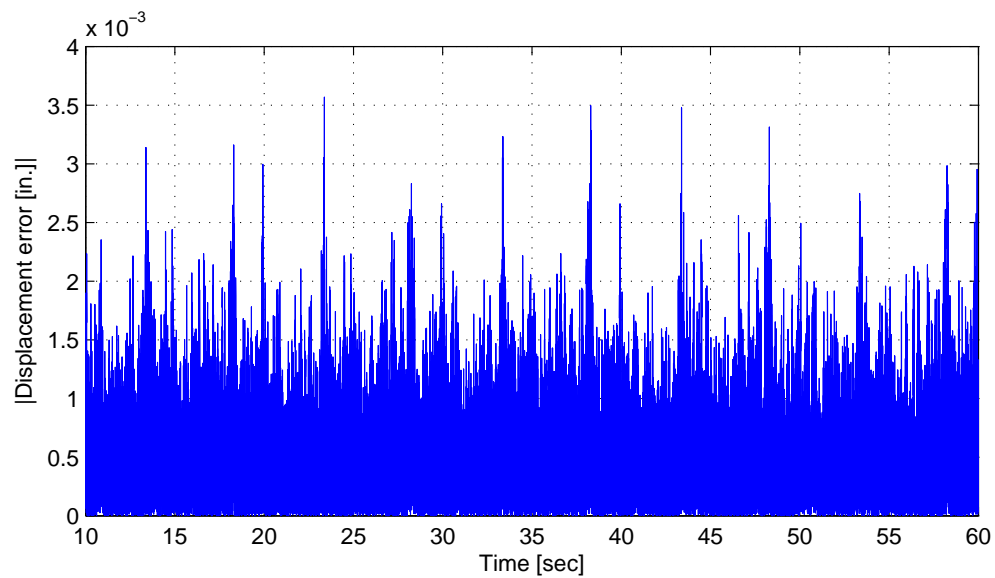


Figure B.8: Absolute value of the displacement error of the plot on Figure B.1. The mean error is  $5.47e - 5$  in. The minimum error is  $1.10e - 8$  in. The maximum error is 0.00357 in.

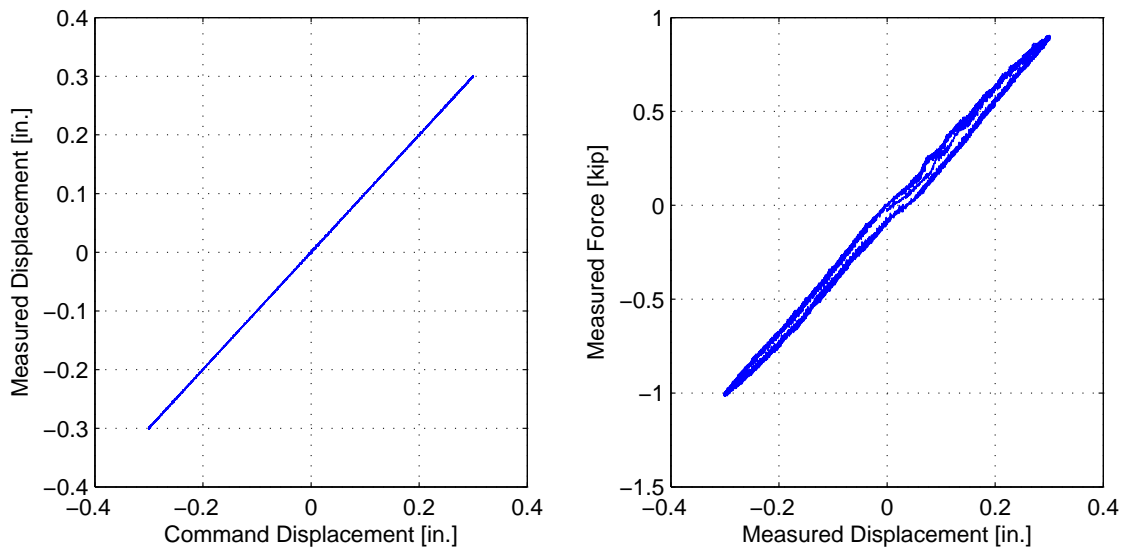


Figure B.9: Left - Tracking plot showing of 1-DOF  $\mu$ -NEES experimental setup using Channel 3 of the MTS-STS controller in displacement control. Right - Hysteresis plot of 1-DOF  $\mu$ -NEES experimental setup using Channel 3 of the MTS-STS controller in displacement control.

# Appendix C

## FC hybrid simulation experimental results

This chapter contains the experimental results for FC hybrid simulation using the  $\mu$ -NEES experimental setup in two different configurations, 1-DOF and 2-DOF setups. Linear and nonlinear hybrid simulations are run on each of the two setups. NME,  $\alpha$ -OS, NMF and NMR are deployed but not all simulations are completed successfully. Some are incomplete because either the STS controller becomes unstable or the time integration tolerance limit is not satisfied. The comparison of the analytical results of these time integration scheme is discussed in Section 3.4.

### C.1 1-DOF setup: Linear hybrid simulation

The El Centro from Section 3.1.2 is scaled to 15% to keep the specimen within its linear range. The hybrid model is shown in Figure 2.1. Element 1, the experimental element, deformations and resisting forces from the FEA level software are plotted vs time. These response quantities are compared against their numerical counterpart from the OpenSees analysis. The command and measured displacements from the STS controller for the DC hybrid simulation is shown. The command and measure forces from the STS controller for CFC and EFC hybrid simulation are plotted vs time as well. The results for these simulations are plotted in Figures C.1 to C.13.

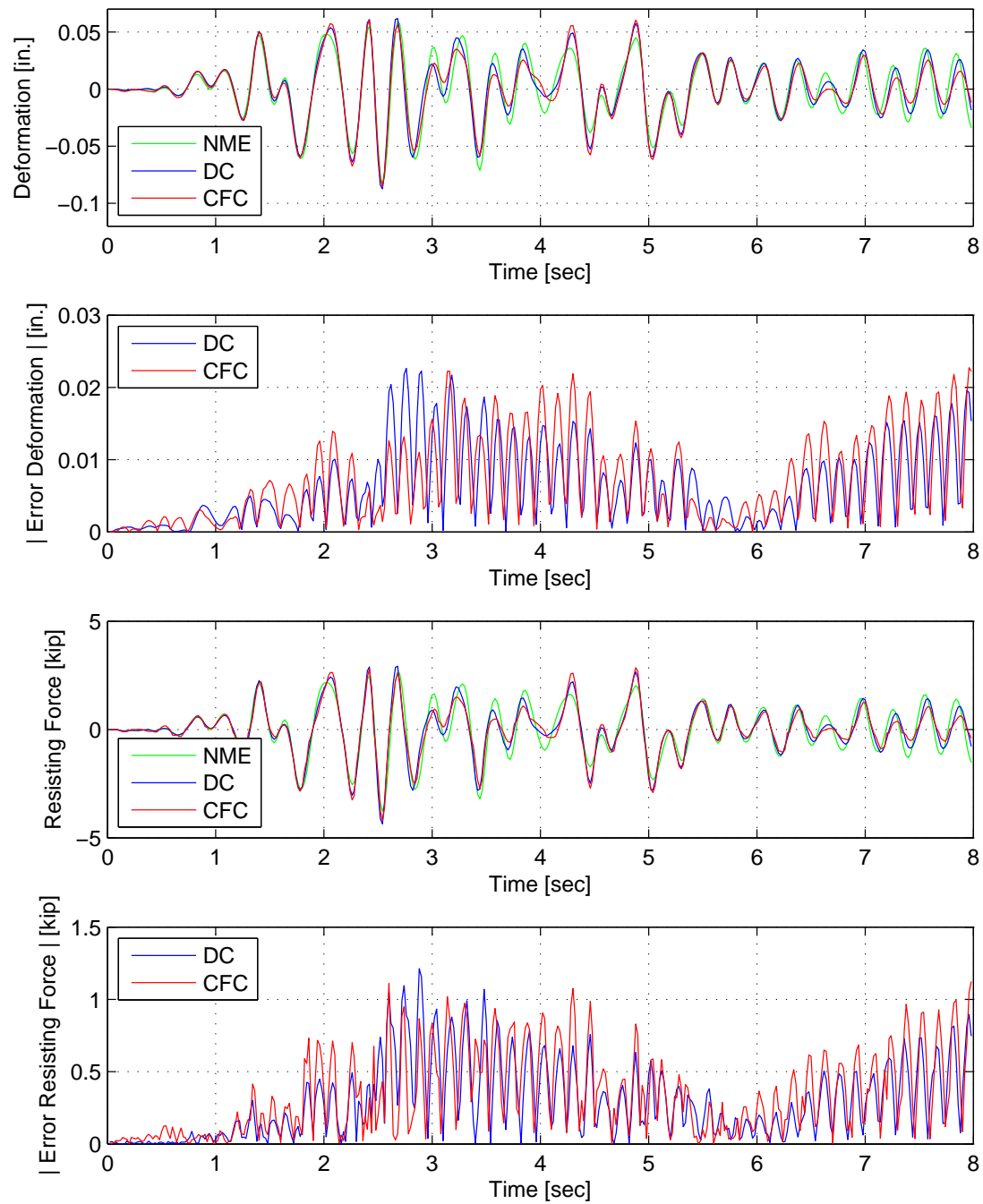


Figure C.1: Displacement-time history plot (top) and displacement error (middle-top) of Element 1 for FC hybrid simulation. Resisting force-time history plot (middle-bottom) and absolute resisting error (bottom) of Element 1. The The ground motion is scaled 15% to keep the simulation in the linear range. The 1-DOF setup uses NME with DC and CFC.

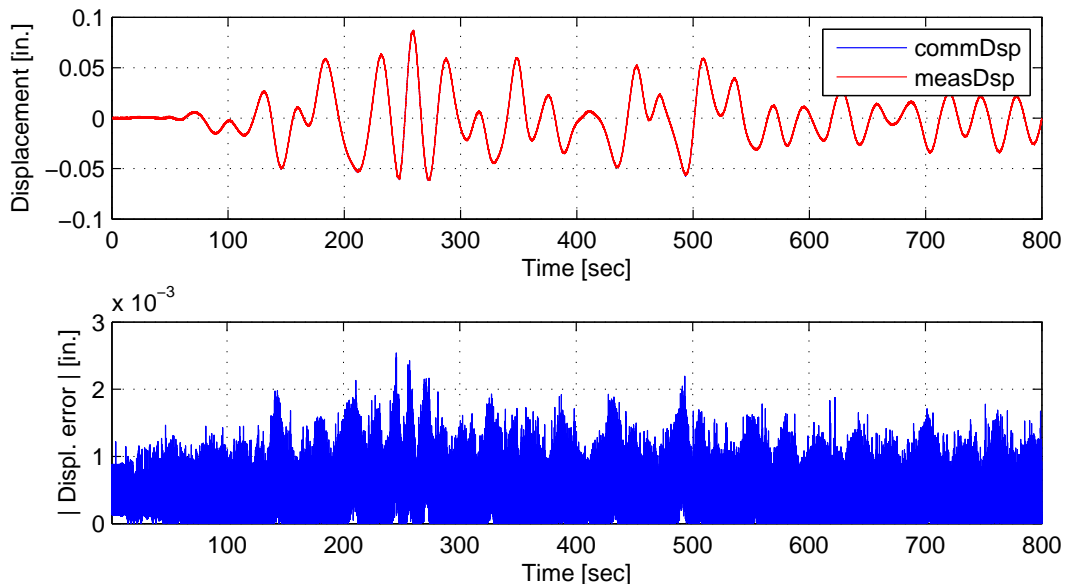


Figure C.2: Top: Command and measured displacement plot from the STS controller for DC hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled 15% to keep the simulation in the linear range. The 1-DOF setup uses NME.

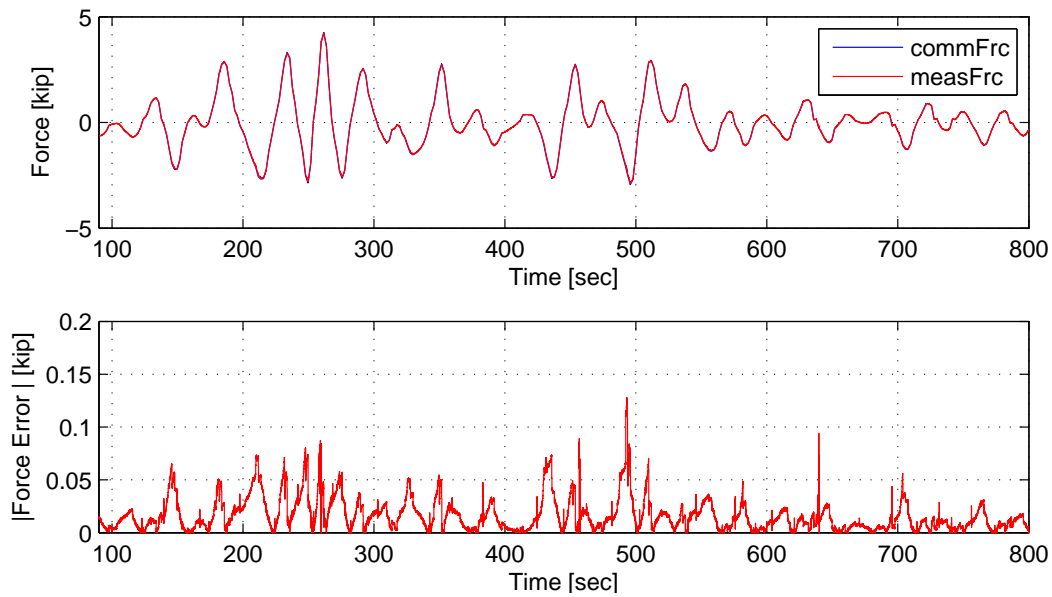


Figure C.3: Top: Command and measured force plot from the STS controller for CFC hybrid simulation. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled 15% to keep the simulation in the linear range. The 1-DOF setup uses NME.

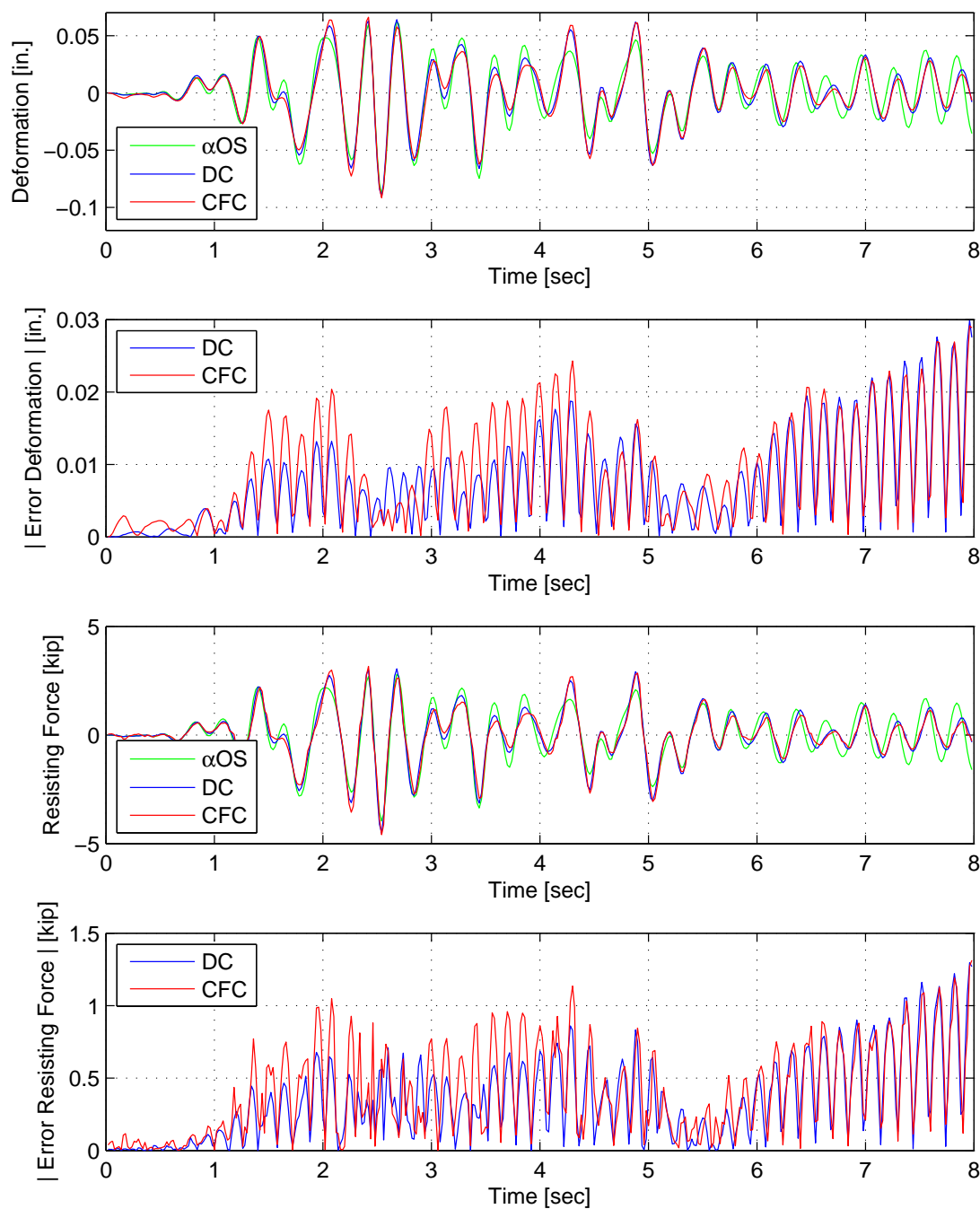


Figure C.4: Displacement-time history plot (top) and displacement error (middle-top) of Element 1 for FC hybrid simulation. Resisting force-time history plot (middle-bottom) and absolute resisting error (bottom) of Element 1. The The ground motion is scaled 15% to keep the simulation in the linear range. The 1-DOF setup uses  $\alpha$ OS with DC and CFC.



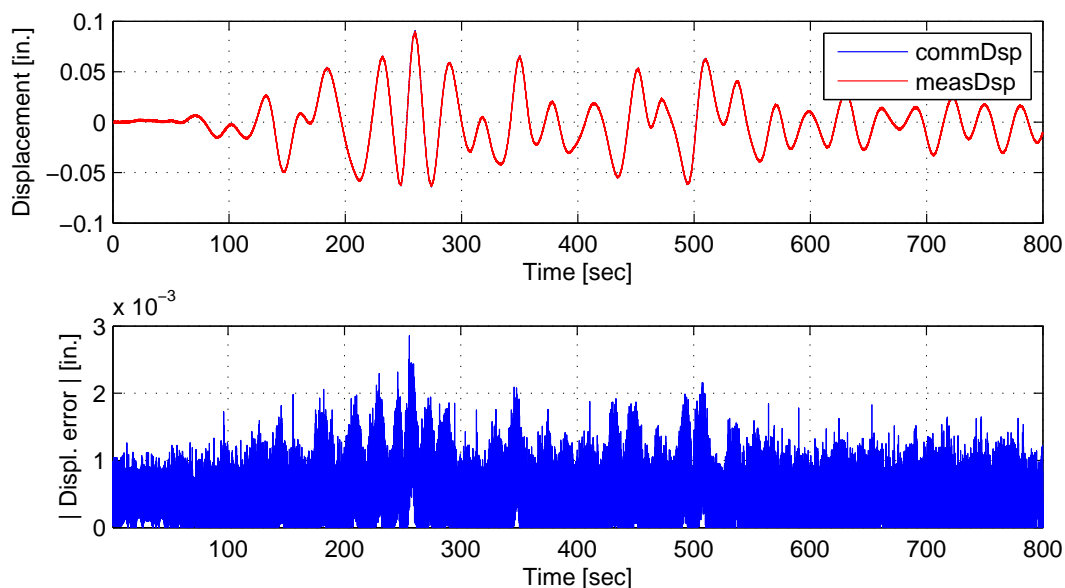


Figure C.5: Top: Command and measured displacement plot from the STS controller for DC hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled 15% to keep the simulation in the linear range. The 1-DOF setup uses  $\alpha$ OS.

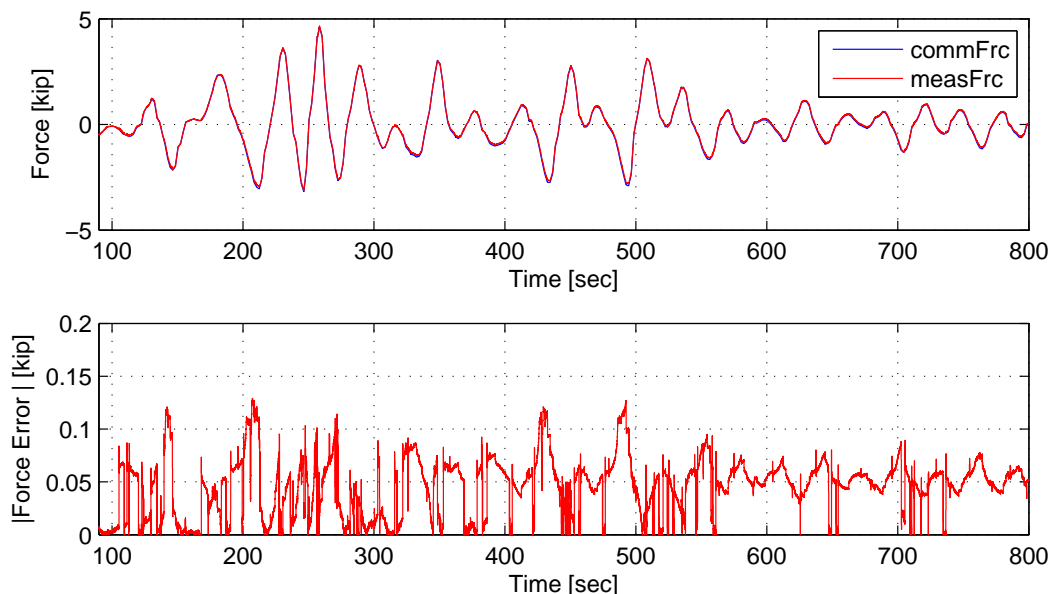


Figure C.6: Top: Command and measured force plot from the STS controller for CFC hybrid simulation. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled 15% to keep the simulation in the linear range. The 1-DOF setup uses  $\alpha$ OS.

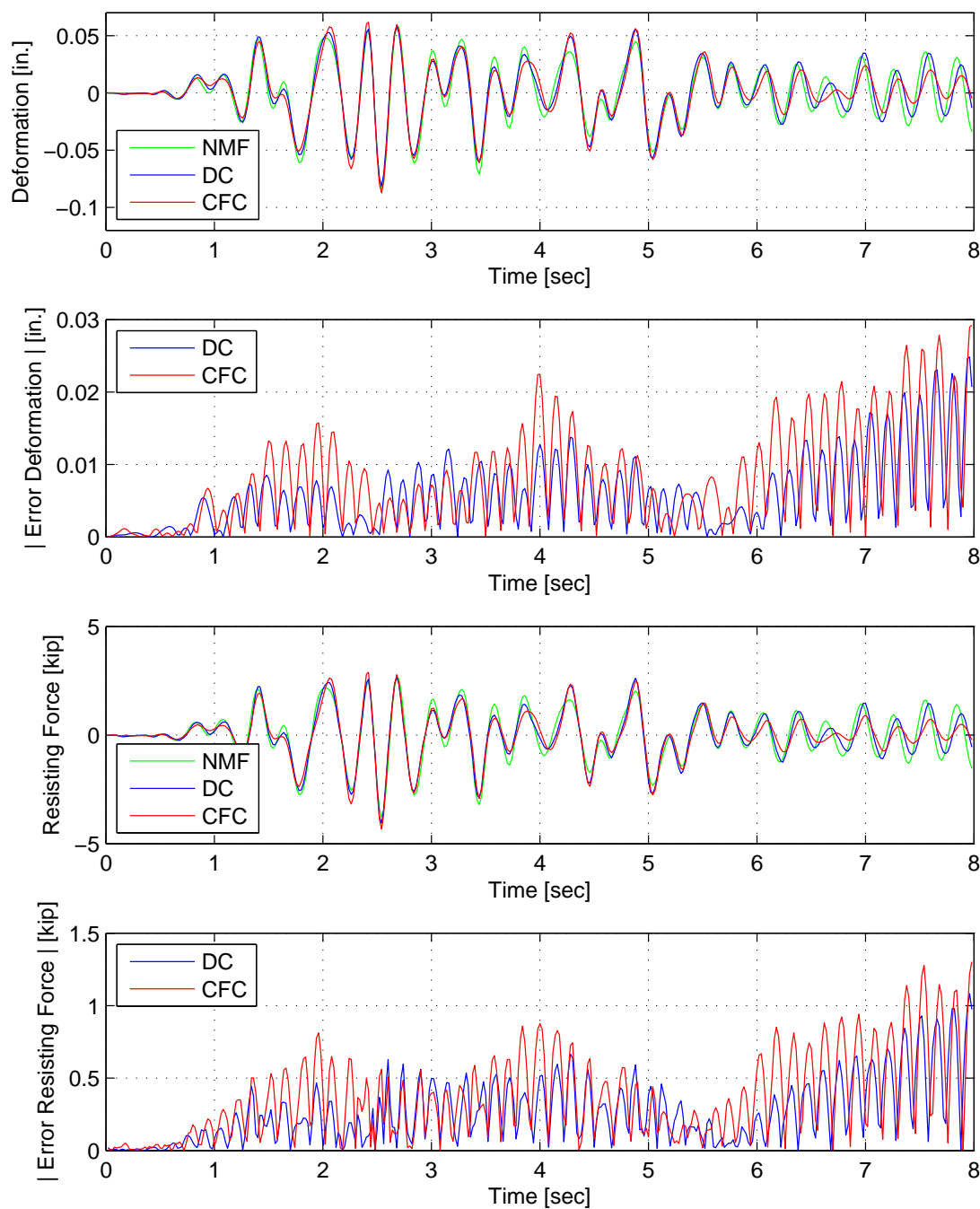


Figure C.7: Displacement-time history plot (top) and displacement error (middle-top) of Element 1 for FC hybrid simulation. Resisting force-time history plot (middle-bottom) and absolute resisting error (bottom) of Element 1. The The ground motion is scaled 15% to keep the simulation in the linear range. The 1-DOF setup uses NMF with DC and CFC.

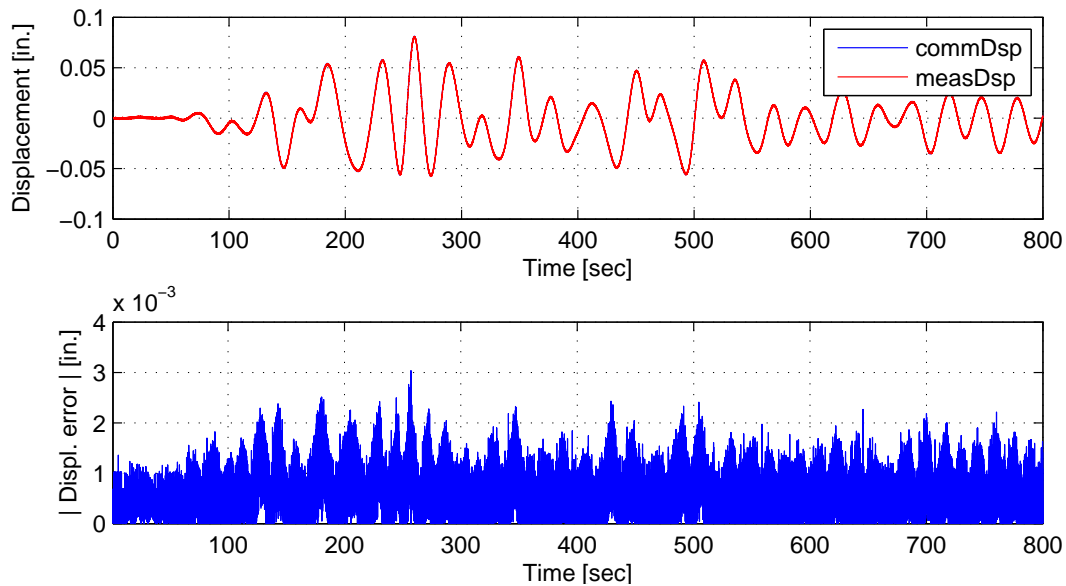


Figure C.8: Top: Command and measured displacement plot from the STS controller for DC hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled 15% to keep the simulation in the linear range. The 1-DOF setup uses NMF.

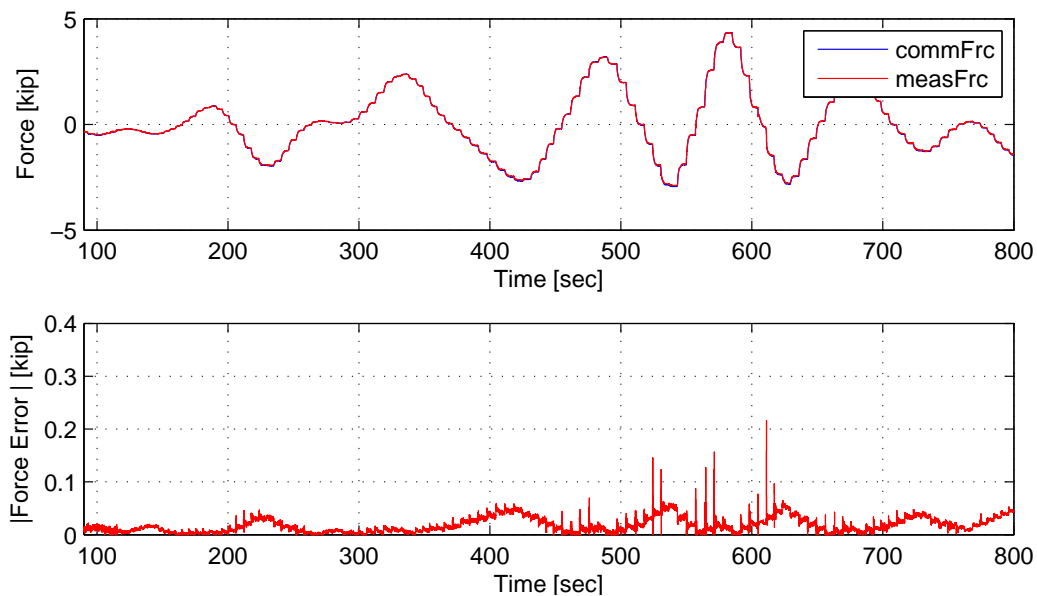


Figure C.9: Top: Command and measured force plot from the STS controller for CFC hybrid simulation. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled 15% to keep the simulation in the linear range. The 1-DOF setup uses NMF.

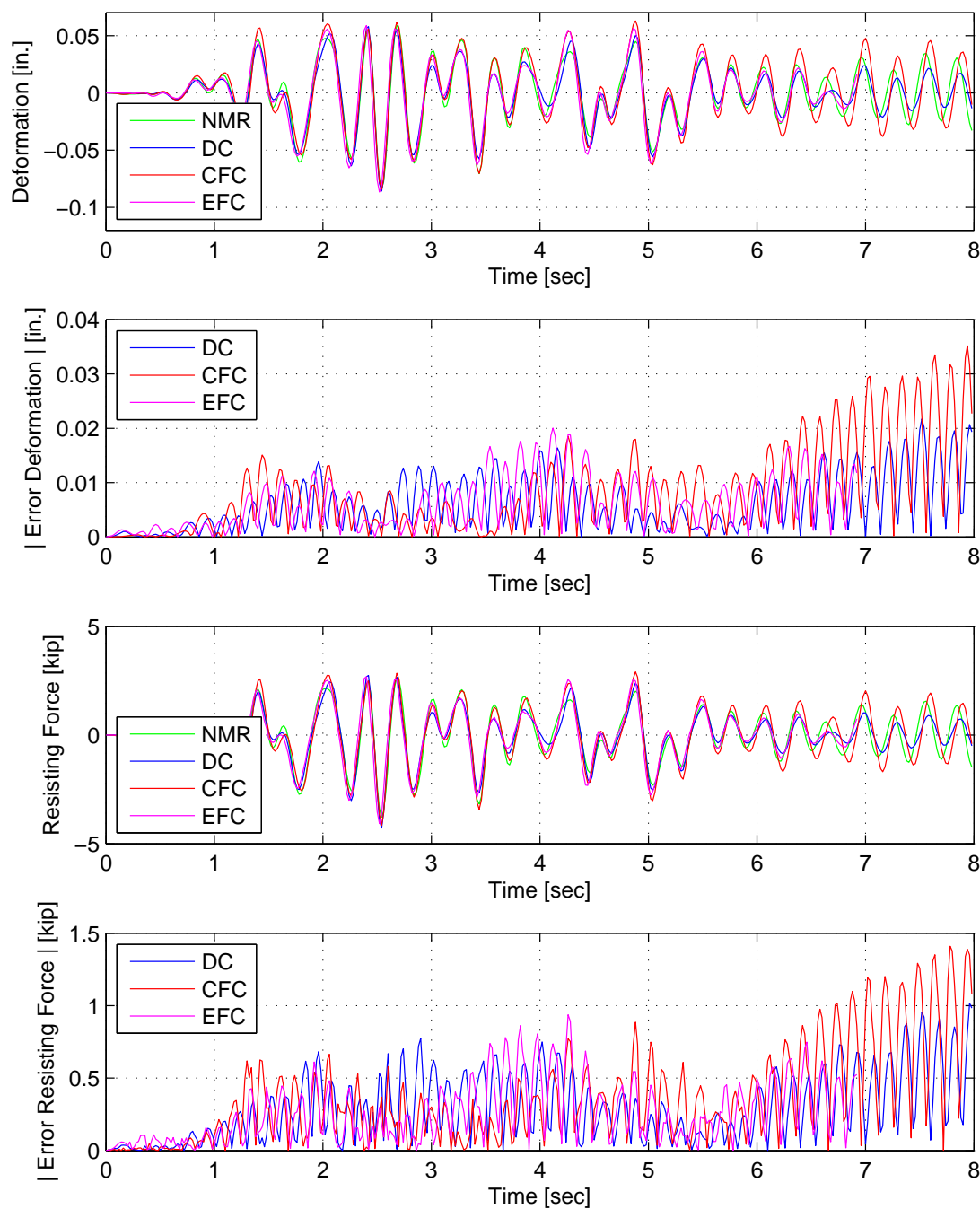


Figure C.10: Displacement-time history plot (top) and displacement error (middle-top) of Element 1 for FC hybrid simulation. Resisting force-time history plot (middle-bottom) and absolute resisting error (bottom) of Element 1. The The ground motion is scaled 15% to keep the simulation in the linear range. The 1-DOF setup uses NMR with DC, CFC and EFC.

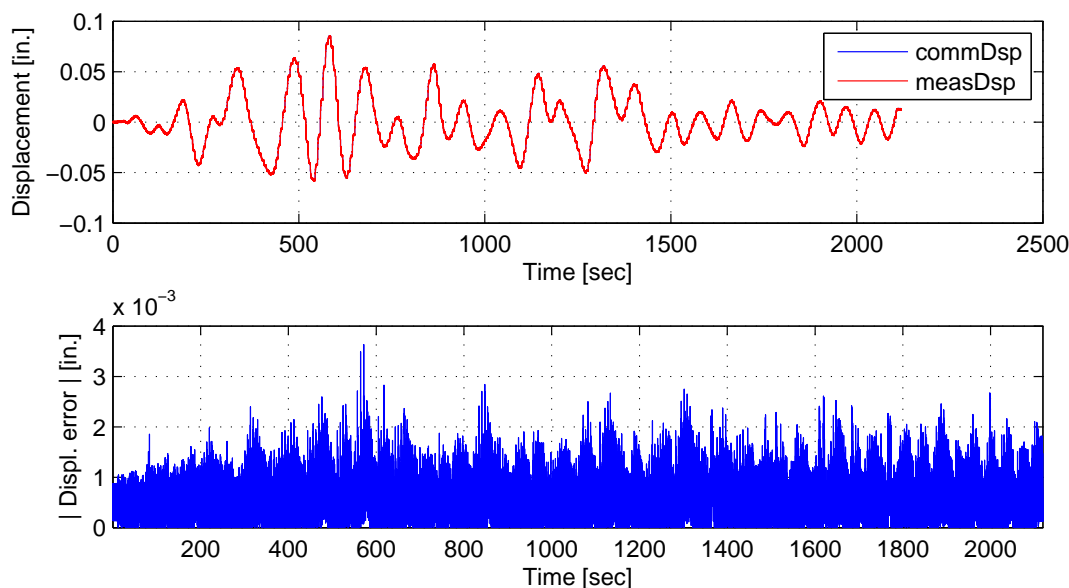


Figure C.11: Top: Command and measured displacement plot from the STS controller for DC hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled 15% to keep the simulation in the linear range. The 1-DOF setup uses NMR.

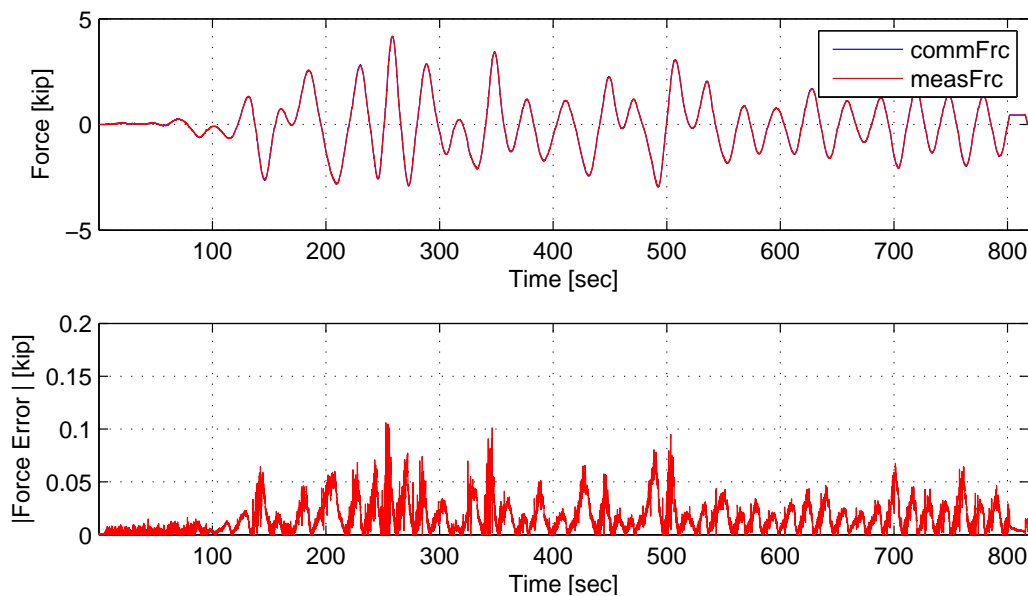


Figure C.12: Top: Command and measured force plot from the STS controller for CFC hybrid simulation. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled 15% to keep the simulation in the linear range. The 1-DOF setup uses NMR.

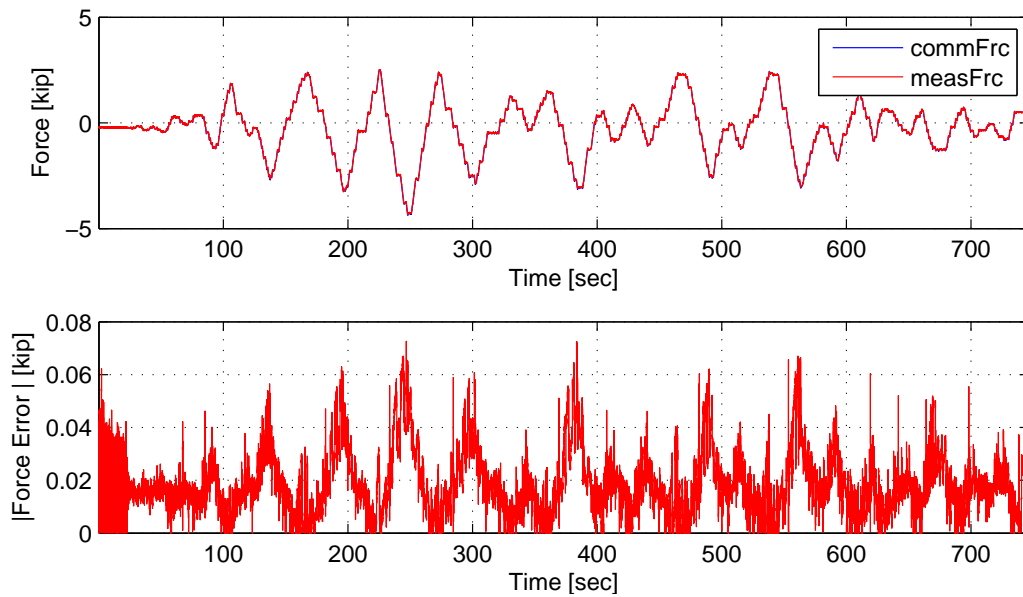


Figure C.13: Top: Command and measured force plot from the STS controller for EFC hybrid simulation. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled 15% to keep the simulation in the linear range. The 1-DOF setup uses NMR.

## C.2 1-DOF setup: Nonlinear hybrid simulation

The El Centro from Section 3.1.2 is scaled to 40%. The hybrid model is shown in Figure 2.1. Element 1, the experimental element, deformations and resisting forces from the FEA software level are plotted vs time. These response quantities are compared against their numerical counterpart from the OpenSees analysis. The command and measured displacements from the STS controller for the DC hybrid simulation is shown. The command and measure forces from the STS controller for CFC and EFC hybrid simulation are plotted vs time as well. The secant estimation is used for the CFC method since all the CFC methods degenerate to estimating the secant when only 1-DOF is used. The results for these simulations are plotted in Figures C.14 to C.19.

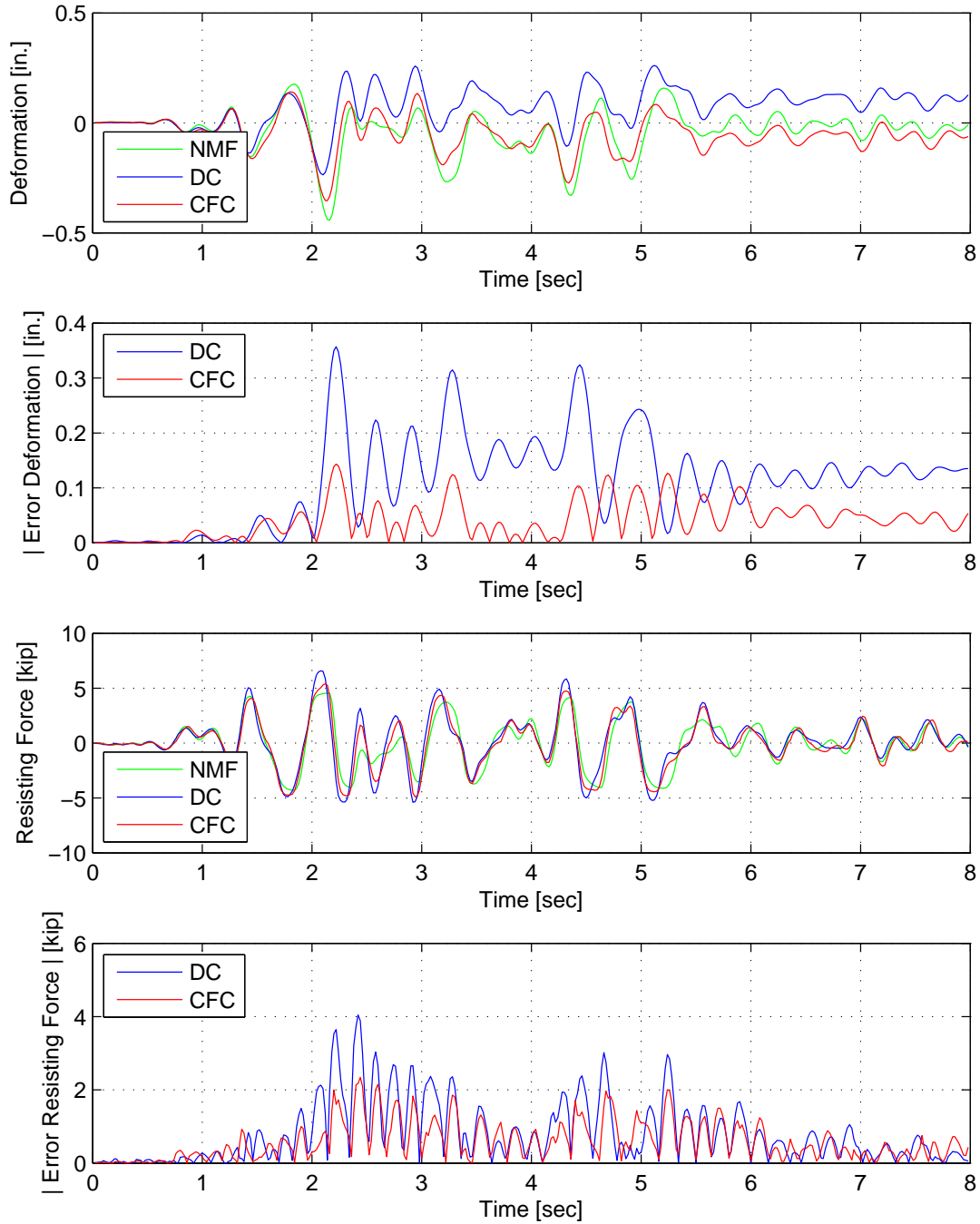


Figure C.14: Displacement-time history plot (top) and displacement error (middle-top) of Element 1 for FC hybrid simulation. Resisting force-time history plot (middle-bottom) and absolute resisting error (bottom) of Element 1. The ground motion is scaled to 40% to push the specimen into the nonlinear range. The 1-DOF setup is used with NMF with DC and CFC.

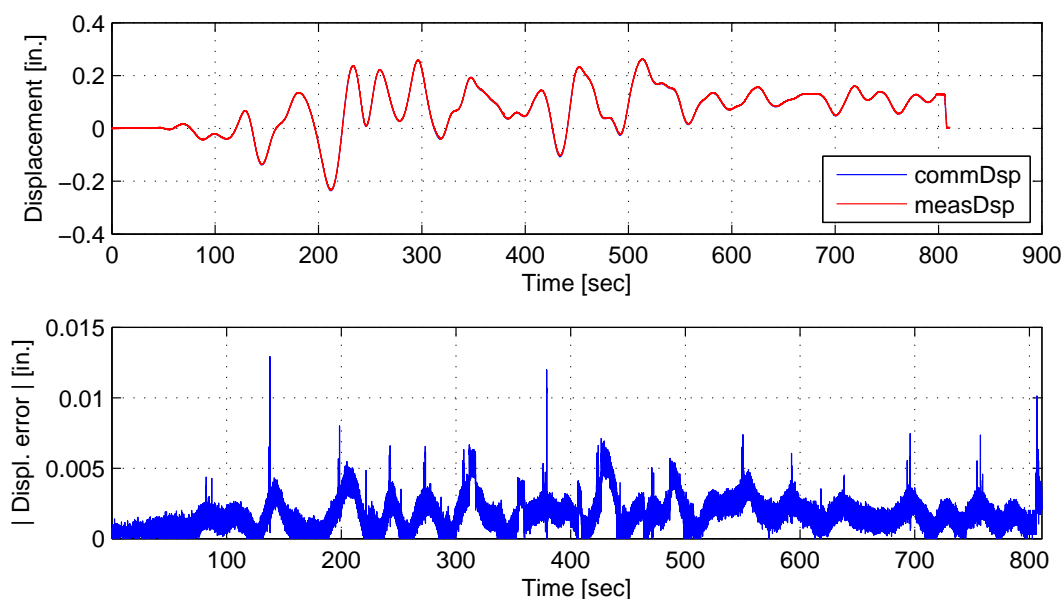


Figure C.15: Top: Command and measured displacement plot from the STS controller for DC hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 40% to push the specimen into the nonlinear range. The 1-DOF setup is used with NMF.

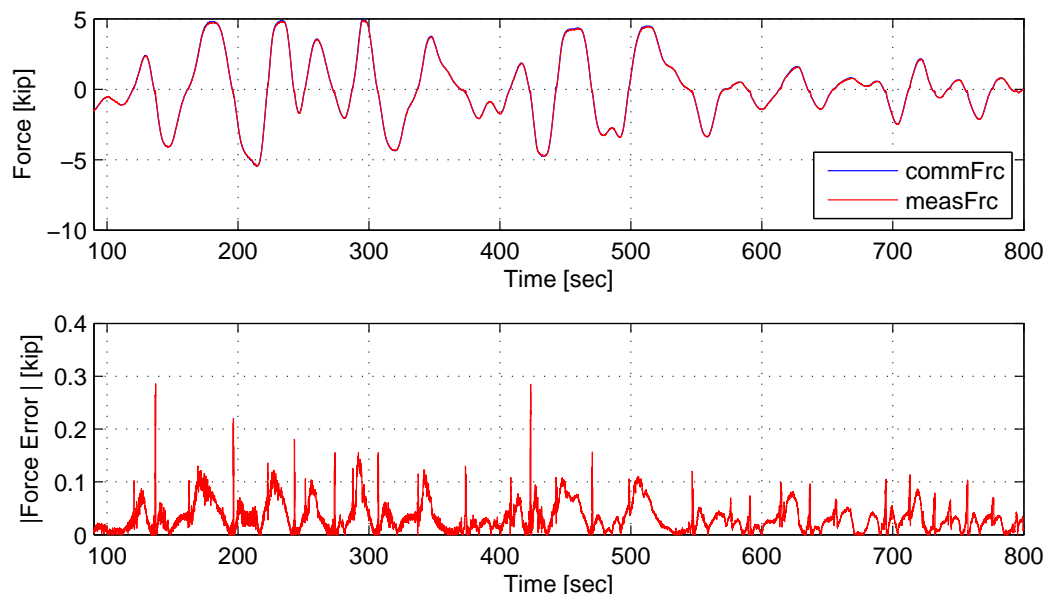


Figure C.16: Top: Command and measured force plot from the STS controller for CFC hybrid simulation. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 40% to push the specimen into the nonlinear range. The 1-DOF setup is used with NMF.



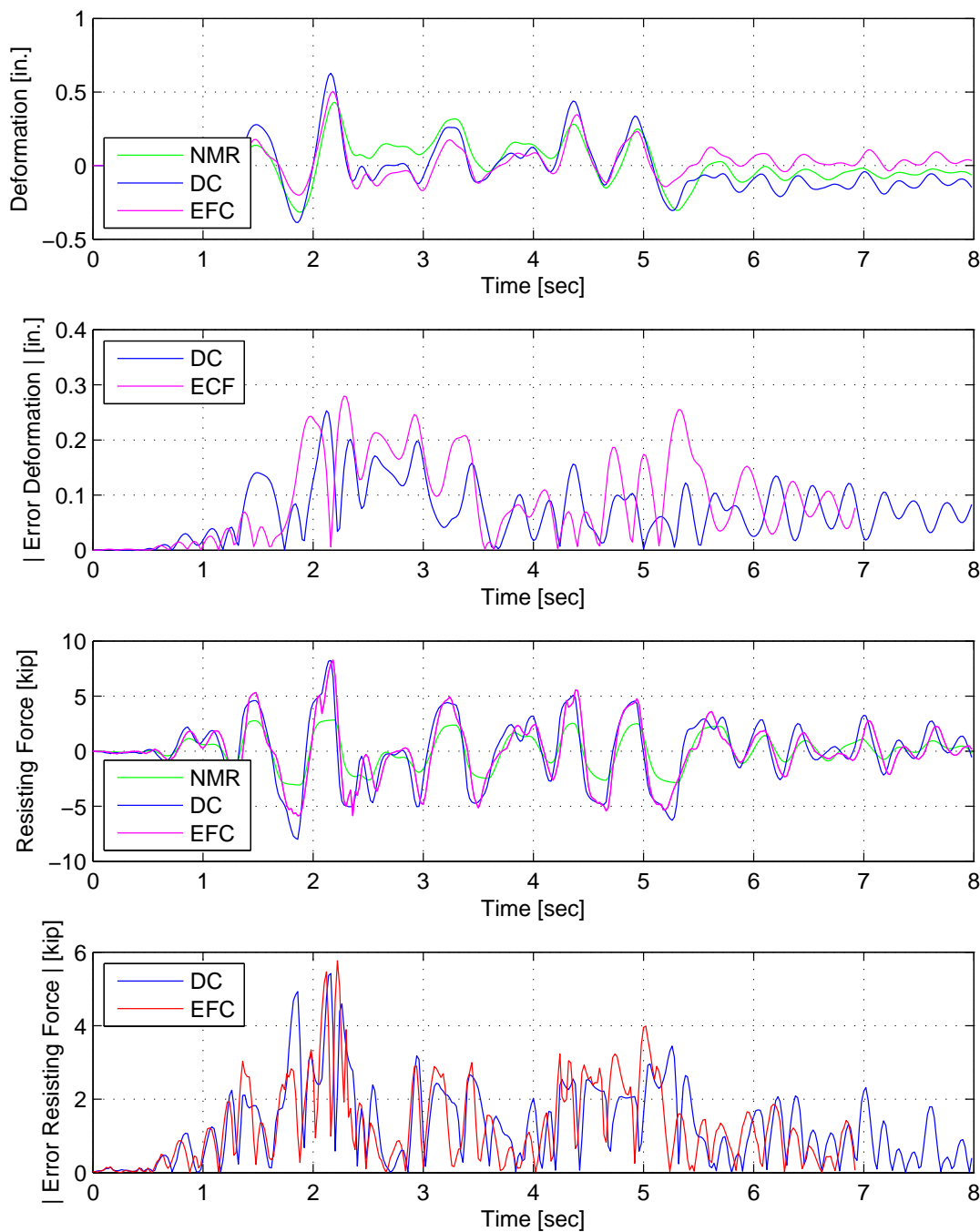


Figure C.17: Displacement-time history plot (top) and displacement error (middle-top) of Element 1 for FC hybrid simulation. Resisting force-time history plot (middle-bottom) and absolute resisting error (bottom) of Element 1. The ground motion is scaled to 40% to push the specimen into the nonlinear range. The 1-DOF setup is used with NMR with DC and CFC.

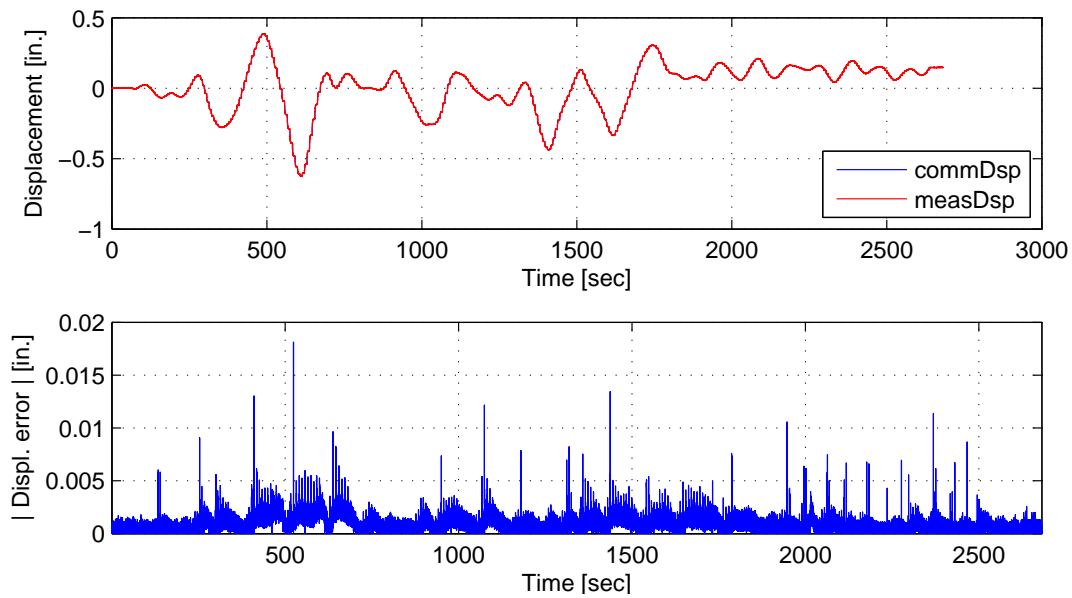


Figure C.18: Top: Command and measured displacement plot from the STS controller for DC hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 40% to push the specimen into the nonlinear range. The 1-DOF setup is used with NMR.

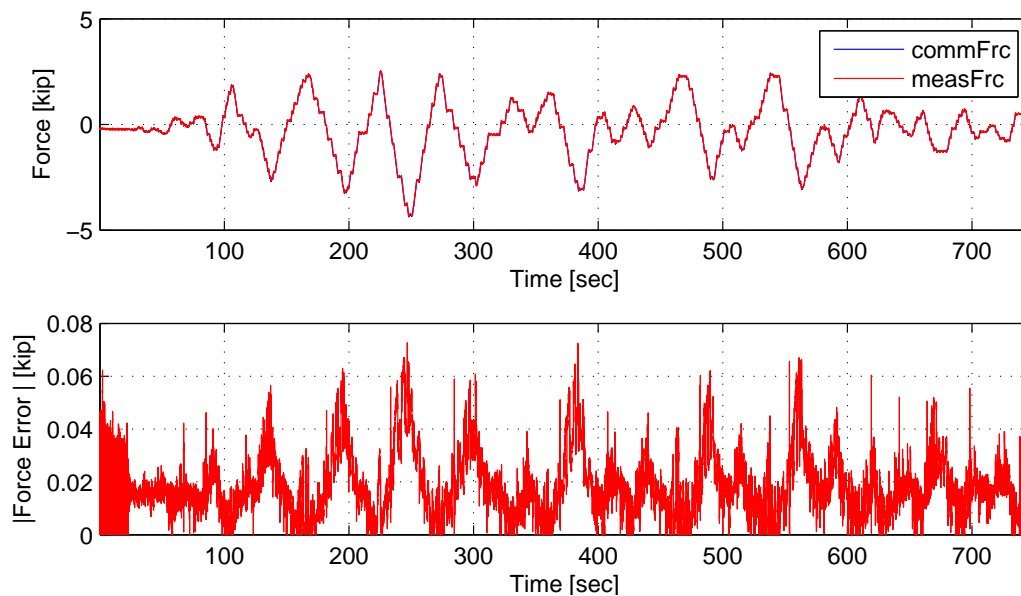


Figure C.19: Top: Command and measured force plot from the STS controller for EFC hybrid simulation. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 40% to push the specimen into the nonlinear range. The 1-DOF setup is used with NMR.

### **C.3 2-DOF setup: Linear hybrid simulation**

The El Centro from Section 3.1.2 is scaled to 10% to keep the specimen within its linear range. The hybrid model is shown in Figure 3.21 (right). The nodal displacements and forces of the experimental element from the FEA software level are plotted vs time. These response quantities are compared against their numerical counterpart from the OpenSees analysis. The command and measured displacements from the STS controller for the DC hybrid simulation is shown for both actuators. The command and measure forces from the STS controller for CFC and EFC hybrid simulation are plotted vs time as well. Only CFC:BFGS results are plot even though the other methods were tested. This is because all the CFC methods use the initial stiffness matrix and no updated to the stiffness matrix are made during th linear simulation. The results for these simulations are plotted in Figures C.20 to C.36.

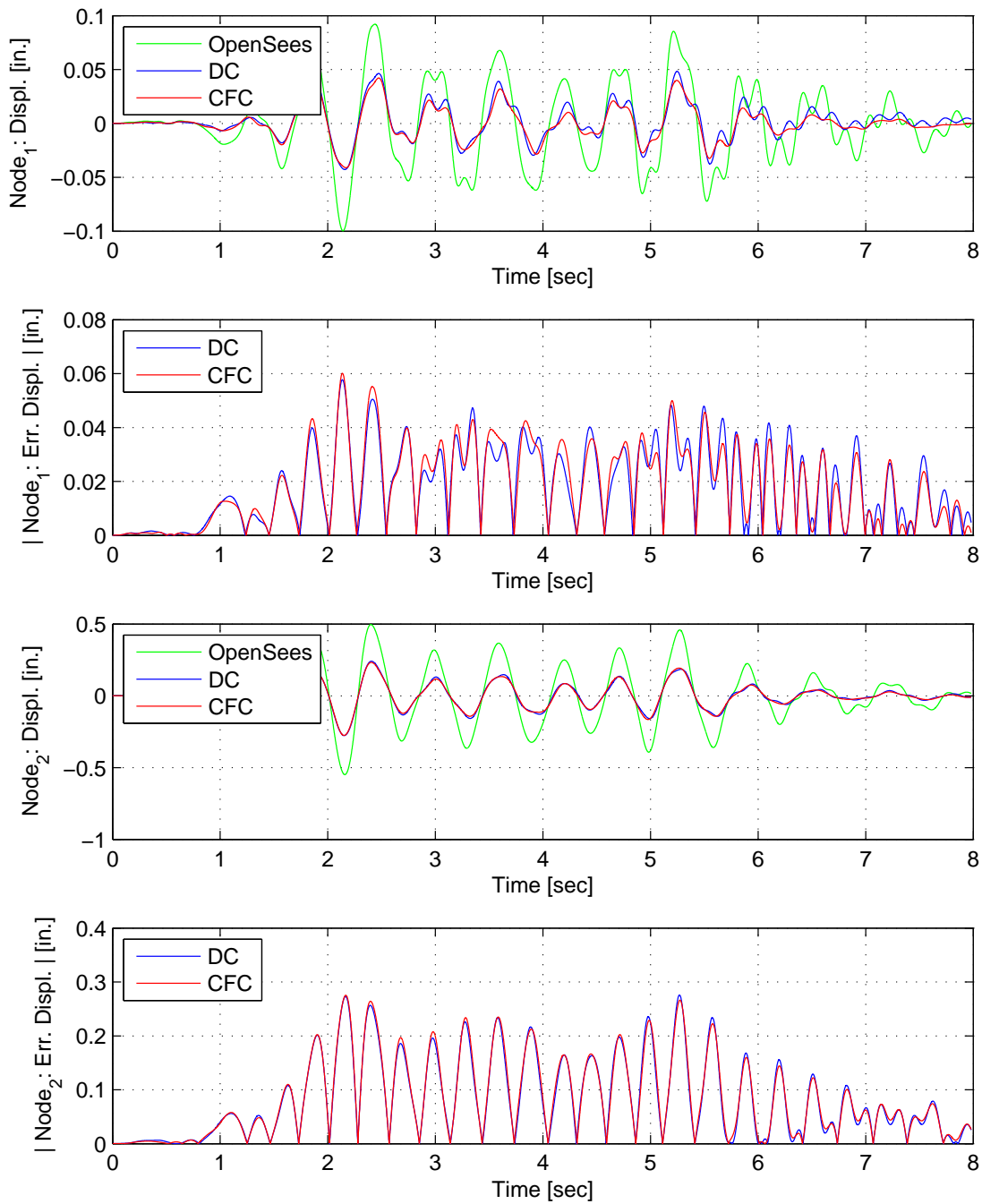


Figure C.20: Node 1 displacement-time history plot (top), Node 1 absolute displacement error (middle-top), Node 2 displacement-time history plot (middle-bottom) and Node 2 absolute displacement error (bottom) for FC hybrid simulation. The ground motion is scaled to 10% to keep the simulation in the linear range. The 2-DOF setup uses NME with DC and CFC.

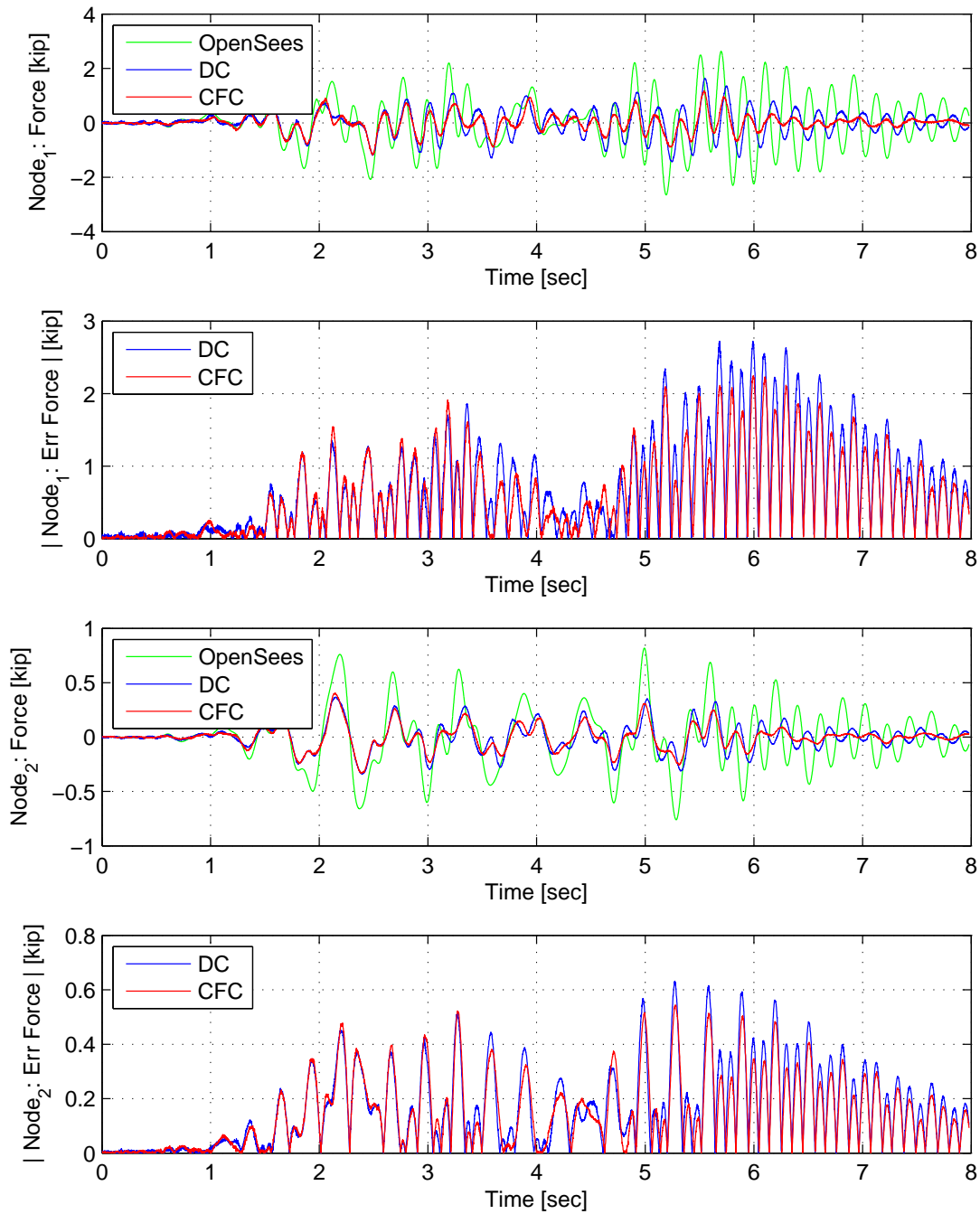


Figure C.21: Node 1 force-time history plot (top), Node 1 absolute force error (middle-top), Node 2 force-time history plot (middle-bottom) and Node 2 absolute force error (bottom) for FC hybrid simulation. The ground motion is scaled to 10% to keep the simulation in the linear range. The 2-DOF setup uses NME with DC and CFC.

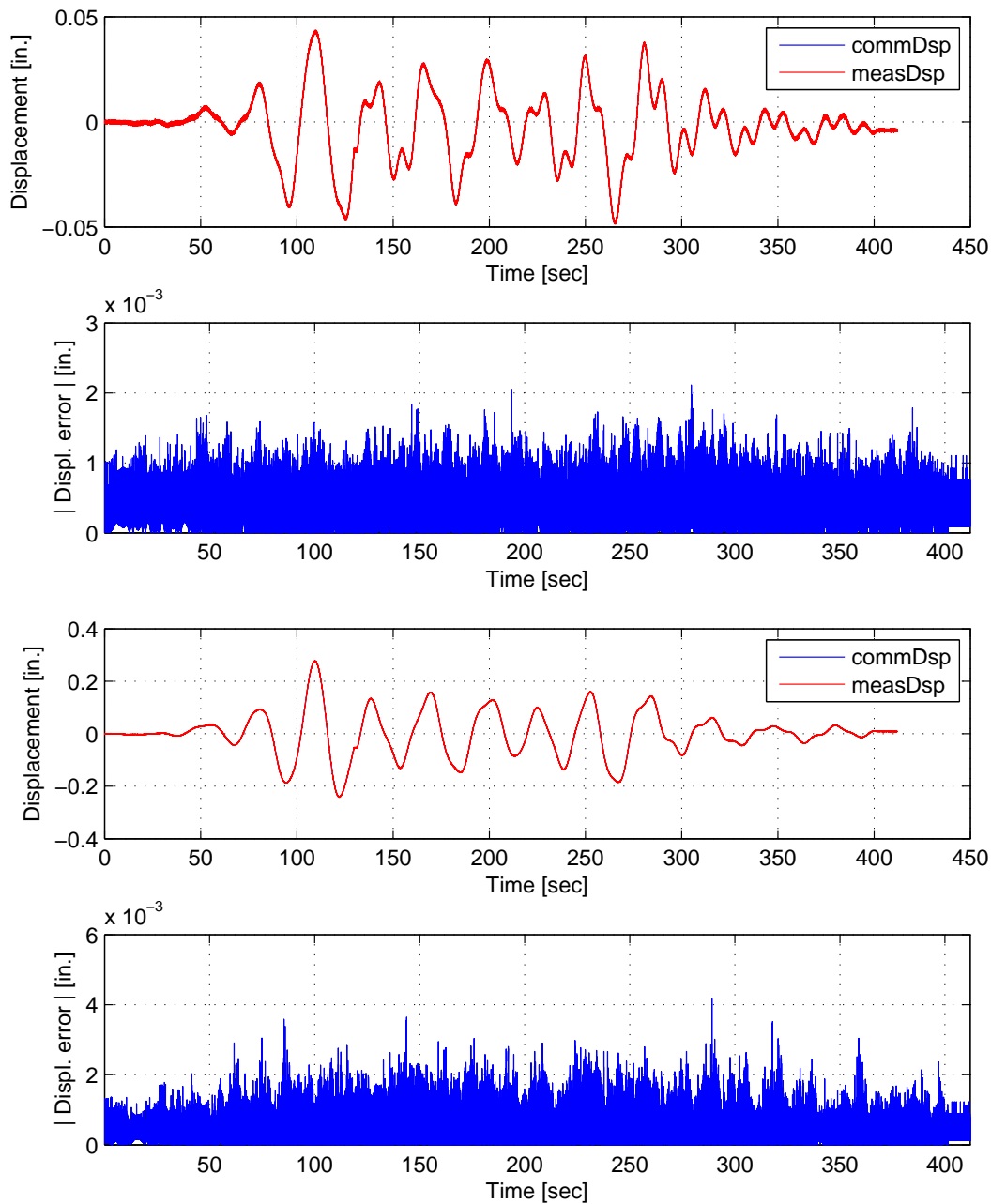


Figure C.22: Bottom actuator command and measured displacement plots from the STS controller in DC mode (top). Bottom actuator absolute error between command and measured displacements from the STS controller (middle-top). Top actuator command and measured displacement plots from the STS controller in DC mode (middle-bottom). Top actuator absolute error between command and measured displacements from the STS controller (bottom). The ground motion is scaled to 10% to keep the simulation in the linear range. The 2-DOF setup uses NME.

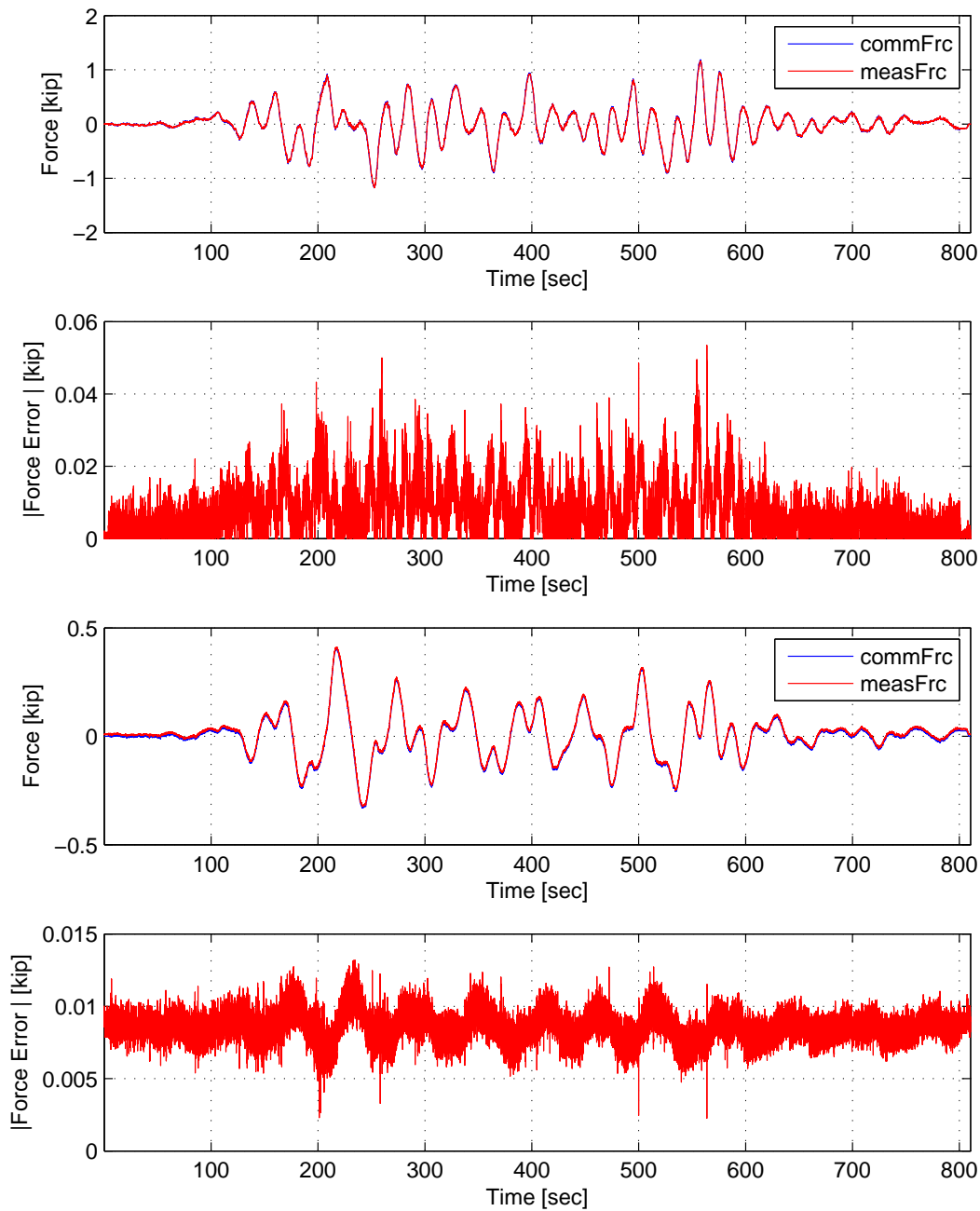


Figure C.23: Bottom actuator command and measured force plots from the STS controller in DC mode (top). Bottom actuator absolute error between command and measured force from the STS controller (middle-top). Top actuator command and measured force plots from the STS controller in DC mode (middle-bottom). Top actuator absolute error between command and measured force from the STS controller (bottom). The ground motion is scaled to 10% to keep the simulation in the linear range. The 2-DOF setup uses NME.

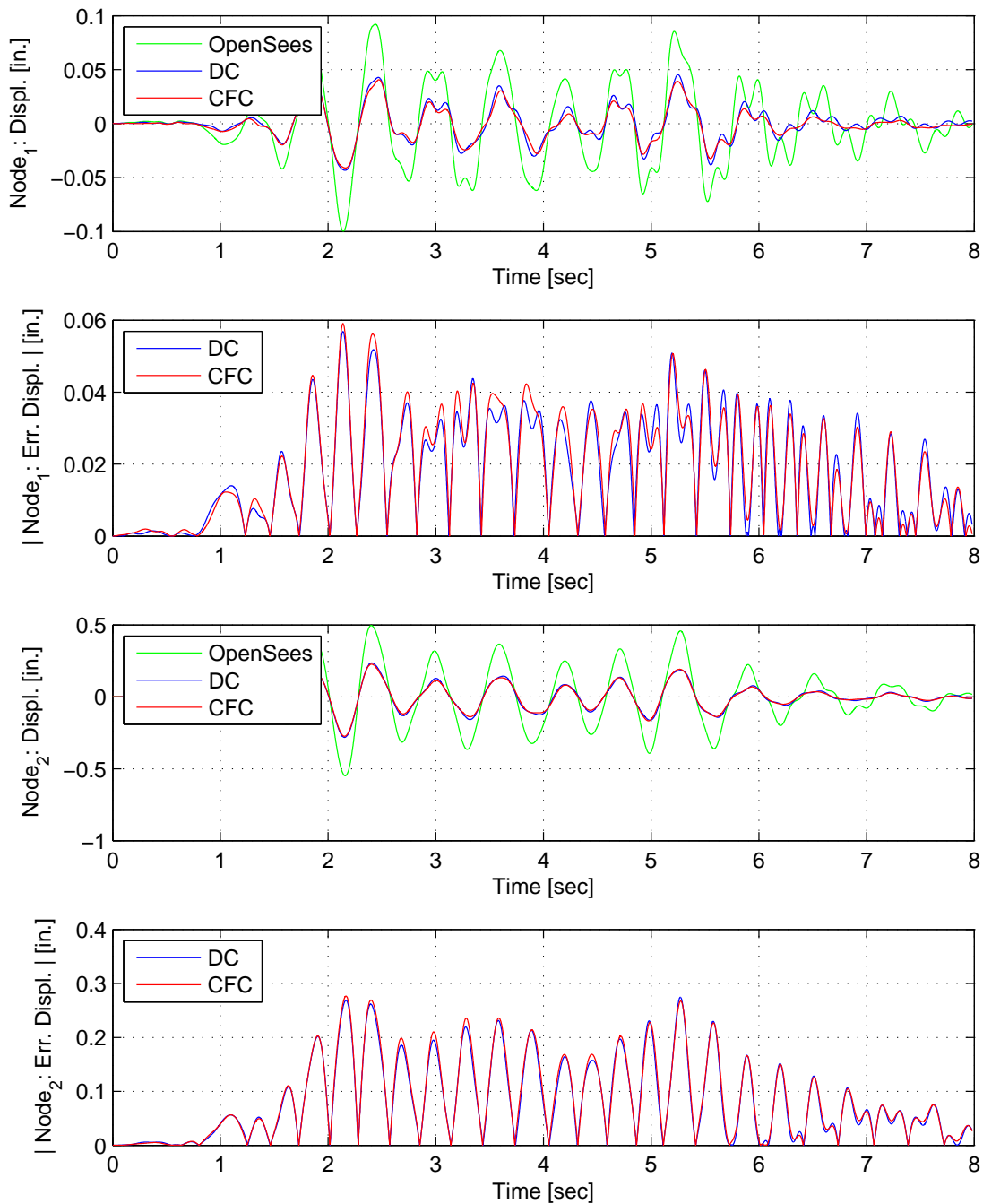


Figure C.24: Node 1 displacement-time history plot (top), Node 1 absolute displacement error (middle-top), Node 2 displacement-time history plot (middle-bottom) and Node 2 absolute displacement error (bottom) for FC hybrid simulation. The ground motion is scaled to 10% to keep the simulation in the linear range. The 2-DOF setup uses  $\alpha$ OS with DC and CFC.



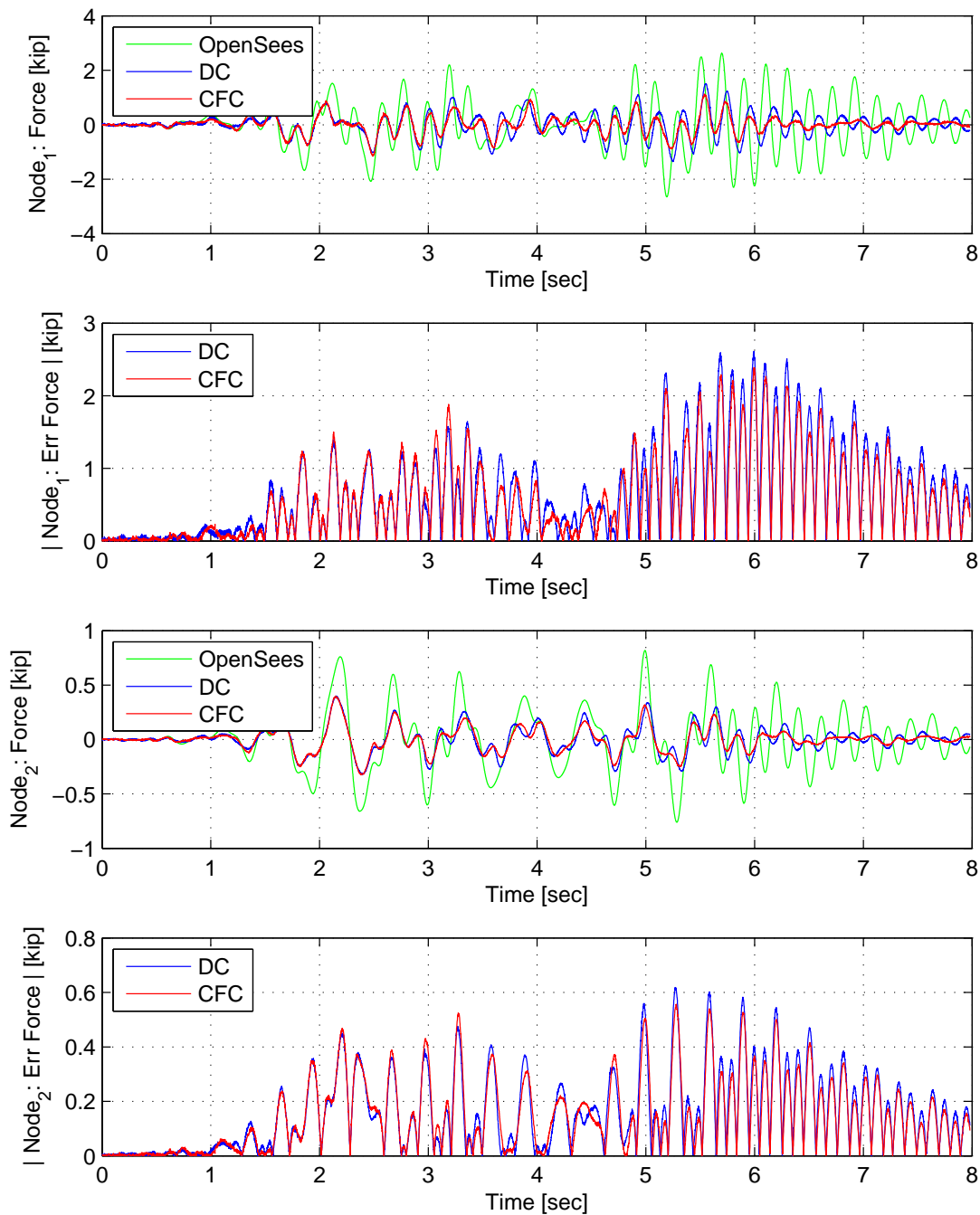


Figure C.25: Node 1 force-time history plot (top), Node 1 absolute force error (middle-top), Node 2 force-time history plot (middle-bottom) and Node 2 absolute force error (bottom) for FC hybrid simulation. The ground motion is scaled to 10% to keep the simulation in the linear range. The 2-DOF setup uses  $\alpha$ OS with DC and CFC.

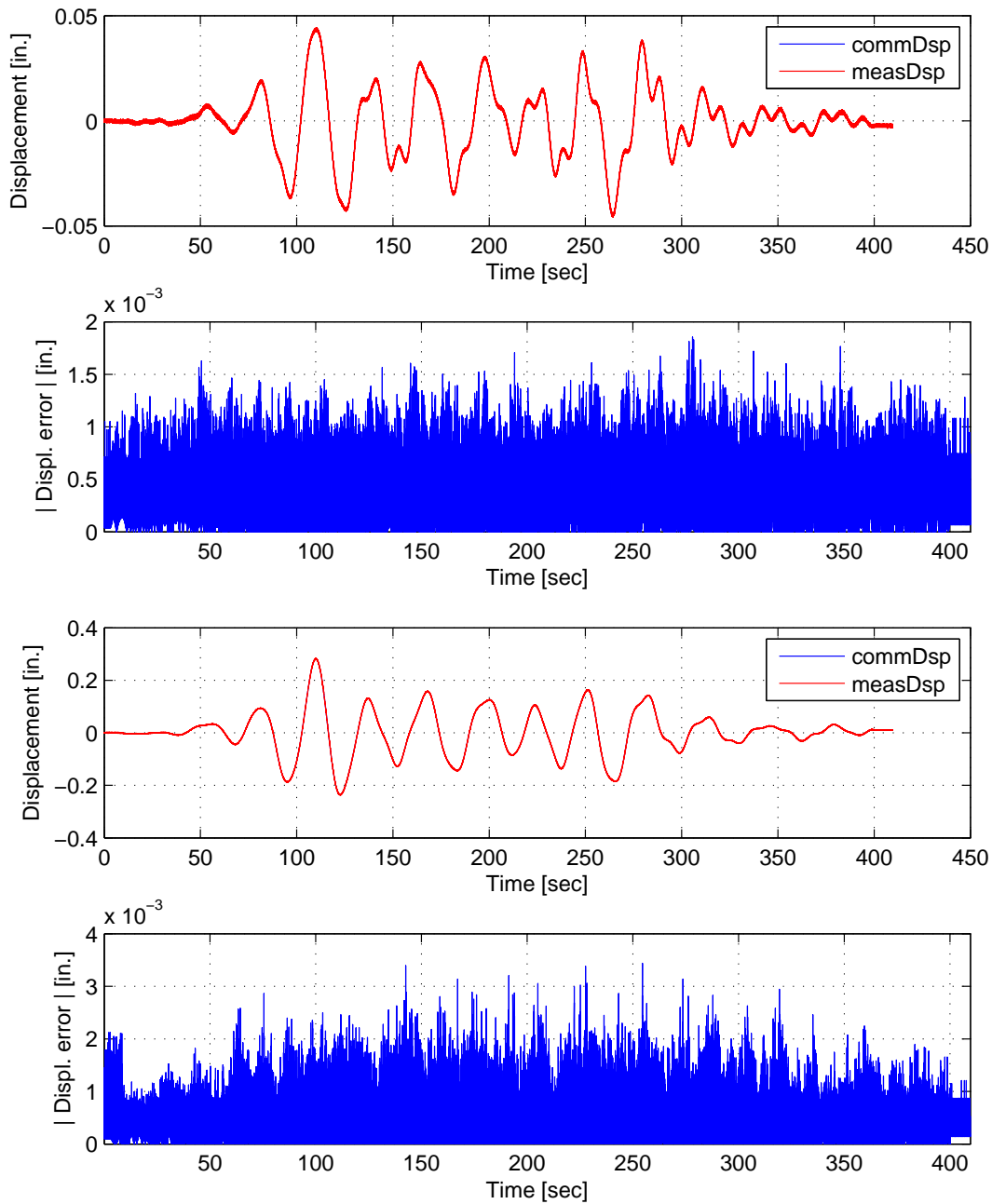


Figure C.26: Bottom actuator command and measured displacement plots from the STS controller in DC mode (top). Bottom actuator absolute error between command and measured displacements from the STS controller (middle-top). Top actuator command and measured displacement plots from the STS controller in DC mode (middle-bottom). Top actuator absolute error between command and measured displacements from the STS controller (bottom). The ground motion is scaled to 10% to keep the simulation in the linear range. The 2-DOF setup uses  $\alpha$ OS.

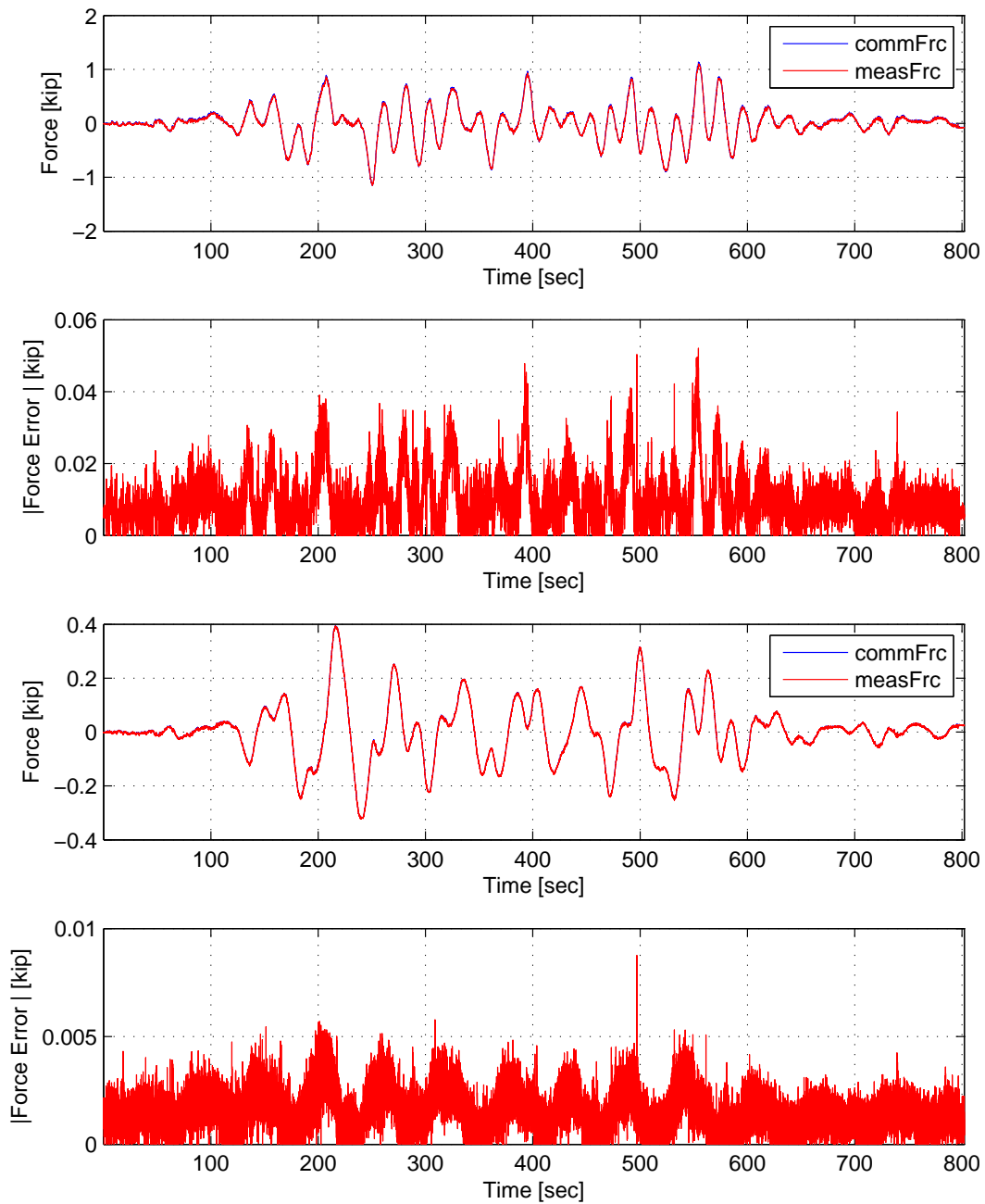


Figure C.27: Bottom actuator command and measured force plots from the STS controller in DC mode (top). Bottom actuator absolute error between command and measured force from the STS controller (middle-top). Top actuator command and measured force plots from the STS controller in DC mode (middle-bottom). Top actuator absolute error between command and measured force from the STS controller (bottom). The ground motion is scaled to 10% to keep the simulation in the linear range. The 2-DOF setup uses  $\alpha OS$ .

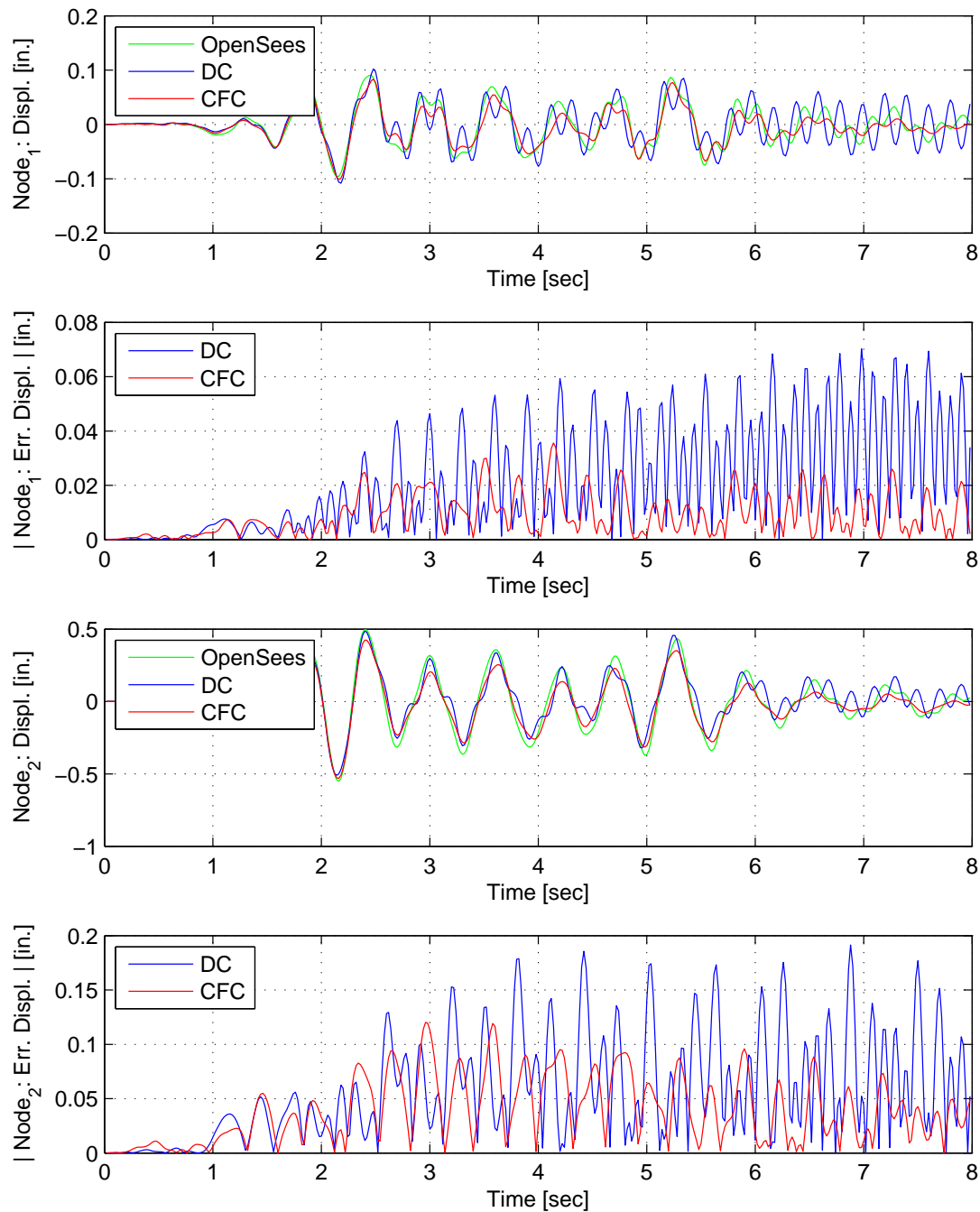


Figure C.28: Node 1 displacement-time history plot (top), Node 1 absolute displacement error (middle-top), Node 2 displacement-time history plot (middle-bottom) and Node 2 absolute displacement error (bottom) for FC hybrid simulation. The ground motion is scaled to 10% to keep the simulation in the linear range. The 2-DOF setup uses NMF with DC and CFC.

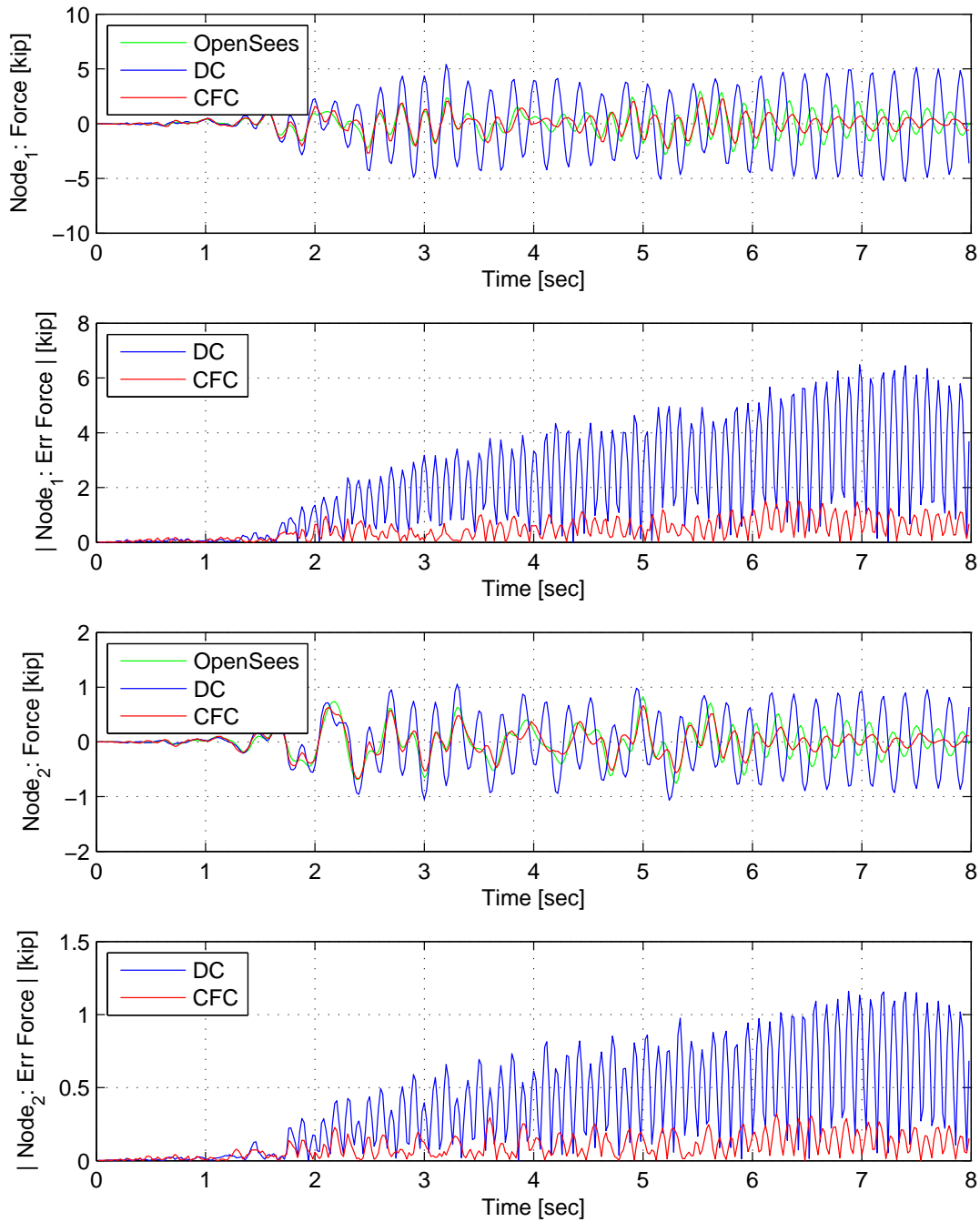


Figure C.29: Node 1 force-time history plot (top), Node 1 absolute force error (middle-top), Node 2 force-time history plot (middle-bottom) and Node 2 absolute force error (bottom) for FC hybrid simulation. The ground motion is scaled to 10% to keep the simulation in the linear range. The 2-DOF setup uses NMF with DC and CFC.

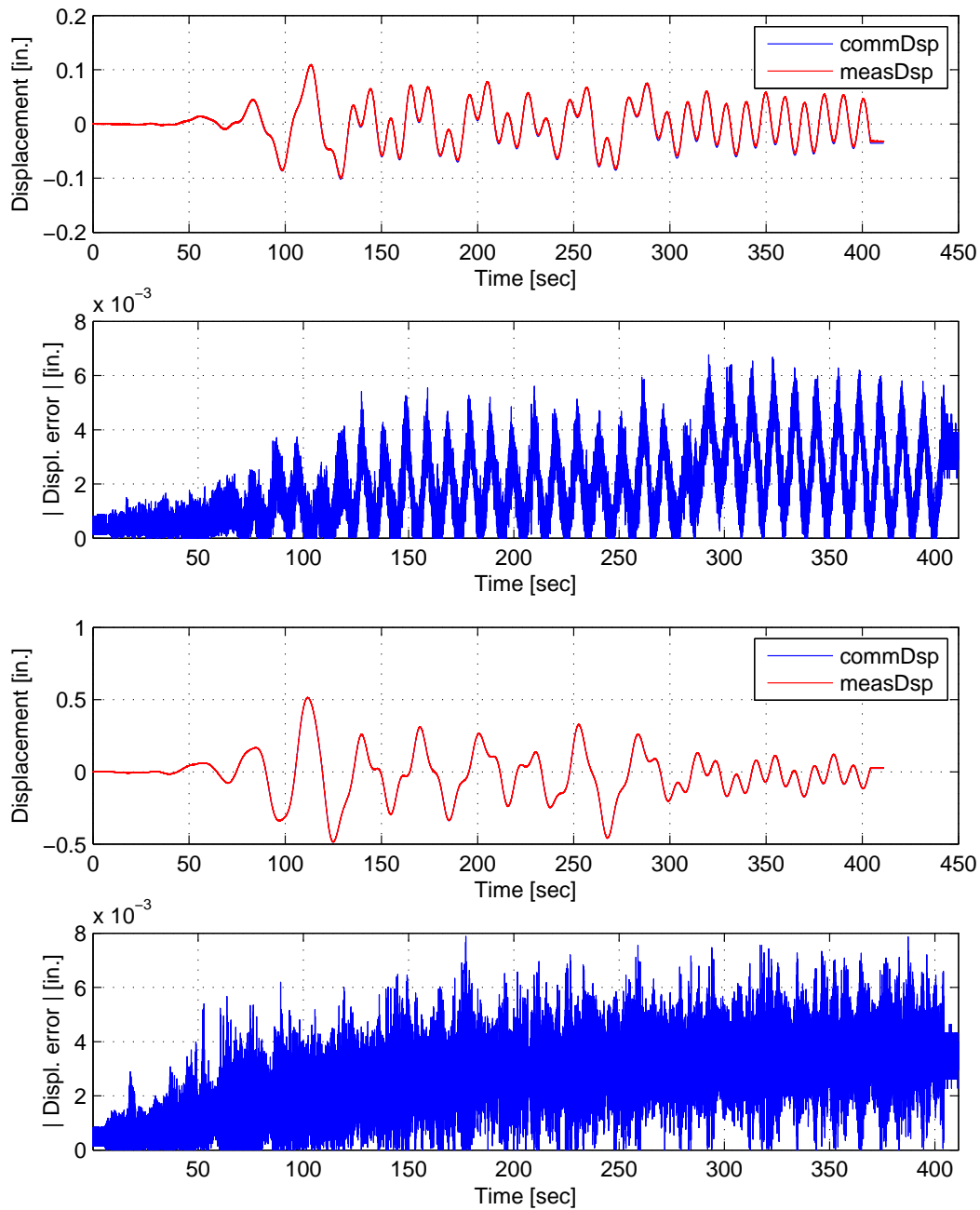


Figure C.30: Bottom actuator command and measured displacement plots from the STS controller in DC mode (top). Bottom actuator absolute error between command and measured displacements from the STS controller (middle-top). Top actuator command and measured displacement plots from the STS controller in DC mode (middle-bottom). Top actuator absolute error between command and measured displacements from the STS controller (bottom). The ground motion is scaled to 10% to keep the simulation in the linear range. The 2-DOF setup uses NMF.

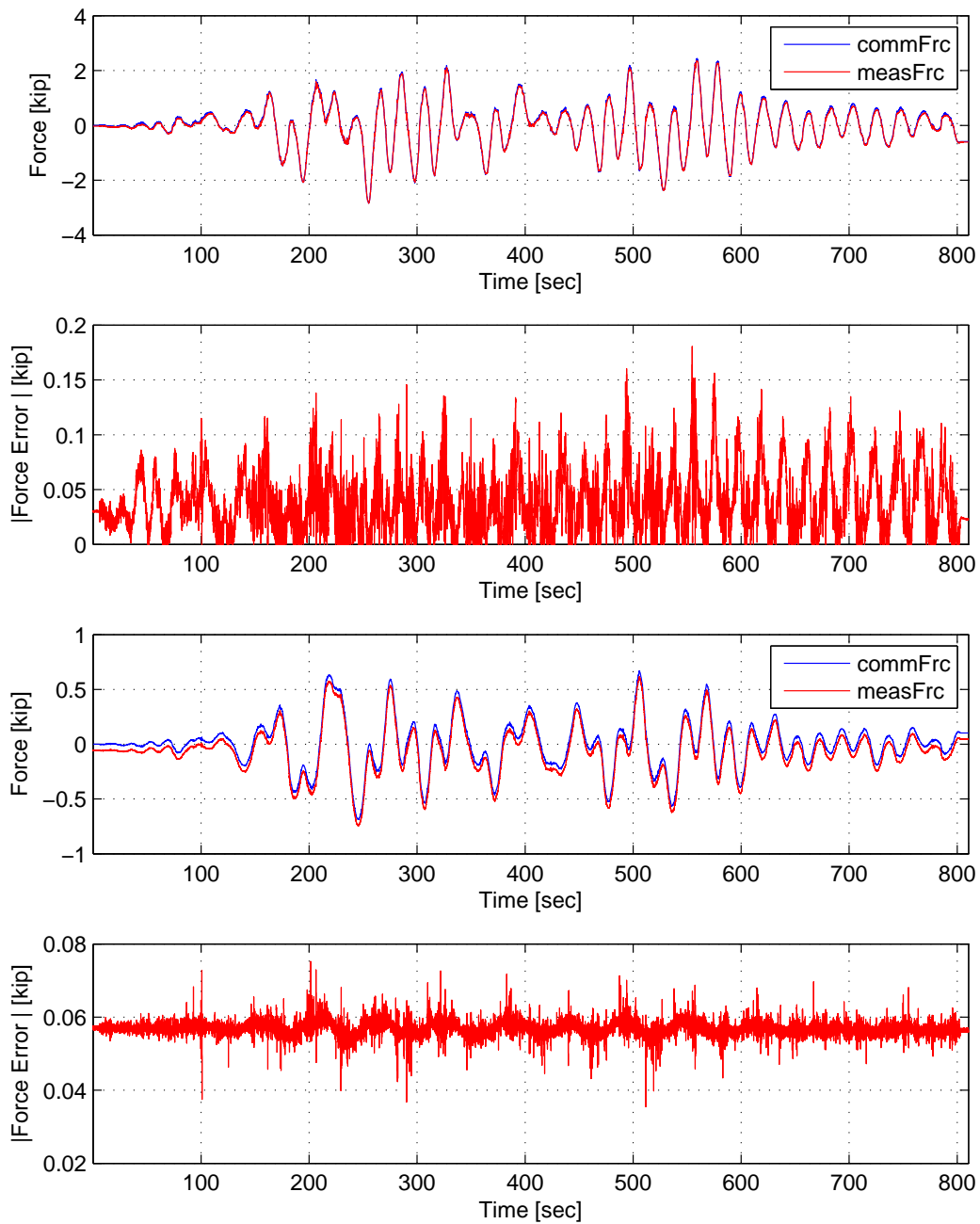


Figure C.31: Bottom actuator command and measured force plots from the STS controller in DC mode (top). Bottom actuator absolute error between command and measured force from the STS controller (middle-top). Top actuator command and measured force plots from the STS controller in DC mode (middle-bottom). Top actuator absolute error between command and measured force from the STS controller (bottom). The ground motion is scaled to 10% to keep the simulation in the linear range. The 2-DOF setup uses NMF.

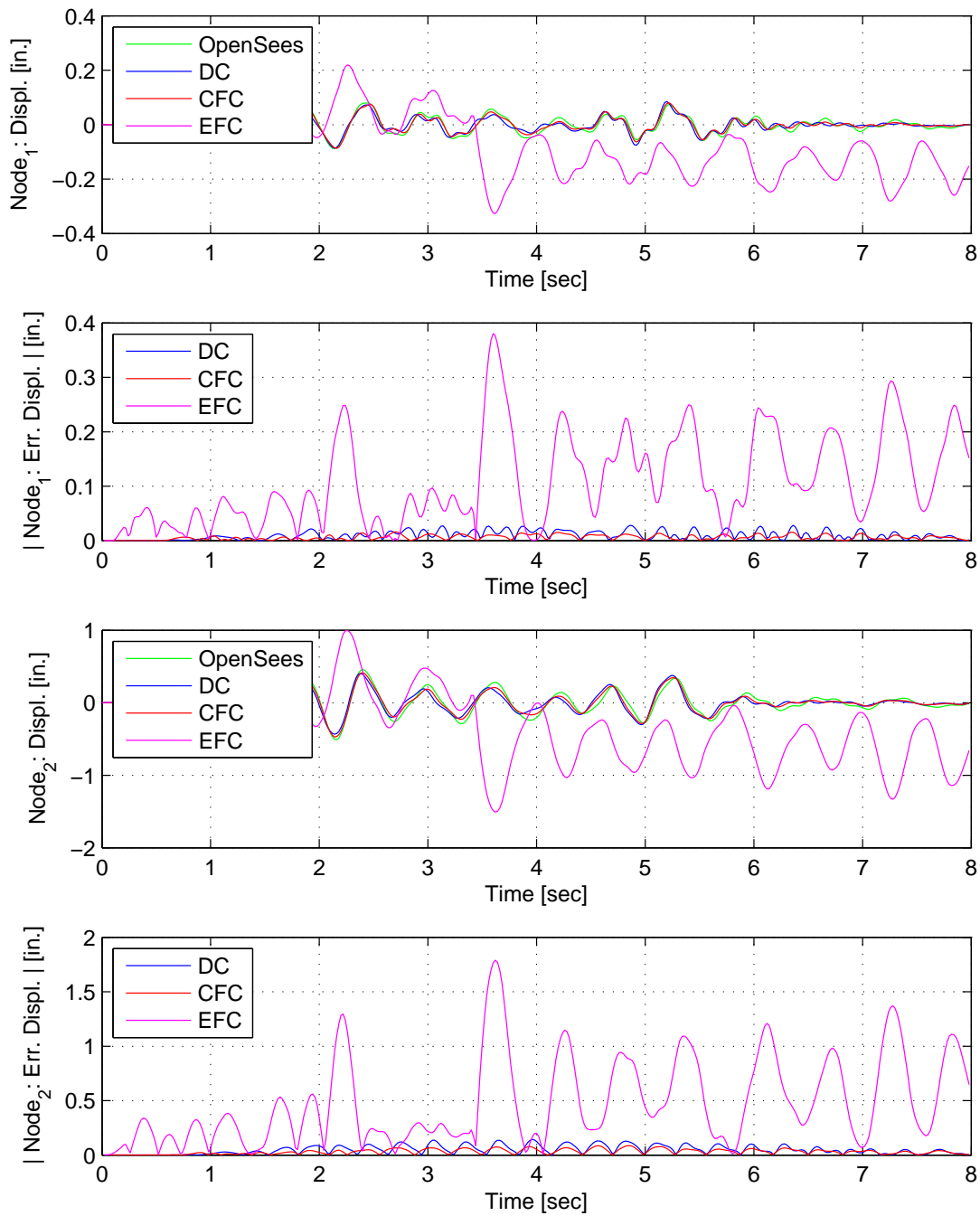


Figure C.32: Node 1 displacement-time history plot (top), Node 1 absolute displacement error (middle-top), Node 2 displacement-time history plot (middle-bottom) and Node 2 absolute displacement error (bottom) for FC hybrid simulation. The ground motion is scaled to 10% to keep the simulation in the linear range. The 2-DOF setup uses NMR with DC, CFC and EFC.



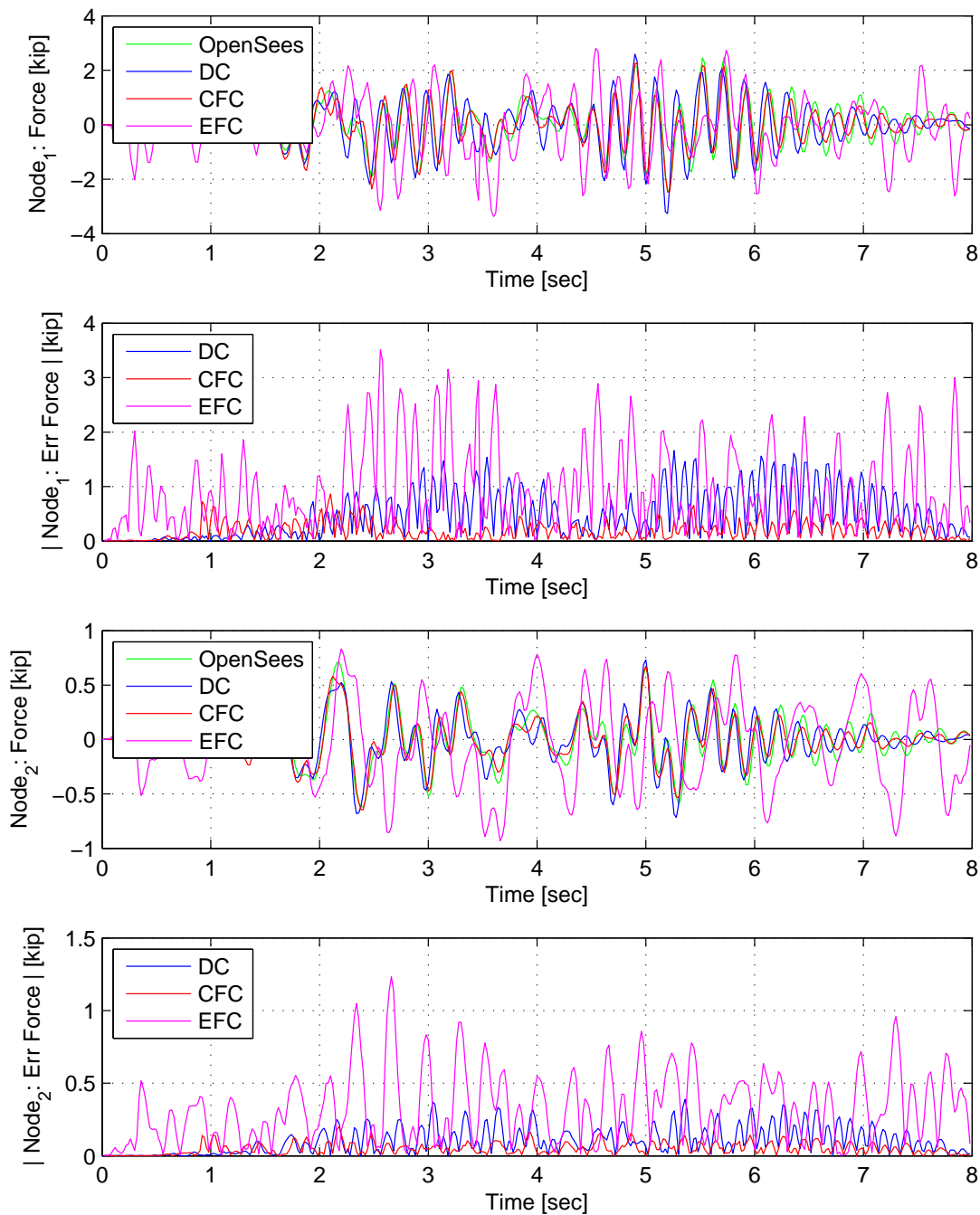


Figure C.33: Node 1 force-time history plot (top), Node 1 absolute force error (middle-top), Node 2 force-time history plot (middle-bottom) and Node 2 absolute force error (bottom) for FC hybrid simulation. The ground motion is scaled to 10% to keep the simulation in the linear range. The 2-DOF setup uses NMR with DC , CFC and EFC.

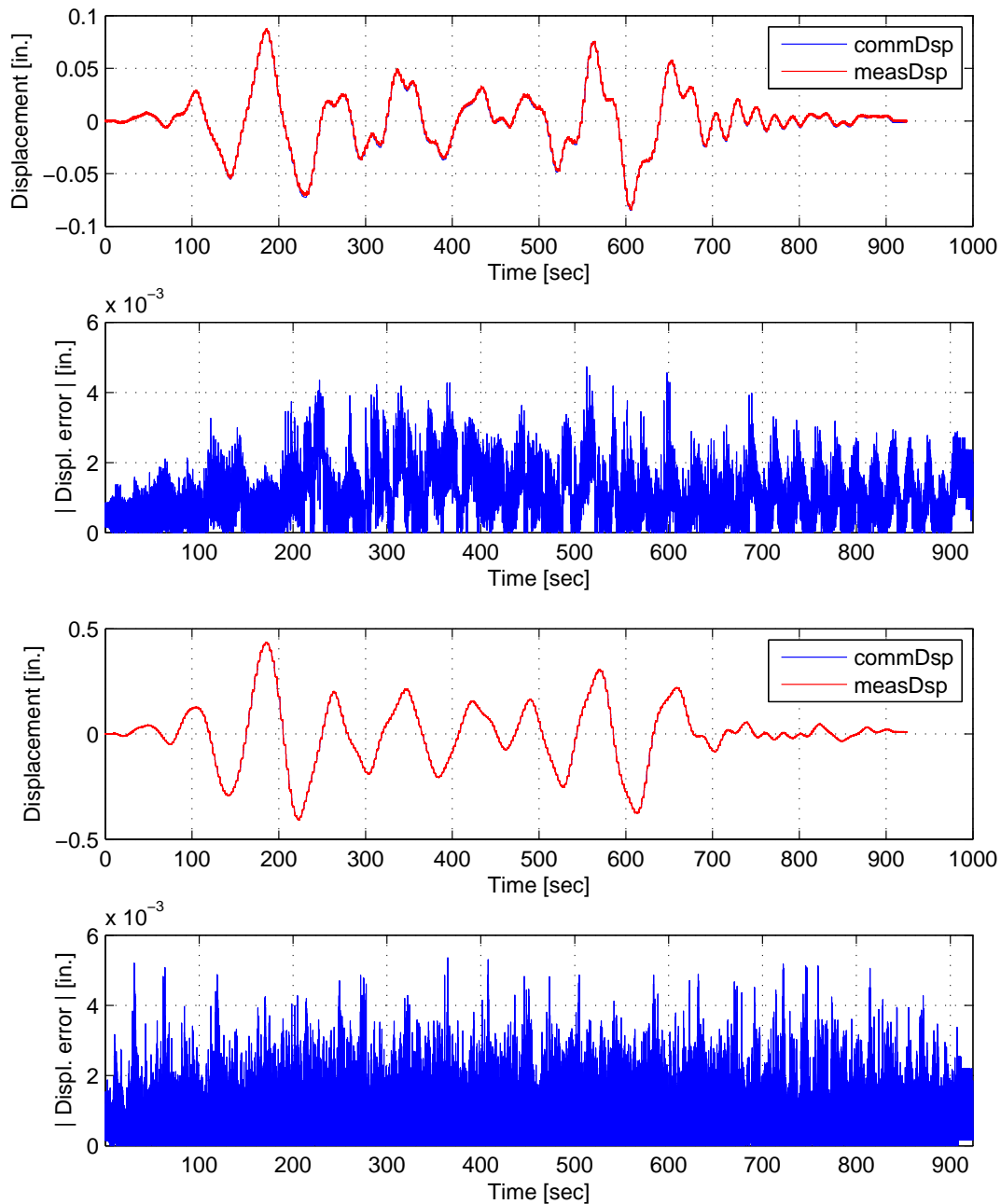


Figure C.34: Bottom actuator command and measured displacement plots from the STS controller in DC mode (top). Bottom actuator absolute error between command and measured displacements from the STS controller (middle-top). Top actuator command and measured displacement plots from the STS controller in DC mode (middle-bottom). Top actuator absolute error between command and measured displacements from the STS controller (bottom). The ground motion is scaled to 10% to keep the simulation in the linear range. The 2-DOF setup uses NMR.

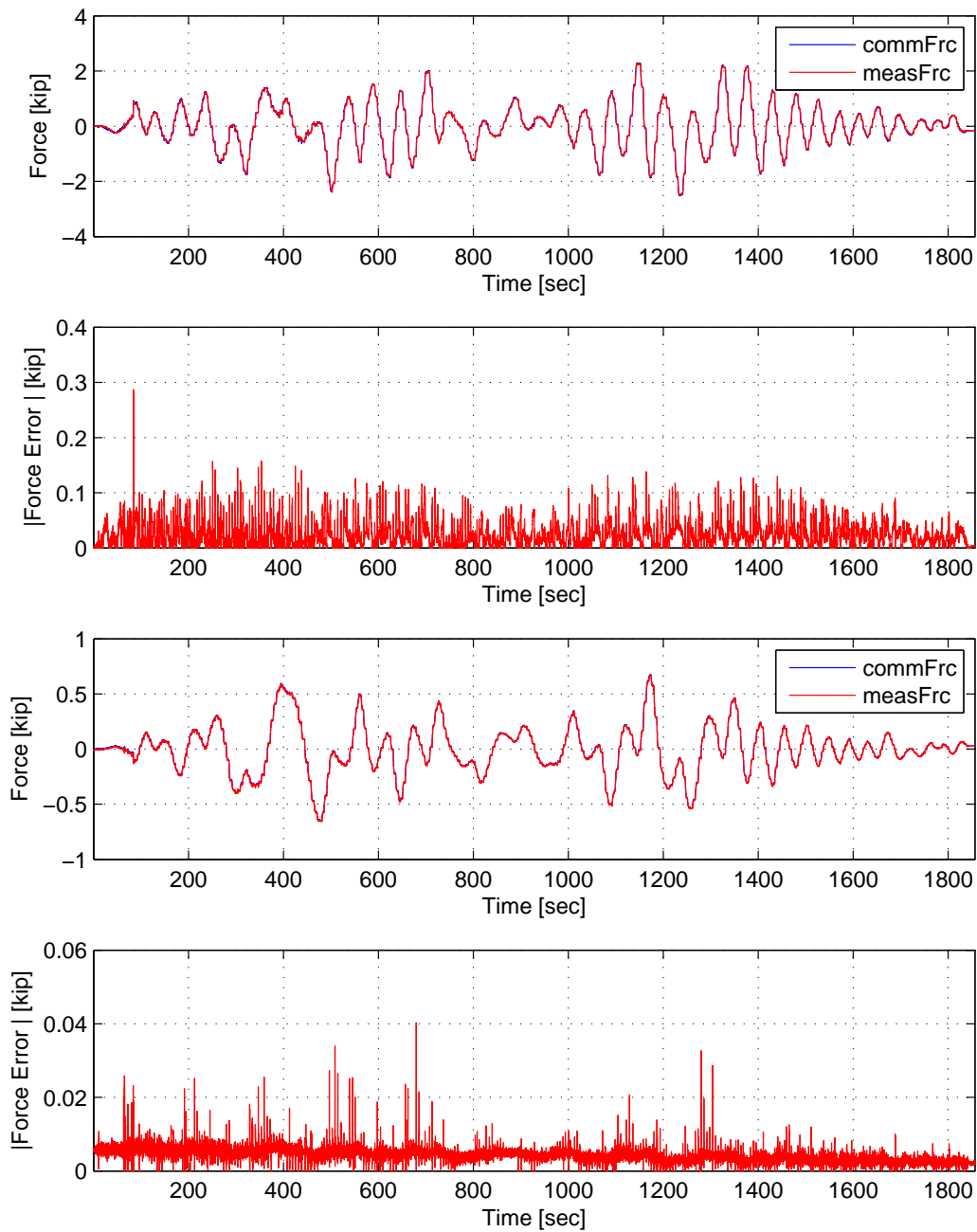


Figure C.35: Bottom actuator command and measured force plots from the STS controller in DC mode (top). Bottom actuator absolute error between command and measured force from the STS controller (middle-top). Top actuator command and measured force plots from the STS controller in DC mode (middle-bottom). Top actuator absolute error between command and measured force from the STS controller (bottom). The ground motion is scaled to 10% to keep the simulation in the linear range. The 2-DOF setup uses NMR with CFC.

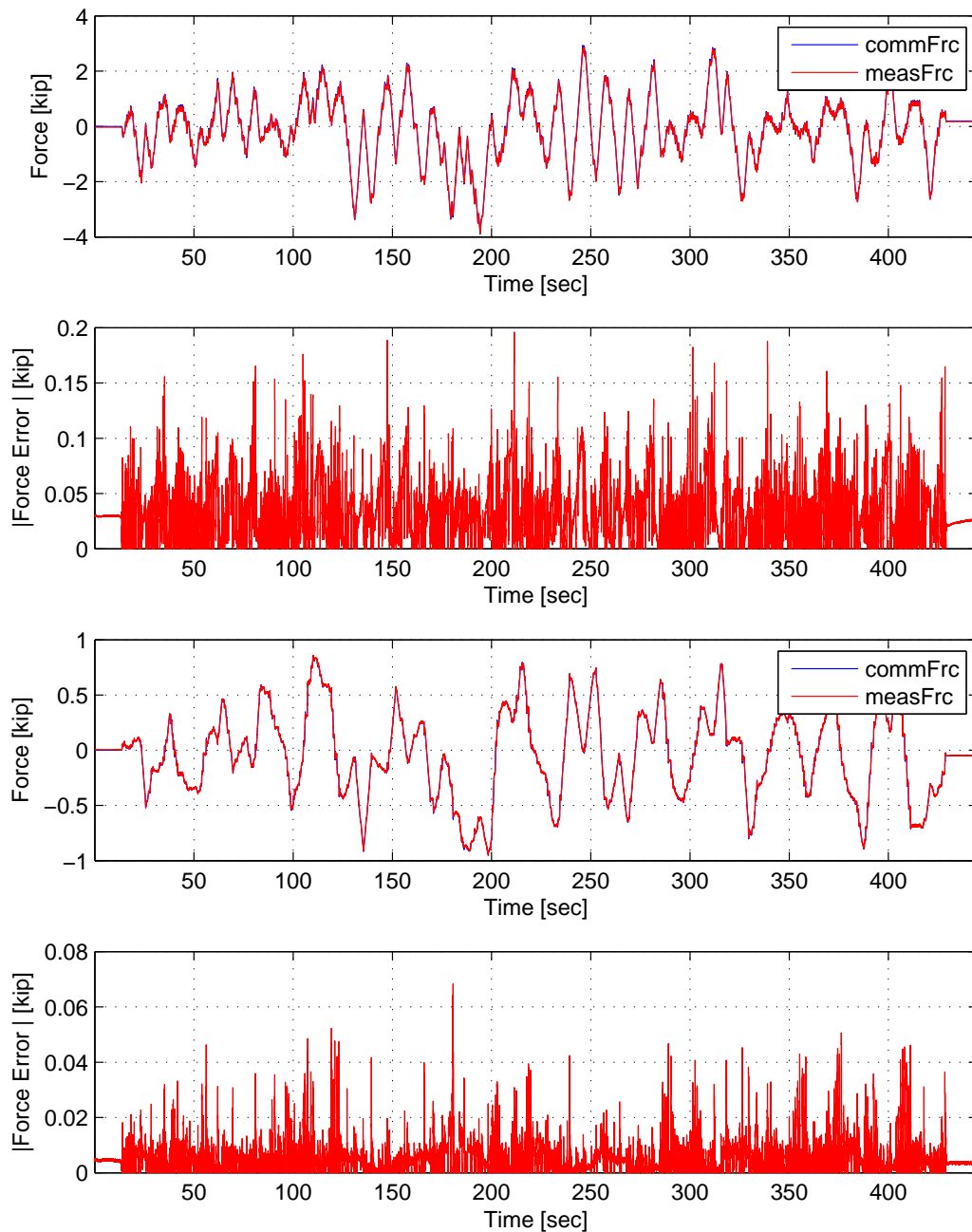


Figure C.36: Bottom actuator command and measured force plots from the STS controller in DC mode (top). Bottom actuator absolute error between command and measured force from the STS controller (middle-top). Top actuator command and measured force plots from the STS controller in DC mode (middle-bottom). Top actuator absolute error between command and measured force from the STS controller (bottom). The ground motion is scaled to 10% to keep the simulation in the linear range. The 2-DOF setup uses NME with EFC.

## **C.4 2-DOF setup: Nonlinear hybrid simulation**

The El Centro from Section 3.1.2 is scaled to 50% . The hybrid model is shown in Figure 3.21 (right). The nodal displacements and forces of the experimental element from the FEA software level are plotted vs time. These response quantities are compared against their numerical counterpart from the OpenSees analysis. The command and measured displacements from the STS controller for the DC hybrid simulation is shown for both actuators. The command and measure forces from the STS controller for CFC and EFC hybrid simulation are plotted vs time as well. Though all methods are tested, only the successful tests are shown. Some tests are not considered successful when the tolerance limit of the time integration scheme is not met or the STS control goes unstable. The results for these simulations are plotted in Figures C.37 to C.54.

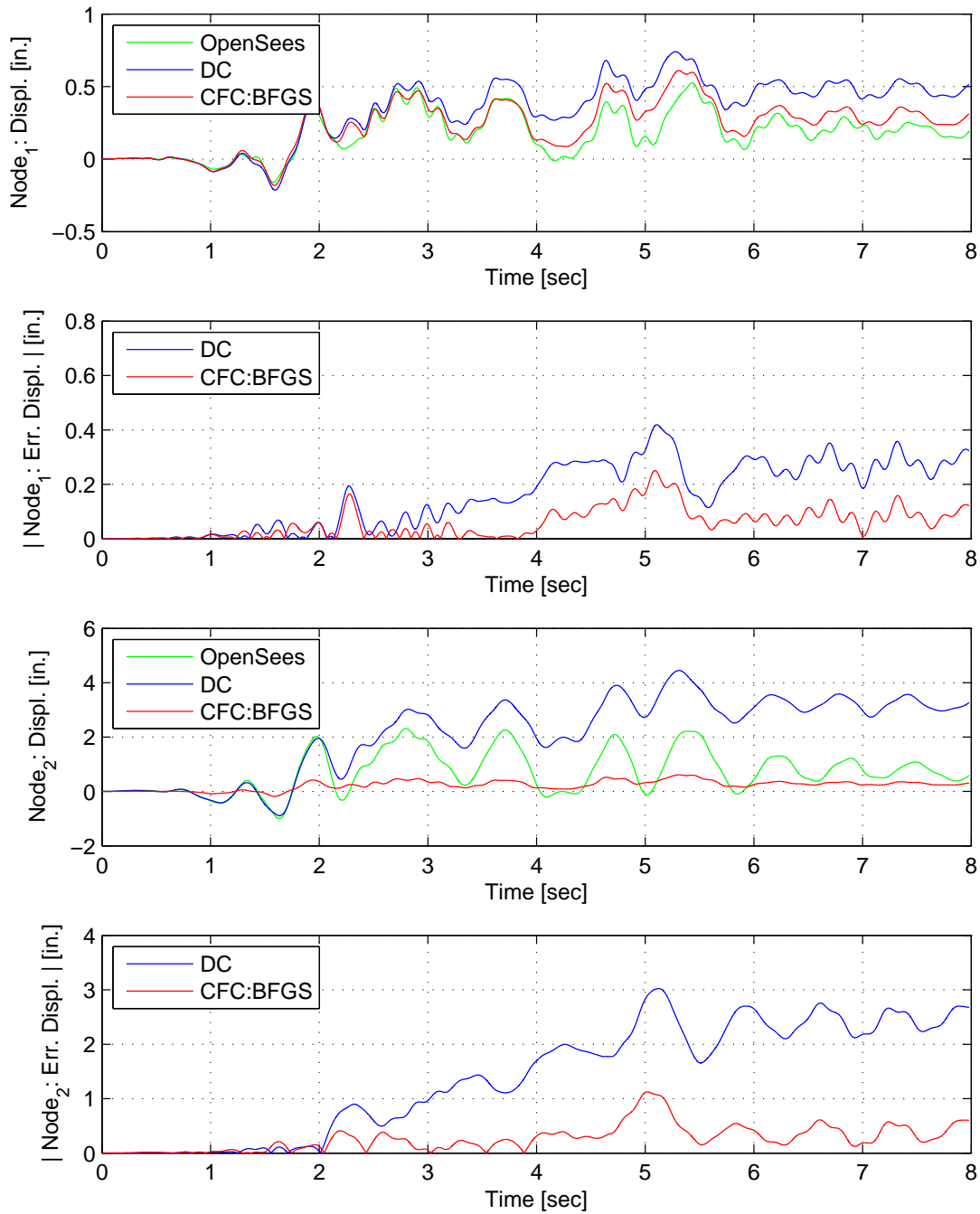


Figure C.37: Node 1 displacement-time history plot (top), Node 1 absolute displacement error (middle-top), Node 2 displacement-time history plot (middle-bottom) and Node 2 absolute displacement error (bottom) for FC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses NME with DC and CFC:BFGS.

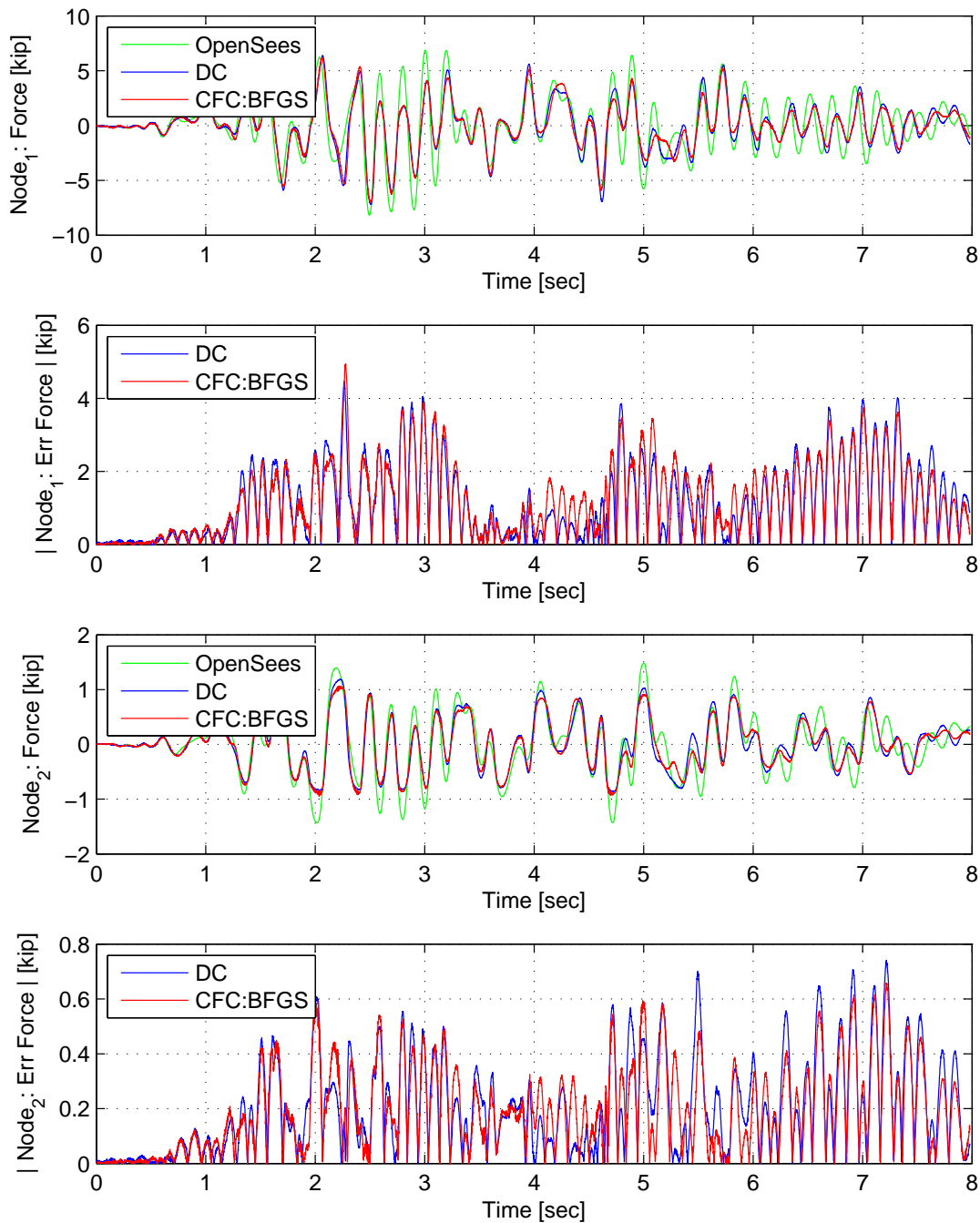


Figure C.38: Node 1 force-time history plot (top), Node 1 absolute force error (middle-top), Node 2 force-time history plot (middle-bottom) and Node 2 absolute force error (bottom) for FC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses NME with DC and CFC:BFGS.

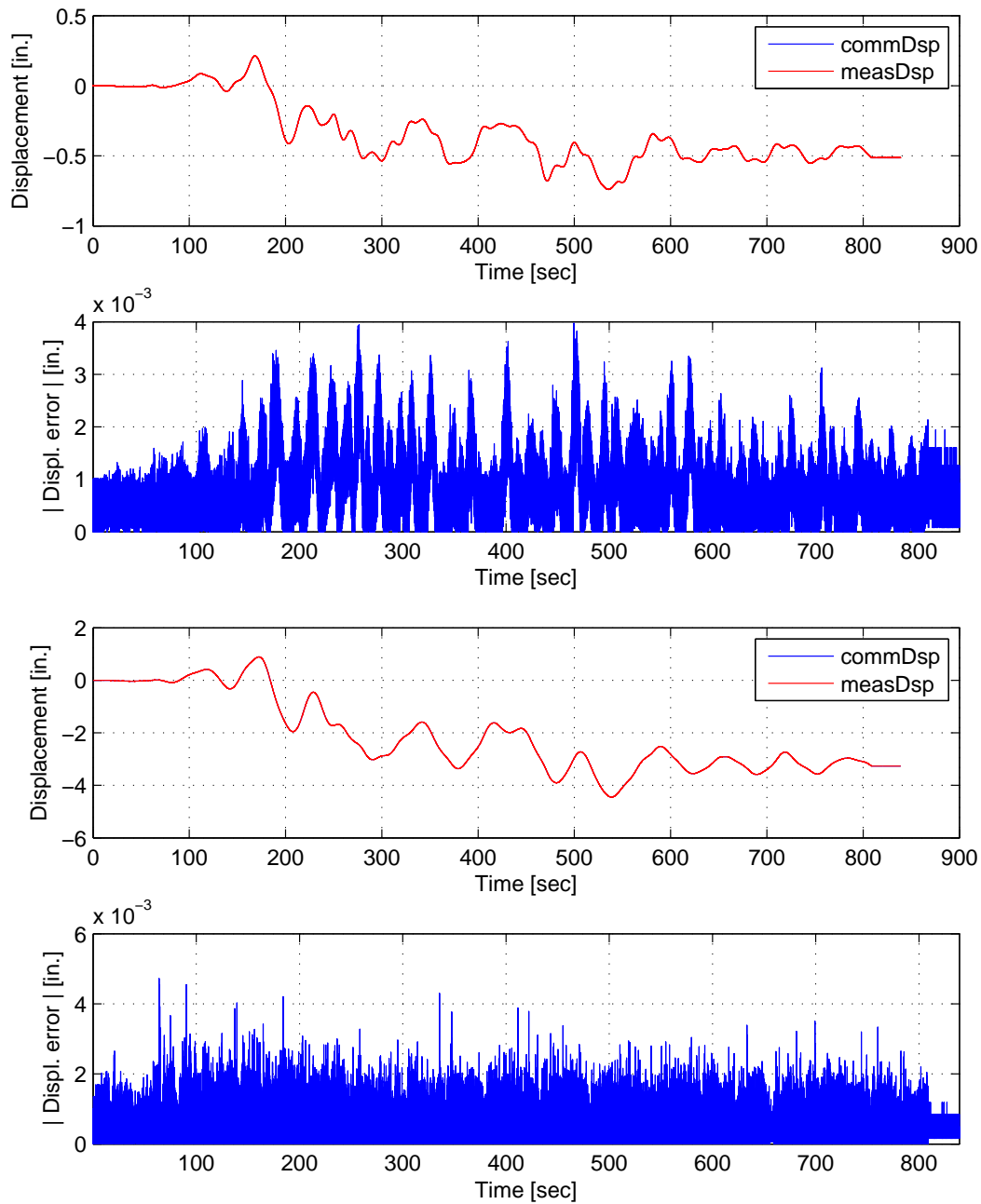


Figure C.39: Bottom actuator command and measured displacement plots from the STS controller in DC mode (top). Bottom actuator absolute error between command and measured displacements from the STS controller (middle-top). Top actuator command and measured displacement plots from the STS controller in DC mode (middle-bottom). Top actuator absolute error between command and measured displacements from the STS controller (bottom). The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses with NME with DC.



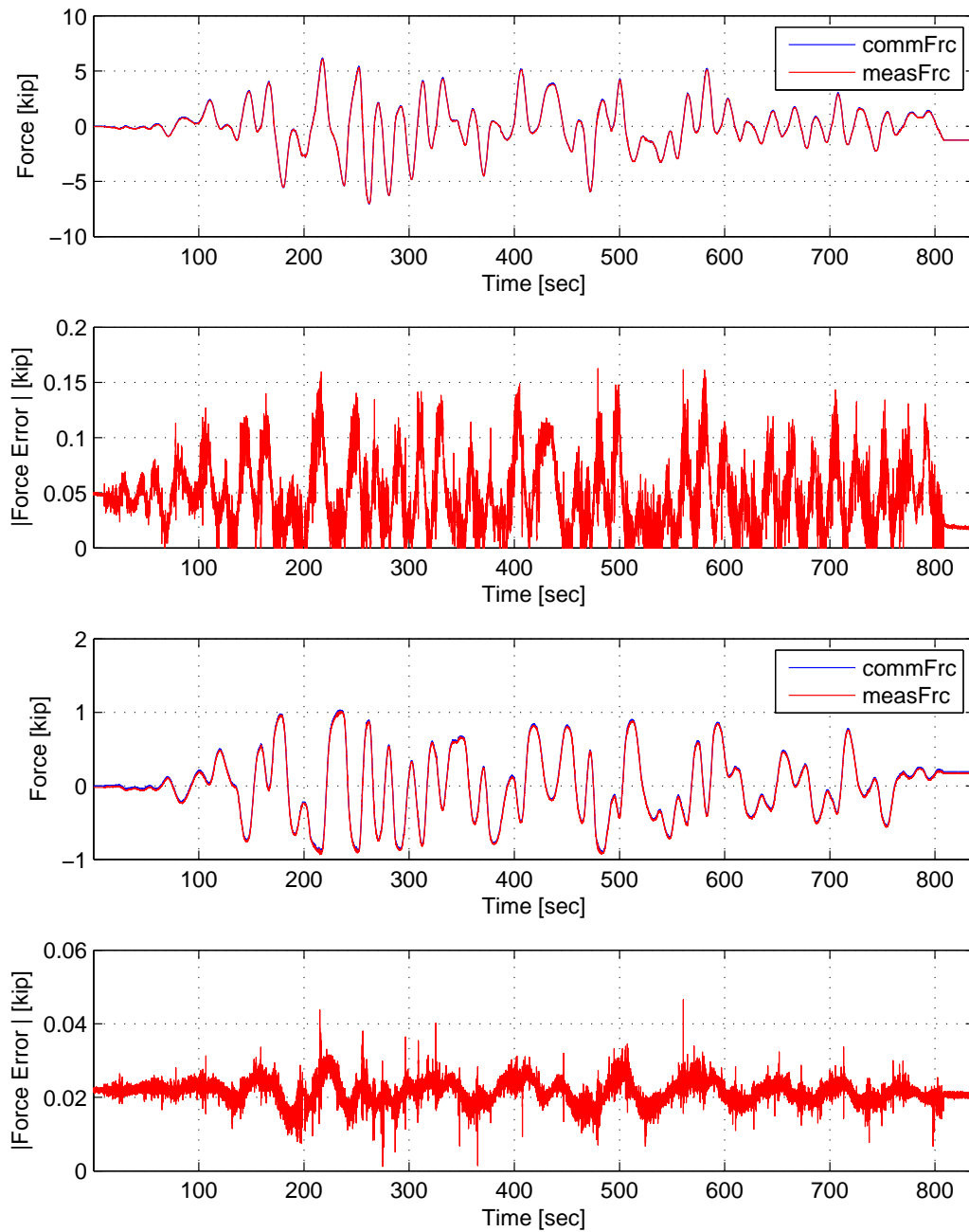


Figure C.40: Bottom actuator command and measured force plots from the STS controller in DC mode (top). Bottom actuator absolute error between command and measured force from the STS controller (middle-top). Top actuator command and measured force plots from the STS controller in DC mode (middle-bottom). Top actuator absolute error between command and measured force from the STS controller (bottom). The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses NME with CFC:BFGS.

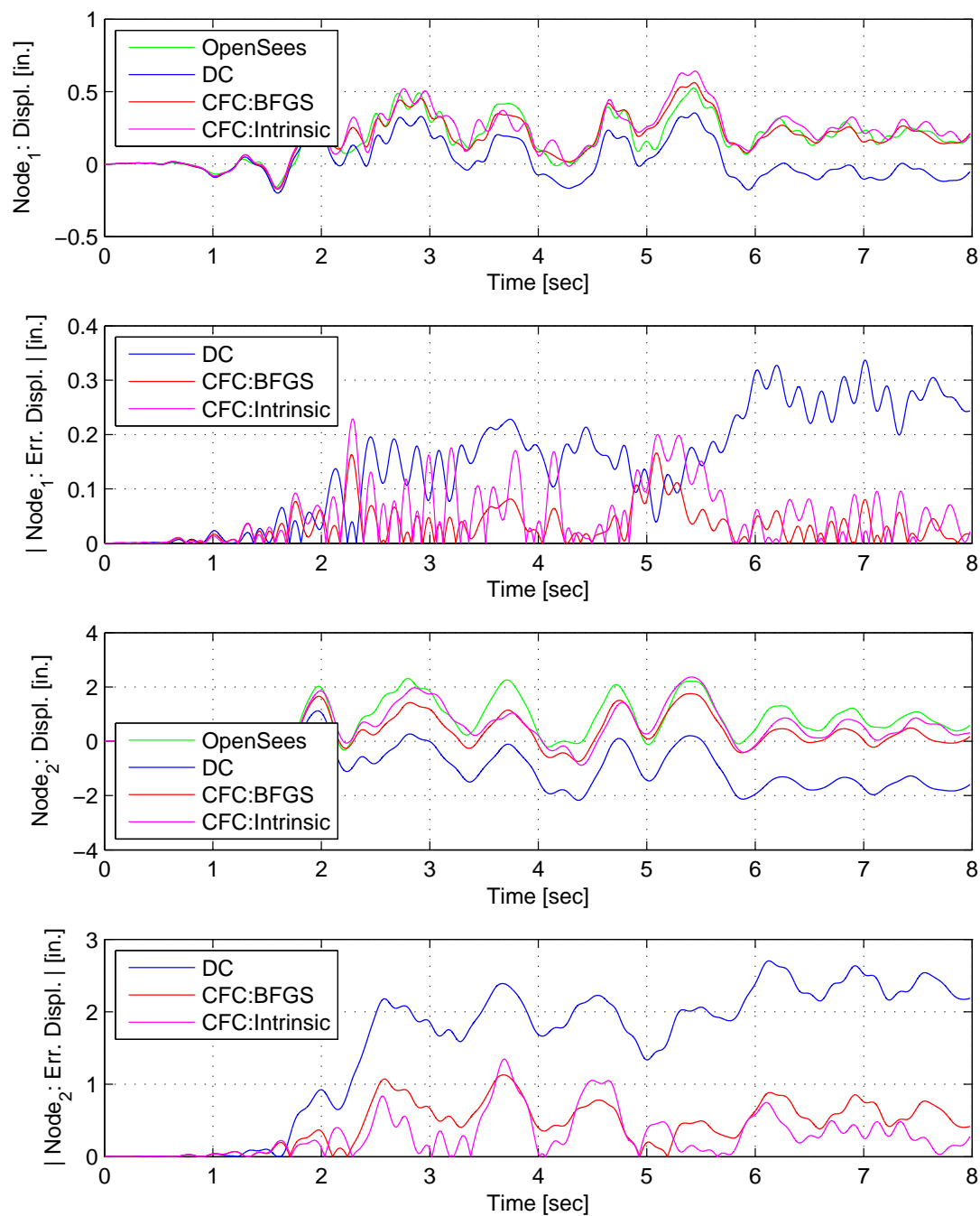


Figure C.41: Node 1 displacement-time history plot (top), Node 1 absolute displacement error (middle-top), Node 2 displacement-time history plot (middle-bottom) and Node 2 absolute displacement error (bottom) for FC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses  $\alpha$ OS with DC, CFC:BFGS and CFC:Intrinsic.

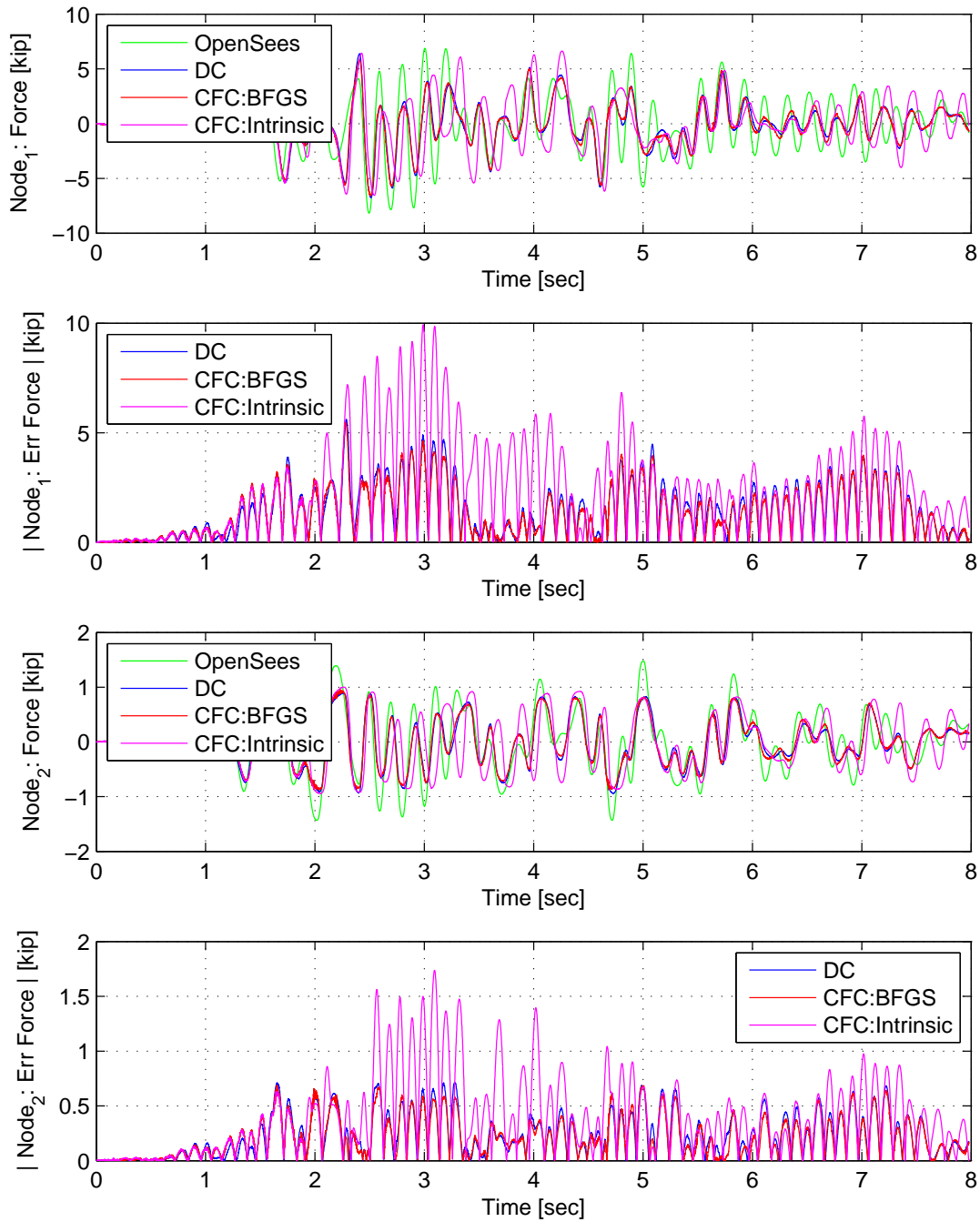


Figure C.42: Node 1 force-time history plot (top), Node 1 absolute force error (middle-top), Node 2 force-time history plot (middle-bottom) and Node 2 absolute force error (bottom) for FC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses  $\alpha$ OS with DC, CFC:BFGS and CFC:Intrinsic.

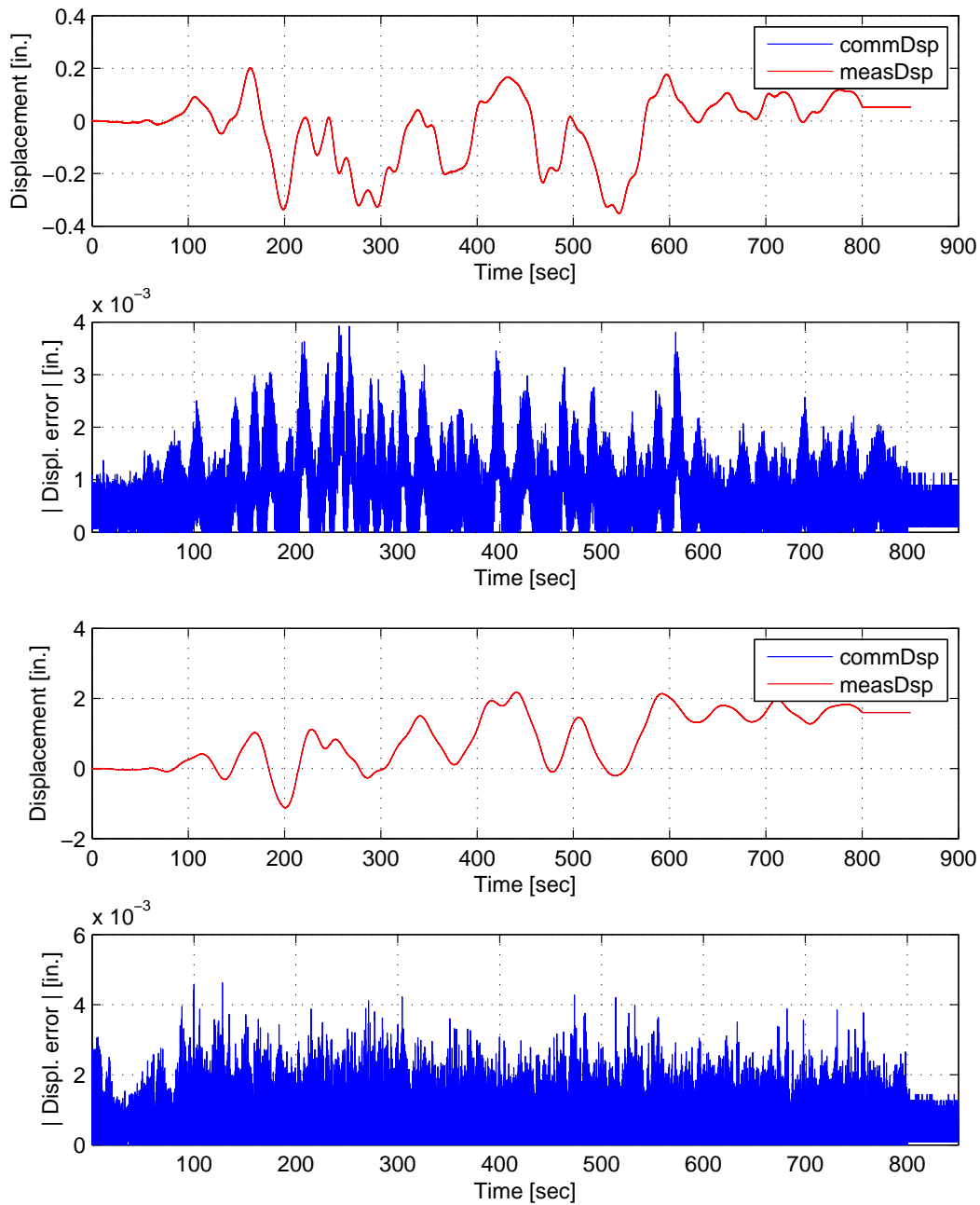


Figure C.43: Bottom actuator command and measured displacement plots from the STS controller in DC mode (top). Bottom actuator absolute error between command and measured displacements from the STS controller (middle-top). Top actuator command and measured displacement plots from the STS controller in DC mode (middle-bottom). Top actuator absolute error between command and measured displacements from the STS controller (bottom). The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses  $\alpha$ OS with DC.

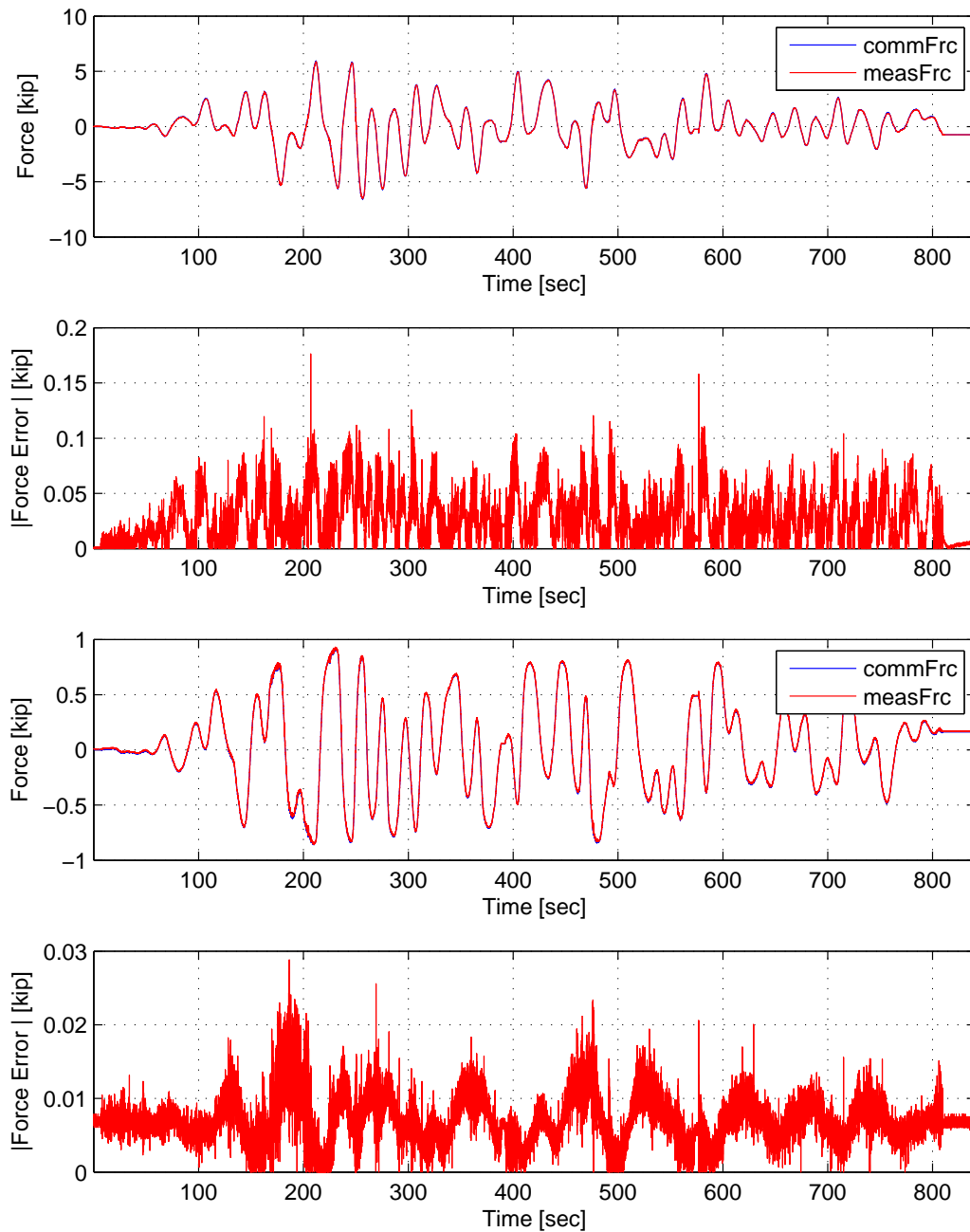


Figure C.44: Bottom actuator command and measured force plots from the STS controller in DC mode (top). Bottom actuator absolute error between command and measured force from the STS controller (middle-top). Top actuator command and measured force plots from the STS controller in DC mode (middle-bottom). Top actuator absolute error between command and measured force from the STS controller (bottom). The 2-DOF setup uses  $\alpha$ OS-CFC:BFGS.

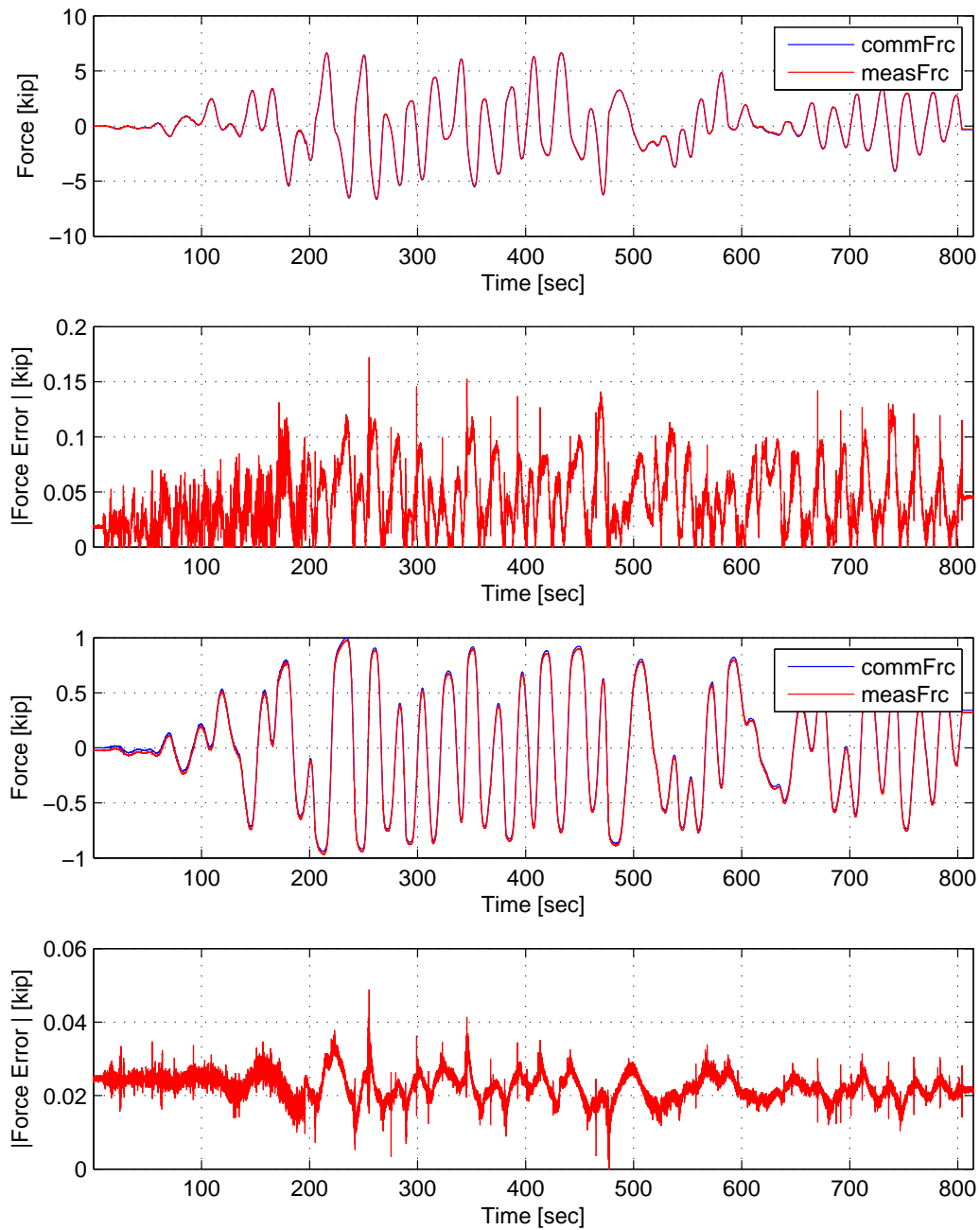


Figure C.45: Bottom actuator command and measured force plots from the STS controller in DC mode (top). Bottom actuator absolute error between command and measured force from the STS controller (middle-top). Top actuator command and measured force plots from the STS controller in DC mode (middle-bottom). Top actuator absolute error between command and measured force from the STS controller (bottom). The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses  $\alpha$ OS-CFC:Intrinsic.

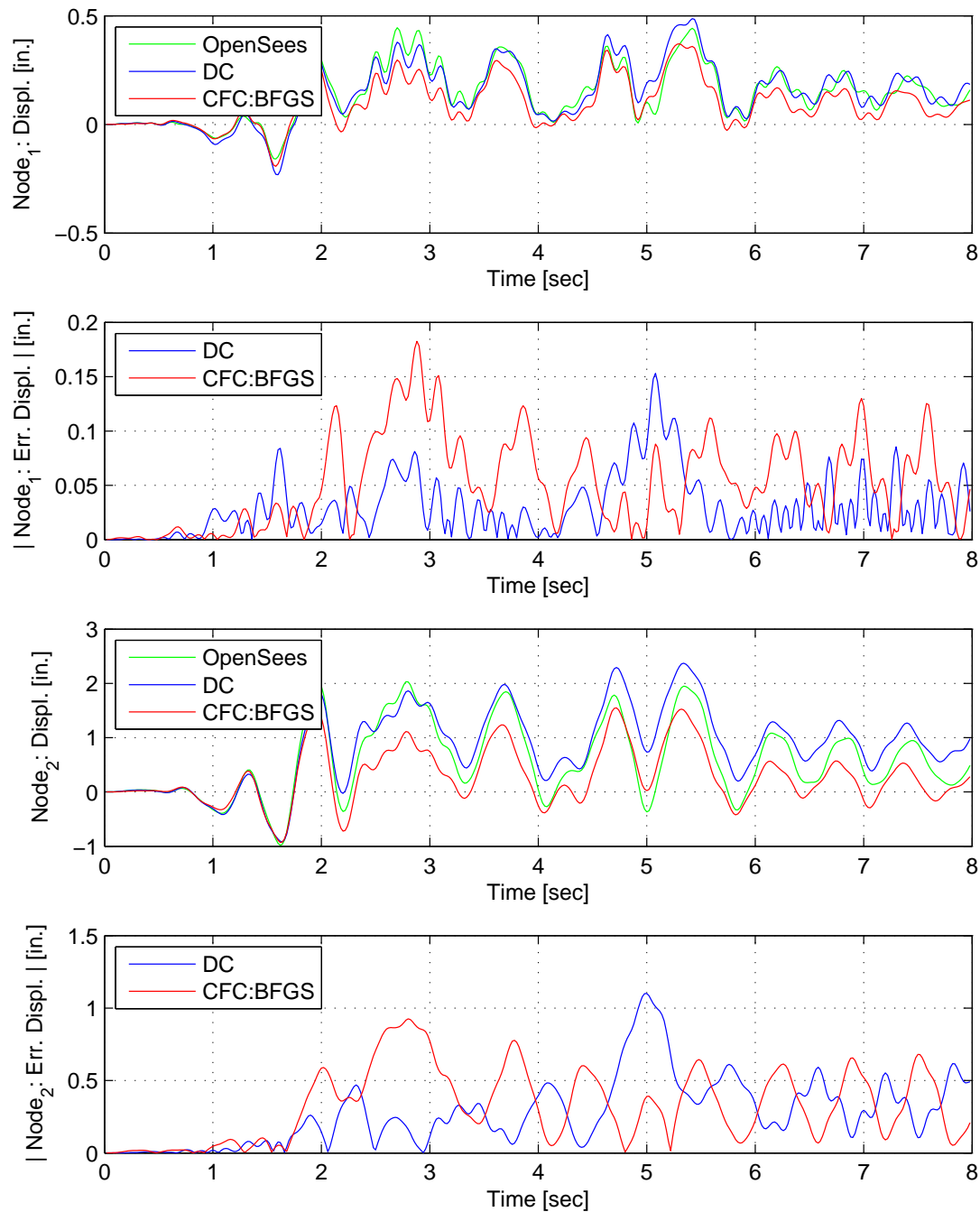


Figure C.46: Node 1 displacement-time history plot (top), Node 1 absolute displacement error (middle-top), Node 2 displacement-time history plot (middle-bottom) and Node 2 absolute displacement error (bottom) for FC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses NMF with DC and CFC:BFGS.

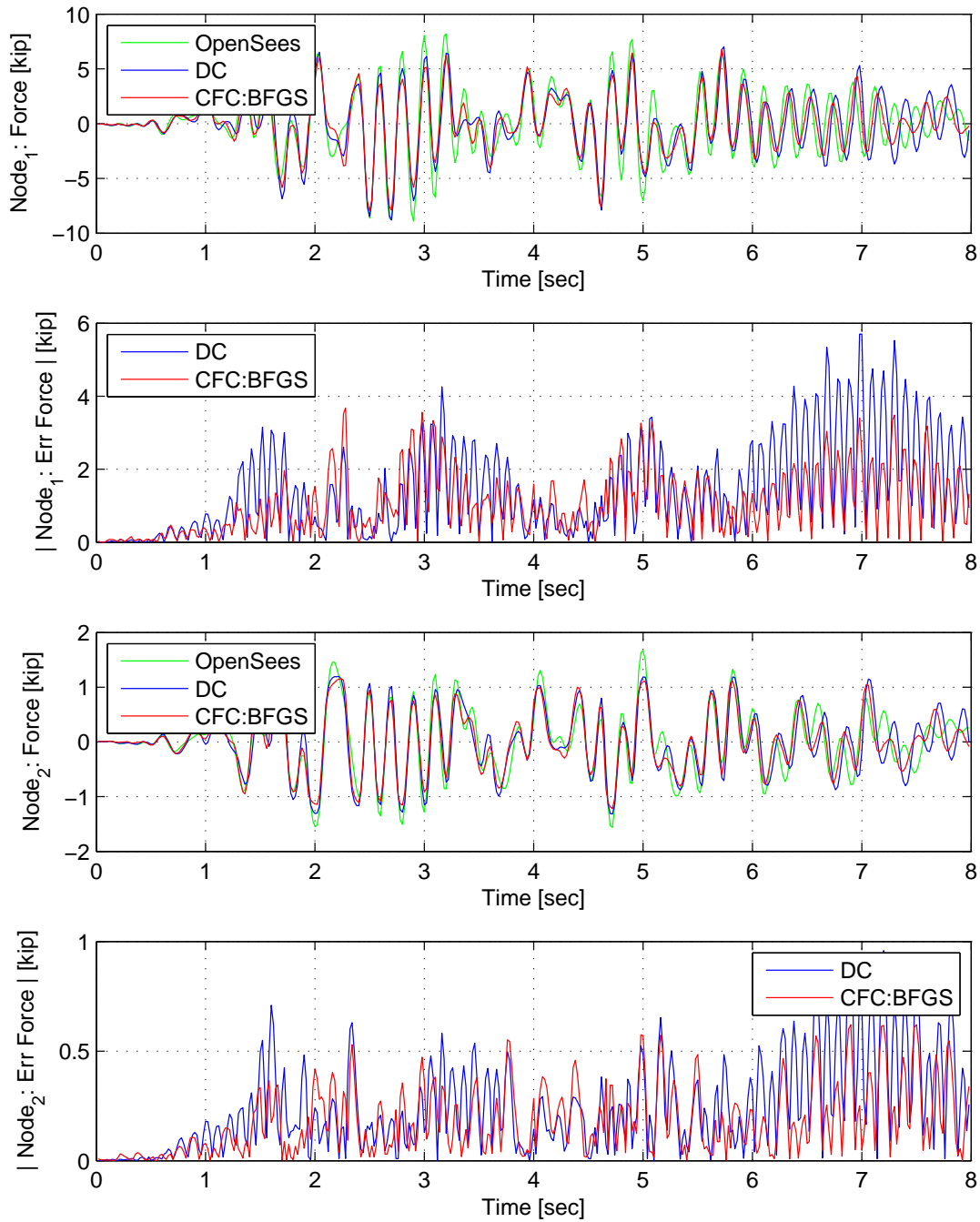


Figure C.47: Node 1 force-time history plot (top), Node 1 absolute force error (middle-top), Node 2 force-time history plot (middle-bottom) and Node 2 absolute force error (bottom) for FC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses NMF with DC and CFC:BFGS.



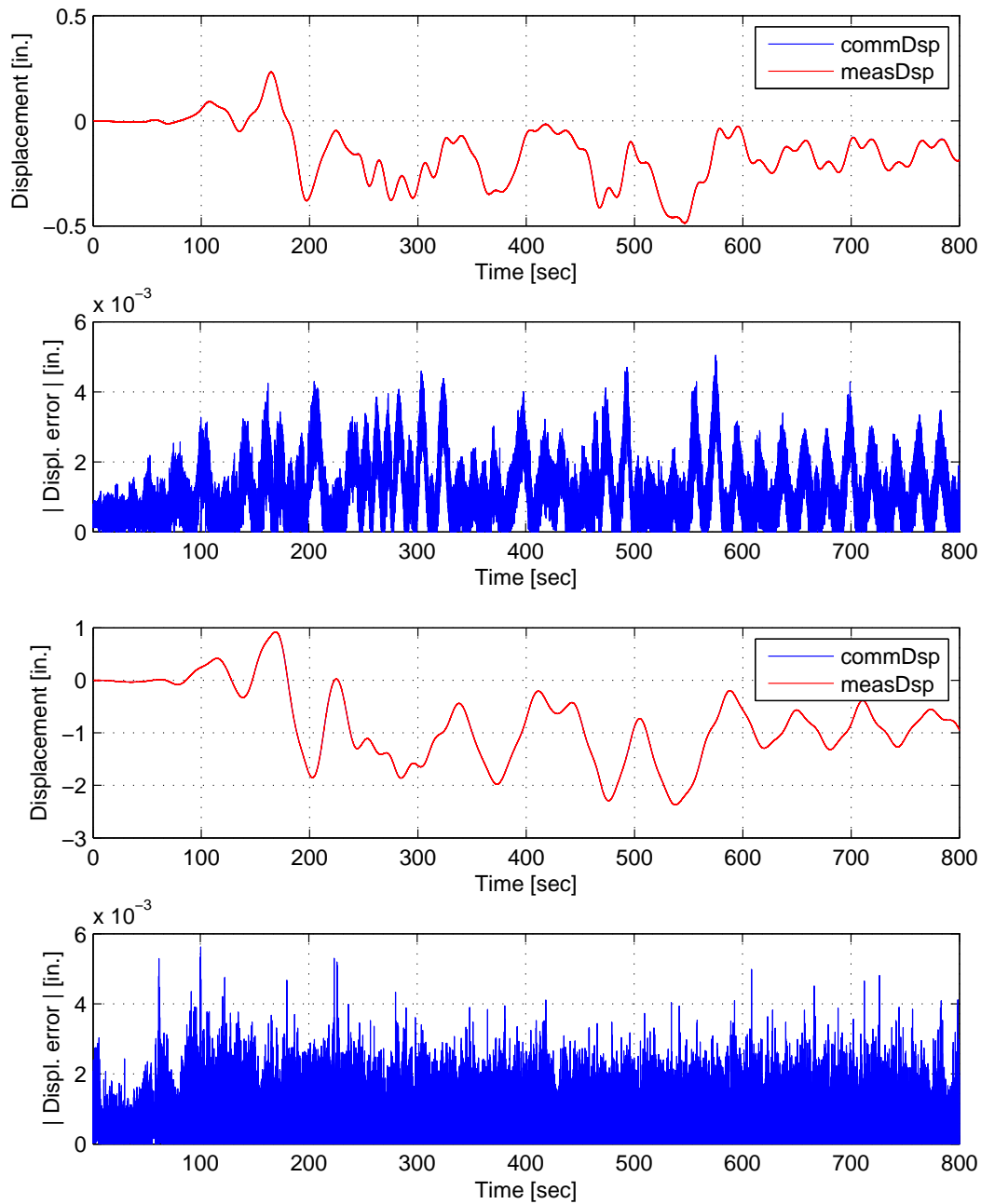


Figure C.48: Bottom actuator command and measured displacement plots from the STS controller in DC mode (top). Bottom actuator absolute error between command and measured displacements from the STS controller (middle-top). Top actuator command and measured displacement plots from the STS controller in DC mode (middle-bottom). Top actuator absolute error between command and measured displacements from the STS controller (bottom). The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses NMF with DC.

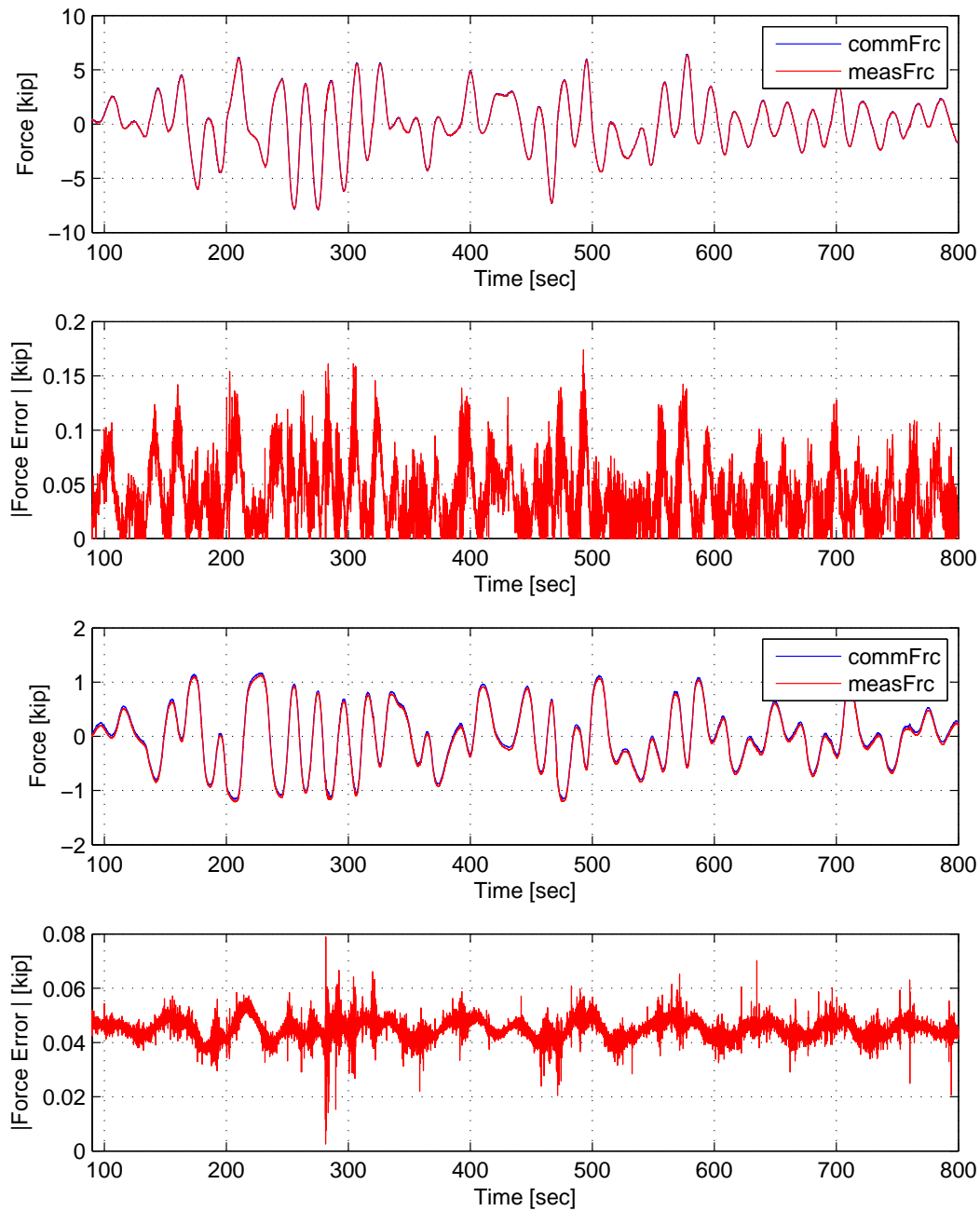


Figure C.49: Bottom actuator command and measured force plots from the STS controller in DC mode (top). Bottom actuator absolute error between command and measured force from the STS controller (middle-top). Top actuator command and measured force plots from the STS controller in DC mode (middle-bottom). Top actuator absolute error between command and measured force from the STS controller (bottom). The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses NMF with CFC:BFGS.

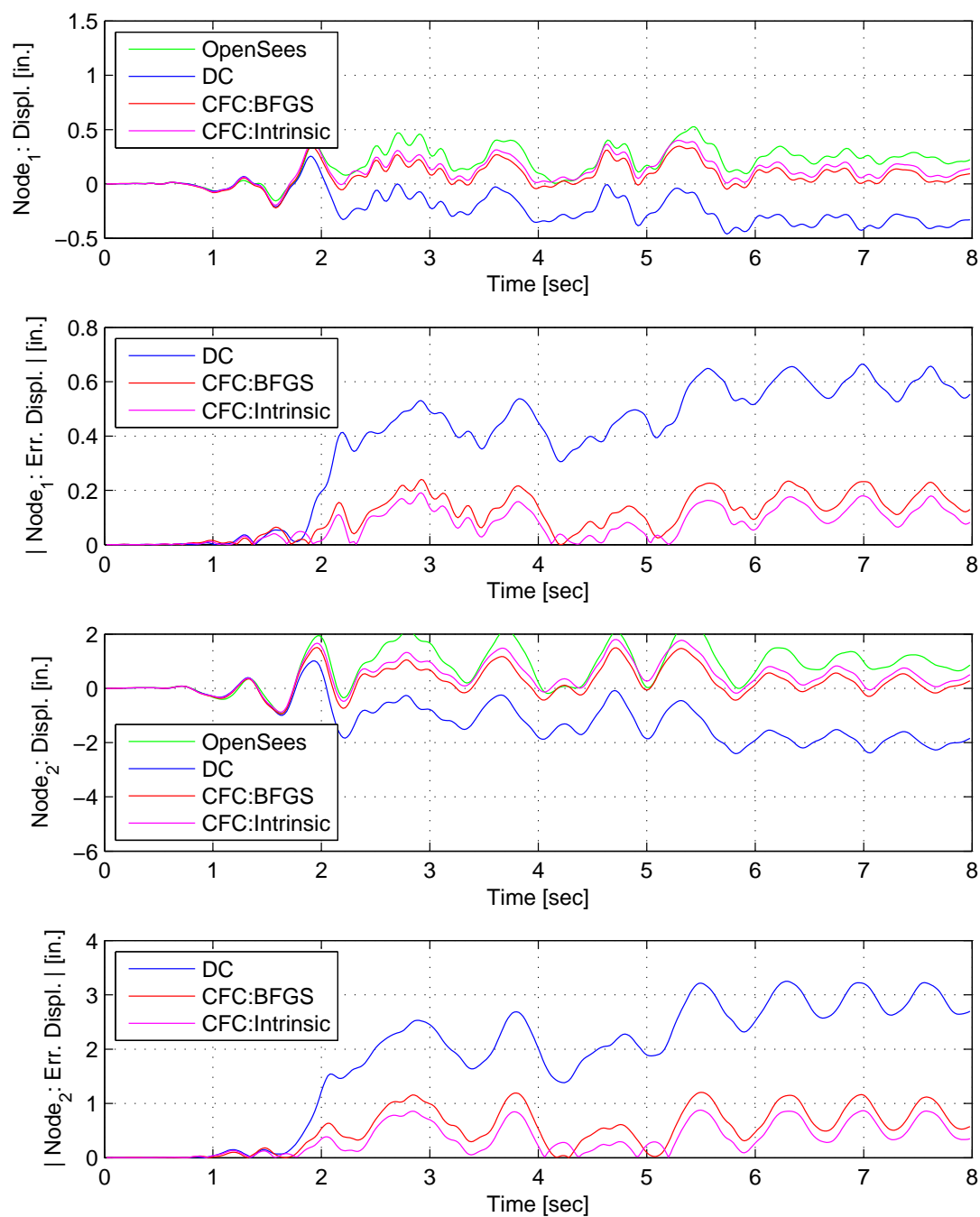


Figure C.50: Node 1 displacement-time history plot (top), Node 1 absolute displacement error (middle-top), Node 2 displacement-time history plot (middle-bottom) and Node 2 absolute displacement error (bottom) for FC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses NMR with DC, CFC:BFGS and CFC:Intrinsic.

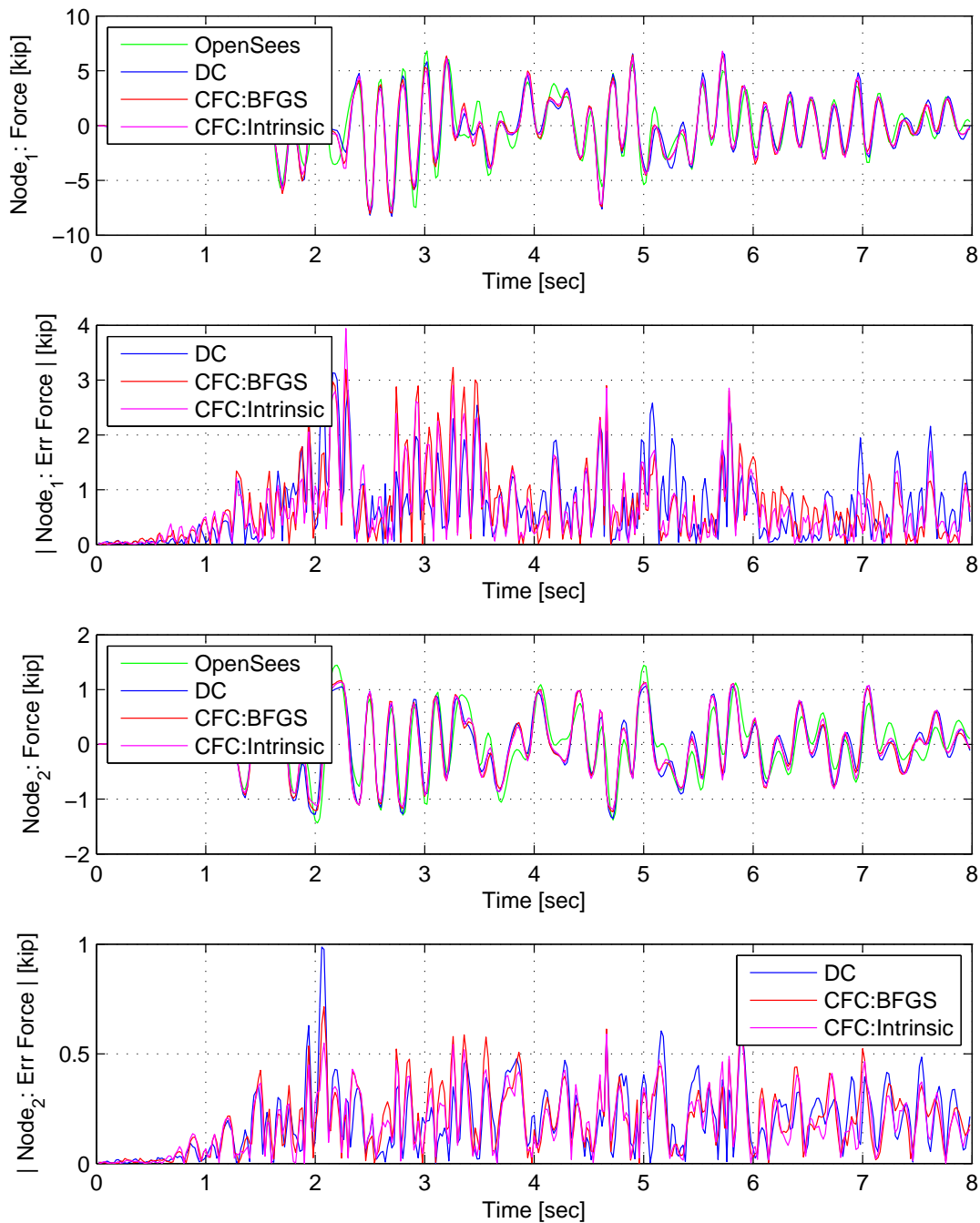


Figure C.51: Node 1 force-time history plot (top), Node 1 absolute force error (middle-top), Node 2 force-time history plot (middle-bottom) and Node 2 absolute force error (bottom) for FC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses NMR with DC , CFC:BFGS and CFC:Intrinsic.

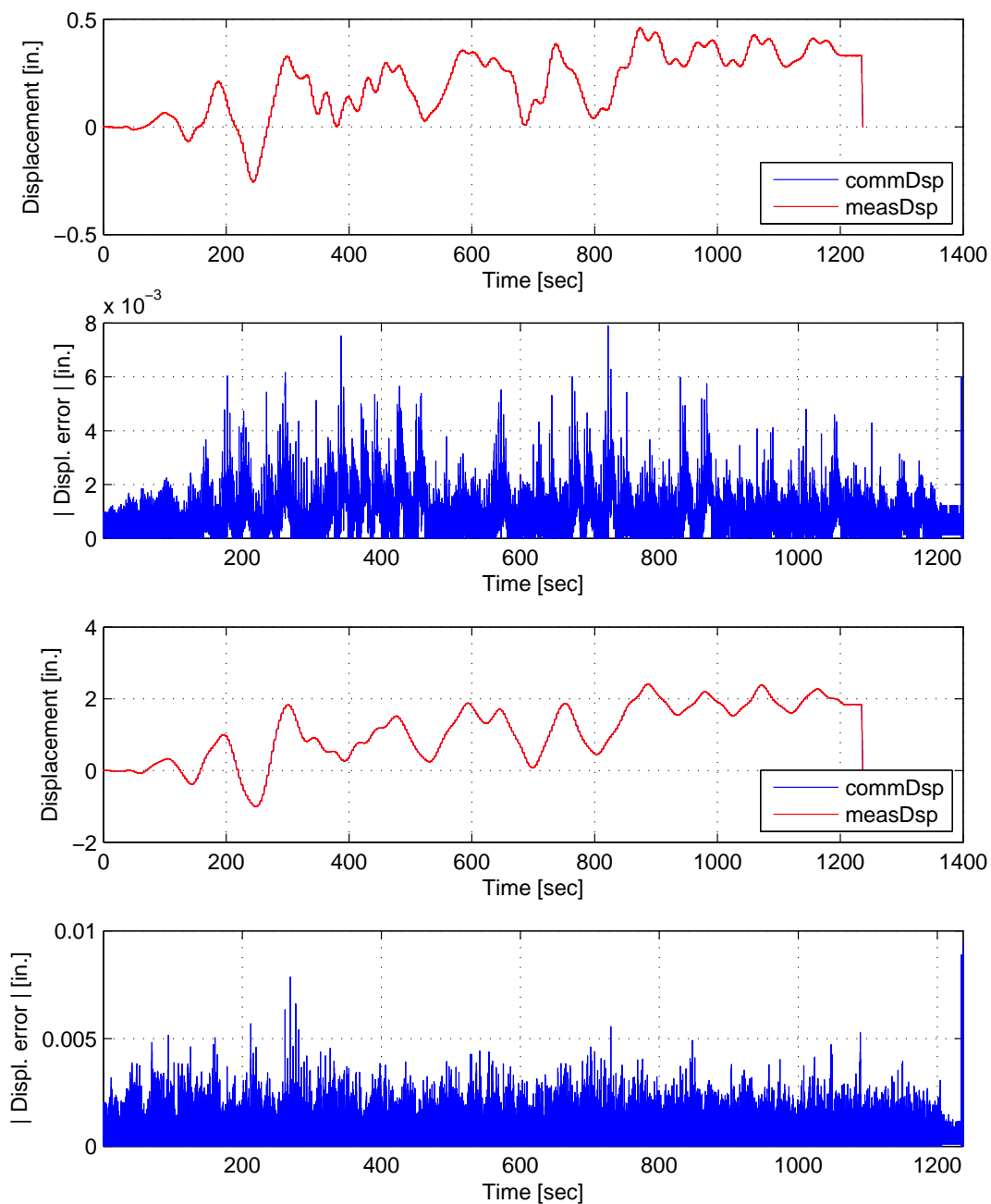


Figure C.52: Bottom actuator command and measured displacement plots from the STS controller in DC mode (top). Bottom actuator absolute error between command and measured displacements from the STS controller (middle-top). Top actuator command and measured displacement plots from the STS controller in DC mode (middle-bottom). Top actuator absolute error between command and measured displacements from the STS controller (bottom). The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses with NMR with DC.

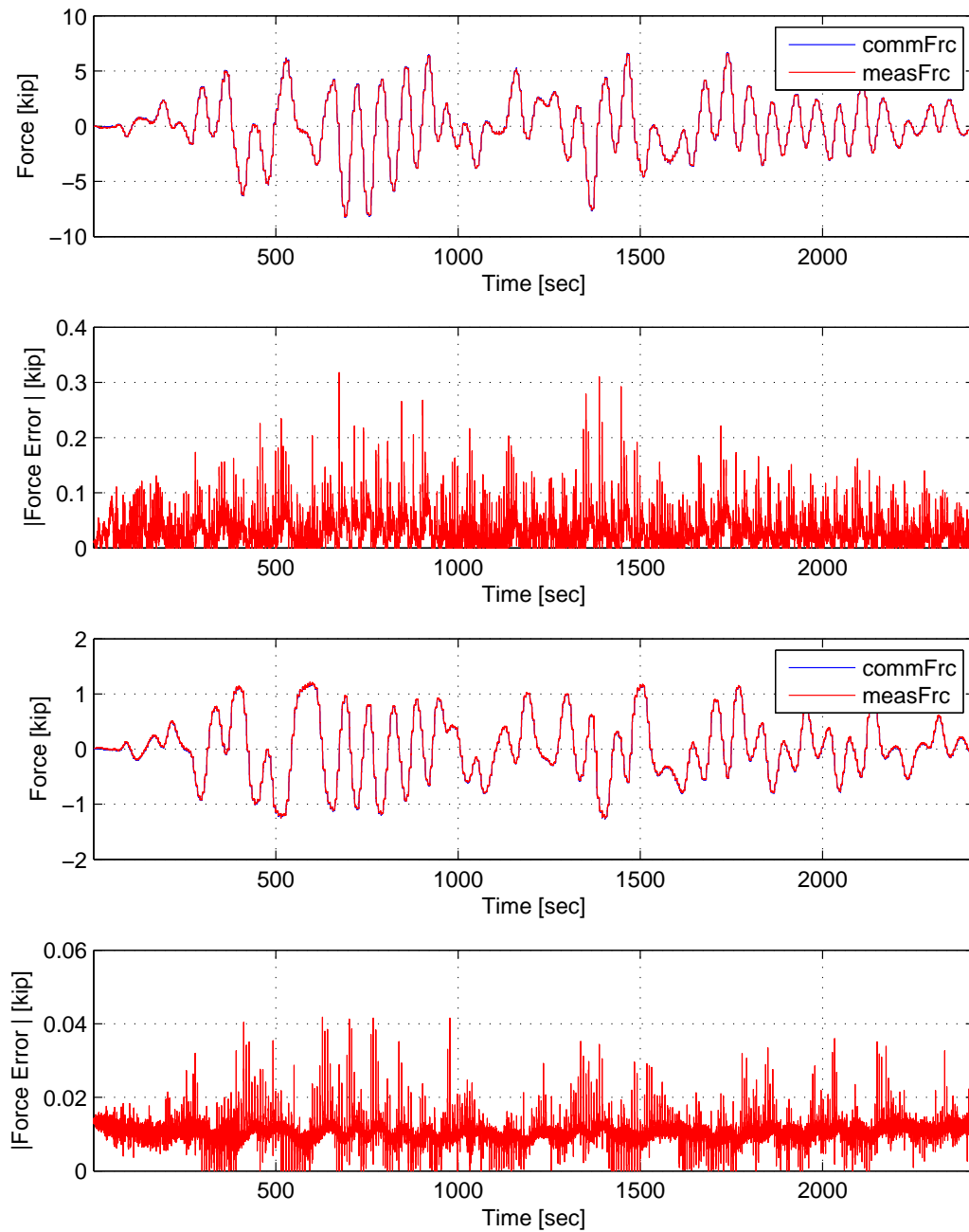


Figure C.53: Bottom actuator command and measured force plots from the STS controller in DC mode (top). Bottom actuator absolute error between command and measured force from the STS controller (middle-top). Top actuator command and measured force plots from the STS controller in DC mode (middle-bottom). Top actuator absolute error between command and measured force from the STS controller (bottom). The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses NMR with CFC:BFGS.

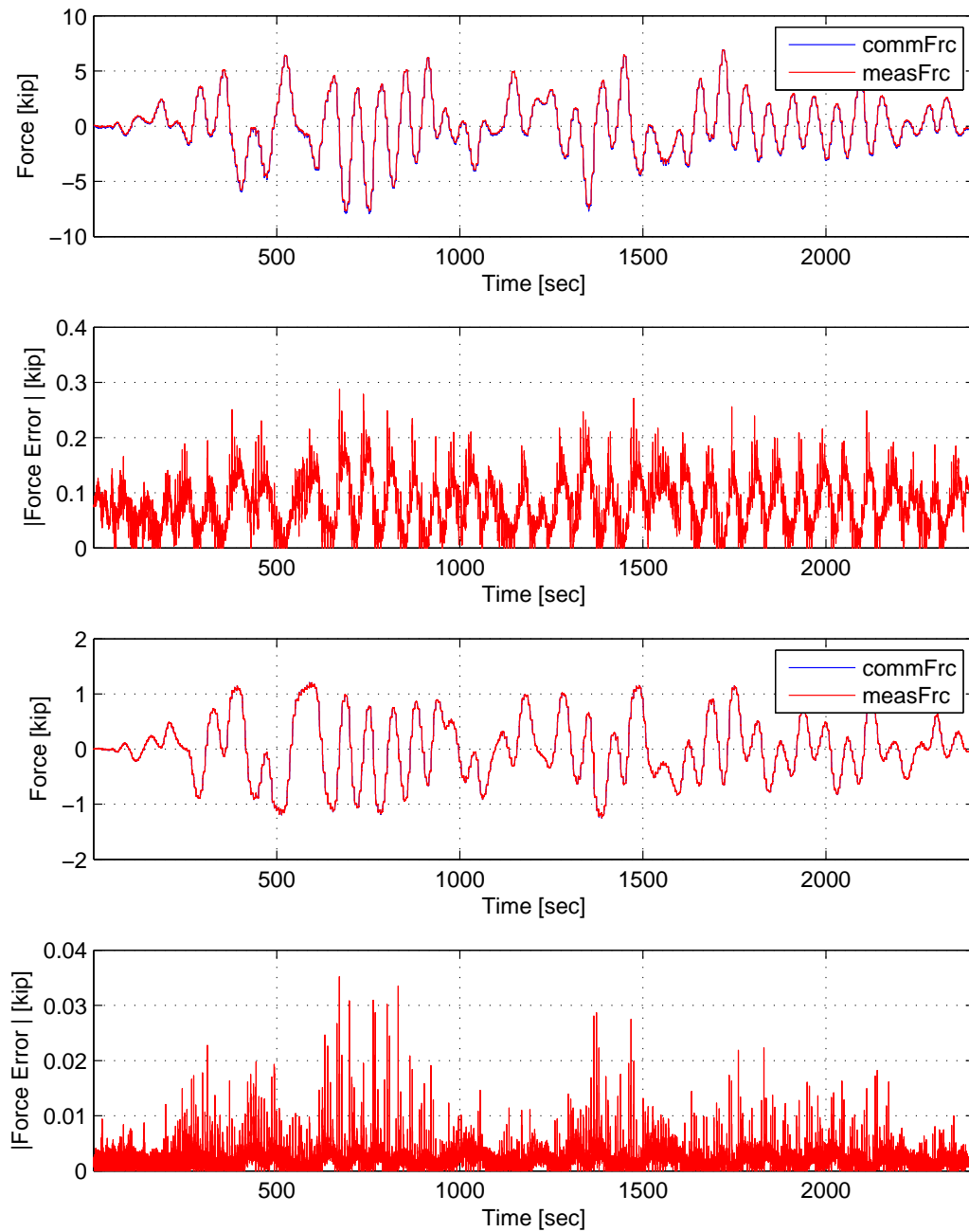


Figure C.54: Bottom actuator command and measured force plots from the STS controller in DC mode (top). Bottom actuator absolute error between command and measured force from the STS controller (middle-top). Top actuator command and measured force plots from the STS controller in DC mode (middle-bottom). Top actuator absolute error between command and measured force from the STS controller (bottom). The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses NMR with CFC:Intrinsic.

## **Appendix D**

### **SC hybrid simulation experimental results**

The El Centro from Section 3.1.2 is scaled to 40%. The hybrid model is shown in Figure XX. Element 1, the experimental element, deformations and resisting forces from the FEA software level are plotted vs time. These response quantities are compared against their numerical counterpart from the OpenSees analysis. The command and measured displacements from the STS controller for the DC hybrid simulation is shown. The command /measure displacement and forces from the STS controller for CFC:SP CFC:SW, EFC:SP and EFC:SW hybrid simulation are plotted vs time as well. The results for these simulations are plotted in Figures D.1 to XX.



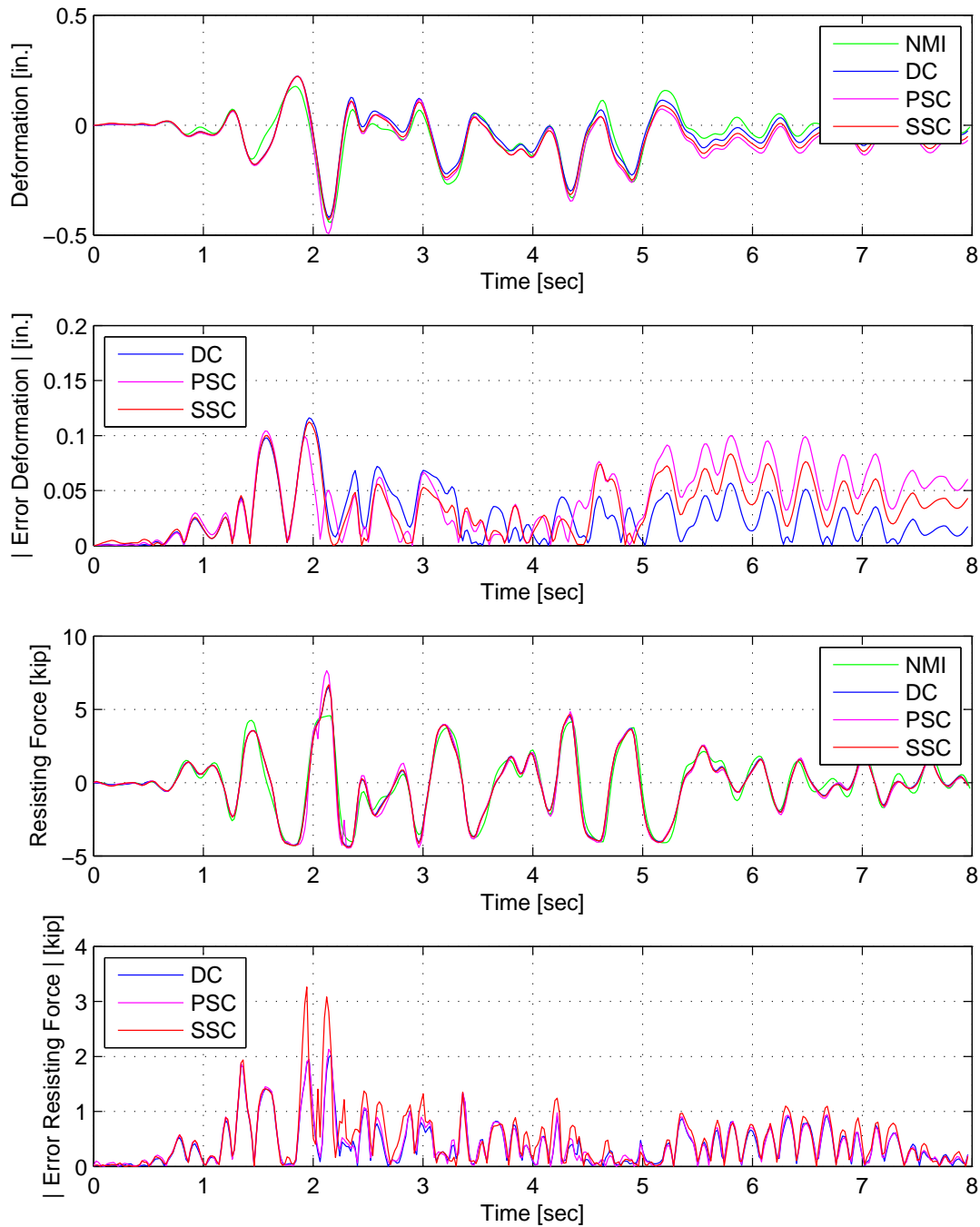


Figure D.1: Displacement-time history plot (top) and displacement error (middle-top) of Element 1 for SC hybrid simulation. Resisting force-time history plot (middle-bottom) and absolute resisting error (bottom) of Element 1. The ground motion is scaled to 40% to push the specimen into the nonlinear range. The 1-DOF setup is used with NME with DC, PSC and SSC.

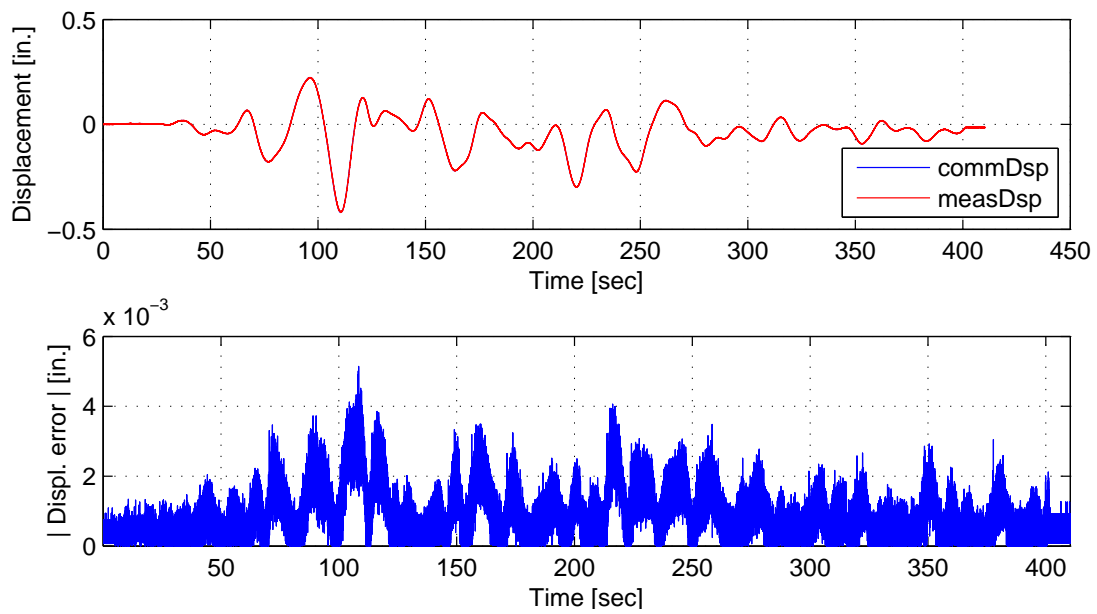


Figure D.2: Top: Command and measured displacement plot from the STS controller for DC hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 40% to push the specimen into the nonlinear range. The 1-DOF setup is used with NME with DC.

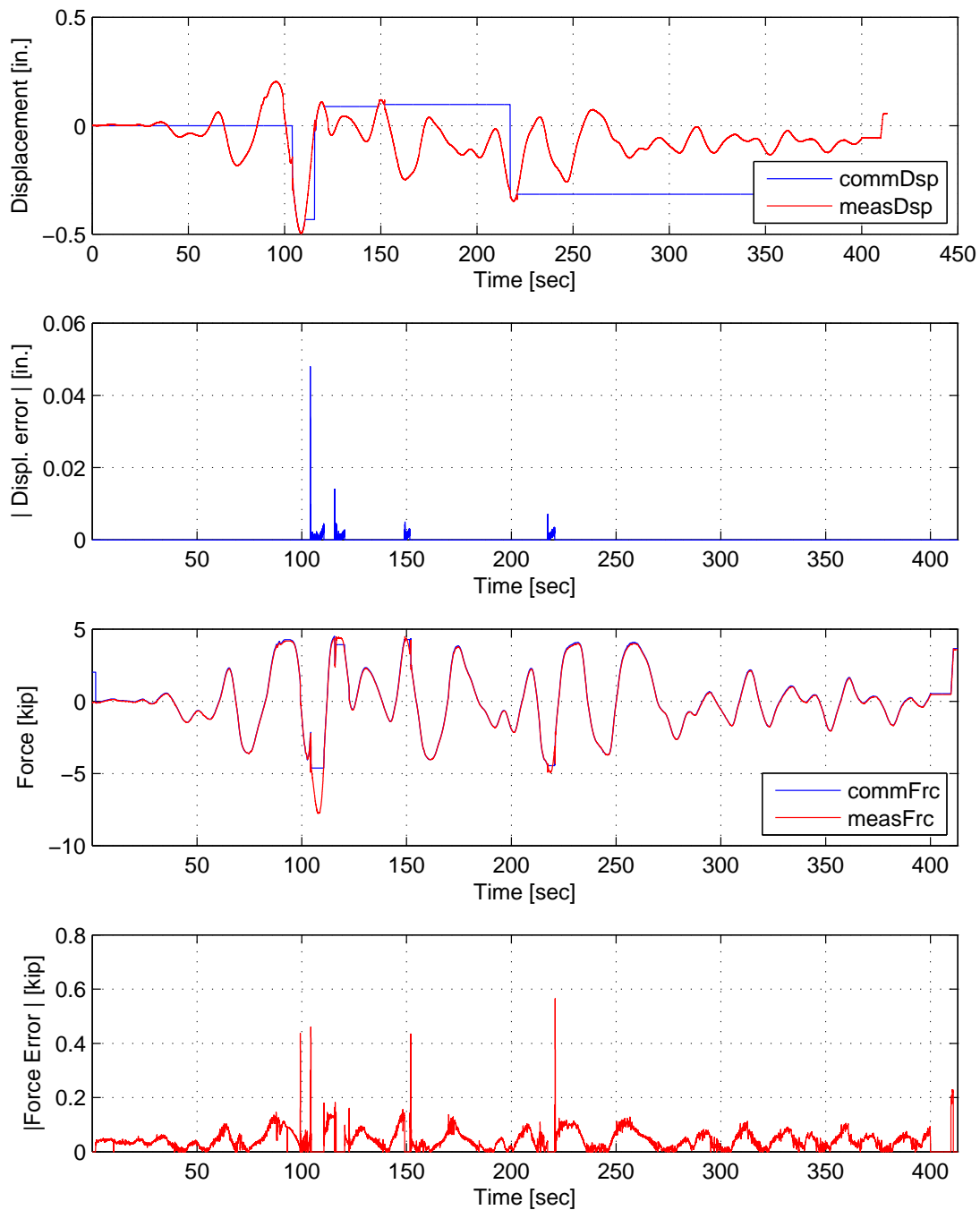


Figure D.3: Top: Command and measured displacement plot from the STS controller for PSC hybrid simulation. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for PSC hybrid simulation. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 40% to push the specimen into the nonlinear range. The 1-DOF setup is used with NME.

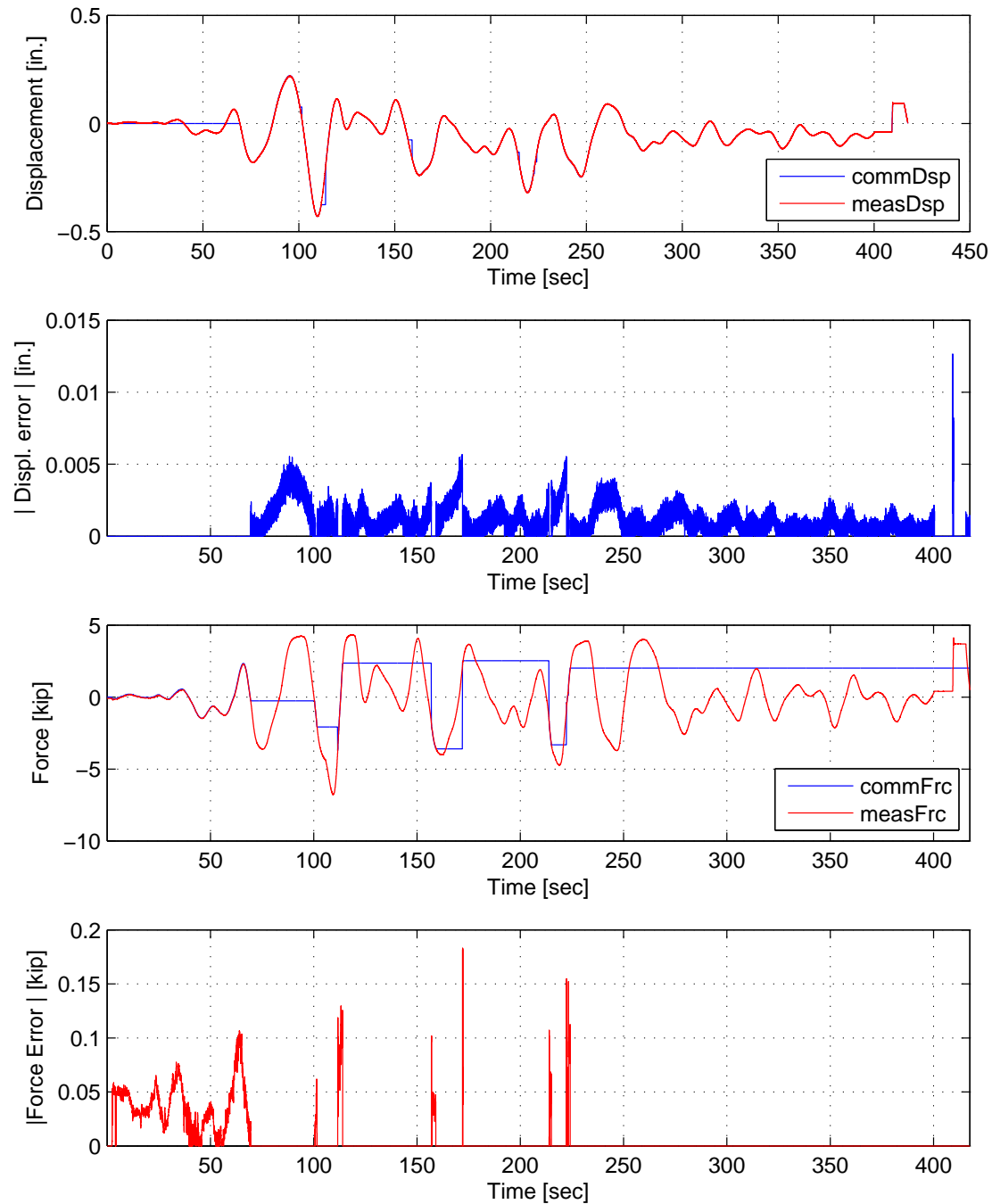


Figure D.4: Top: Command and measured displacement plot from the STS controller for SSC hybrid simulation. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for SSC hybrid simulation. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 40% to push the specimen into the nonlinear range. The 1-DOF setup is used with NME.

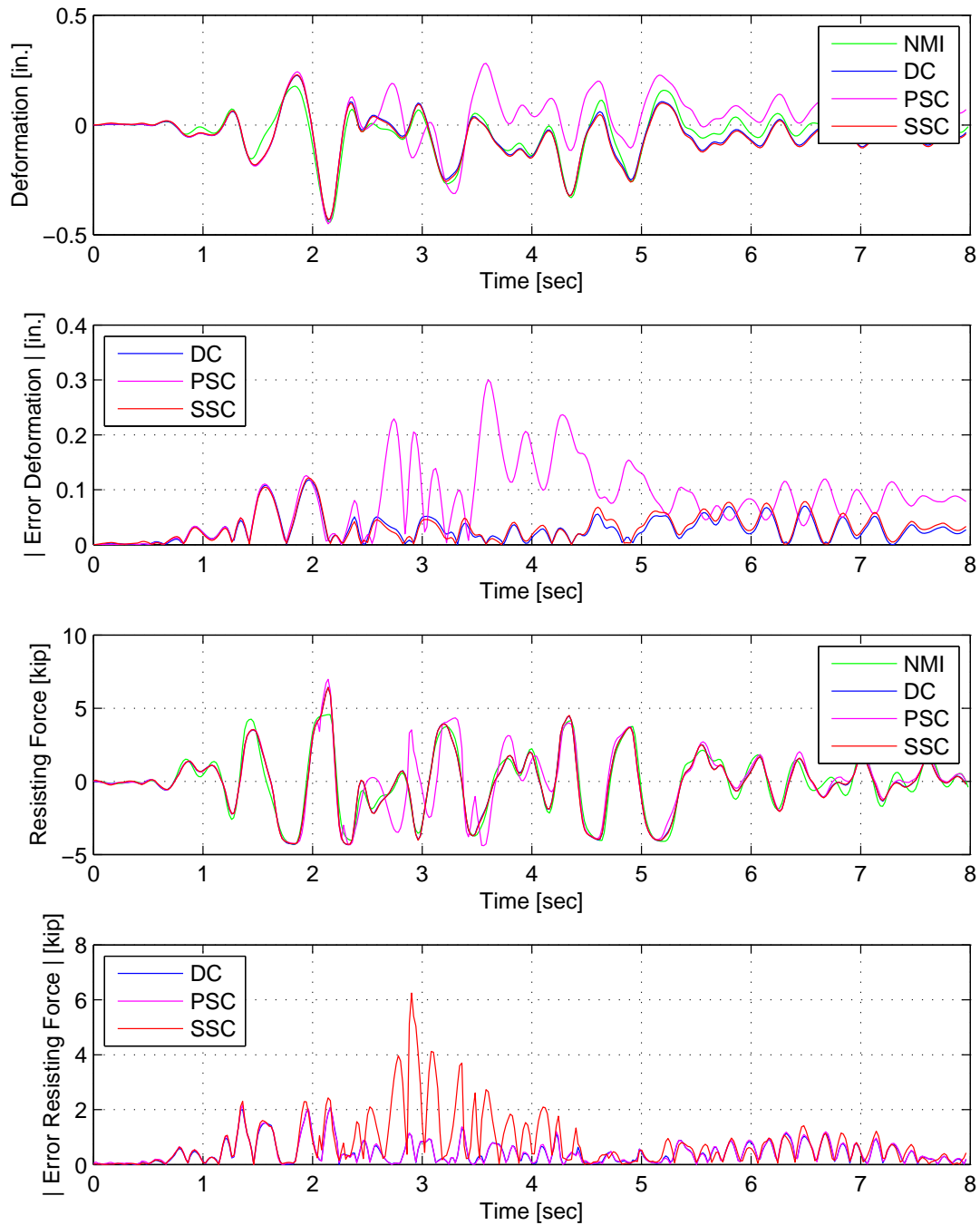


Figure D.5: Displacement-time history plot (top) and displacement error (middle-top) of Element 1 for SC hybrid simulation. Resisting force-time history plot (middle-bottom) and absolute resisting error (bottom) of Element 1. The ground motion is scaled to 40% to push the specimen into the nonlinear range. The 1-DOF setup is used with  $\alpha$ OS with DC, PSC and SSC.

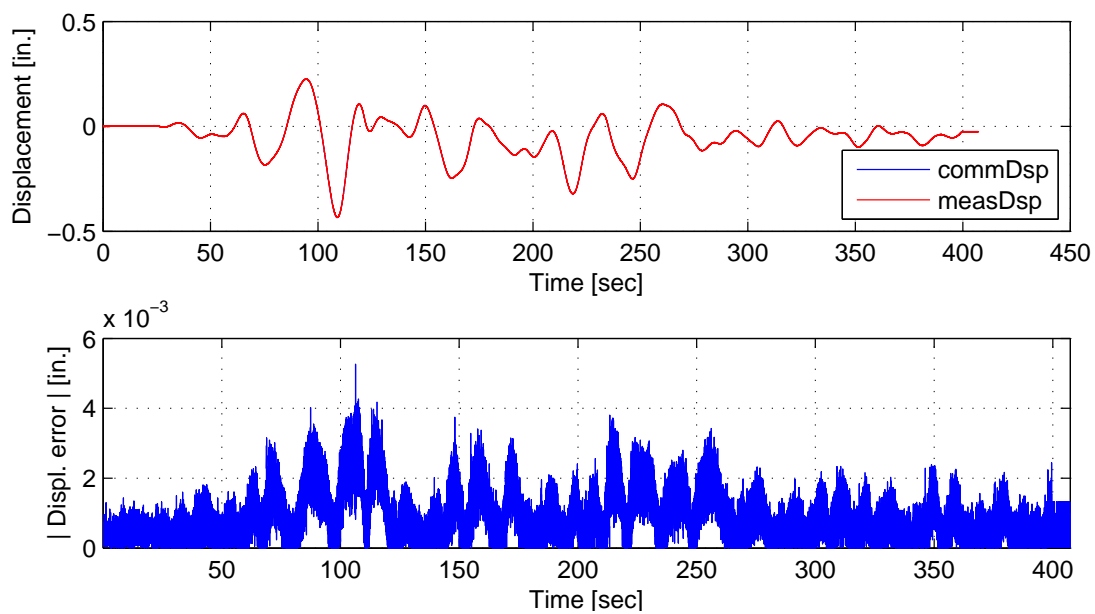


Figure D.6: Top: Command and measured displacement plot from the STS controller for DC hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 40% to push the specimen into the nonlinear range. The 1-DOF setup is used with  $\alpha$ OS with DC.

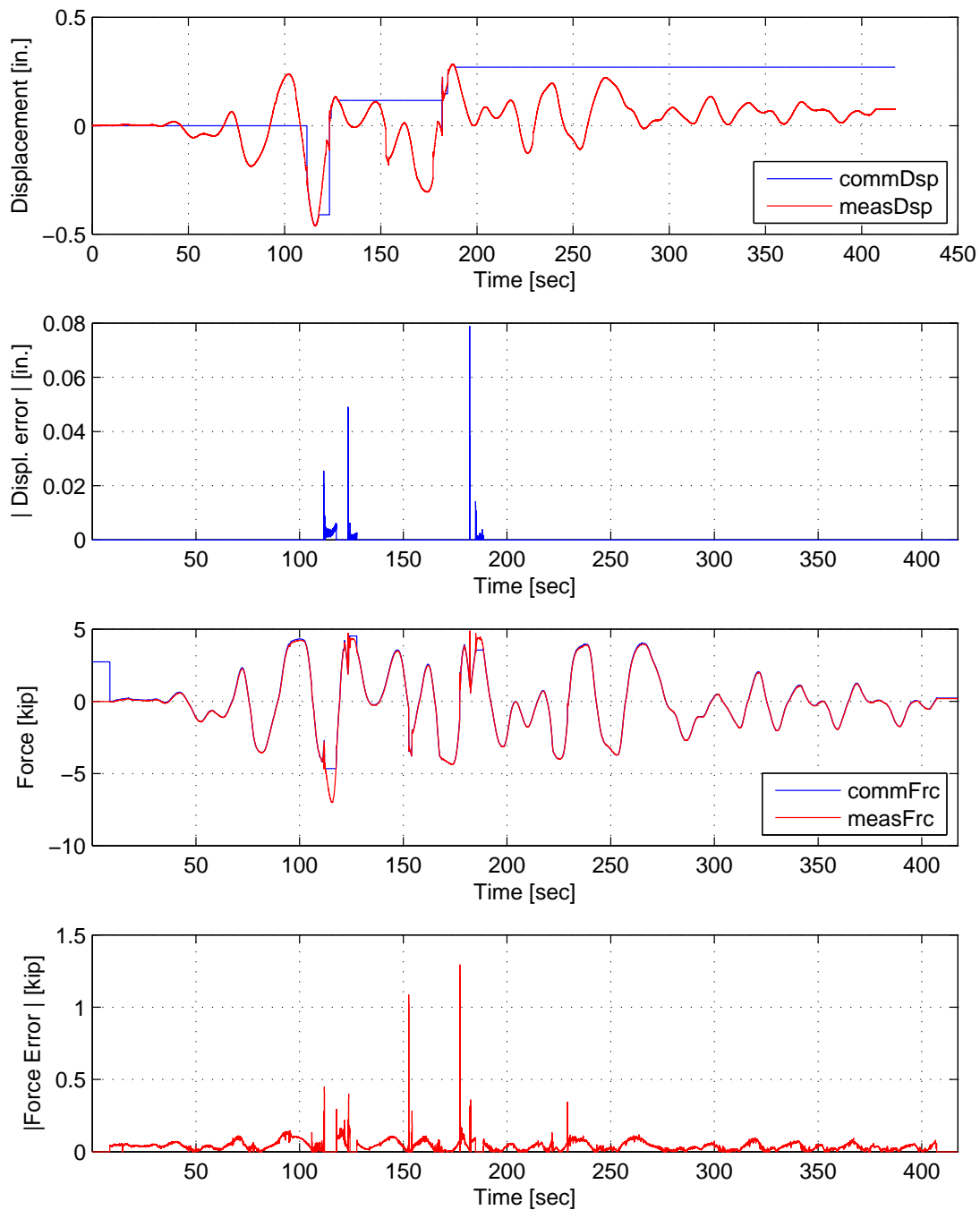


Figure D.7: Top: Command and measured displacement plot from the STS controller for PSC hybrid simulation. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for PSC hybrid simulation. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 40% to push the specimen into the nonlinear range. The 1-DOF setup is used with  $\alpha$ OS.

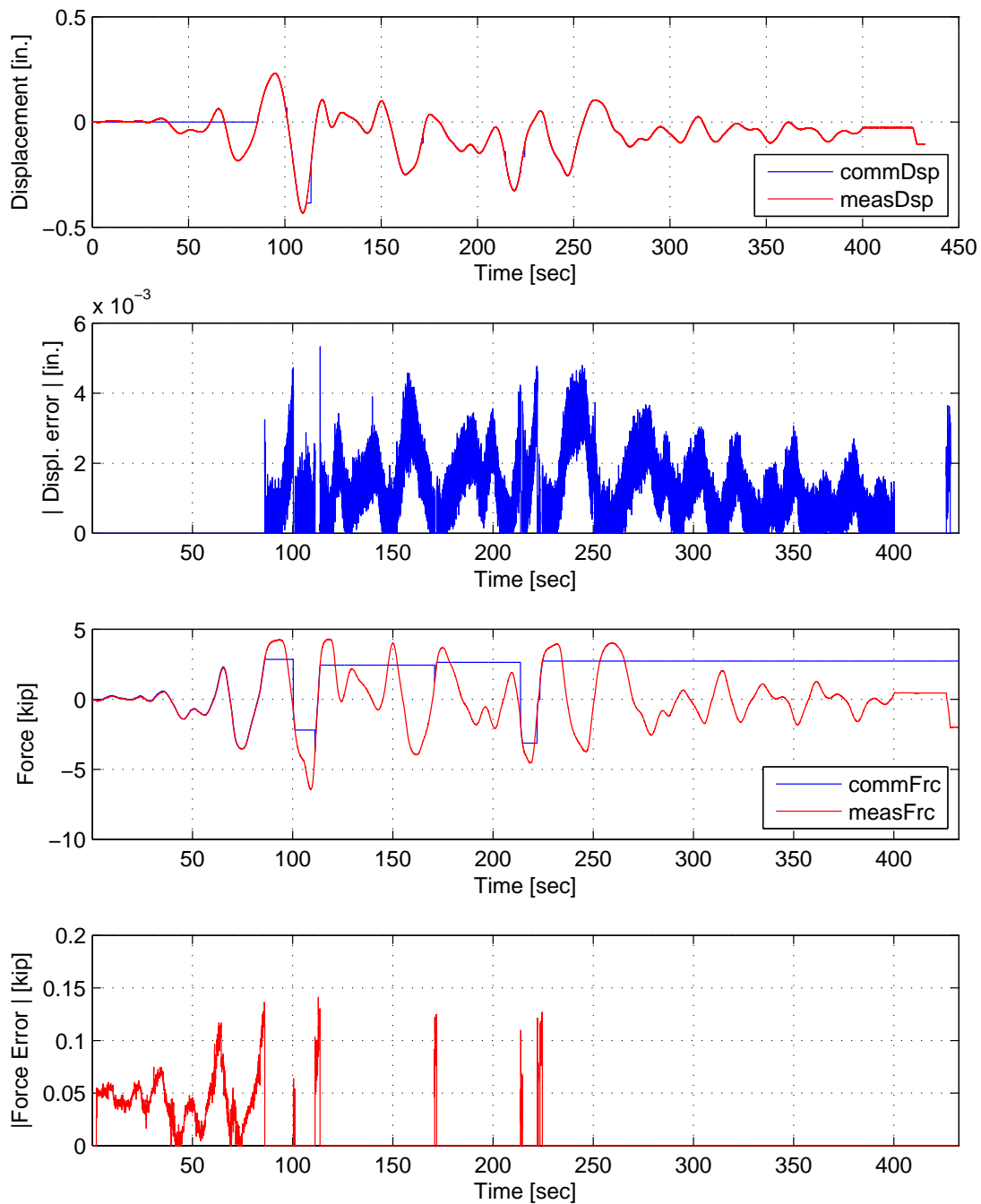


Figure D.8: Top: Command and measured displacement plot from the STS controller for SSC hybrid simulation. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for SSC hybrid simulation. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 40% to push the specimen into the nonlinear range. The 1-DOF setup is used with  $\alpha OS$ .



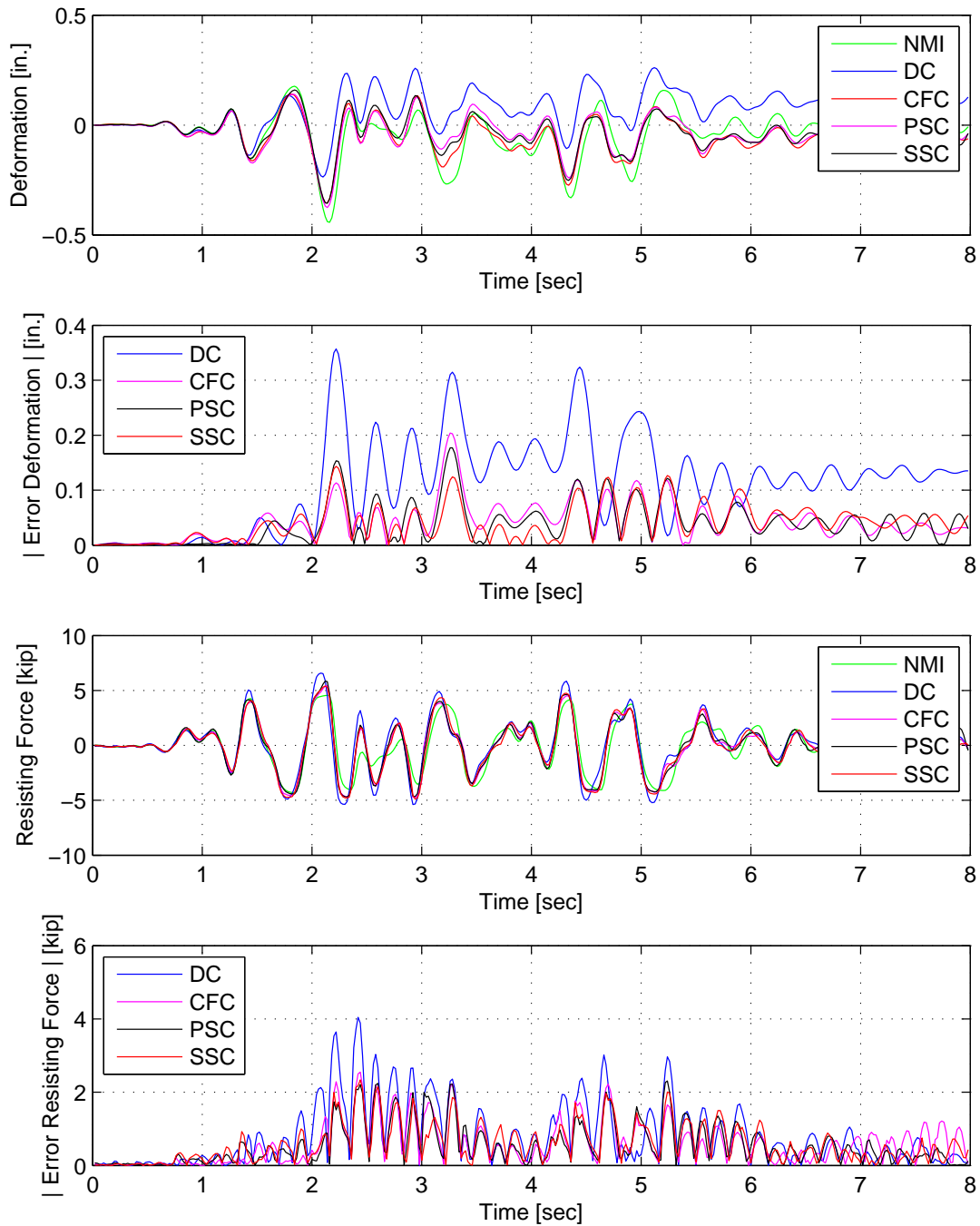


Figure D.9: Displacement-time history plot (top) and displacement error (middle-top) of Element 1 for SC hybrid simulation. Resisting force-time history plot (middle-bottom) and absolute resisting error (bottom) of Element 1. The ground motion is scaled to 40% to push the specimen into the nonlinear range. The 1-DOF setup is used with NMF with PSC and SSC.

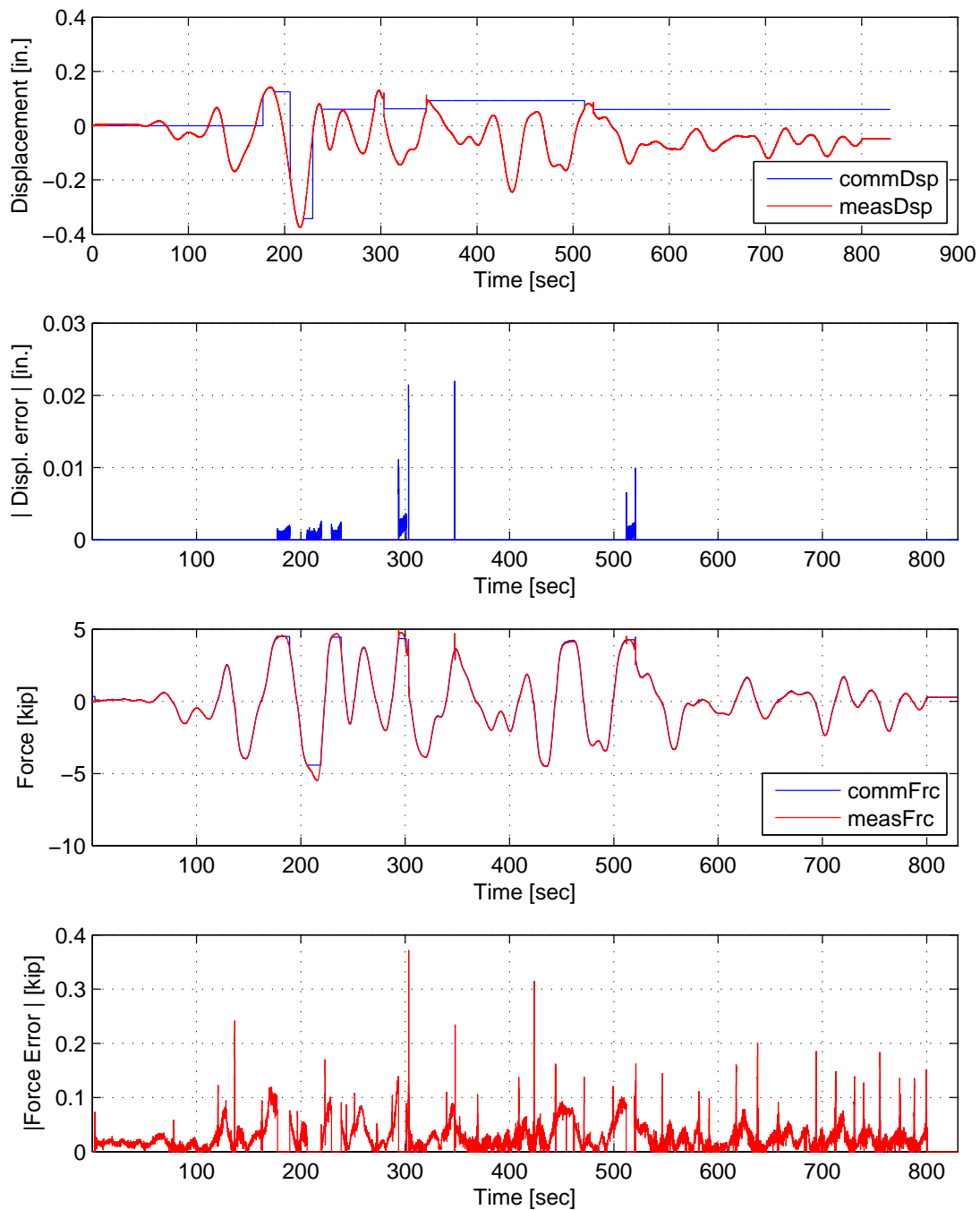


Figure D.10: Top: Command and measured displacement plot from the STS controller for PSC hybrid simulation. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for PSC hybrid simulation. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 40% to push the specimen into the nonlinear range. The 1-DOF setup is used with NMF.

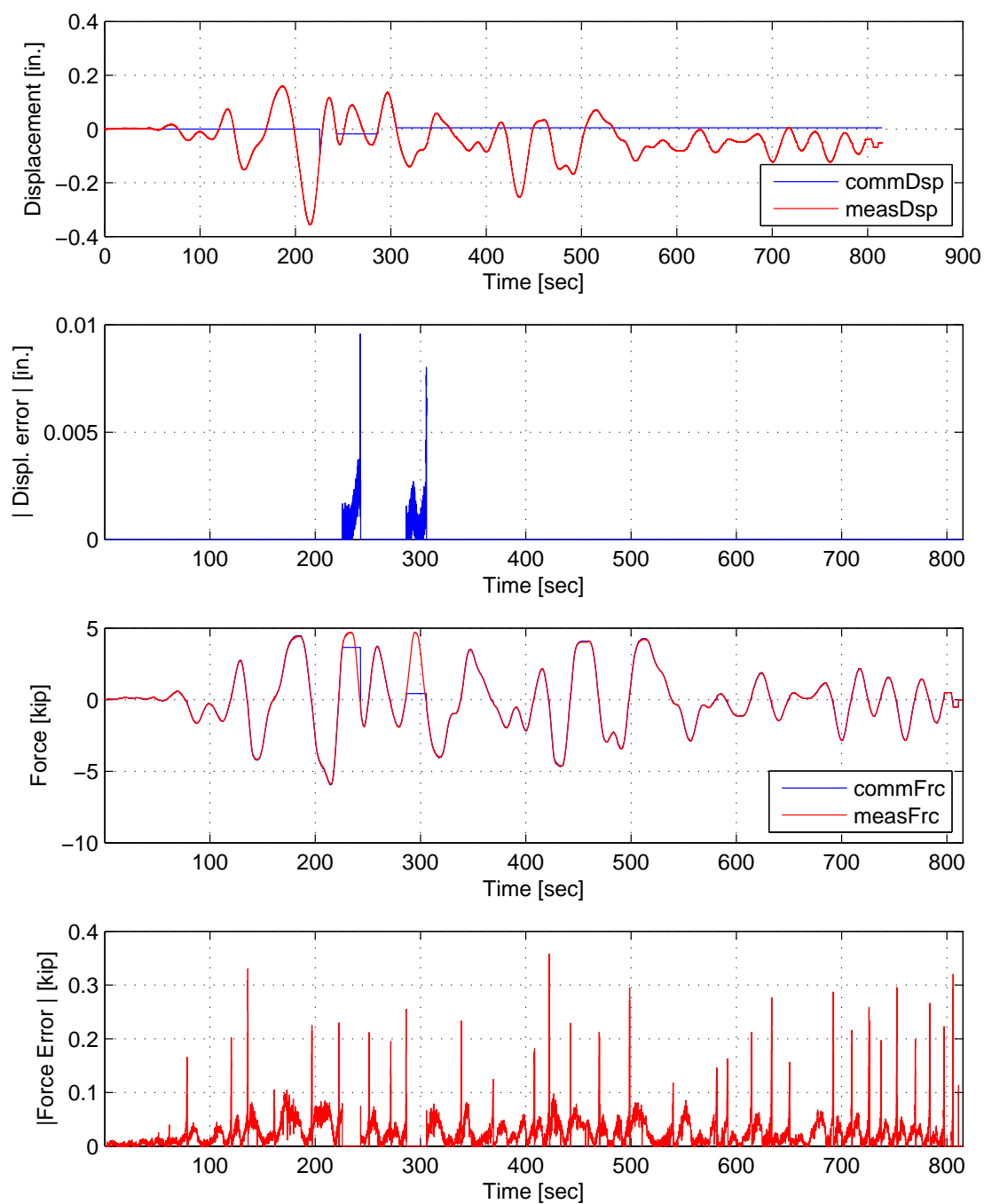


Figure D.11: Top: Command and measured displacement plot from the STS controller for SSC hybrid simulation. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for SSC hybrid simulation. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 40% to push the specimen into the nonlinear range. The 1-DOF setup is used with NMF.

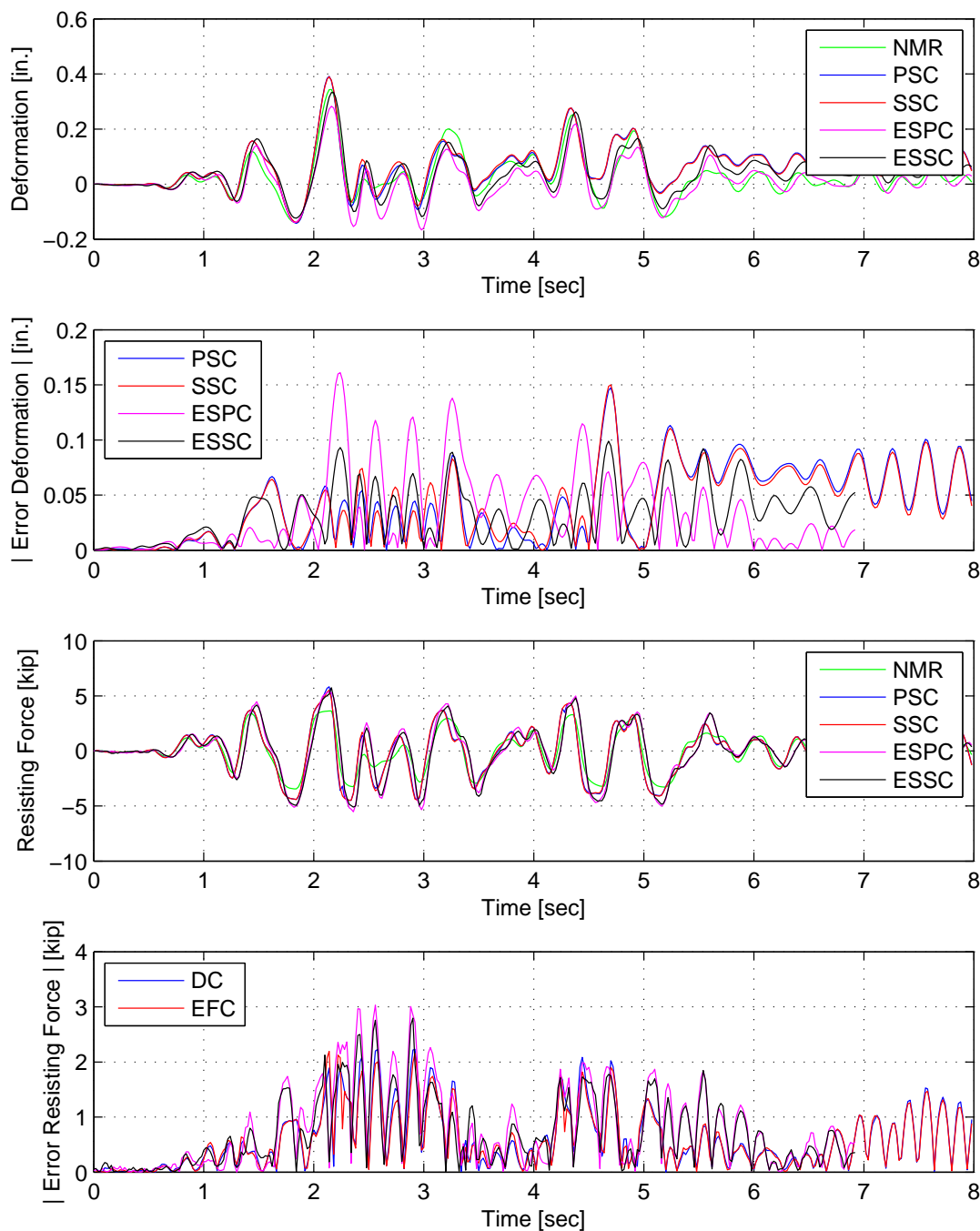


Figure D.12: Displacement-time history plot (top) and displacement error (middle-top) of Element 1 for SC hybrid simulation. Resisting force-time history plot (middle-bottom) and absolute resisting error (bottom) of Element 1. The ground motion is scaled to 40% to push the specimen into the nonlinear range. The 1-DOF setup is used with NMR with PSC, SSC, ESPC and ESSC.

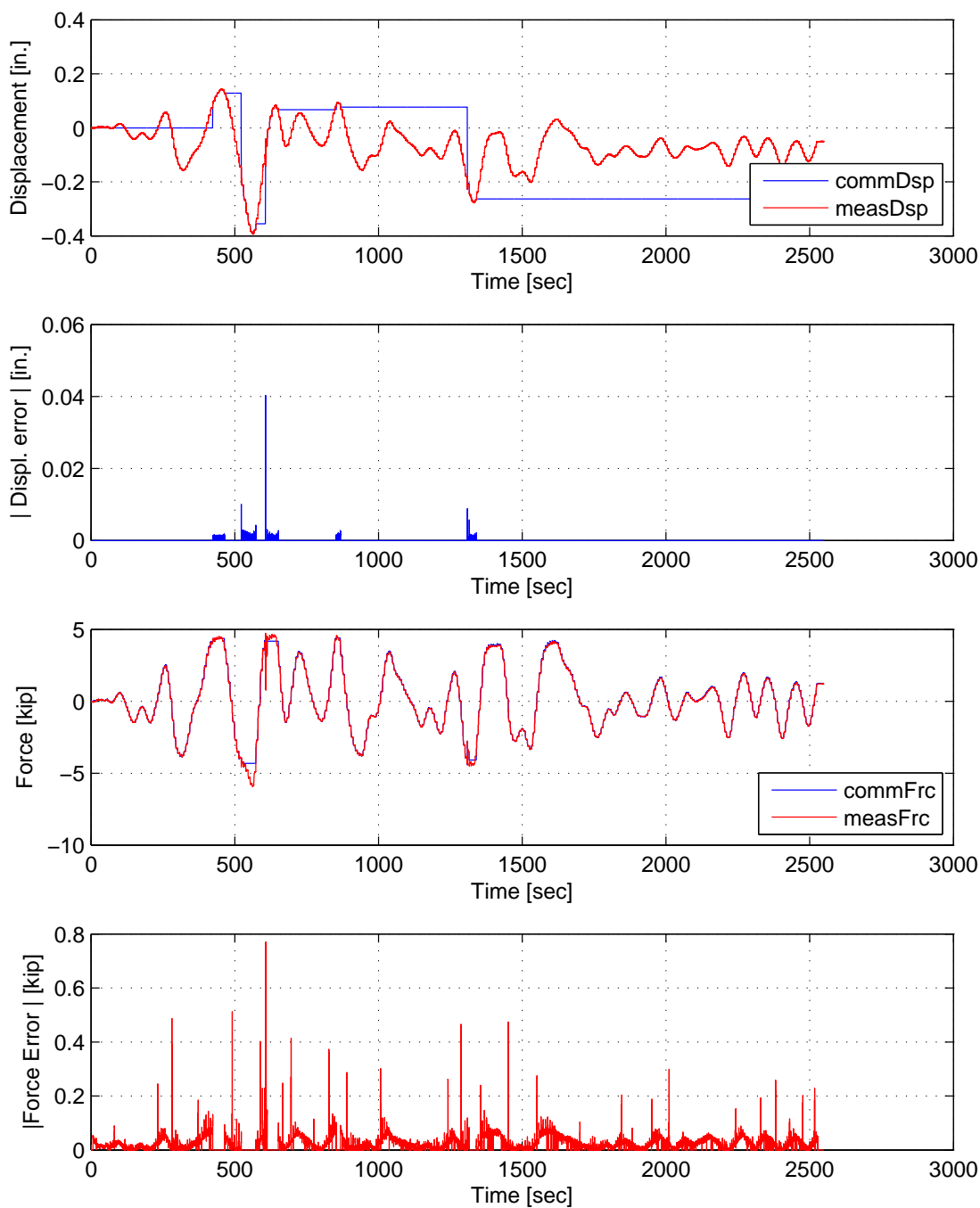


Figure D.13: Top: Command and measured displacement plot from the STS controller for PSC hybrid simulation. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for PSC hybrid simulation. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 40% to push the specimen into the nonlinear range. The 1-DOF setup is used with NMR.

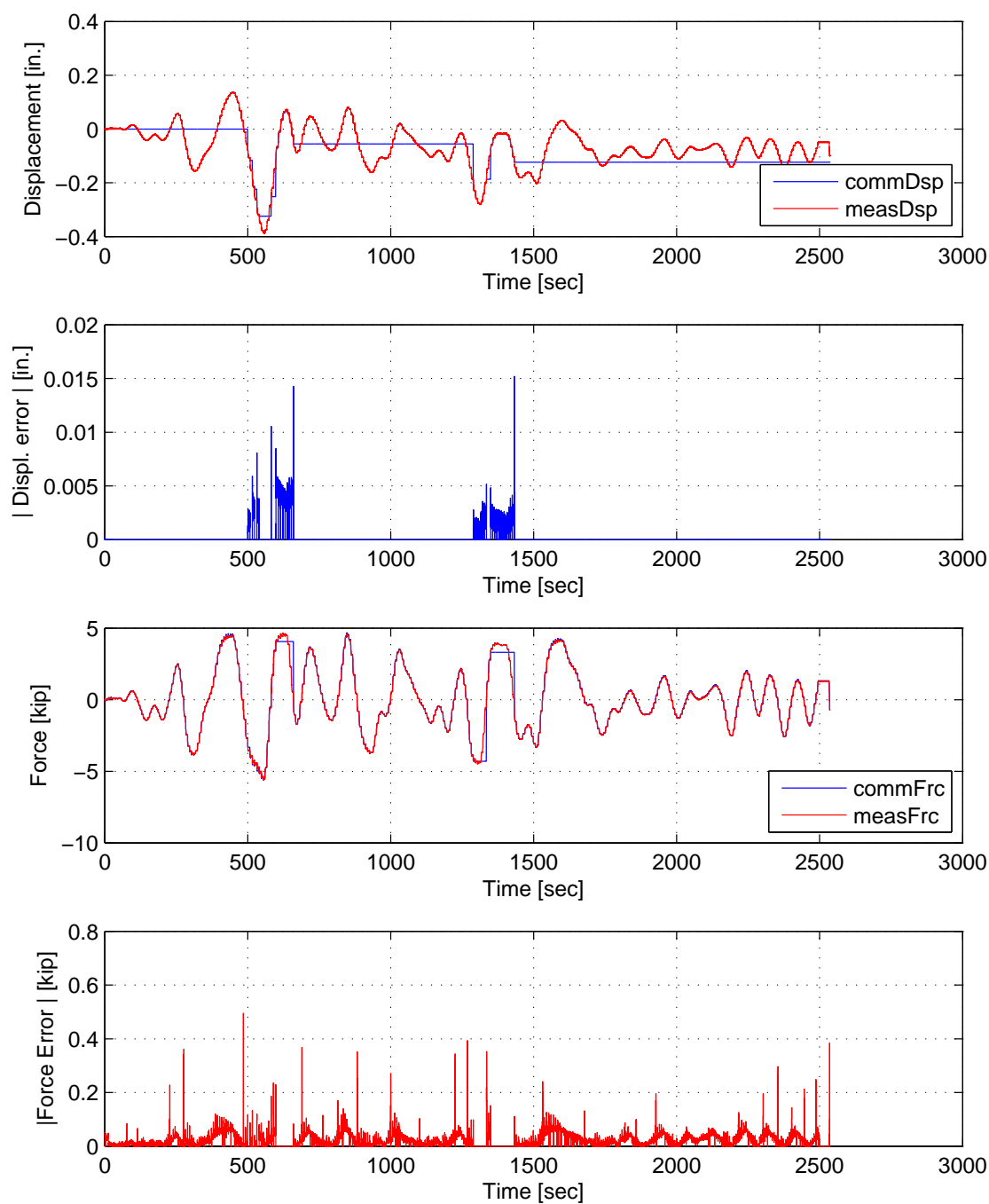


Figure D.14: Top: Command and measured displacement plot from the STS controller for SSC hybrid simulation. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for SSC hybrid simulation. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 40% to push the specimen into the nonlinear range. The 1-DOF setup is used with NMR.

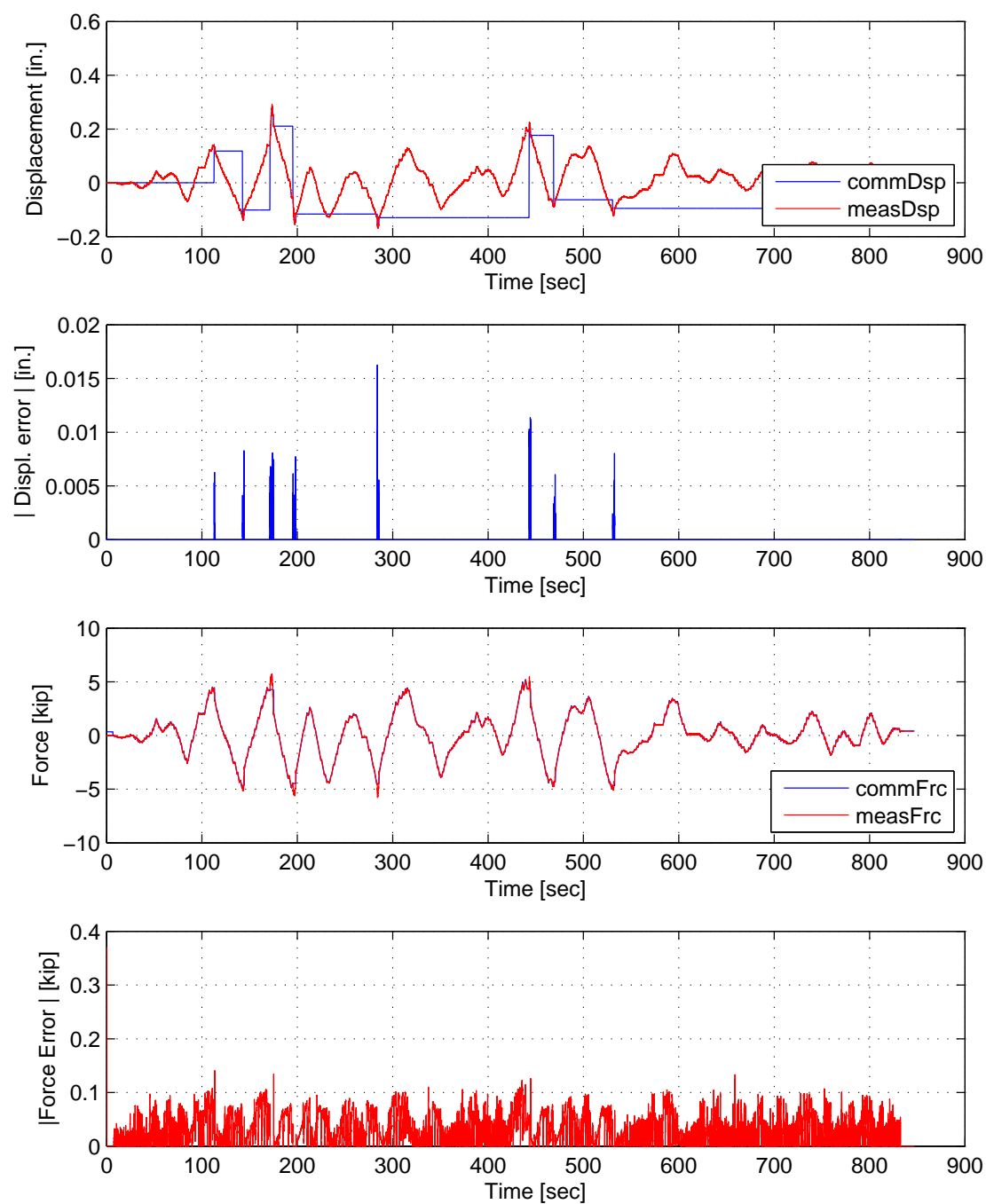


Figure D.15: Top: Command and measured displacement plot from the STS controller for EPSC hybrid simulation. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for EPSC hybrid simulation. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 40% to push the specimen into the nonlinear range. The 1-DOF setup is used with NMR.

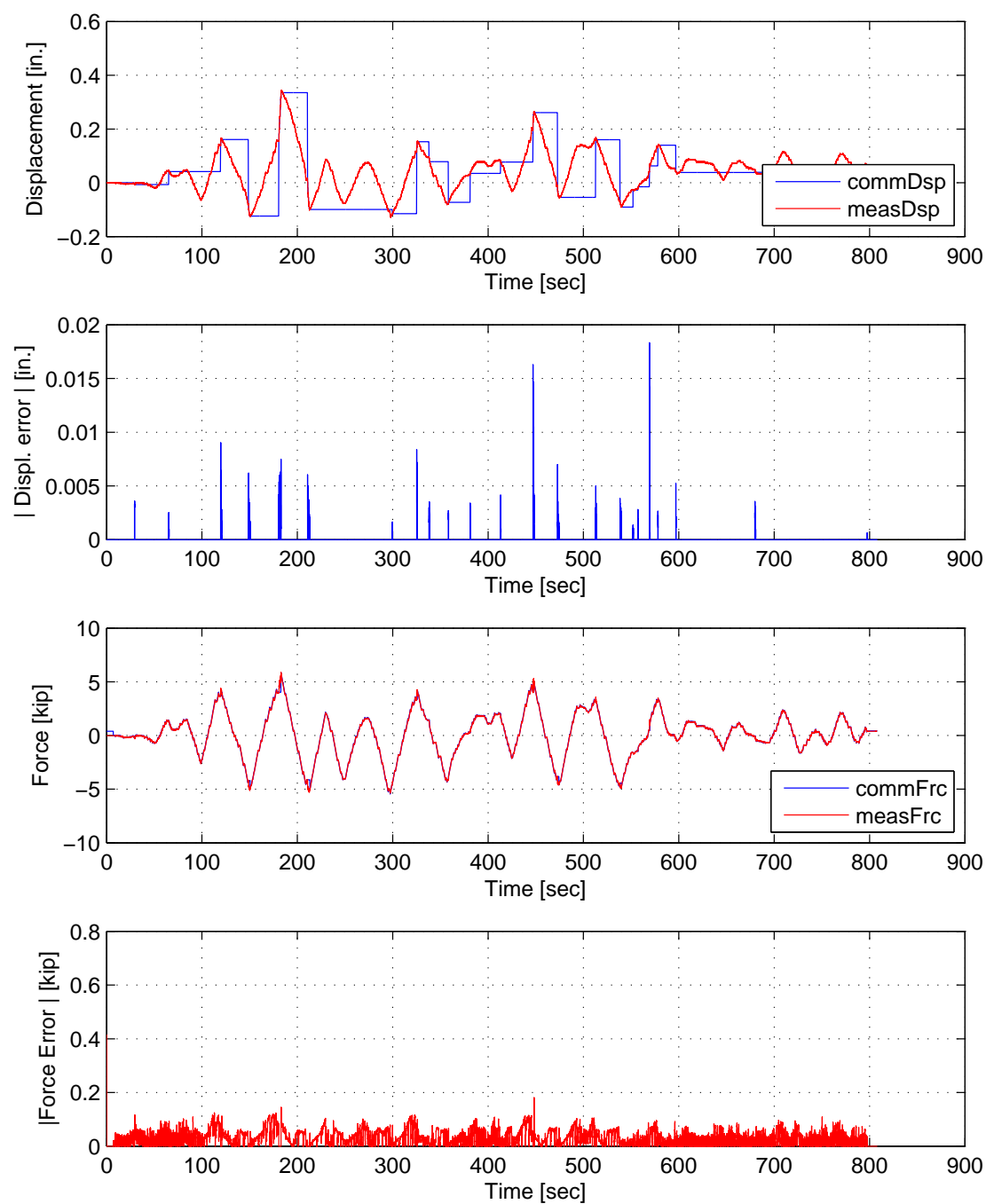


Figure D.16: Top: Command and measured displacement plot from the STS controller for ESSC hybrid simulation. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for ESSC hybrid simulation. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 40% to push the specimen into the nonlinear range. The 1-DOF setup is used with NMR.



# Appendix E

## MC hybrid simulation experimental results

The El Centro from Section 3.1.2 is scaled to 40%. The hybrid model is shown in Figure XX. Element 1, the experimental element, deformations and resisting forces from the FEA software level are plotted vs time. These response quantities are compared against their numerical counterpart from the OpenSees analysis. The command and measured displacements from the STS controller for the DC hybrid simulation is shown. The command /measure displacement and forces from the STS controller for CFC:SP CFC:SW, EFC:SP and EFC:SW hybrid simulation are plotted vs time as well. The results for these simulations are plotted in Figures XX to XX.

### E.1 Point switching hybrid simulation

#### E.1.1 NME time integration scheme

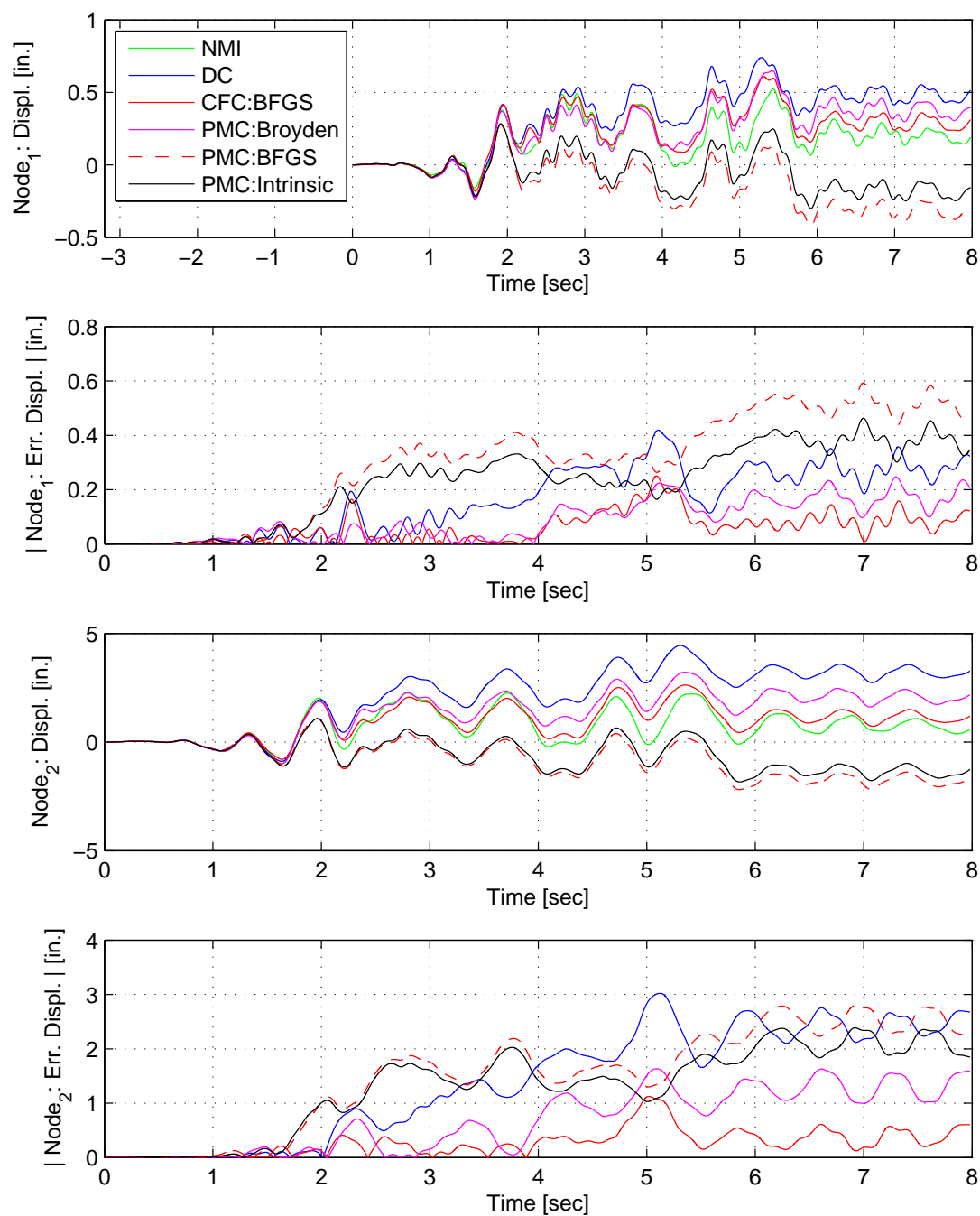


Figure E.1: Node 1 displacement-time history plot (top), Node 1 absolute displacement error (middle-top), Node 2 displacement-time history plot (middle-bottom) and Node 2 absolute displacement error (bottom) for PMC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses NME with DC, CFC:BFGS, PMC:Broyden, PMC: BFGS, PMC:Intrinsic. All experimental result errors are calculated using the NMI numerical result.

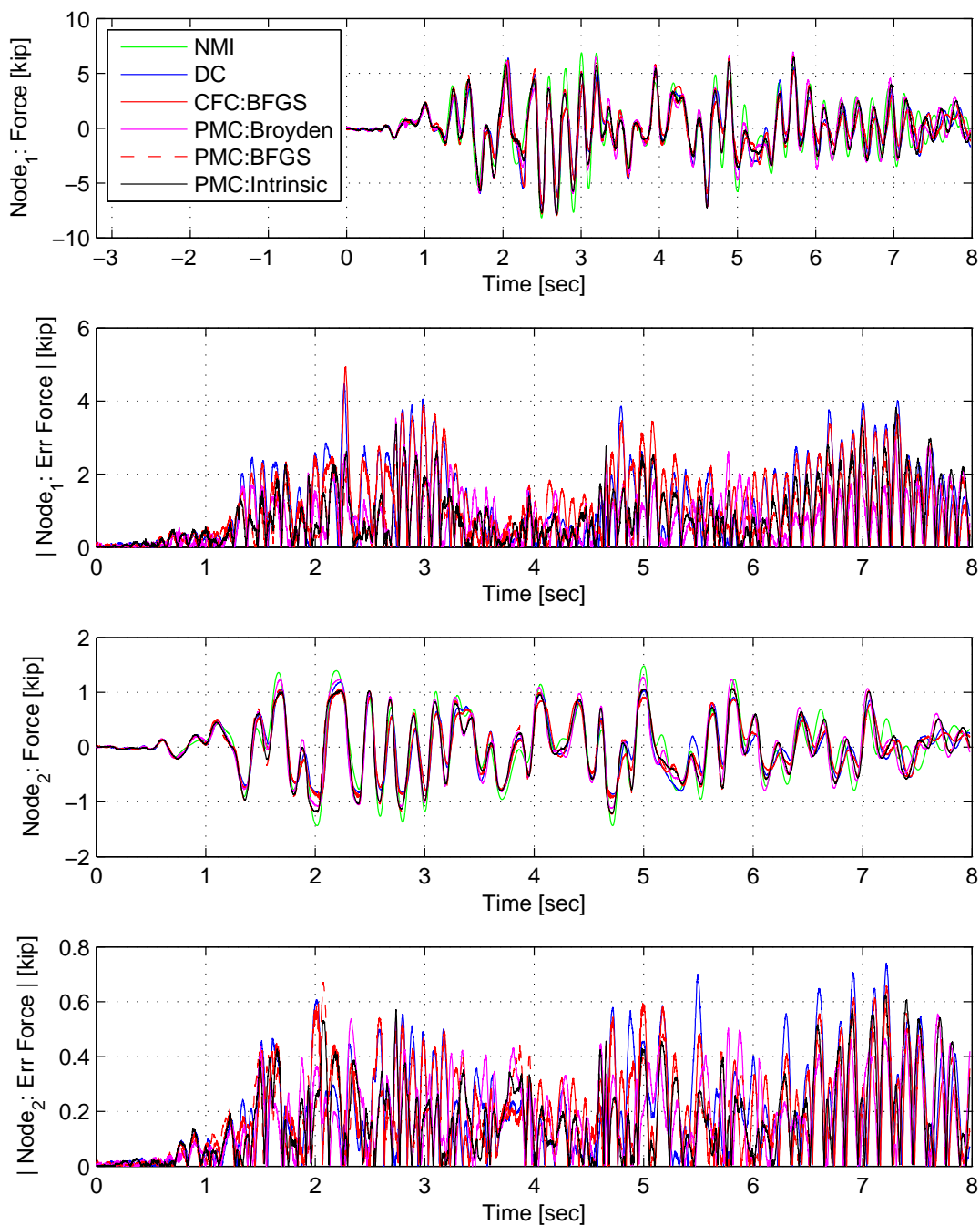


Figure E.2: Node 1 force-time history plot (top), Node 1 absolute force error (middle-top), Node 2 force-time history plot (middle-bottom) and Node 2 absolute force error (bottom) for PMC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses NME with DC, CFC:BFGS, PMC:Broyden, PMC: BFGS, PMC:Intrinsic. All experimental result errors are calculated using the NMI numerical result.

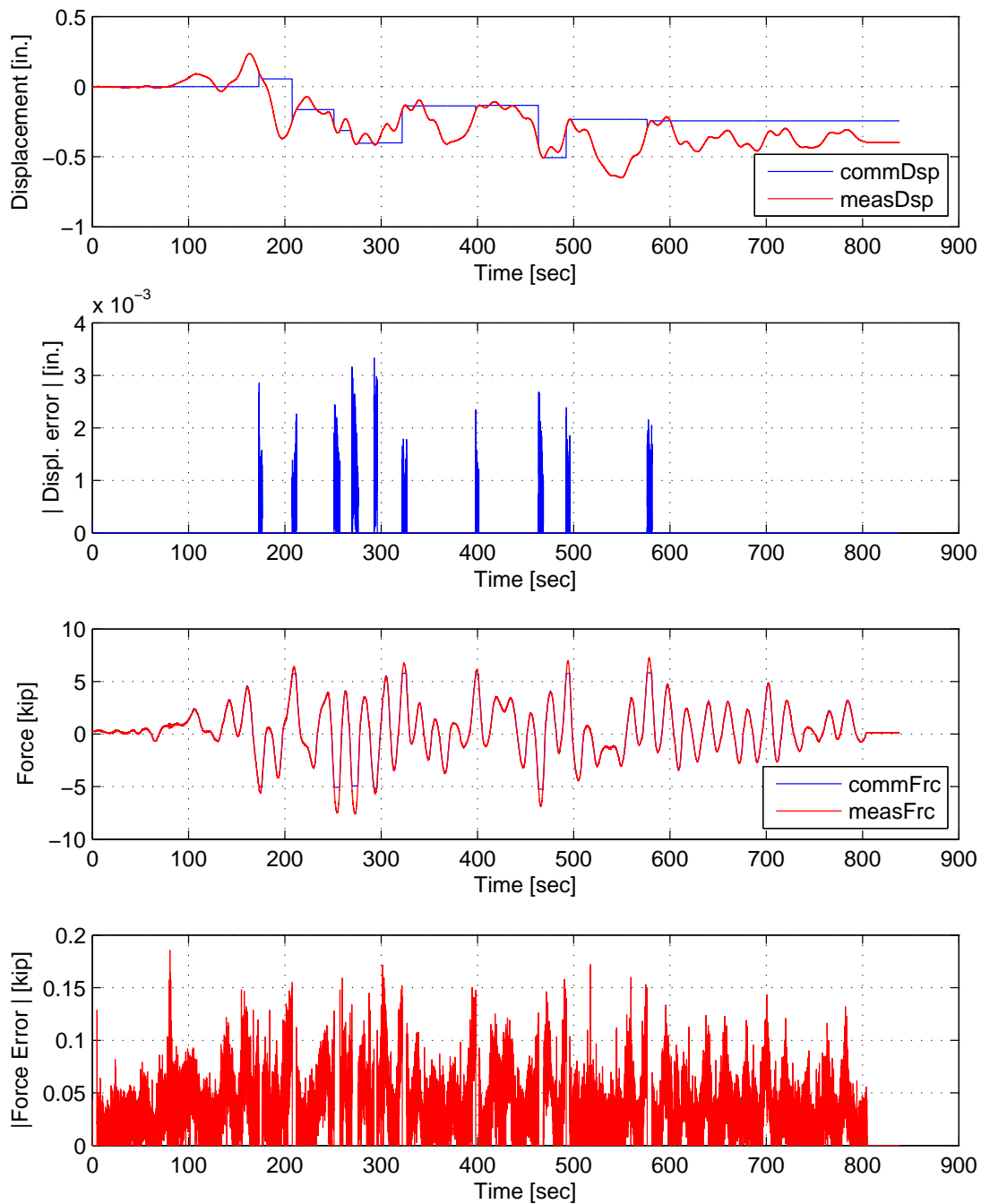


Figure E.3: STS controller recorded values for bottom actuator. Top: Command and measured displacement plot from the STS controller for PMC:Broyden hybrid simulation in DC. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for PMC:Broyden hybrid simulation in FC. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NME with PMC:Broyden.

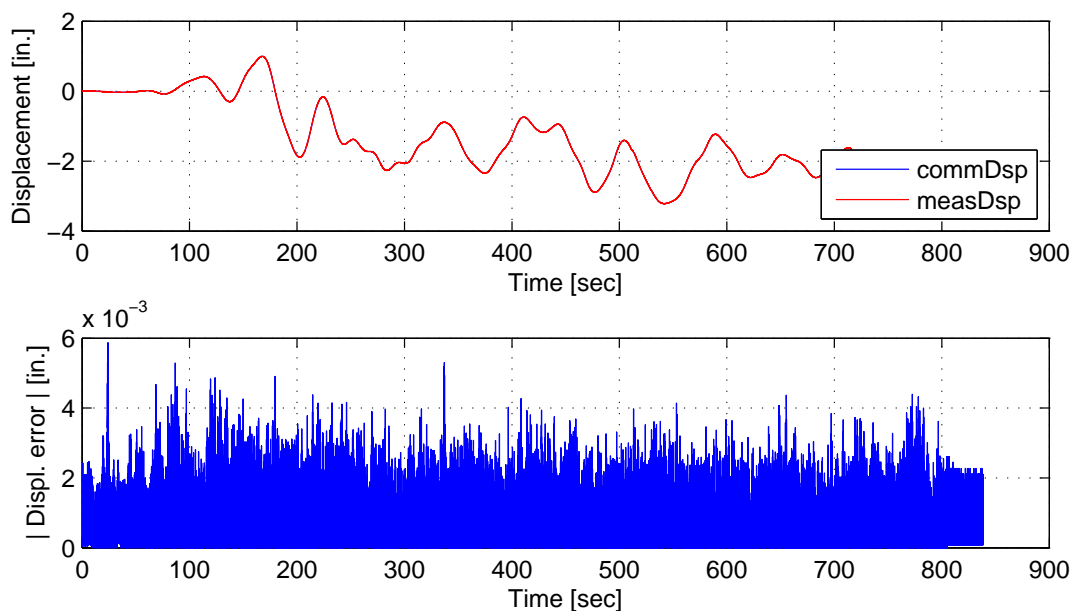


Figure E.4: STS controller recorded values for top actuator. Top: Command and measured displacement plot from the STS controller for PMC:Broyden hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NME with PMC:Broyden.

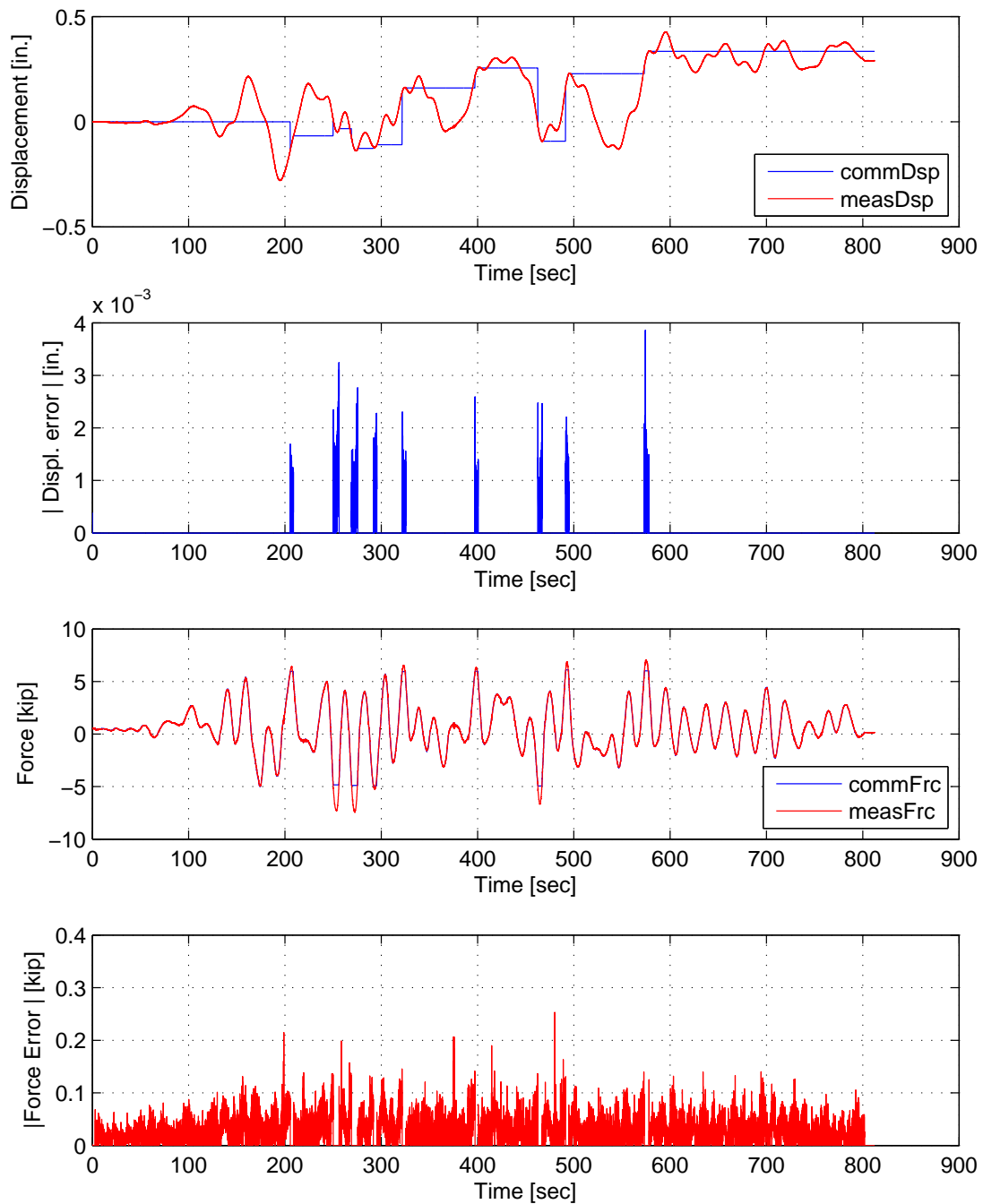


Figure E.5: STS controller recorded values for bottom actuator. Top: Command and measured displacement plot from the STS controller for PMC:BFGS hybrid simulation in DC. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for PMC:BFGS hybrid simulation in FC. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NME with PMC:BFGS.

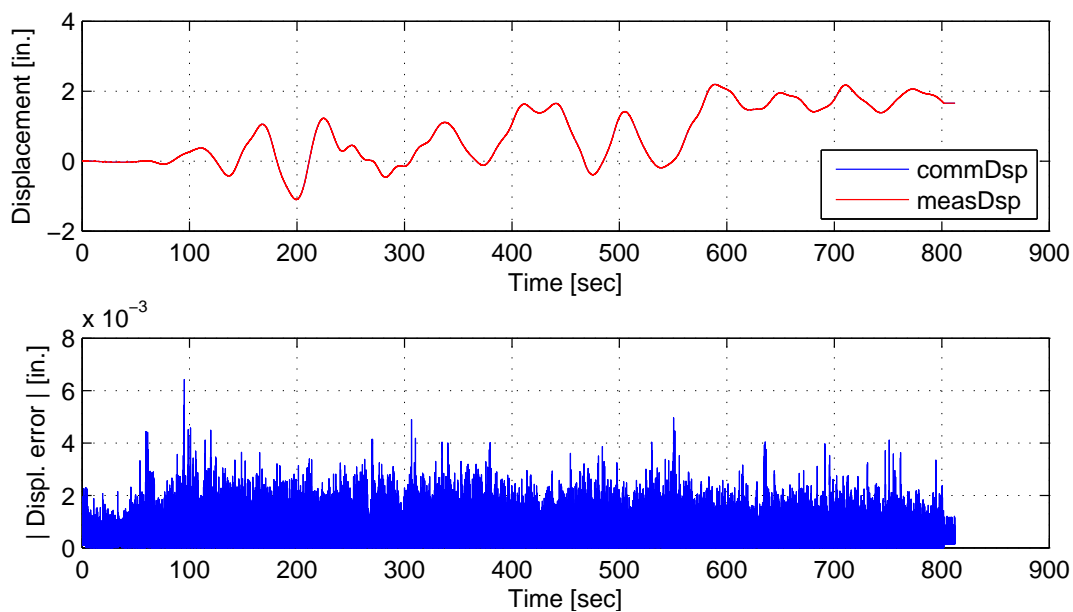


Figure E.6: STS controller recorded values for top actuator. Top: Command and measured displacement plot from the STS controller for PMC:BFGS hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NME with PMC:BFGS.

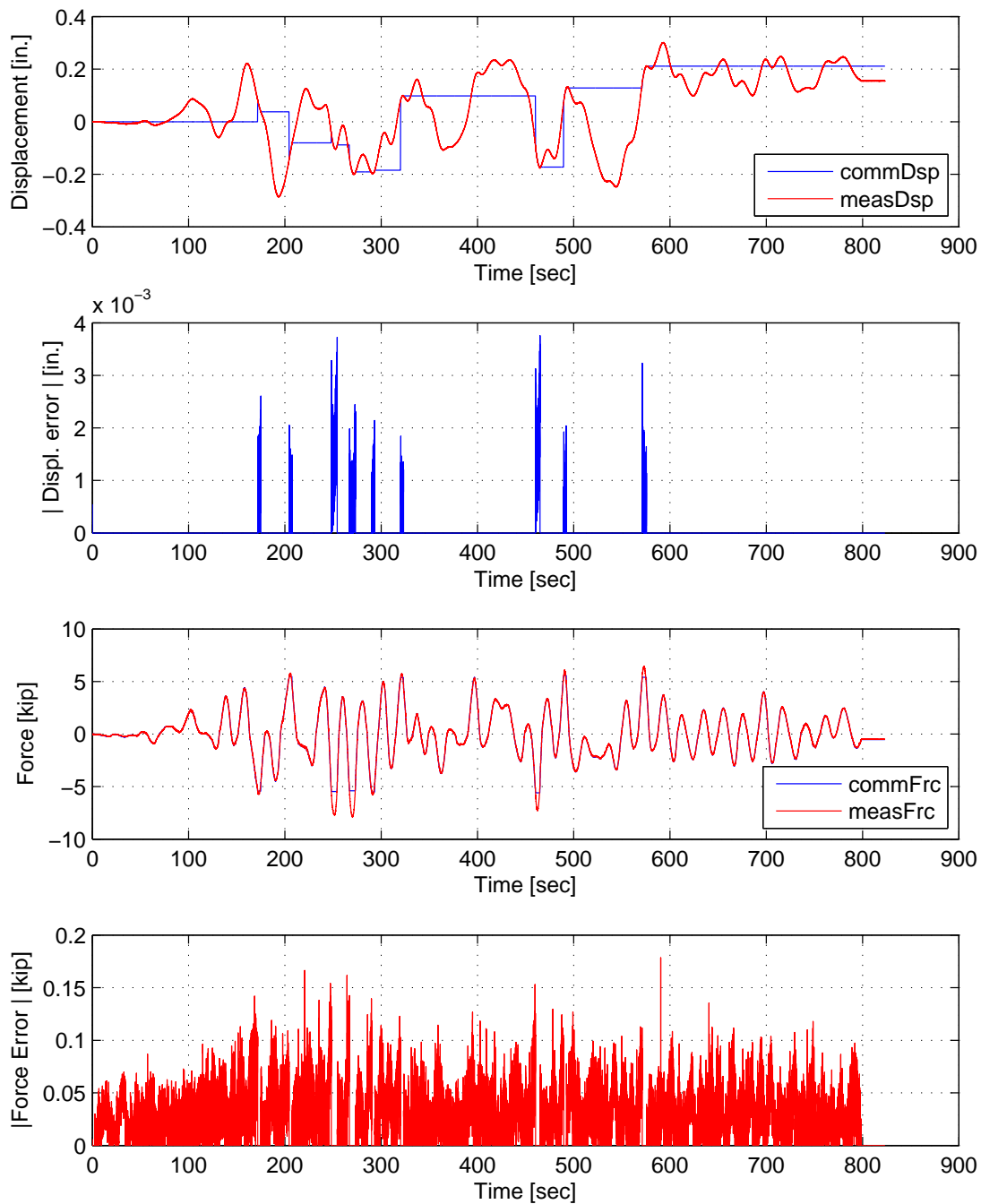


Figure E.7: STS controller recorded values for bottom actuator. Top: Command and measured displacement plot from the STS controller for PMC:Intrinsic hybrid simulation in DC. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for PMC:Intrinsic hybrid simulation in FC. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NME with PMC:Intrinsic.



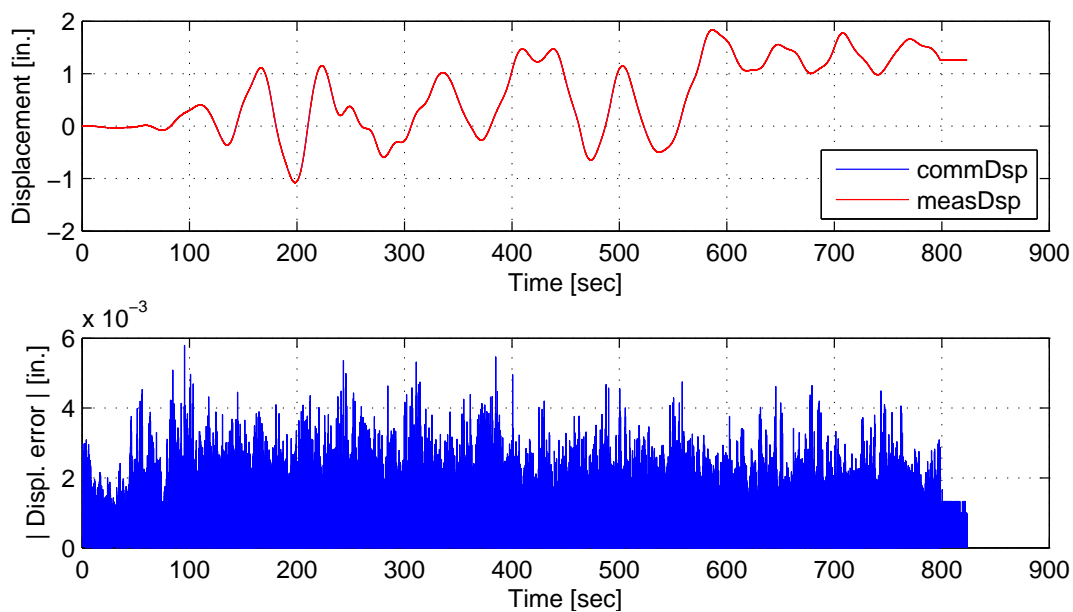


Figure E.8: STS controller recorded values for top actuator. Top: Command and measured displacement plot from the STS controller for PMC:Intrinsic hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NME with PMC:Intrinsic.

### **E.1.2 $\alpha$ time integration scheme**

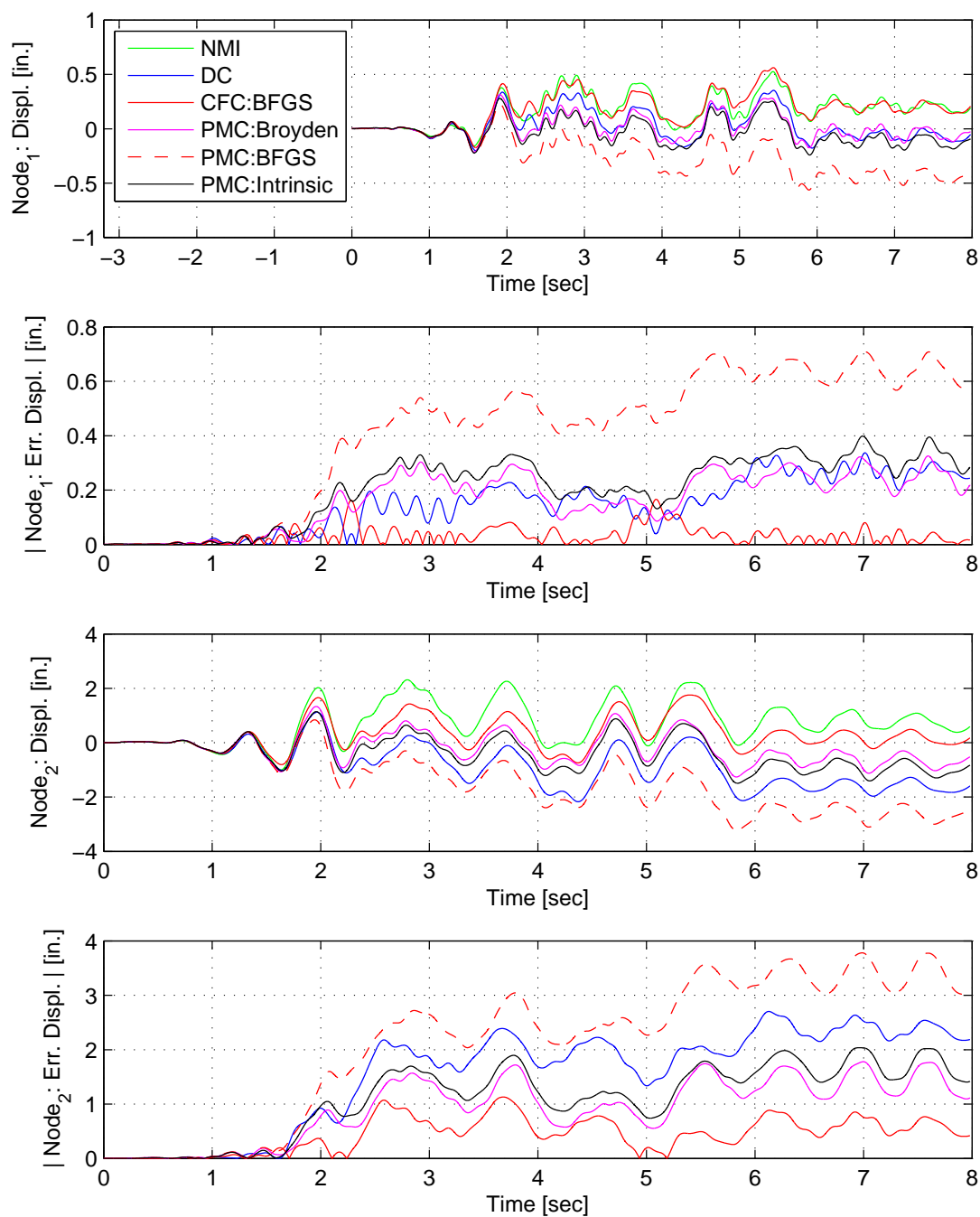


Figure E.9: Node 1 displacement-time history plot (top), Node 1 absolute displacement error (middle-top), Node 2 displacement-time history plot (middle-bottom) and Node 2 absolute displacement error (bottom) for PMC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses  $\alpha$ OS with DC, CFC:BFGS, PMC:Broyden, PMC: BFGS, PMC:Intrinsic. All experimental result errors are calculated using the NMI numerical result.

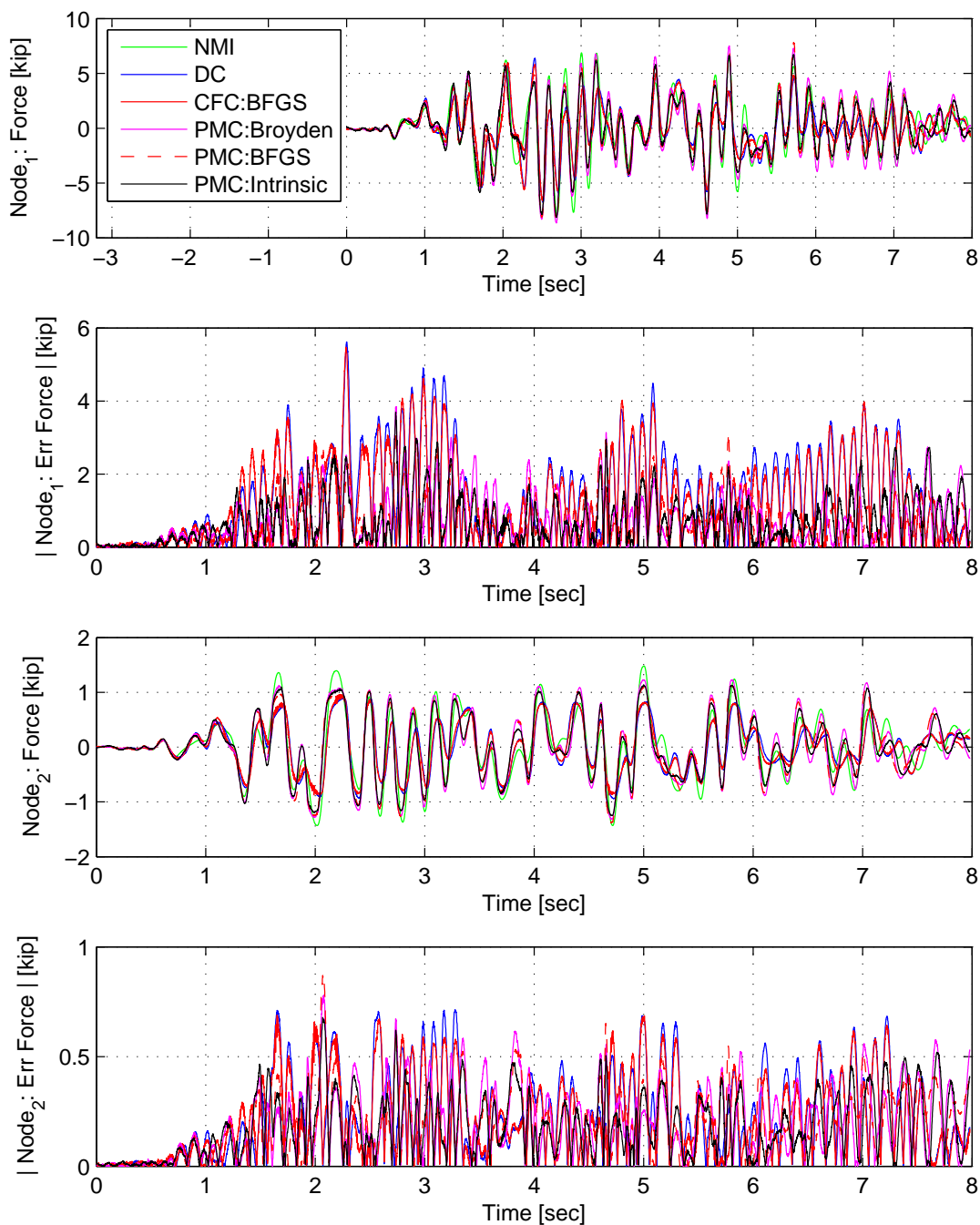


Figure E.10: Node 1 force-time history plot (top), Node 1 absolute force error (middle-top), Node 2 force-time history plot (middle-bottom) and Node 2 absolute force error (bottom) for PMC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses  $\alpha$ OS with DC, CFC:BFGS, PMC:Broyden, PMC: BFGS, PMC:Intrinsic. All experimental result errors are calculated using the NMI numerical result.

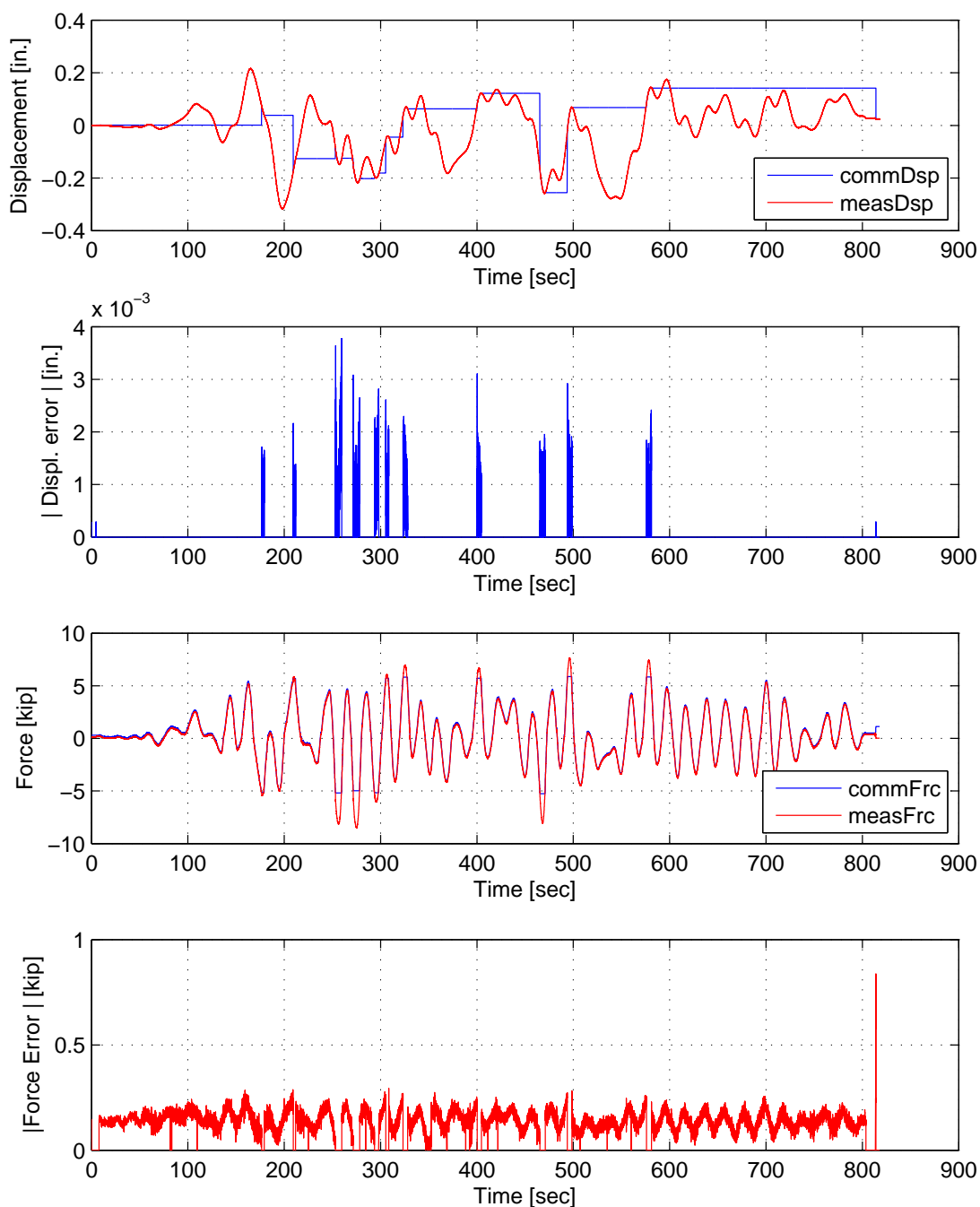


Figure E.11: STS controller recorded values for bottom actuator. Top: Command and measured displacement plot from the STS controller for PMC:Broyden hybrid simulation in DC. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for PMC:Broyden hybrid simulation in FC. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with  $\alpha$ OS with PMC:Broyden.

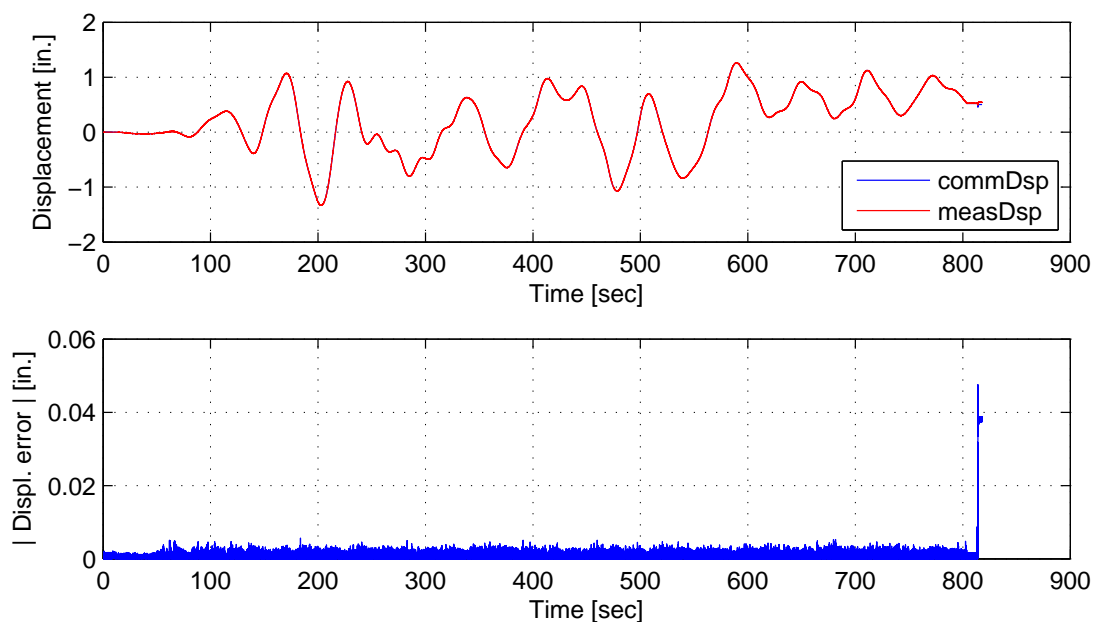


Figure E.12: STS controller recorded values for top actuator. Top: Command and measured displacement plot from the STS controller for PMC:Broyden hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with  $\alpha$ OS with PMC:Broyden.

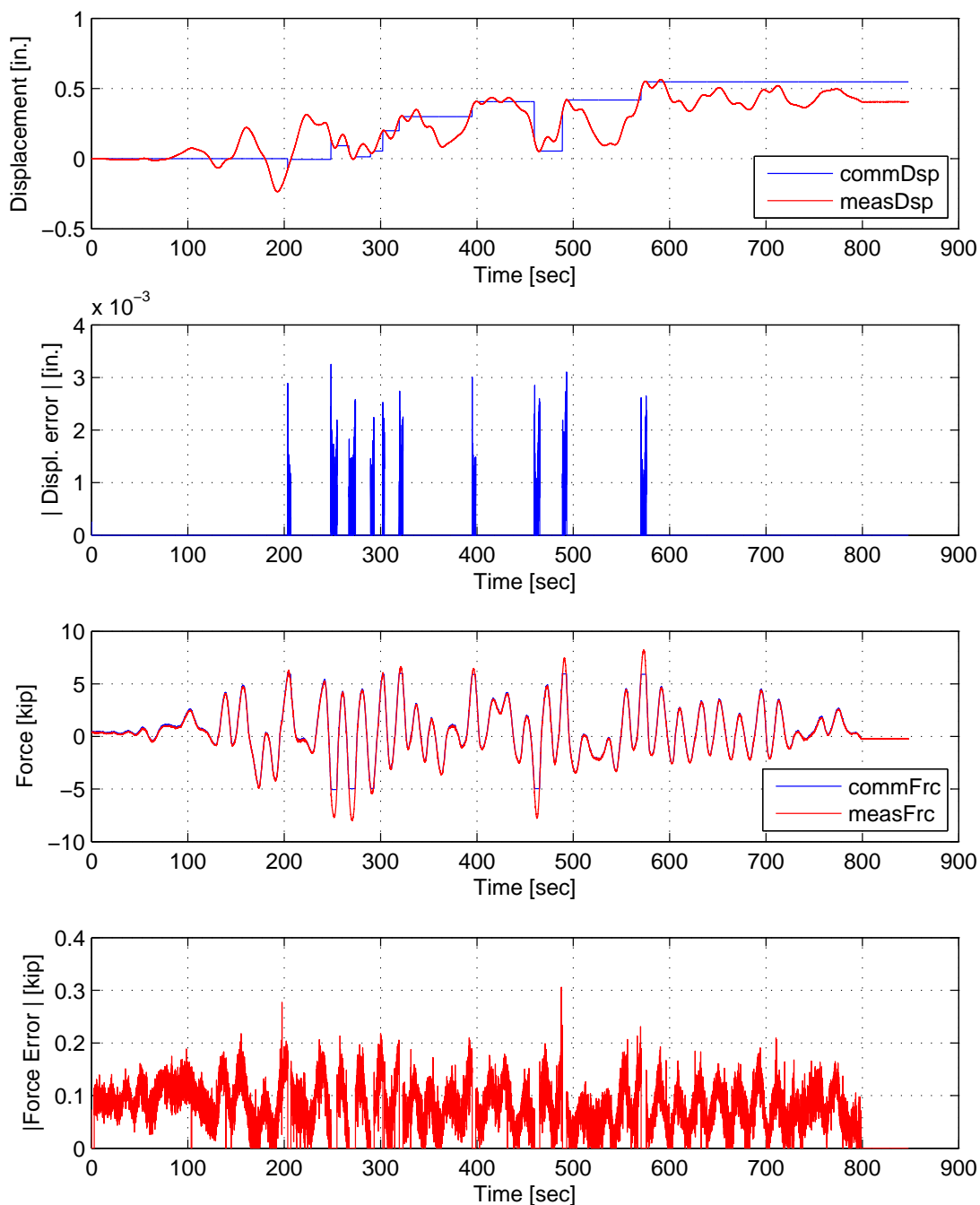


Figure E.13: STS controller recorded values for bottom actuator. Top: Command and measured displacement plot from the STS controller for PMC:BFGS hybrid simulation in DC. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for PMC:BFGS hybrid simulation in FC. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with  $\alpha$ OS with PMC:BFGS.

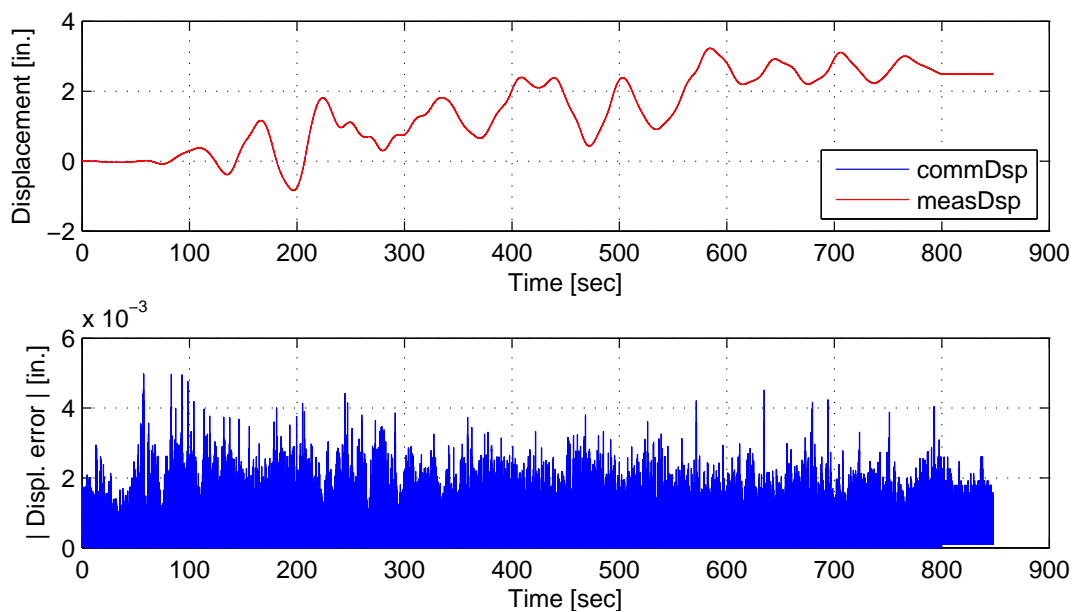


Figure E.14: STS controller recorded values for top actuator. Top: Command and measured displacement plot from the STS controller for PMC:BFGS hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with  $\alpha$ OS with PMC:BFGS.



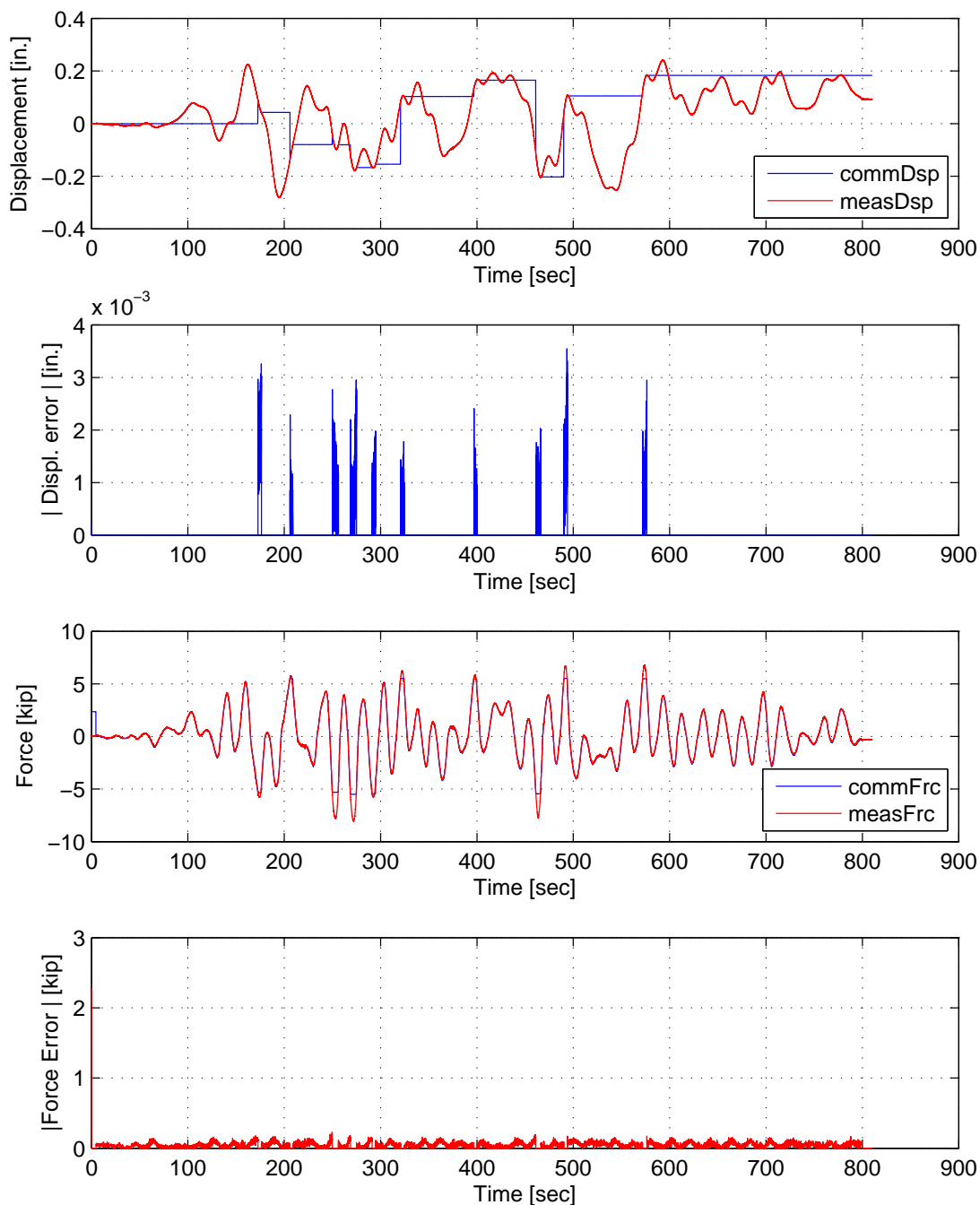


Figure E.15: STS controller recorded values for bottom actuator. Top: Command and measured displacement plot from the STS controller for PMC:Intrinsic hybrid simulation in DC. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for PMC:Intrinsic hybrid simulation in FC. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with  $\alpha$ OS with PMC:Intrinsic.

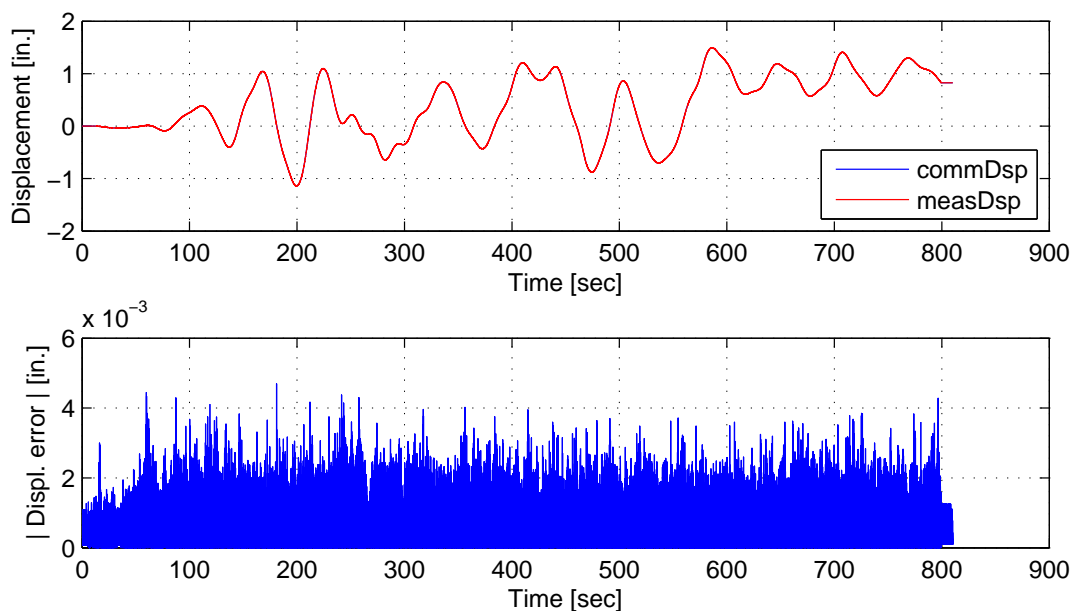


Figure E.16: STS controller recorded values for top actuator. Top: Command and measured displacement plot from the STS controller for PMC:Intrinsic hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with  $\alpha$ OS with PMC:Intrinsic.

### **E.1.3 NMF time integration scheme**

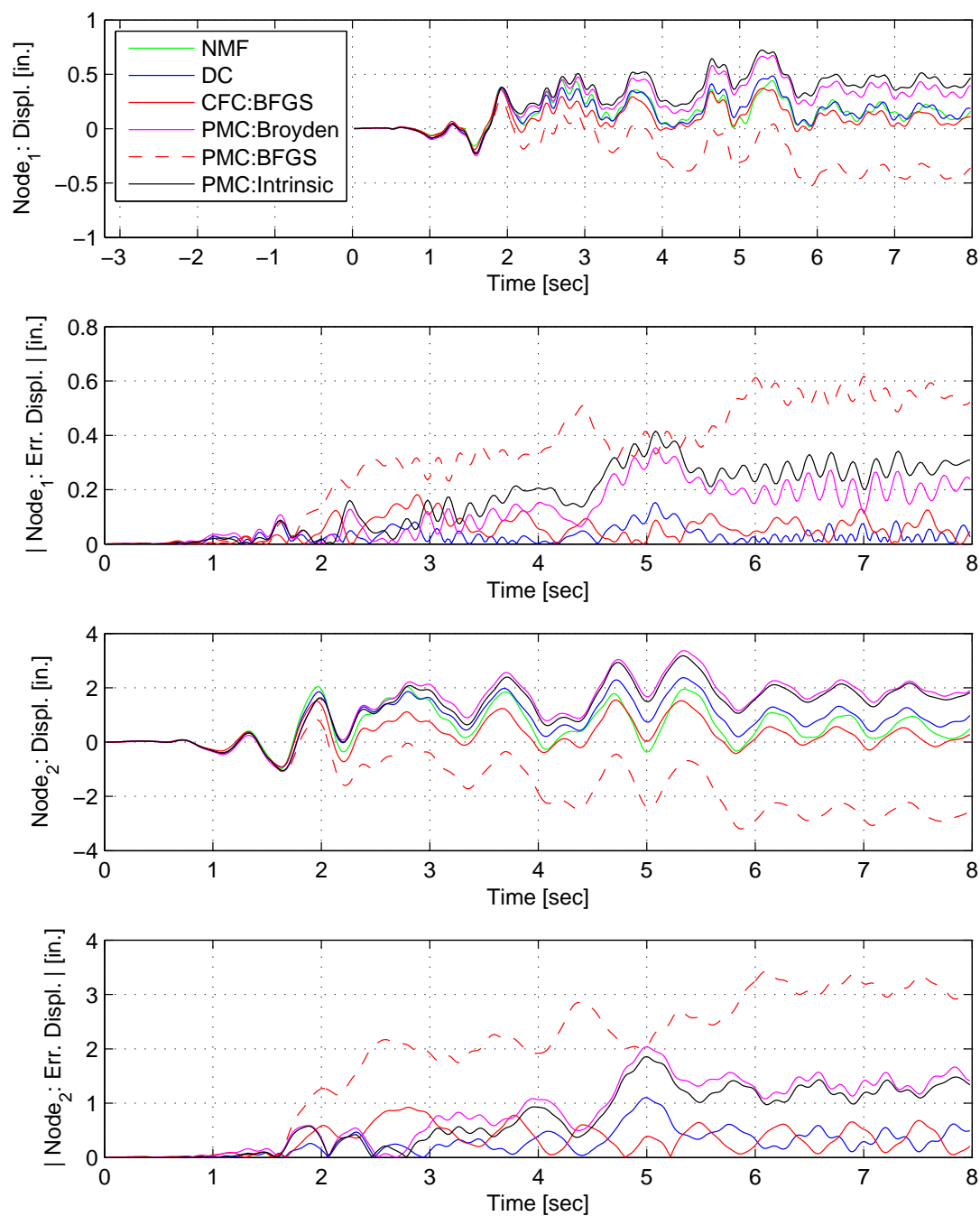


Figure E.17: Node 1 displacement-time history plot (top), Node 1 absolute displacement error (middle-top), Node 2 displacement-time history plot (middle-bottom) and Node 2 absolute displacement error (bottom) for PMC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses NMF with DC, CFC:BFGS, PMC:Broyden, PMC: BFGS, PMC:Intrinsic. All experimental result errors are calculated using the NMI numerical result.

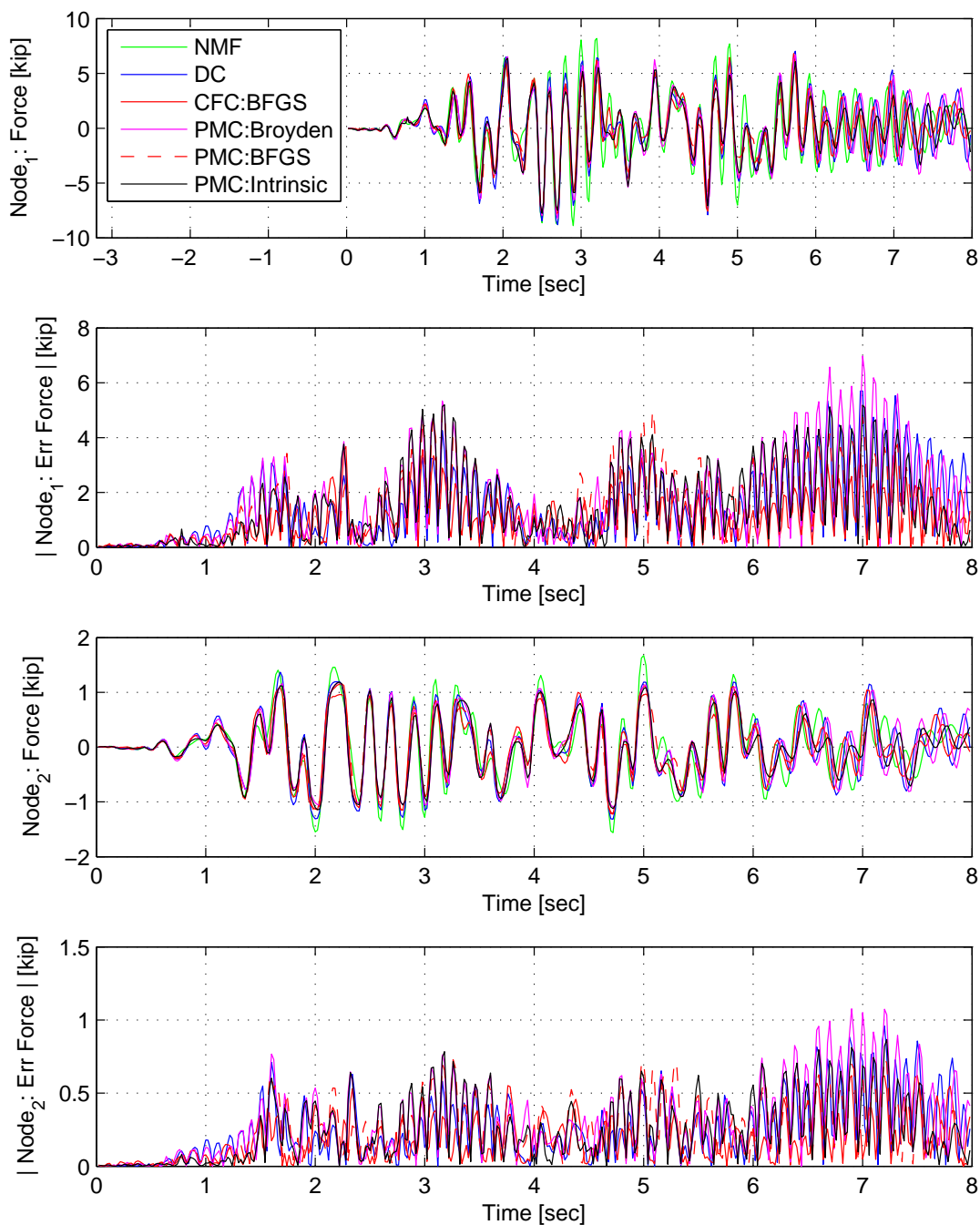


Figure E.18: Node 1 force-time history plot (top), Node 1 absolute force error (middle-top), Node 2 force-time history plot (middle-bottom) and Node 2 absolute force error (bottom) for PMC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses NMF with DC, CFC:BFGS, PMC:Broyden, PMC: BFGS, PMC:Intrinsic. All experimental result errors are calculated using the NMI numerical result.

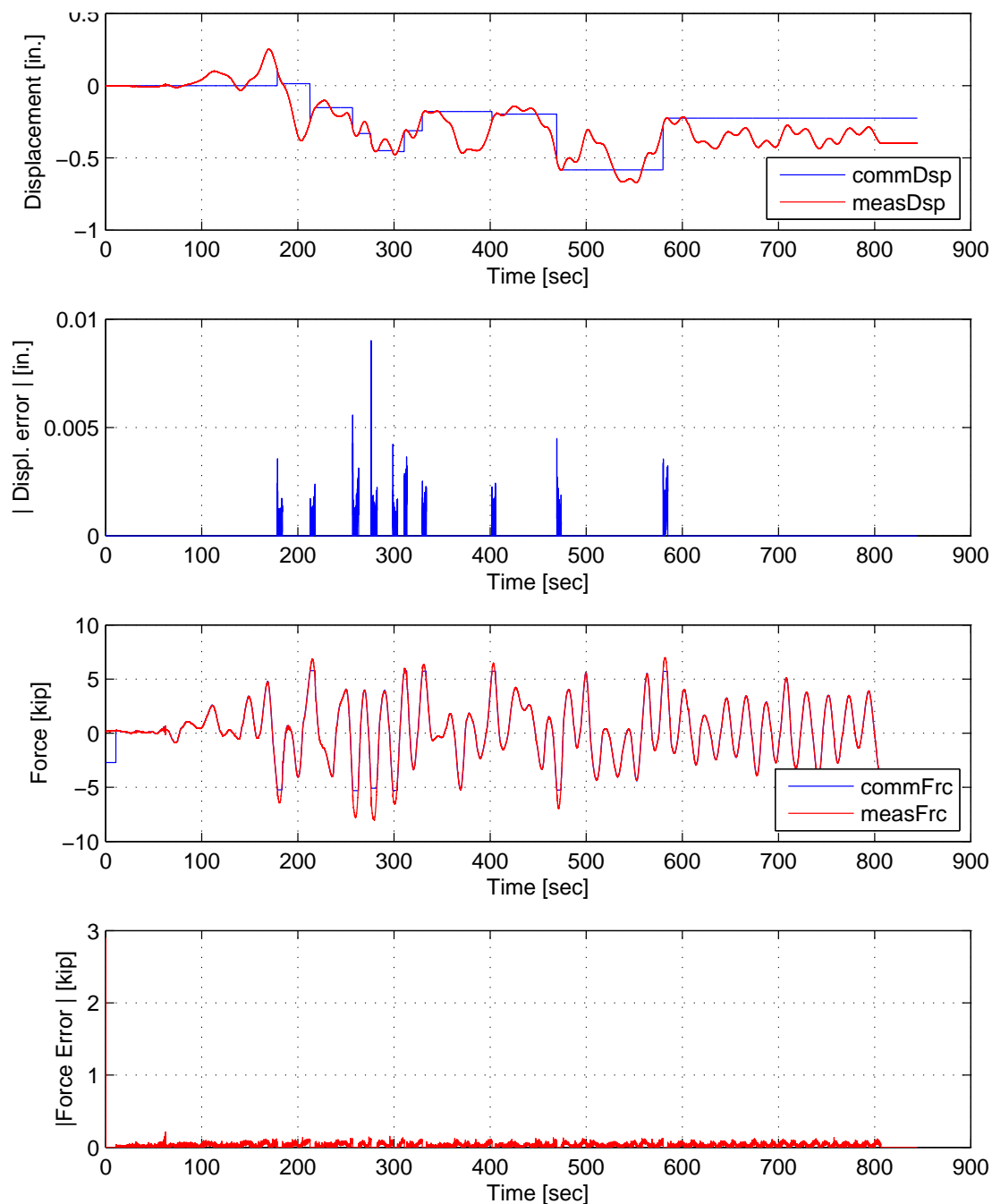


Figure E.19: STS controller recorded values for bottom actuator. Top: Command and measured displacement plot from the STS controller for PMC:Broyden hybrid simulation in DC. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for PMC:Broyden hybrid simulation in FC. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NMF with PMC:Broyden.

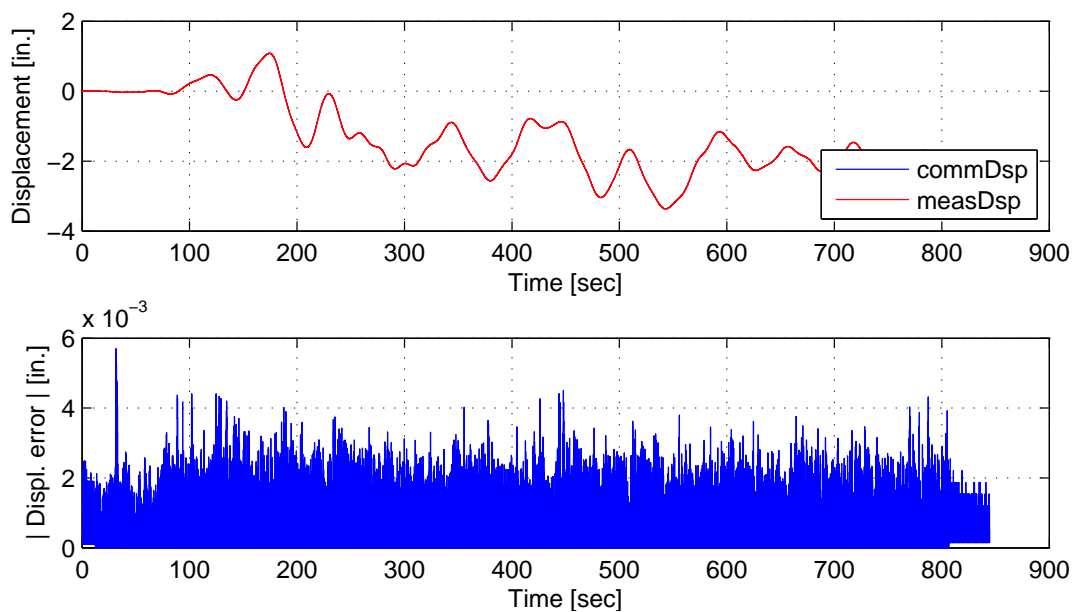


Figure E.20: STS controller recorded values for top actuator. Top: Command and measured displacement plot from the STS controller for PMC:Broyden hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NMF with PMC:Broyden.

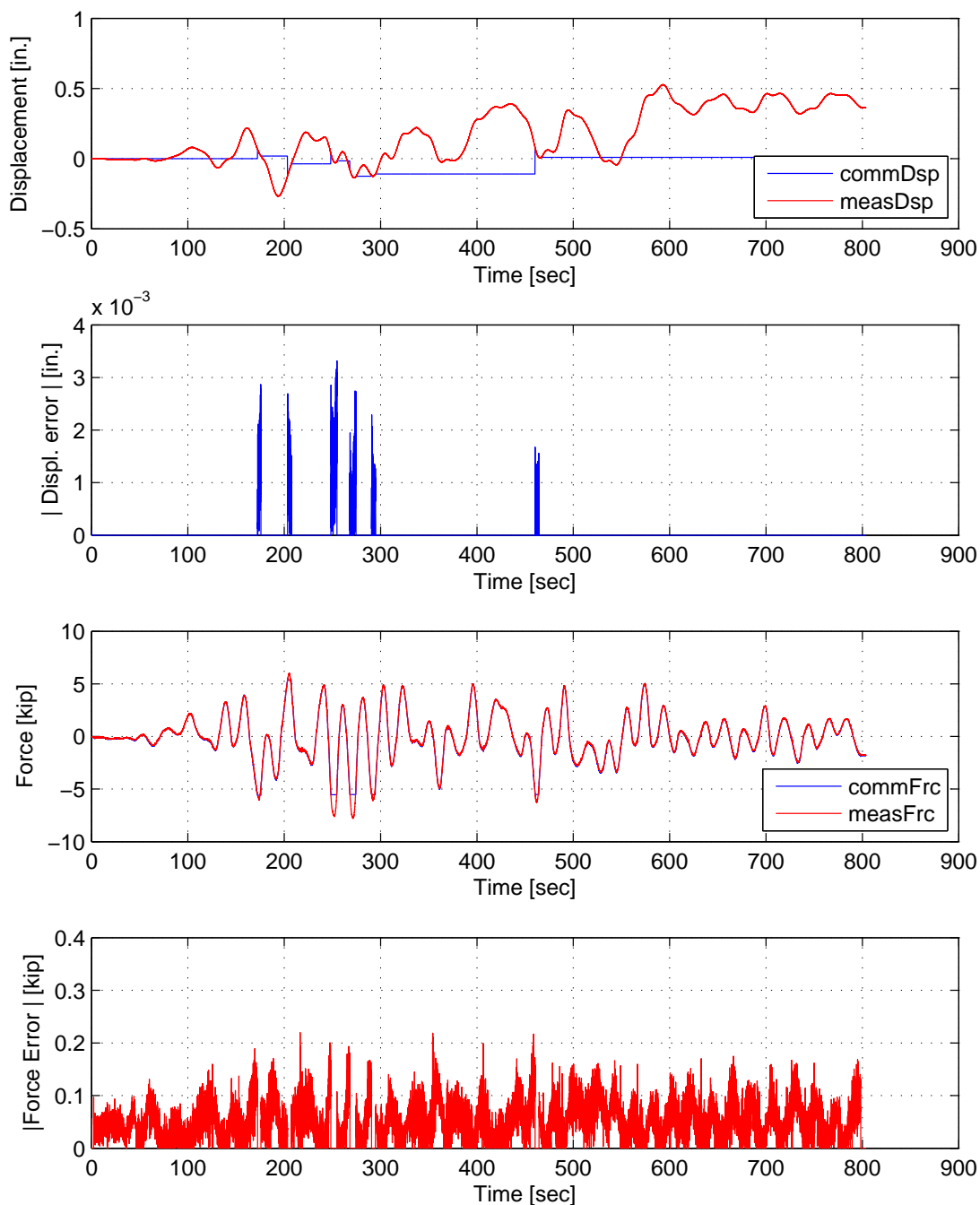


Figure E.21: STS controller recorded values for bottom actuator. Top: Command and measured displacement plot from the STS controller for PMC:BFGS hybrid simulation in DC. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for PMC:BFGS hybrid simulation in FC. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NMF with PMC:BFGS.



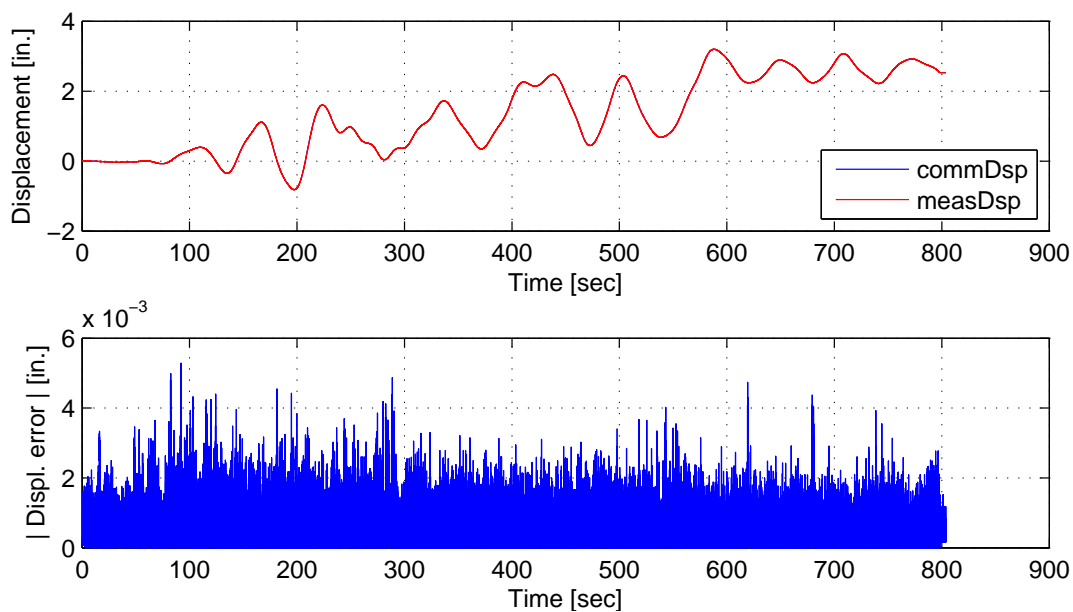


Figure E.22: STS controller recorded values for top actuator. Top: Command and measured displacement plot from the STS controller for PMC:BFGS hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NMF with PMC:BFGS.

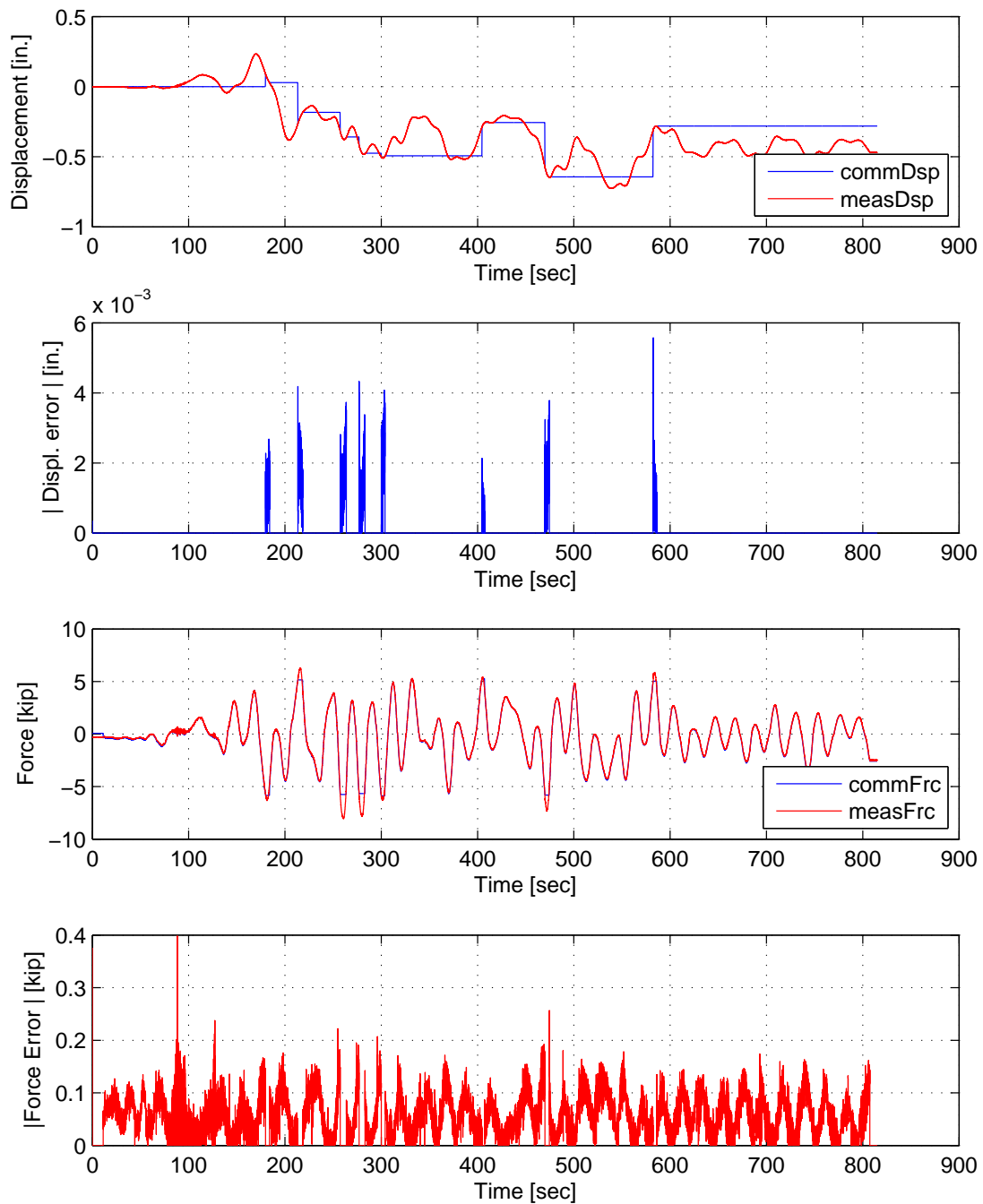


Figure E.23: STS controller recorded values for bottom actuator. Top: Command and measured displacement plot from the STS controller for PMC:Intrinsic hybrid simulation in DC. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for PMC:Intrinsic hybrid simulation in FC. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NMF with PMC:Intrinsic.

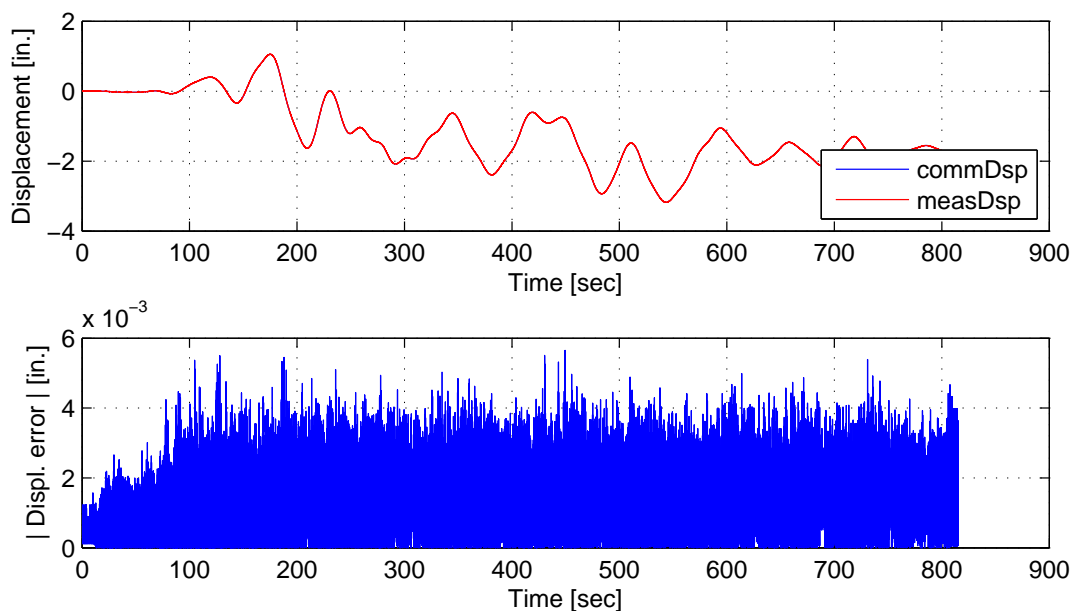


Figure E.24: STS controller recorded values for top actuator. Top: Command and measured displacement plot from the STS controller for PMC:Intrinsic hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NMF with PMC:Intrinsic.

## **E.1.4 NMR time integration scheme**

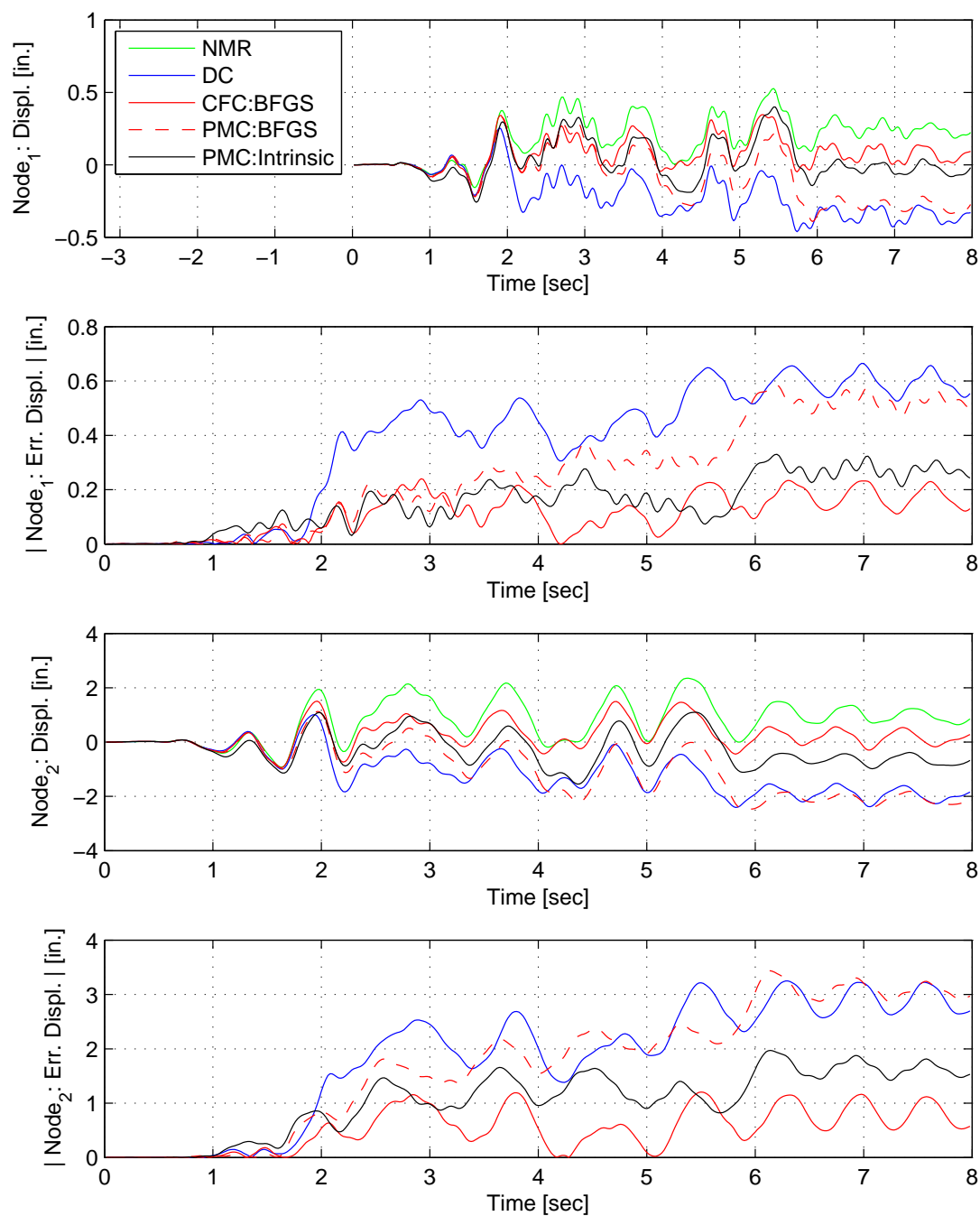


Figure E.25: Node 1 displacement-time history plot (top), Node 1 absolute displacement error (middle-top), Node 2 displacement-time history plot (middle-bottom) and Node 2 absolute displacement error (bottom) for PMC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses NMR with DC, CFC:BFGS, PMC: BFGS, PMC:Intrinsic. All experimental result errors are calculated using the NMI numerical result.

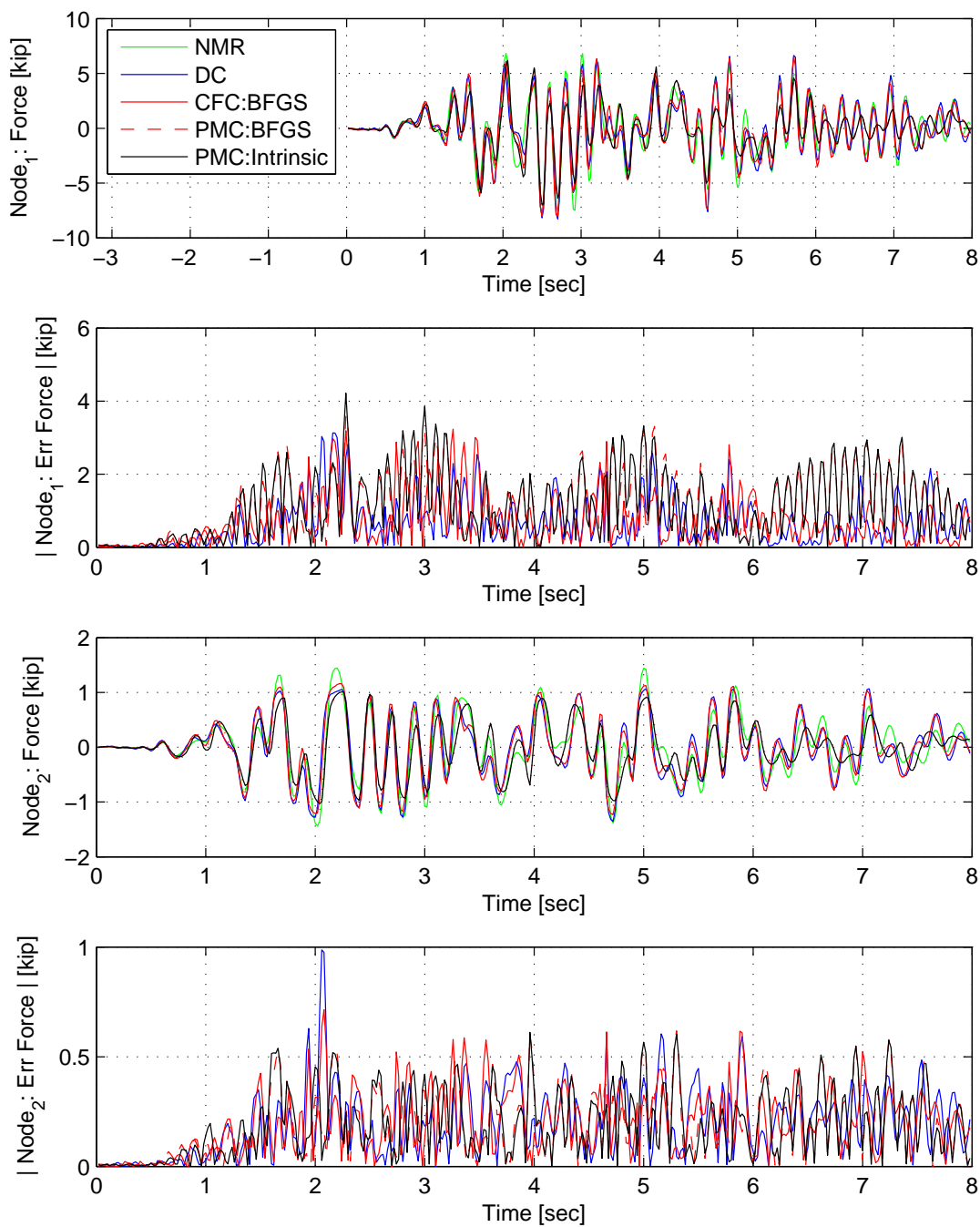


Figure E.26: Node 1 force-time history plot (top), Node 1 absolute force error (middle-top), Node 2 force-time history plot (middle-bottom) and Node 2 absolute force error (bottom) for PMC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses NMR with DC, CFC:BFGS, PMC: BFGS, PMC:Intrinsic. All experimental result errors are calculated using the NMI numerical result.

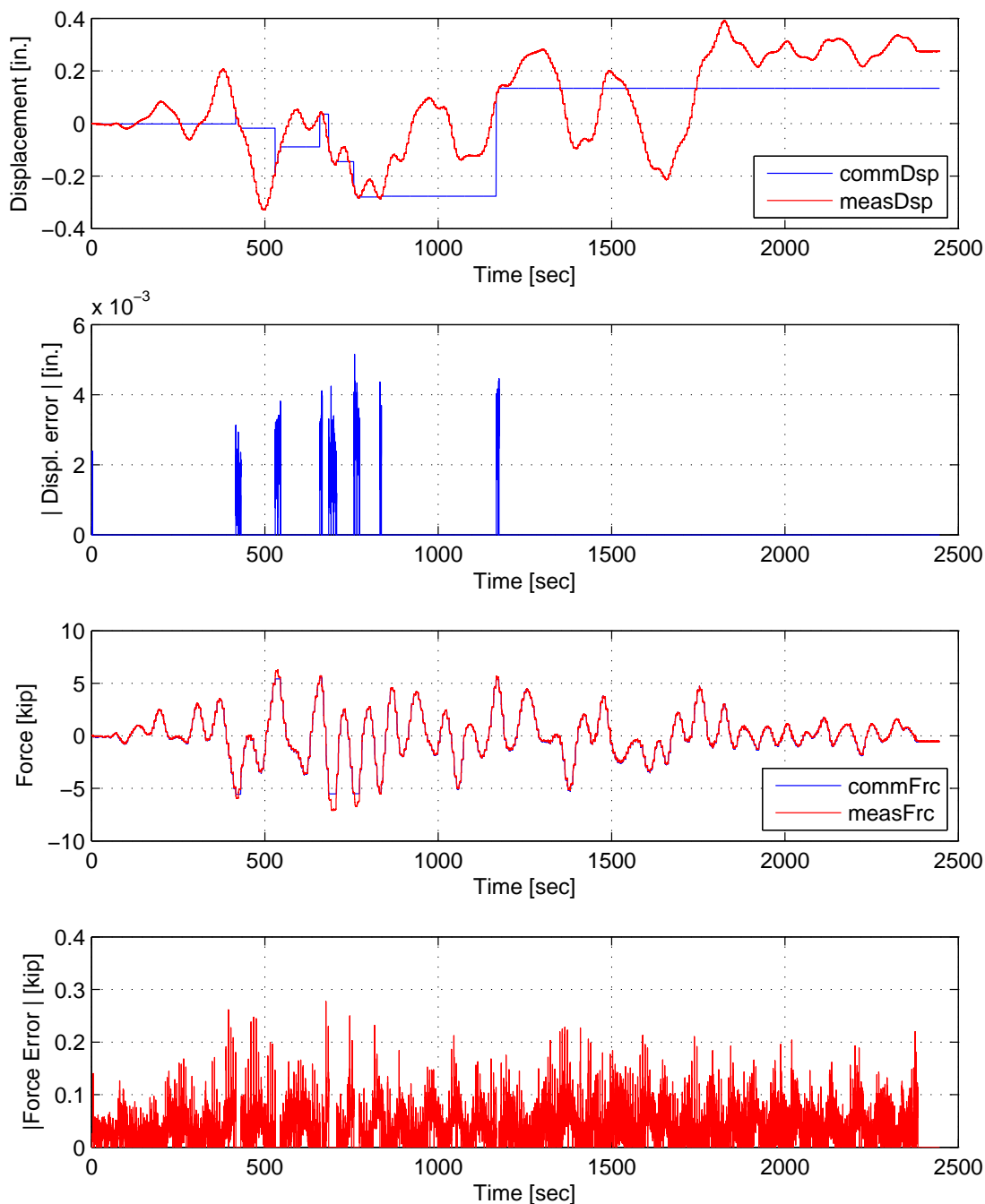


Figure E.27: STS controller recorded values for bottom actuator. Top: Command and measured displacement plot from the STS controller for PMC:BFGS hybrid simulation in DC. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for PMC:BFGS hybrid simulation in FC. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NMR with PMC:BFGS.

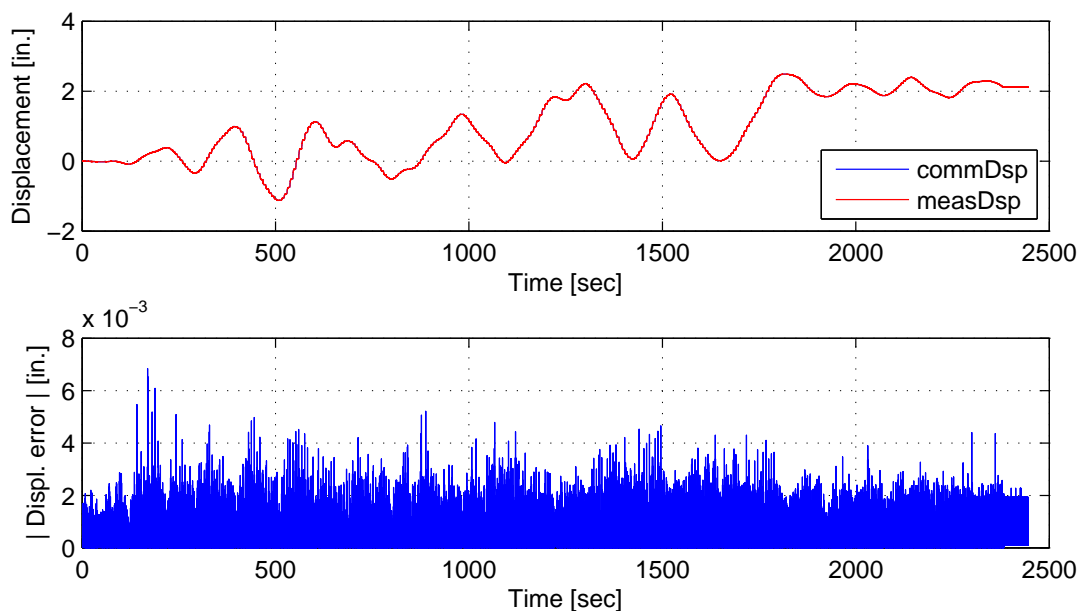


Figure E.28: STS controller recorded values for top actuator. Top: Command and measured displacement plot from the STS controller for PMC:BFGS hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NMR with PMC:BFGS.



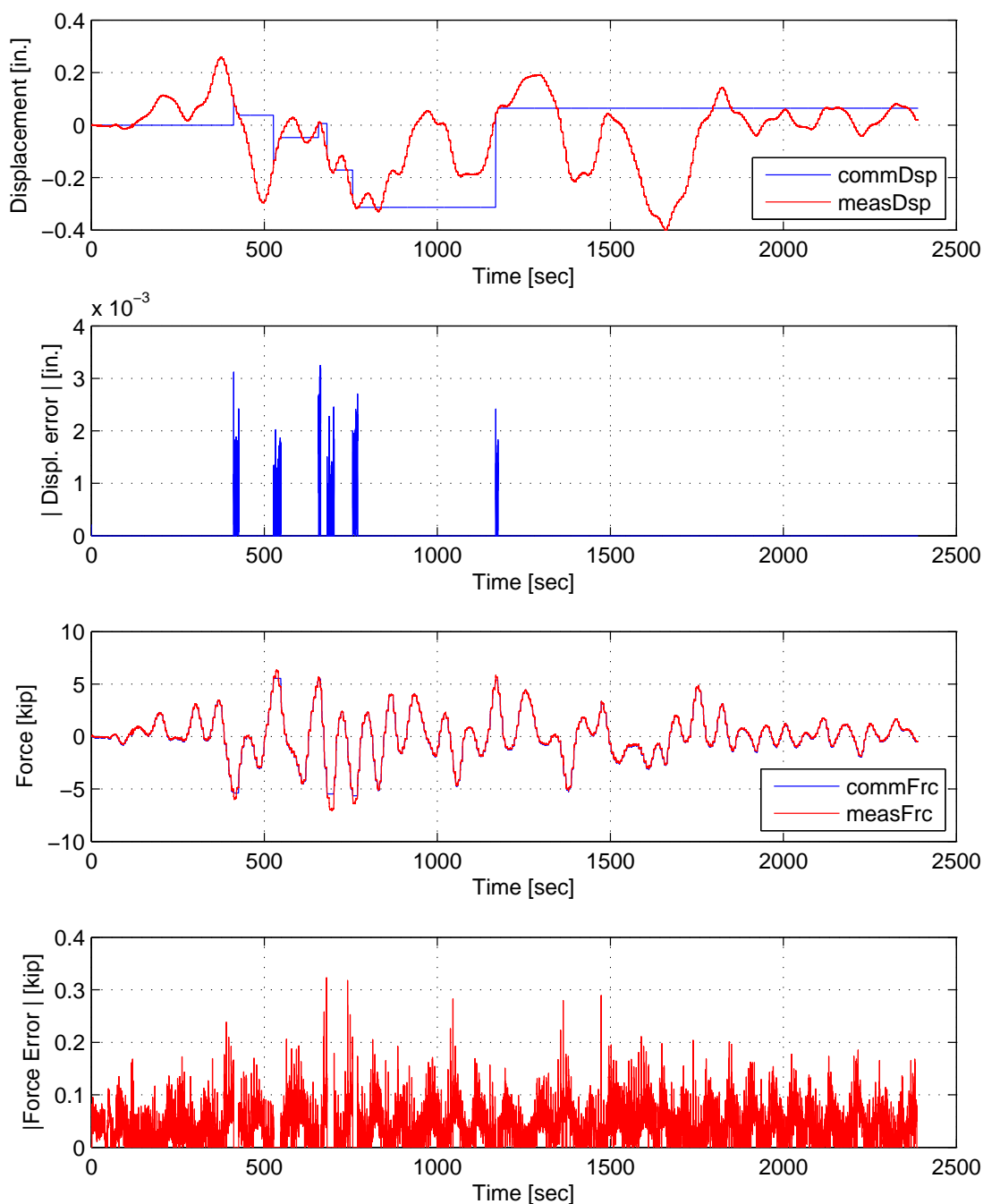


Figure E.29: STS controller recorded values for bottom actuator. Top: Command and measured displacement plot from the STS controller for PMC:Intrinsic hybrid simulation in DC. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for PMC:Intrinsic hybrid simulation in FC. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NMR with PMC:Intrinsic.

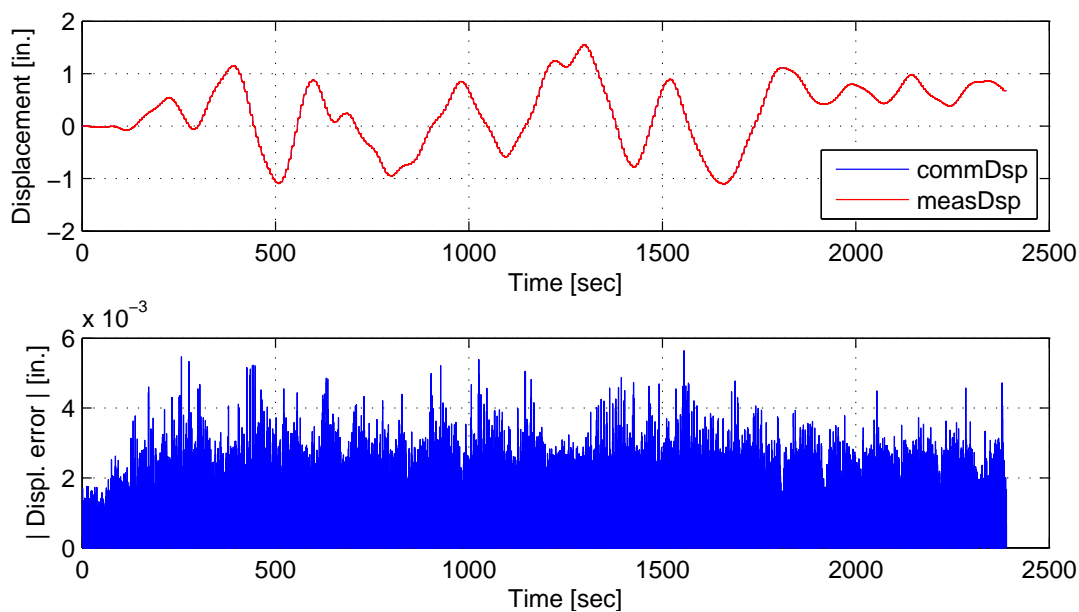


Figure E.30: STS controller recorded values for top actuator. Top: Command and measured displacement plot from the STS controller for PMC:Intrinsic hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NMR with PMC:Intrinsic.

## **E.2 Secant switching hybrid simulation**

### **E.2.1 NME time integration scheme**

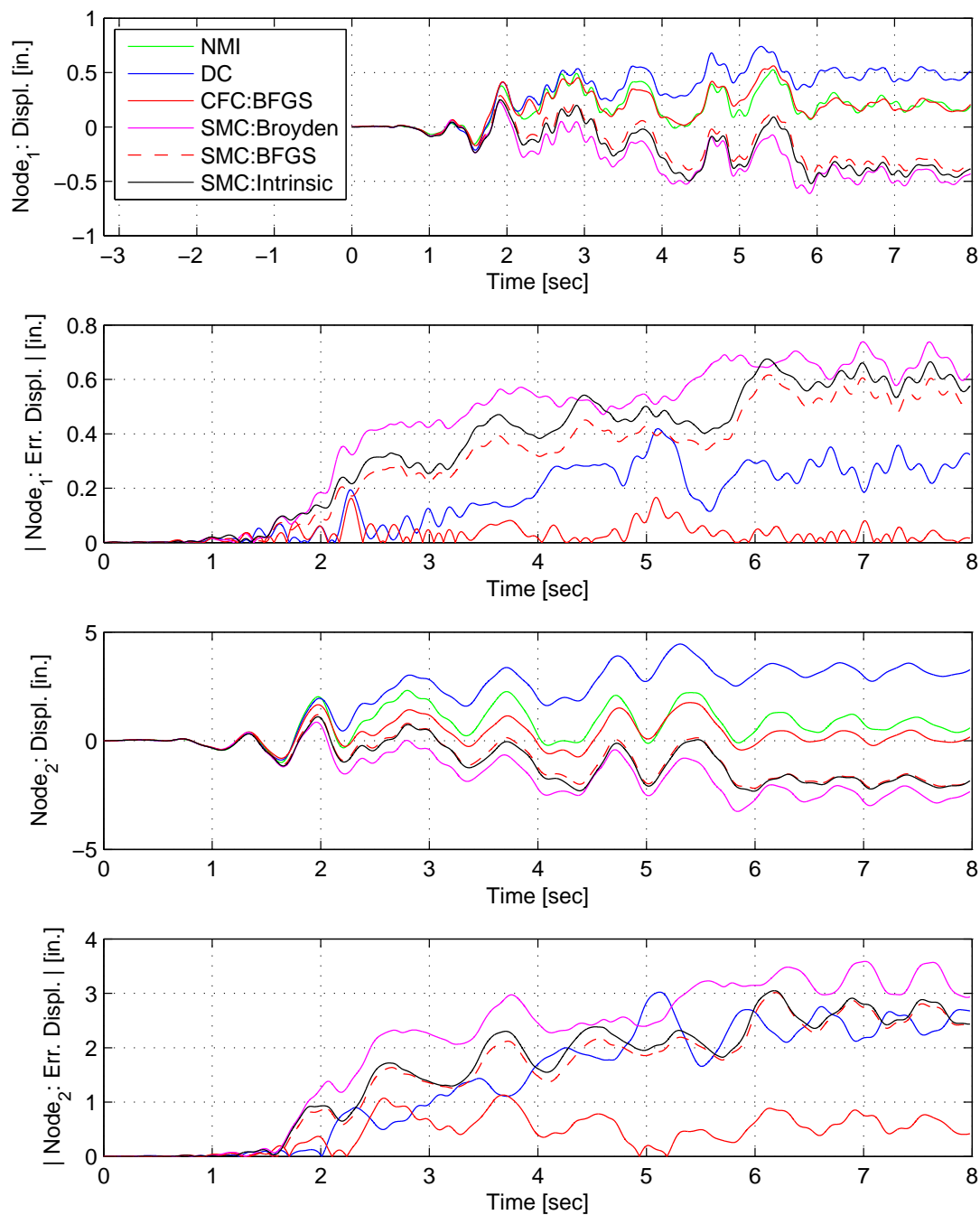


Figure E.31: Node 1 displacement-time history plot (top), Node 1 absolute displacement error (middle-top), Node 2 displacement-time history plot (middle-bottom) and Node 2 absolute displacement error (bottom) for SMC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses NME with DC, CFC:BFGS, SMC:Broyden, SMC: BFGS, SMC:Intrinsic. All experimental result errors are calculated using the NMI numerical result.

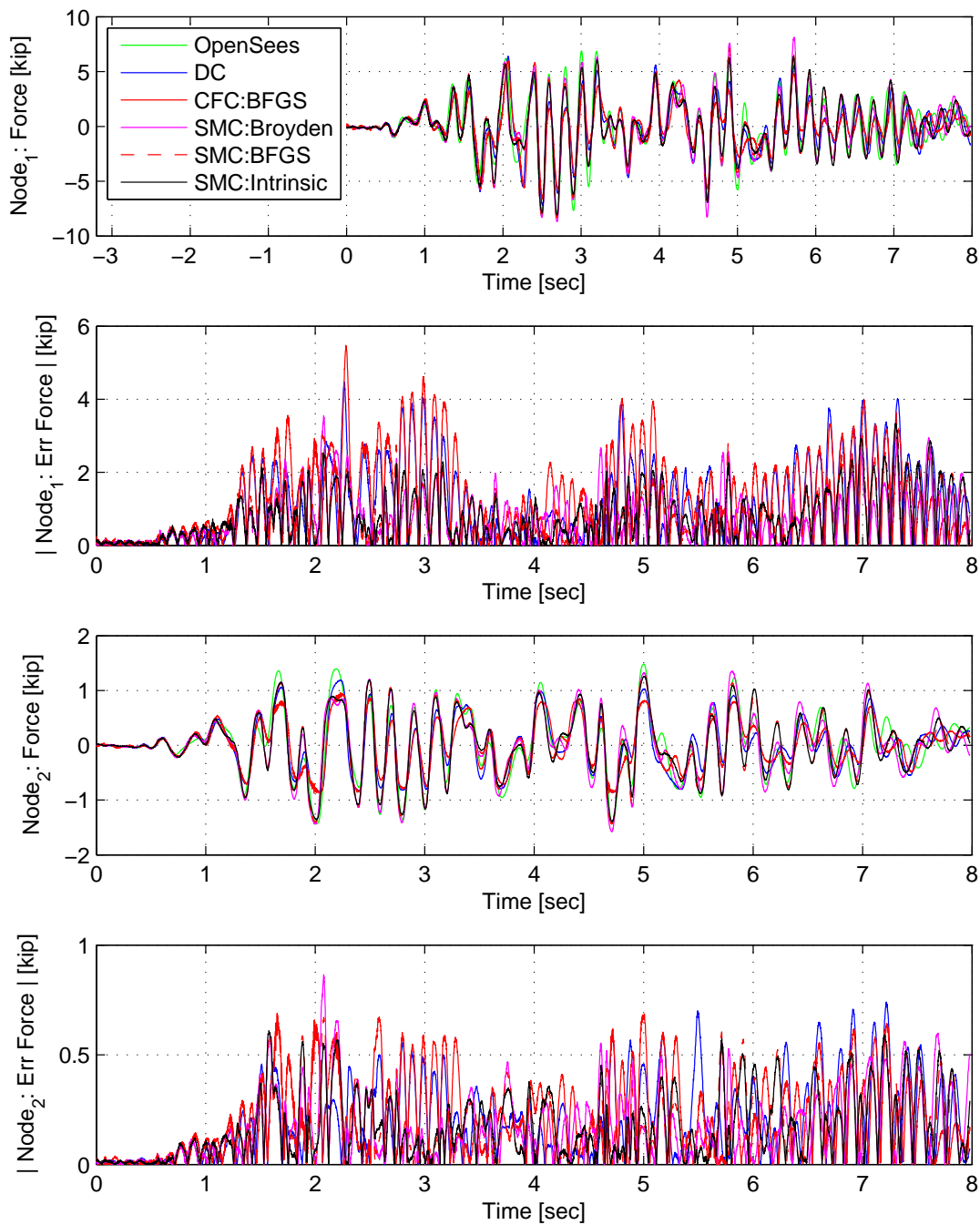


Figure E.32: Node 1 force-time history plot (top), Node 1 absolute force error (middle-top), Node 2 force-time history plot (middle-bottom) and Node 2 absolute force error (bottom) for SMC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses NME with DC, CFC:BFGS, SMC:Broyden, SMC: BFGS, SMC:Intrinsic. All experimental result errors are calculated using the NMI numerical result.

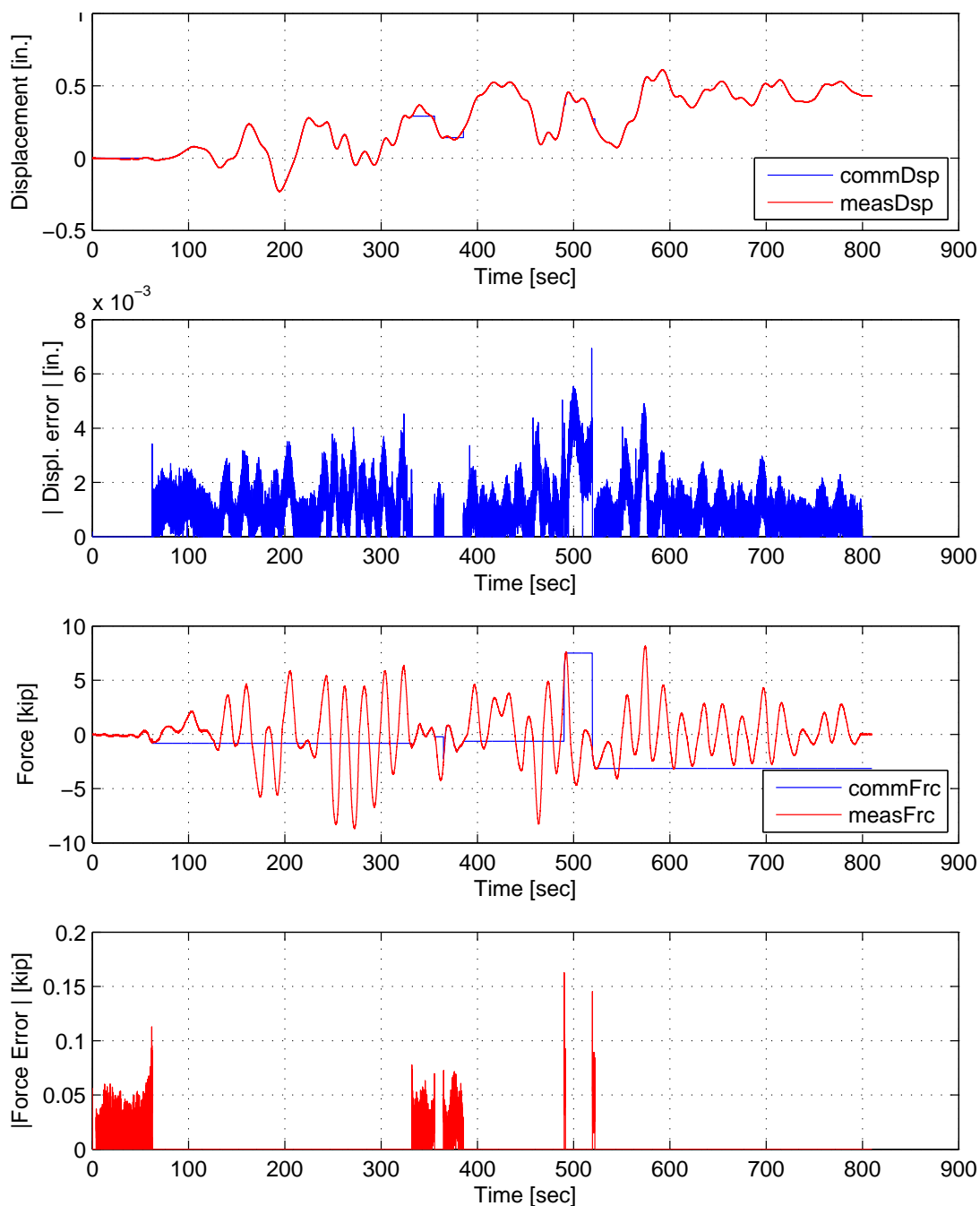


Figure E.33: STS controller recorded values for bottom actuator. Top: Command and measured displacement plot from the STS controller for SMC:Broyden hybrid simulation in DC. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for SMC:Broyden hybrid simulation in FC. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NME with SMC:Broyden.

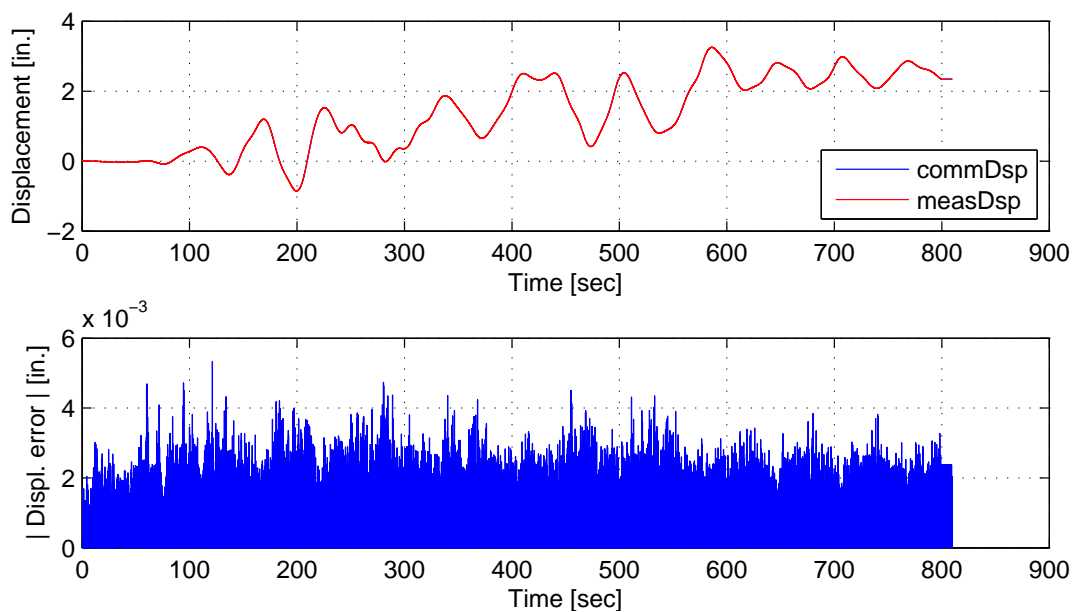


Figure E.34: STS controller recorded values for top actuator. Top: Command and measured displacement plot from the STS controller for SMC:Broyden hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NME with SMC:Broyden.

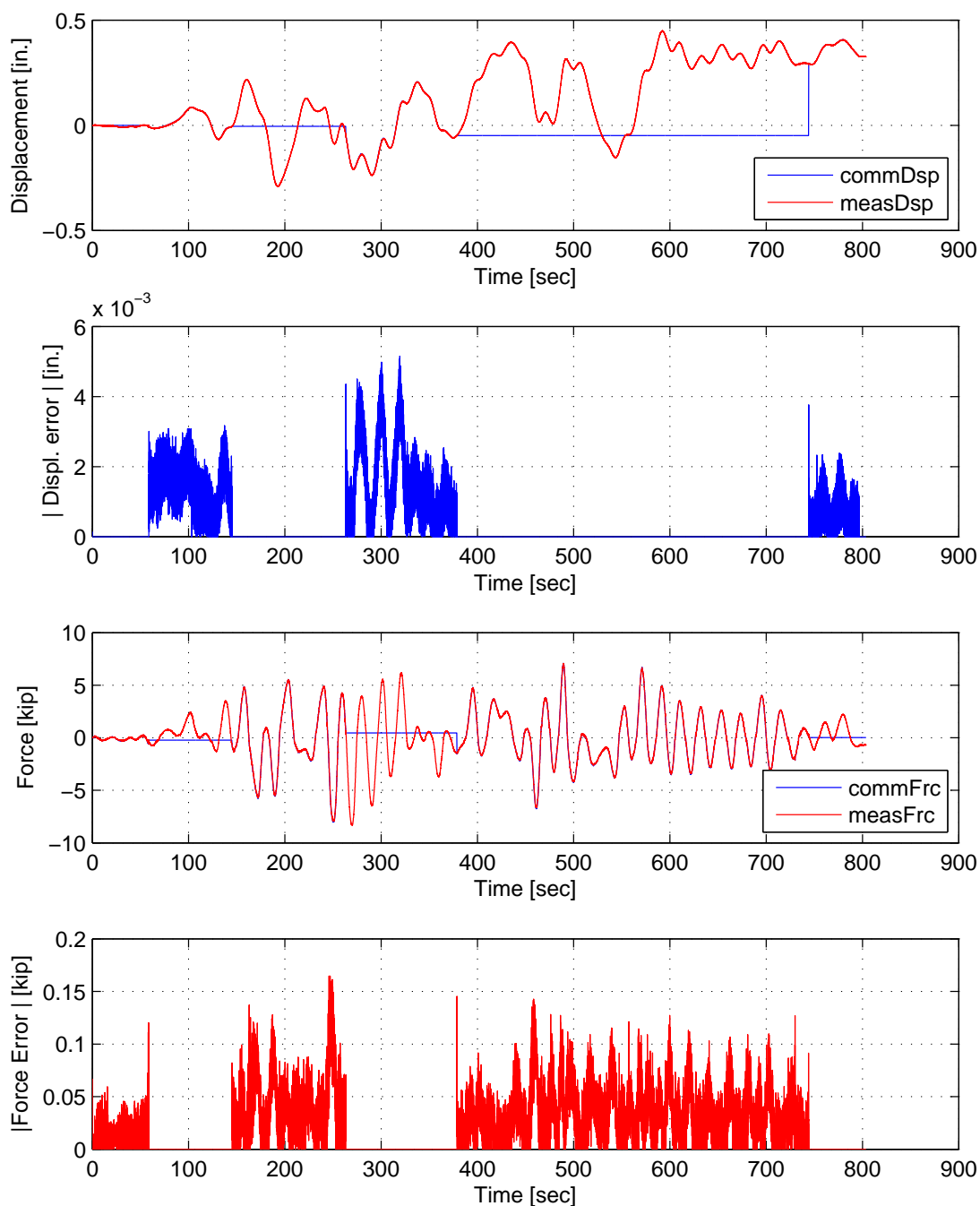


Figure E.35: STS controller recorded values for bottom actuator. Top: Command and measured displacement plot from the STS controller for SMC:BFGS hybrid simulation in DC. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for SMC:BFGS hybrid simulation in FC. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NME with SMC:BFGS.



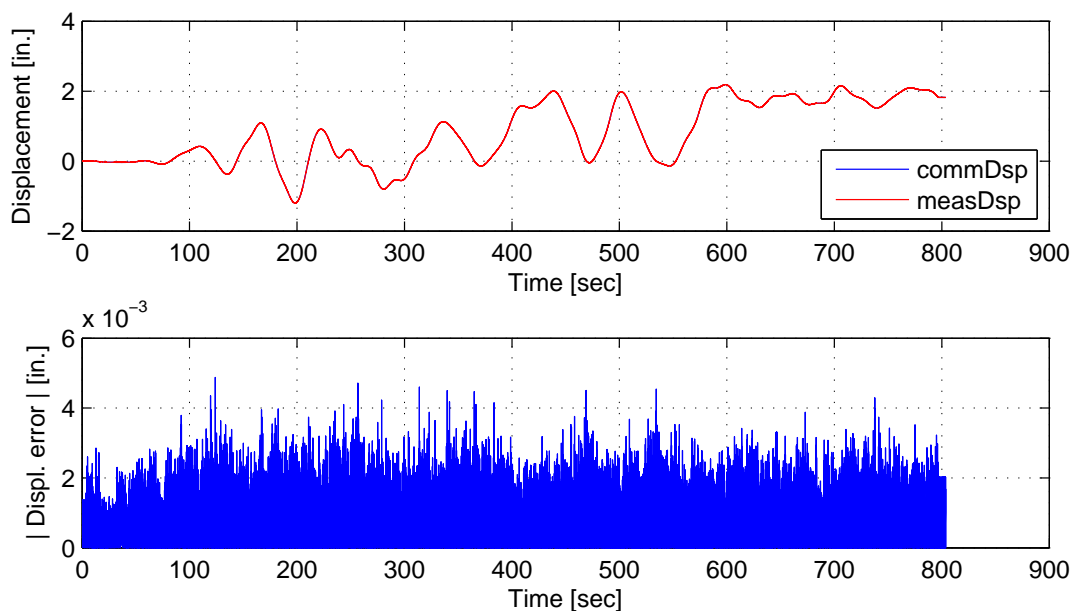


Figure E.36: STS controller recorded values for top actuator. Top: Command and measured displacement plot from the STS controller for SMC:BFGS hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NME with SMC:BFGS.

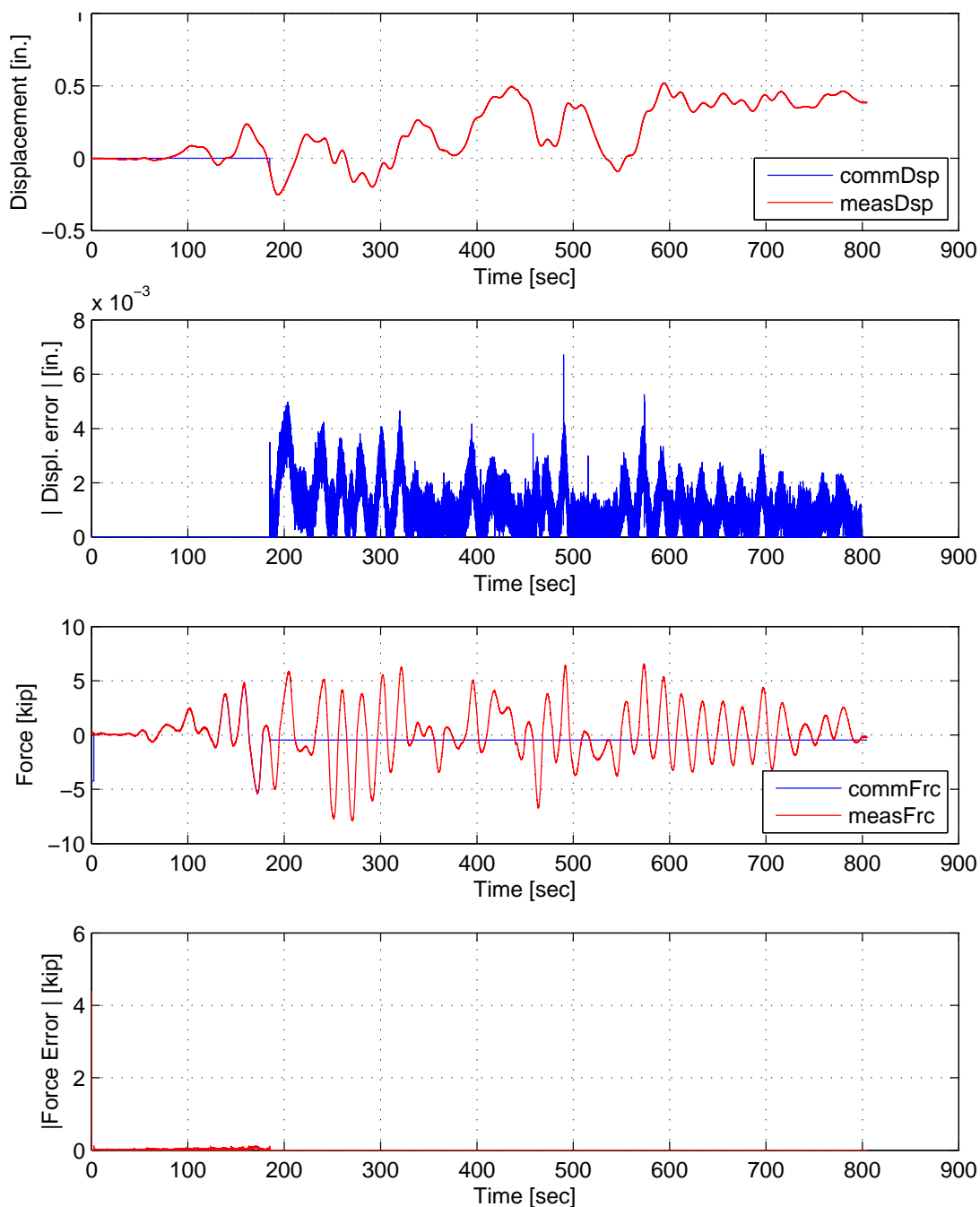


Figure E.37: STS controller recorded values for bottom actuator. Top: Command and measured displacement plot from the STS controller for SMC:Intrinsic hybrid simulation in DC. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for SMC:Intrinsic hybrid simulation in FC. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NME with SMC:Intrinsic.

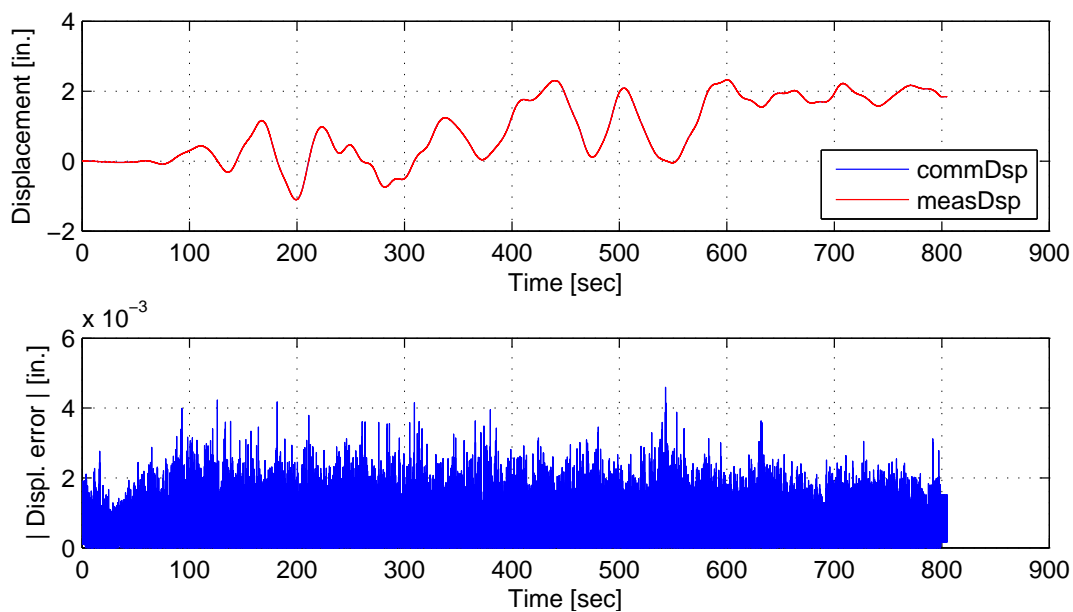


Figure E.38: STS controller recorded values for top actuator. Top: Command and measured displacement plot from the STS controller for SMC:Intrinsic hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NME with SMC:Intrinsic.

## E.2.2 $\alpha$ time integration scheme

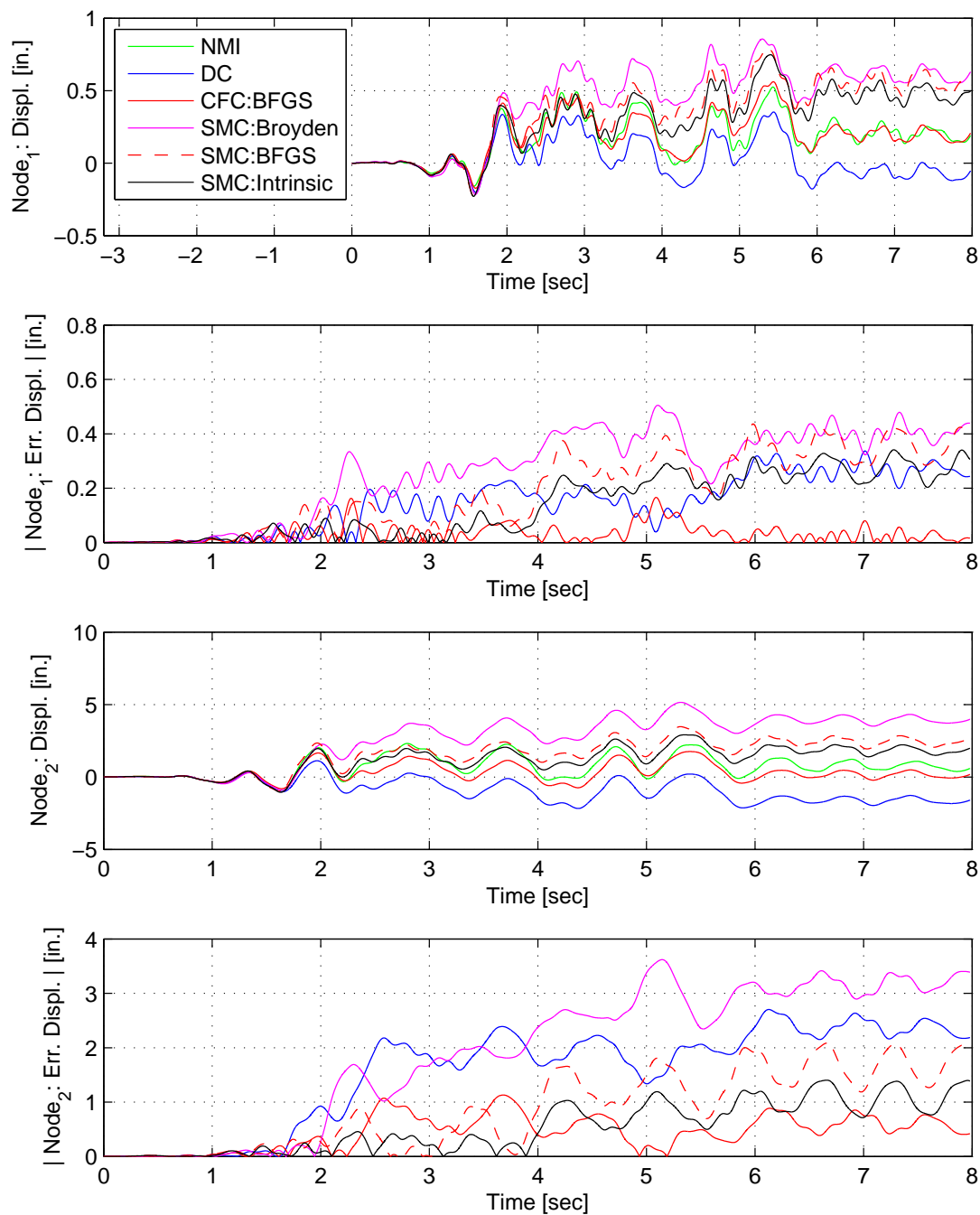


Figure E.39: Node 1 displacement-time history plot (top), Node 1 absolute displacement error (middle-top), Node 2 displacement-time history plot (middle-bottom) and Node 2 absolute displacement error (bottom) for SMC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses  $\alpha$ OS with DC, CFC:BFGS, SMC:Broyden, SMC: BFGS, SMC:Intrinsic. All experimental result errors are calculated using the NMI numerical result.

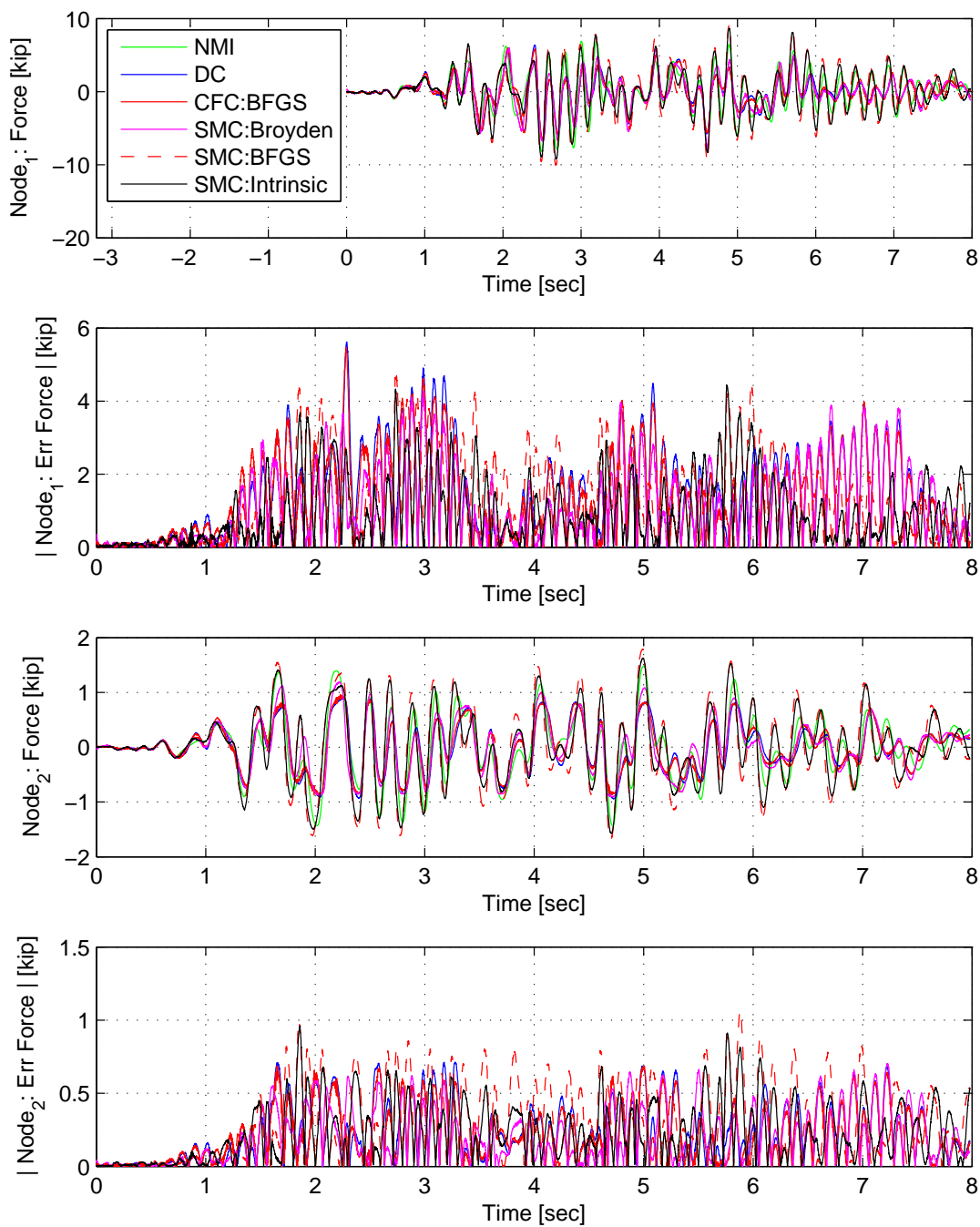


Figure E.40: Node 1 force-time history plot (top), Node 1 absolute force error (middle-top), Node 2 force-time history plot (middle-bottom) and Node 2 absolute force error (bottom) for SMC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses  $\alpha$ OS with DC, CFC:BFGS, SMC:Broyden, SMC: BFGS, SMC:Intrinsic. All experimental result errors are calculated using the NMI numerical result.

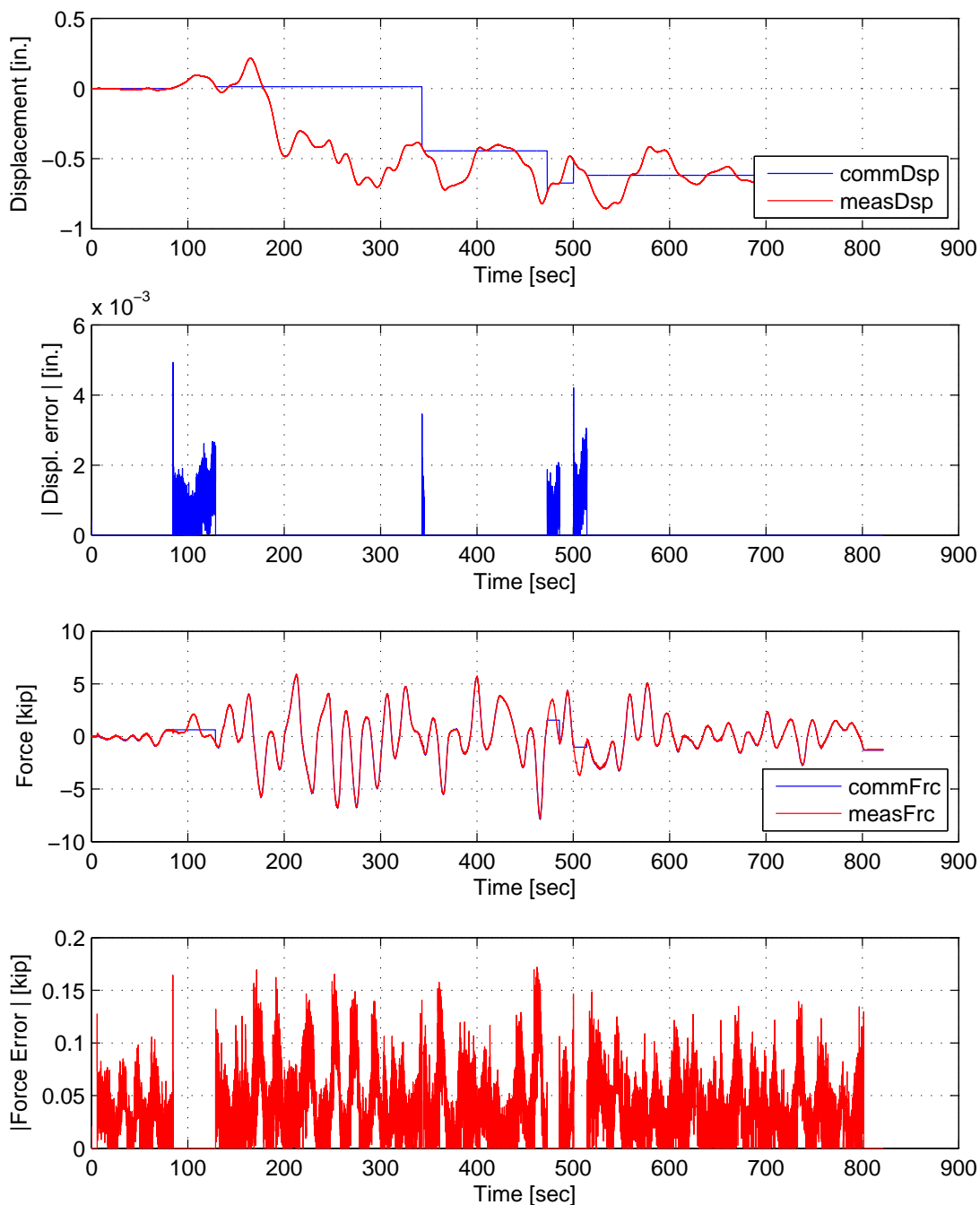


Figure E.41: STS controller recorded values for bottom actuator. Top: Command and measured displacement plot from the STS controller for SMC:Broyden hybrid simulation in DC. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for SMC:Broyden hybrid simulation in FC. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with  $\alpha$ OS with SMC:Broyden.

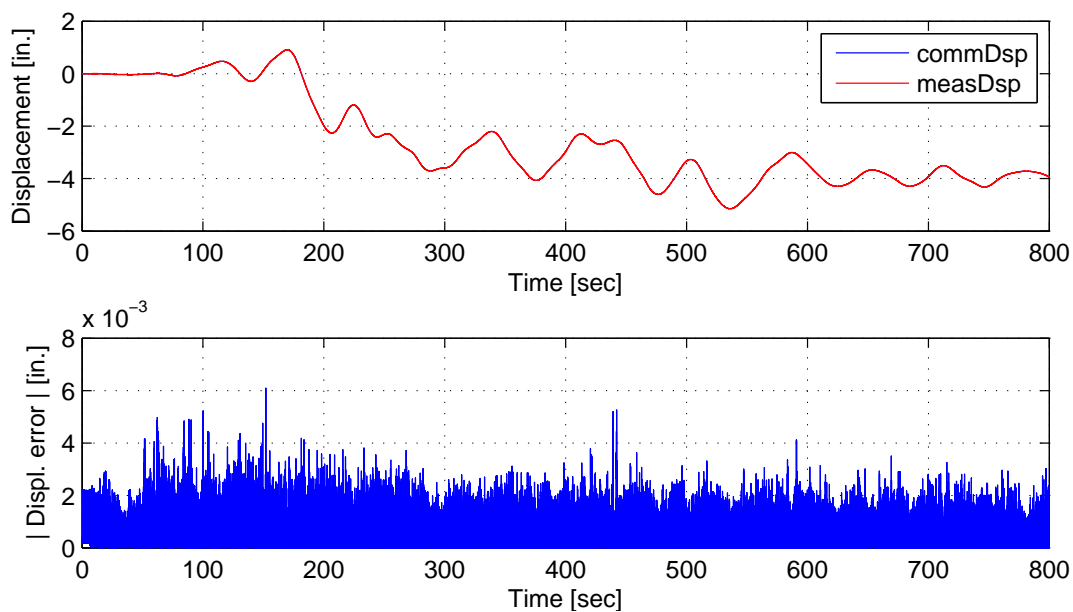


Figure E.42: STS controller recorded values for top actuator. Top: Command and measured displacement plot from the STS controller for SMC:Broyden hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with  $\alpha$ OS with SMC:Broyden.



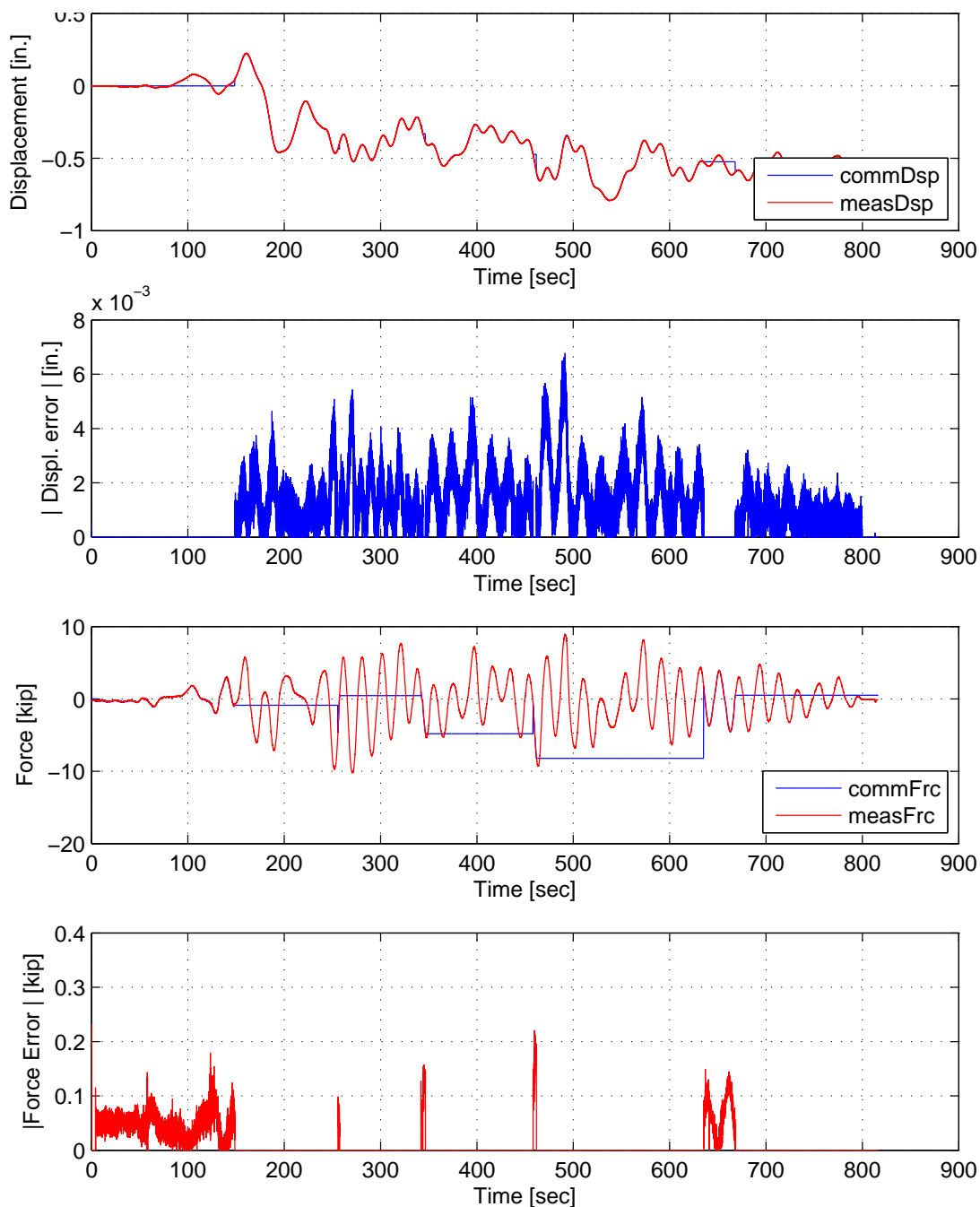


Figure E.43: STS controller recorded values for bottom actuator. Top: Command and measured displacement plot from the STS controller for SMC:BFGS hybrid simulation in DC. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for SMC:BFGS hybrid simulation in FC. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with  $\alpha$ OS with SMC:BFGS.

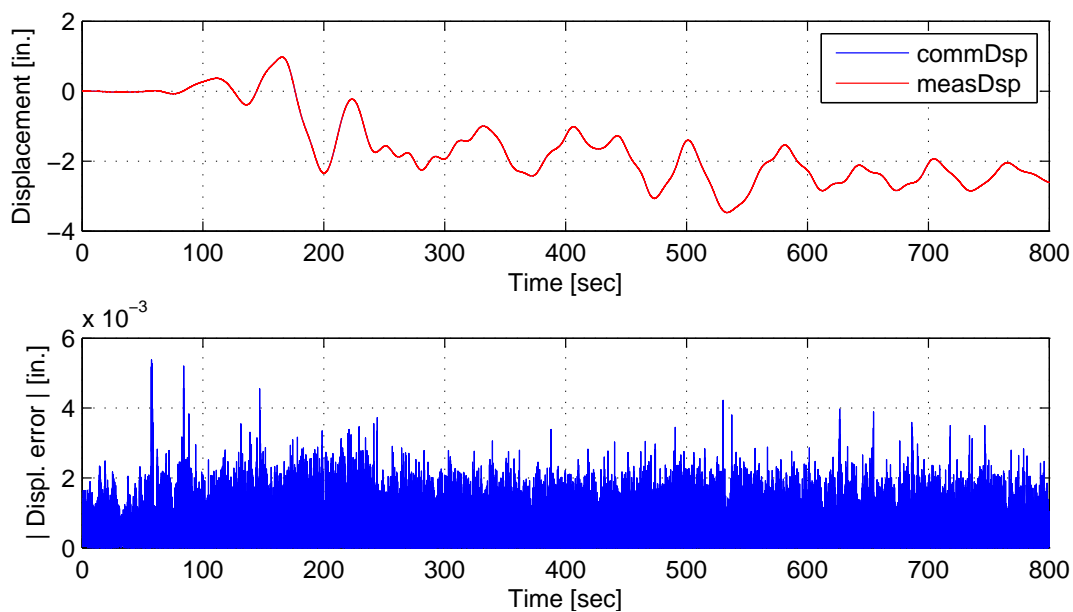


Figure E.44: STS controller recorded values for top actuator. Top: Command and measured displacement plot from the STS controller for SMC:BFGS hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with  $\alpha$ OS with SMC:BFGS.

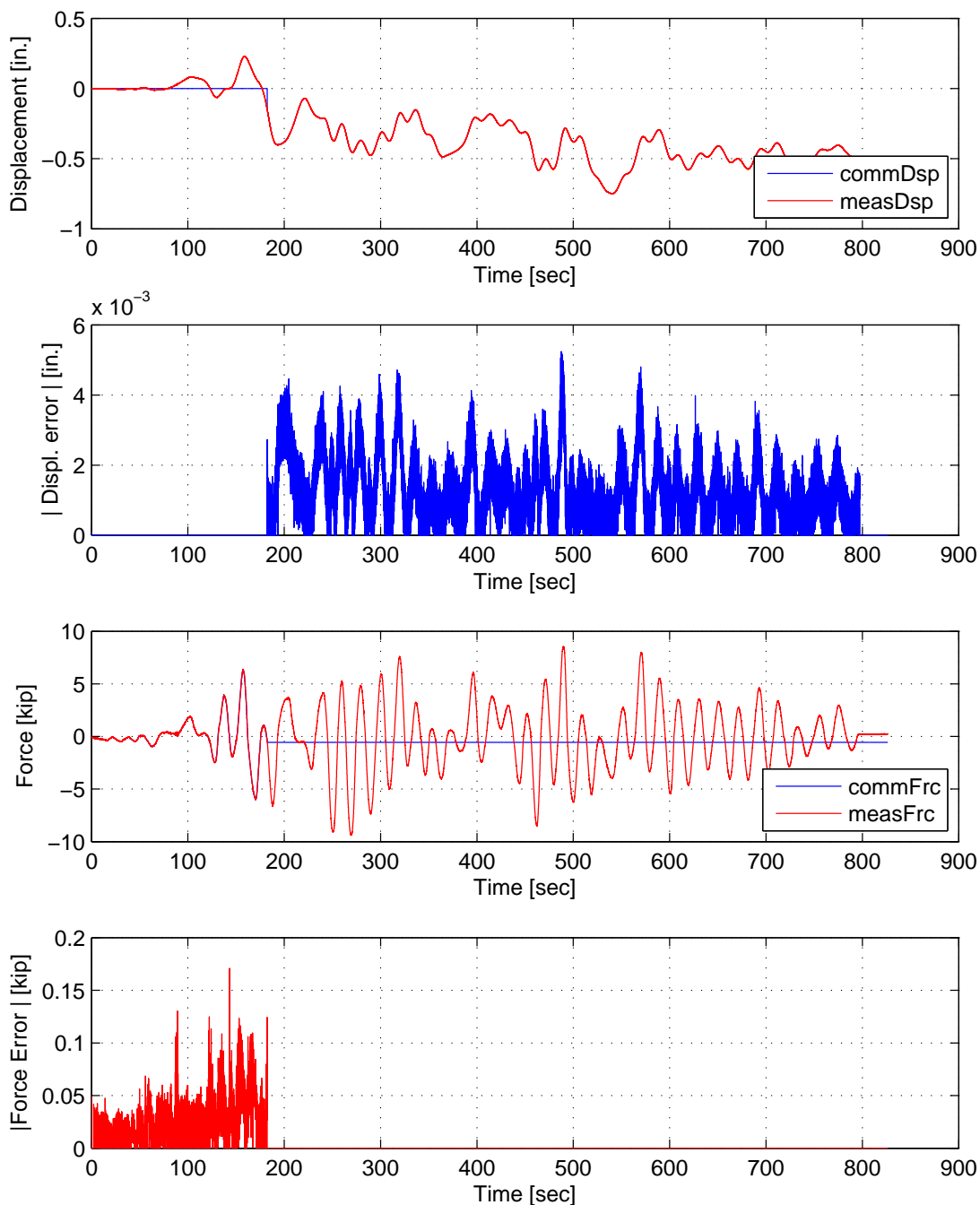


Figure E.45: STS controller recorded values for bottom actuator. Top: Command and measured displacement plot from the STS controller for SMC:Intrinsic hybrid simulation in DC. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for SMC:Intrinsic hybrid simulation in FC. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with  $\alpha$ OS with SMC:Intrinsic.

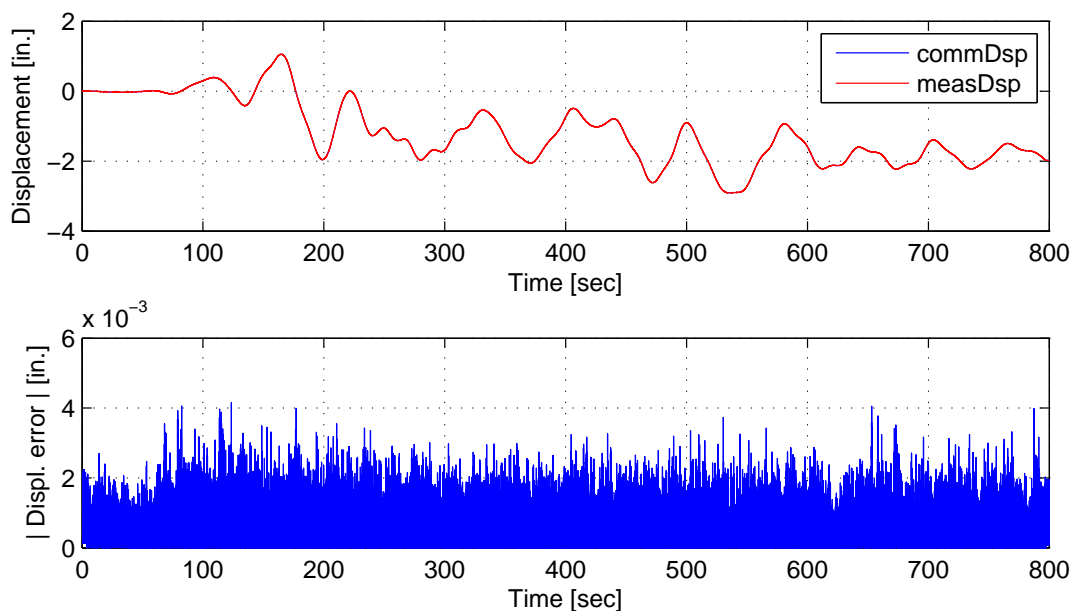


Figure E.46: STS controller recorded values for top actuator. Top: Command and measured displacement plot from the STS controller for SMC:Intrinsic hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with  $\alpha$ OS with SMC:Intrinsic.

### **E.2.3 NMF time integration scheme**

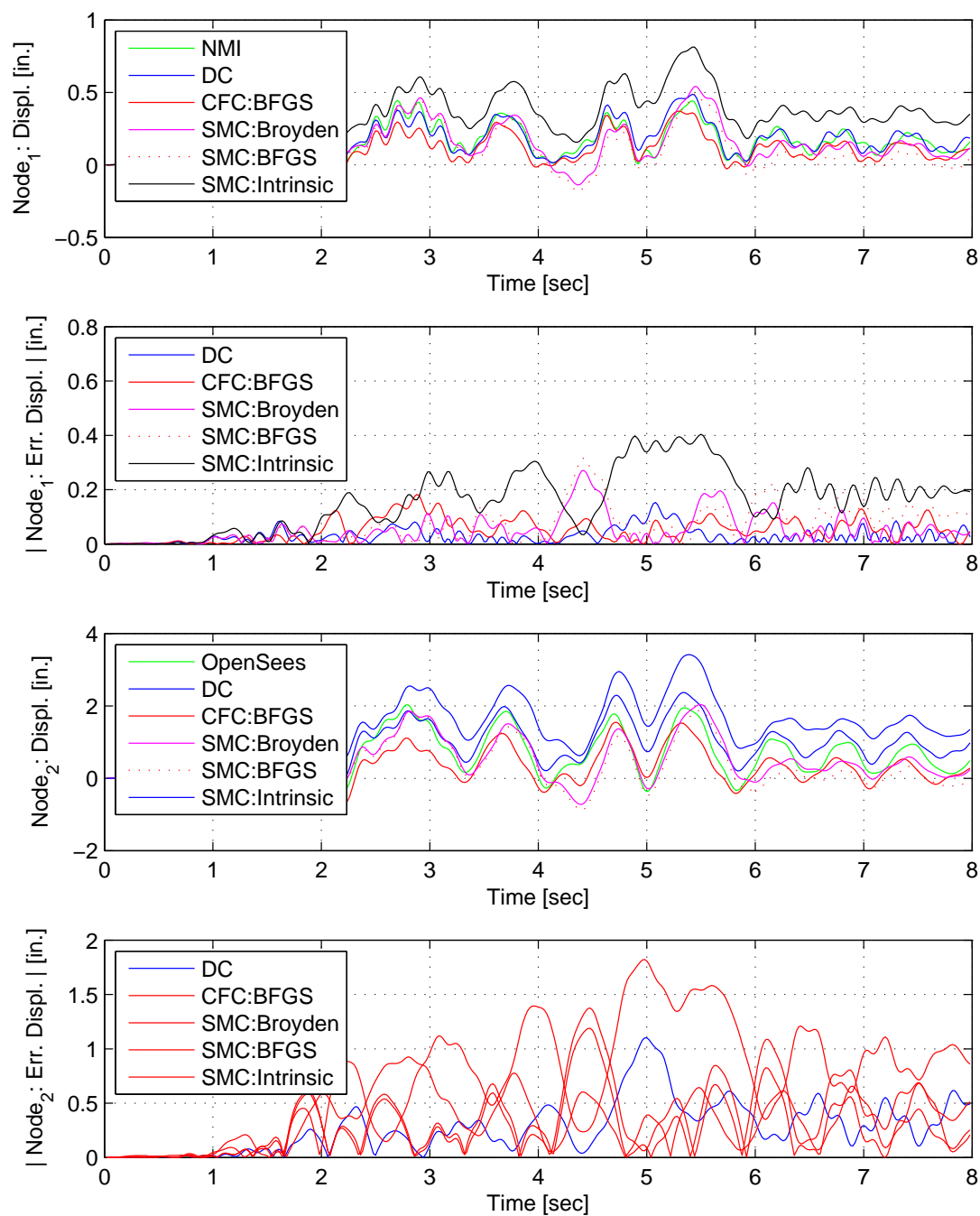


Figure E.47: Node 1 displacement-time history plot (top), Node 1 absolute displacement error (middle-top), Node 2 displacement-time history plot (middle-bottom) and Node 2 absolute displacement error (bottom) for SMC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses NMF with DC, CFC:BFGS, SMC:Broyden, SMC: BFGS, SMC:Intrinsic. All experimental result errors are calculated using the NMI numerical result.

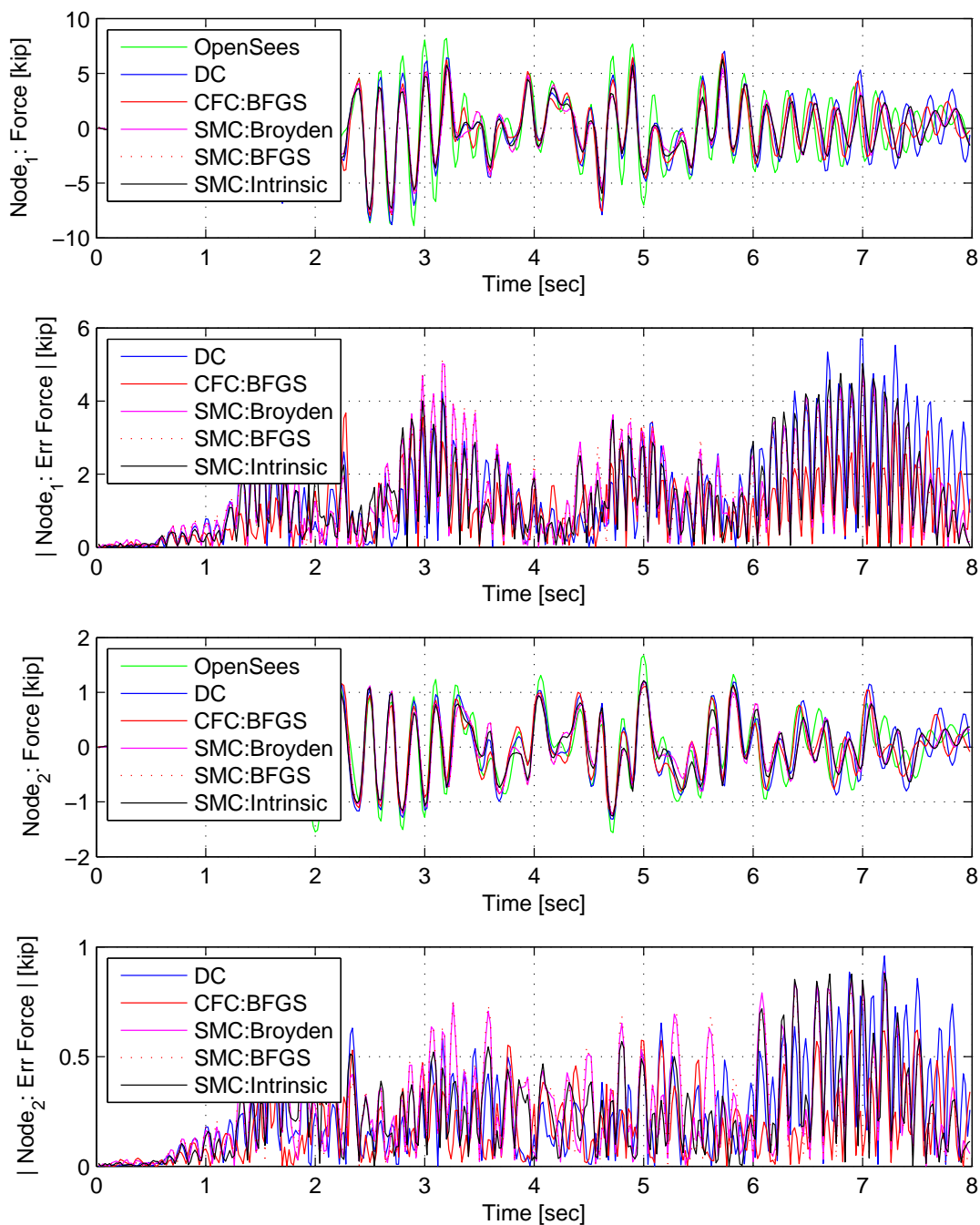


Figure E.48: Node 1 force-time history plot (top), Node 1 absolute force error (middle-top), Node 2 force-time history plot (middle-bottom) and Node 2 absolute force error (bottom) for SMC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses NMF with DC, CFC:BFGS, SMC:Broyden, SMC: BFGS, SMC:Intrinsic. All experimental result errors are calculated using the NMI numerical result.

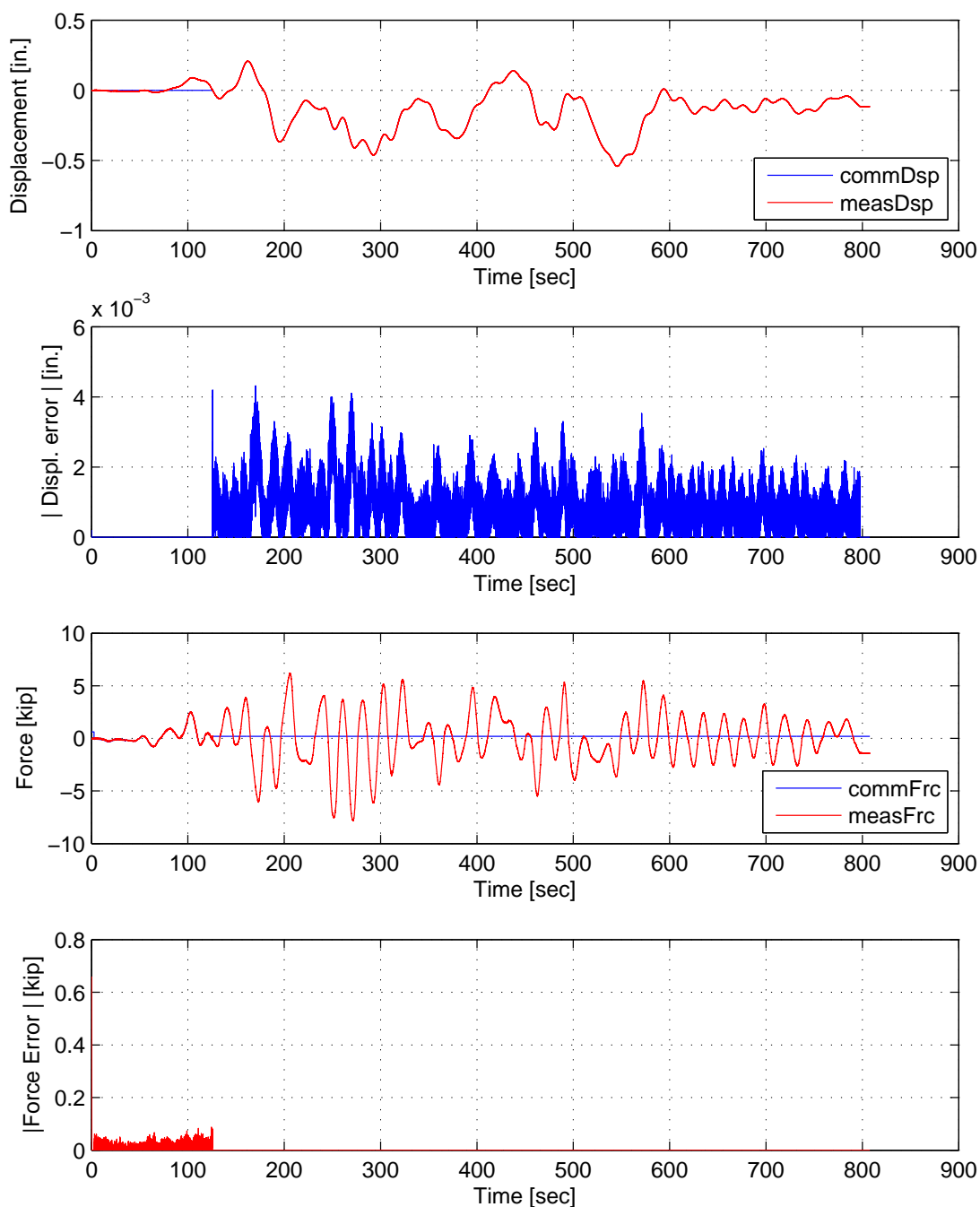


Figure E.49: STS controller recorded values for bottom actuator. Top: Command and measured displacement plot from the STS controller for SMC:Broyden hybrid simulation in DC. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for SMC:Broyden hybrid simulation in FC. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NMF with SMC:Broyden.



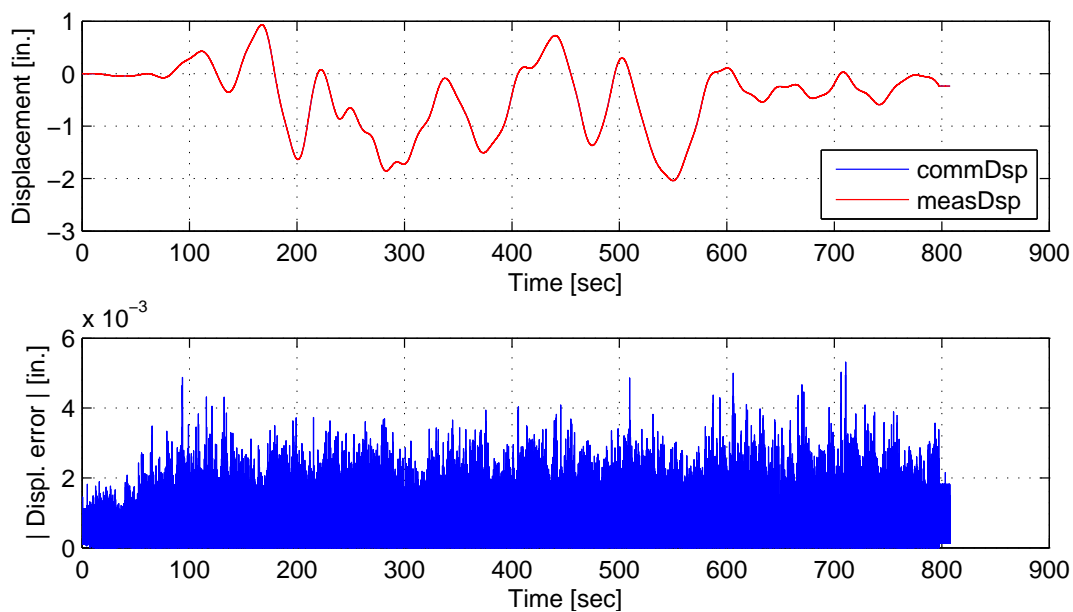


Figure E.50: STS controller recorded values for top actuator. Top: Command and measured displacement plot from the STS controller for SMC:Broyden hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NMF with SMC:Broyden.

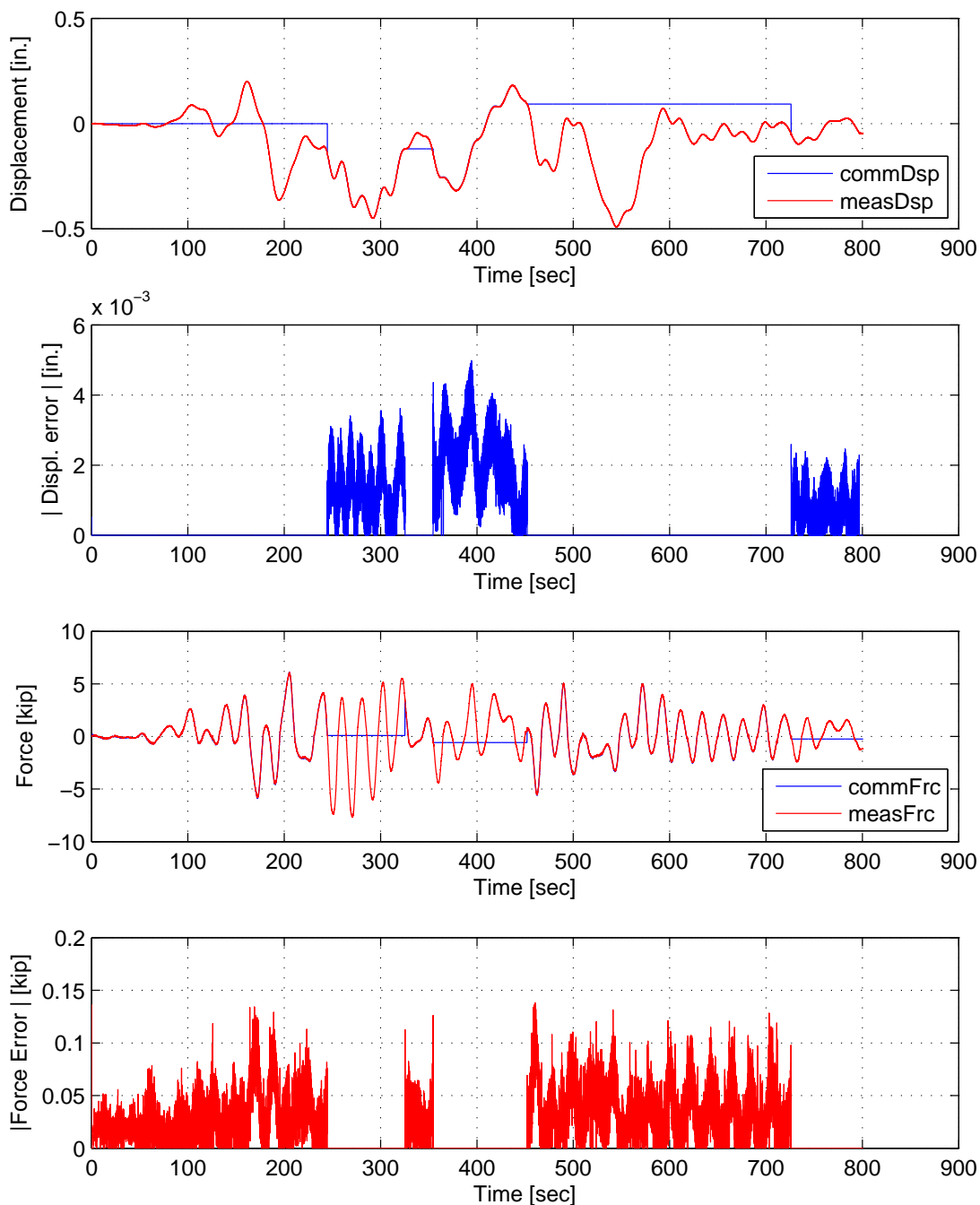


Figure E.51: STS controller recorded values for bottom actuator. Top: Command and measured displacement plot from the STS controller for SMC:BFGS hybrid simulation in DC. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for SMC:BFGS hybrid simulation in FC. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NMF with SMC:BFGS.

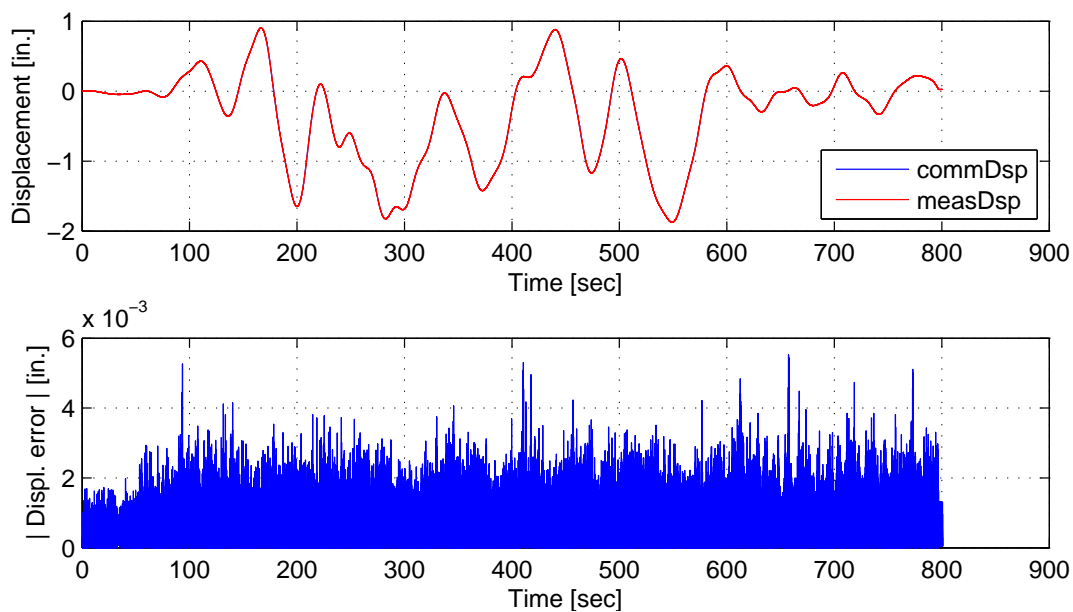


Figure E.52: STS controller recorded values for top actuator. Top: Command and measured displacement plot from the STS controller for SMC:BFGS hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NMF with SMC:BFGS.

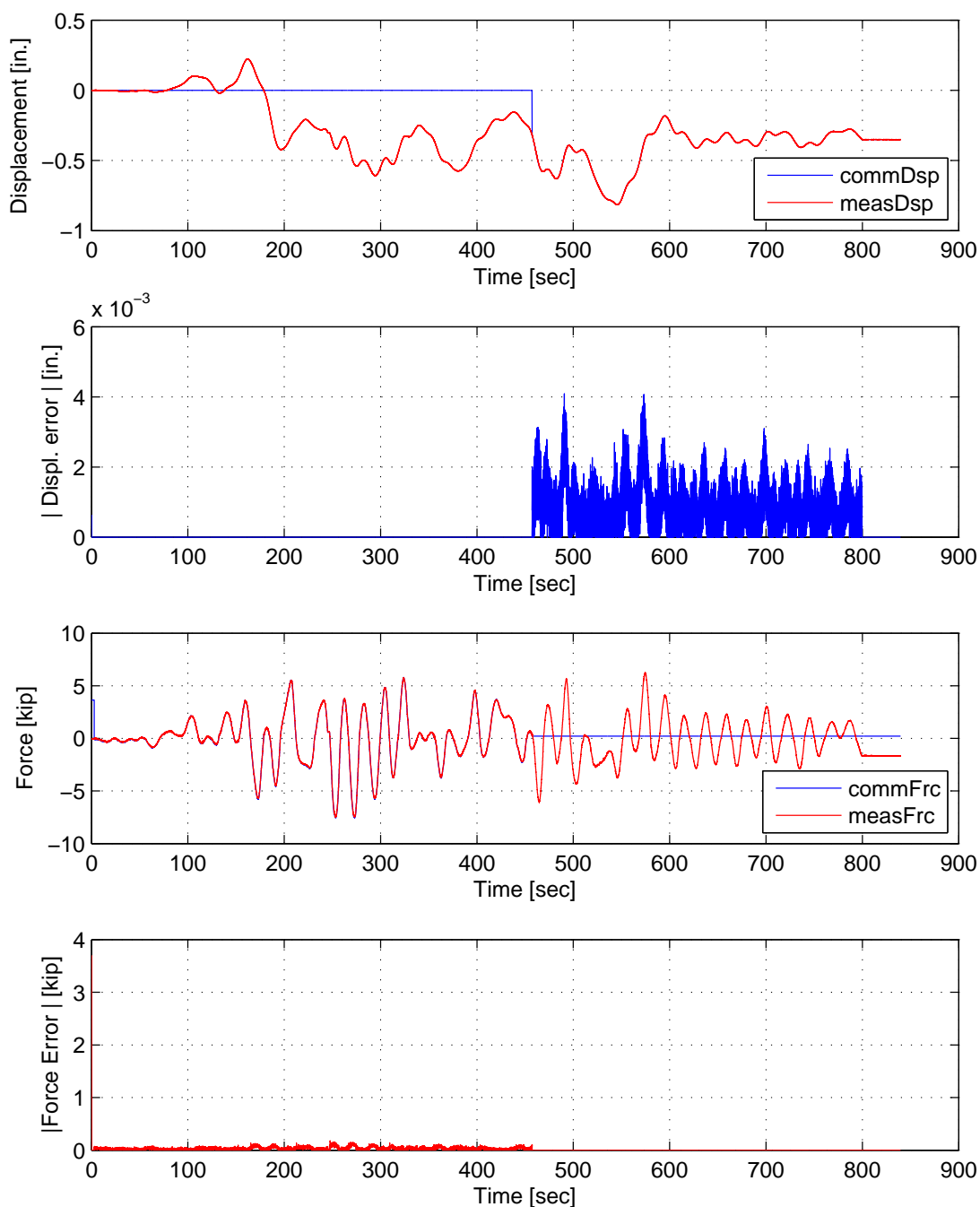


Figure E.53: STS controller recorded values for bottom actuator. Top: Command and measured displacement plot from the STS controller for SMC:Intrinsic hybrid simulation in DC. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for SMC:Intrinsic hybrid simulation in FC. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NMF with SMC:Intrinsic.

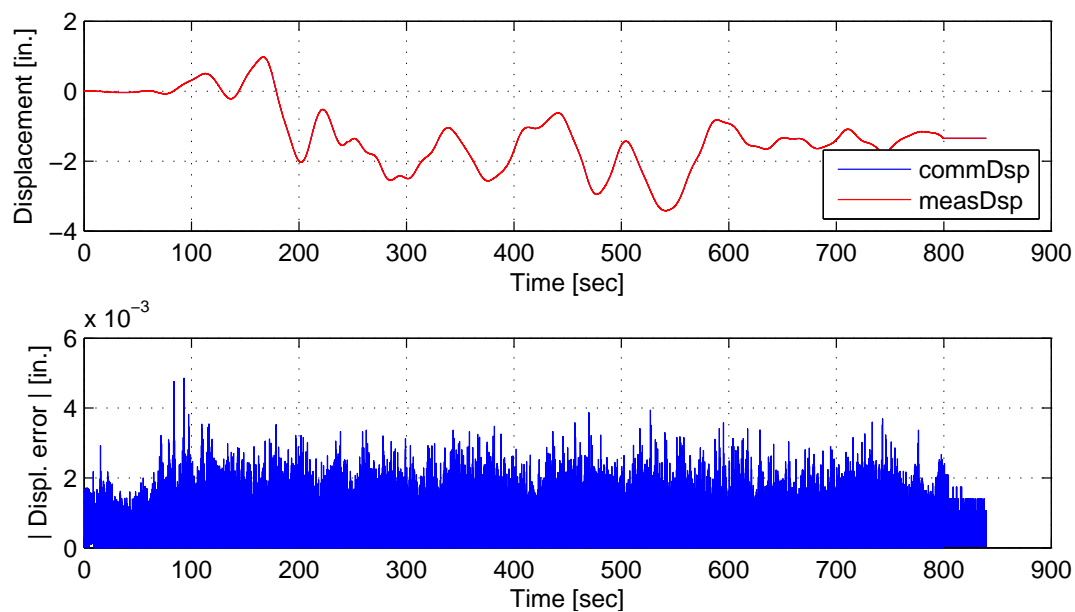


Figure E.54: STS controller recorded values for top actuator. Top: Command and measured displacement plot from the STS controller for SMC:Intrinsic hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NMF with SMC:Intrinsic.

## **E.2.4 NMR time integration scheme**

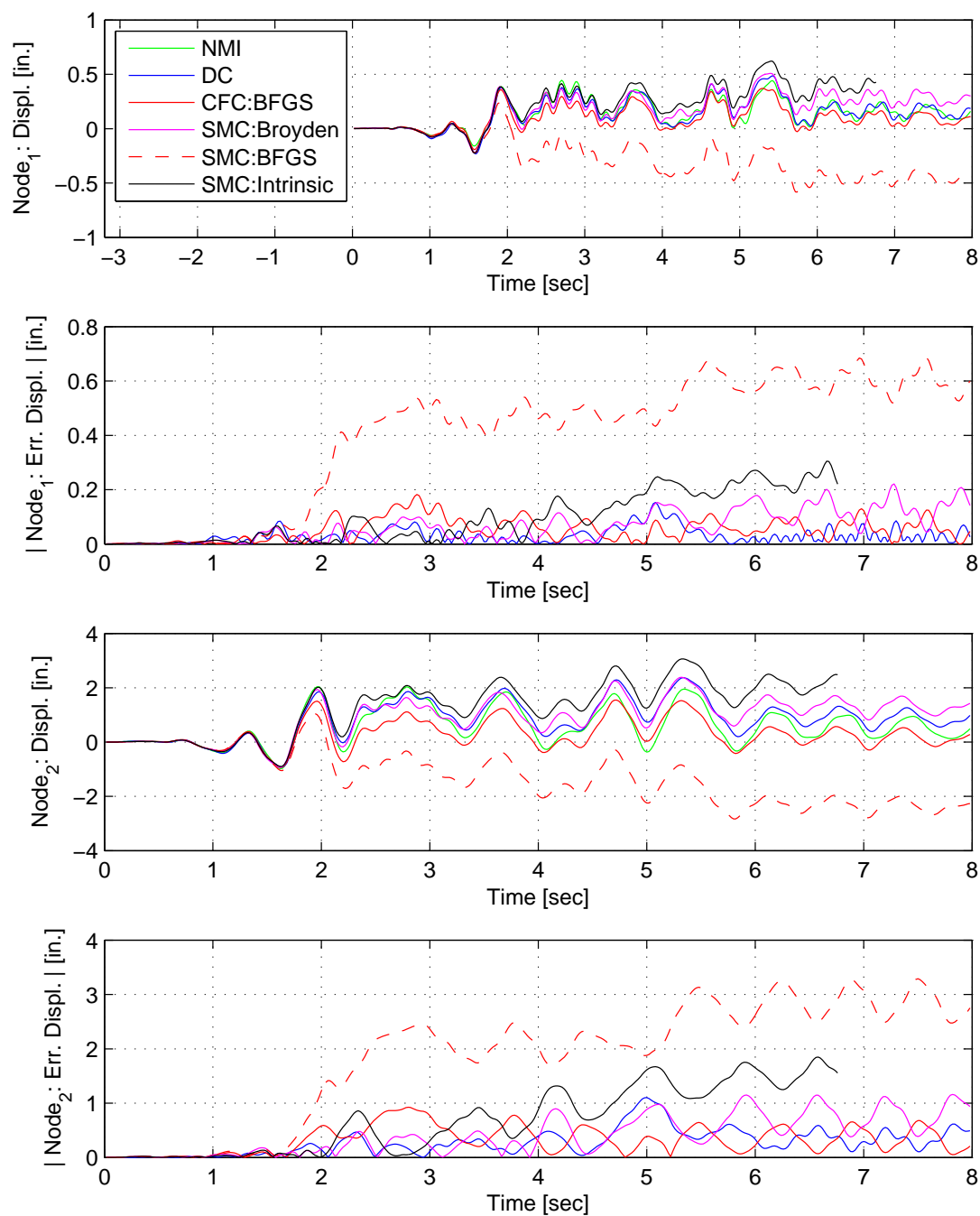


Figure E.55: Node 1 displacement-time history plot (top), Node 1 absolute displacement error (middle-top), Node 2 displacement-time history plot (middle-bottom) and Node 2 absolute displacement error (bottom) for SMC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses NMR with DC, CFC:BFGS, SMC:Broyden, SMC: BFGS, SMC:Intrinsic. All experimental result errors are calculated using the NMI numerical result.

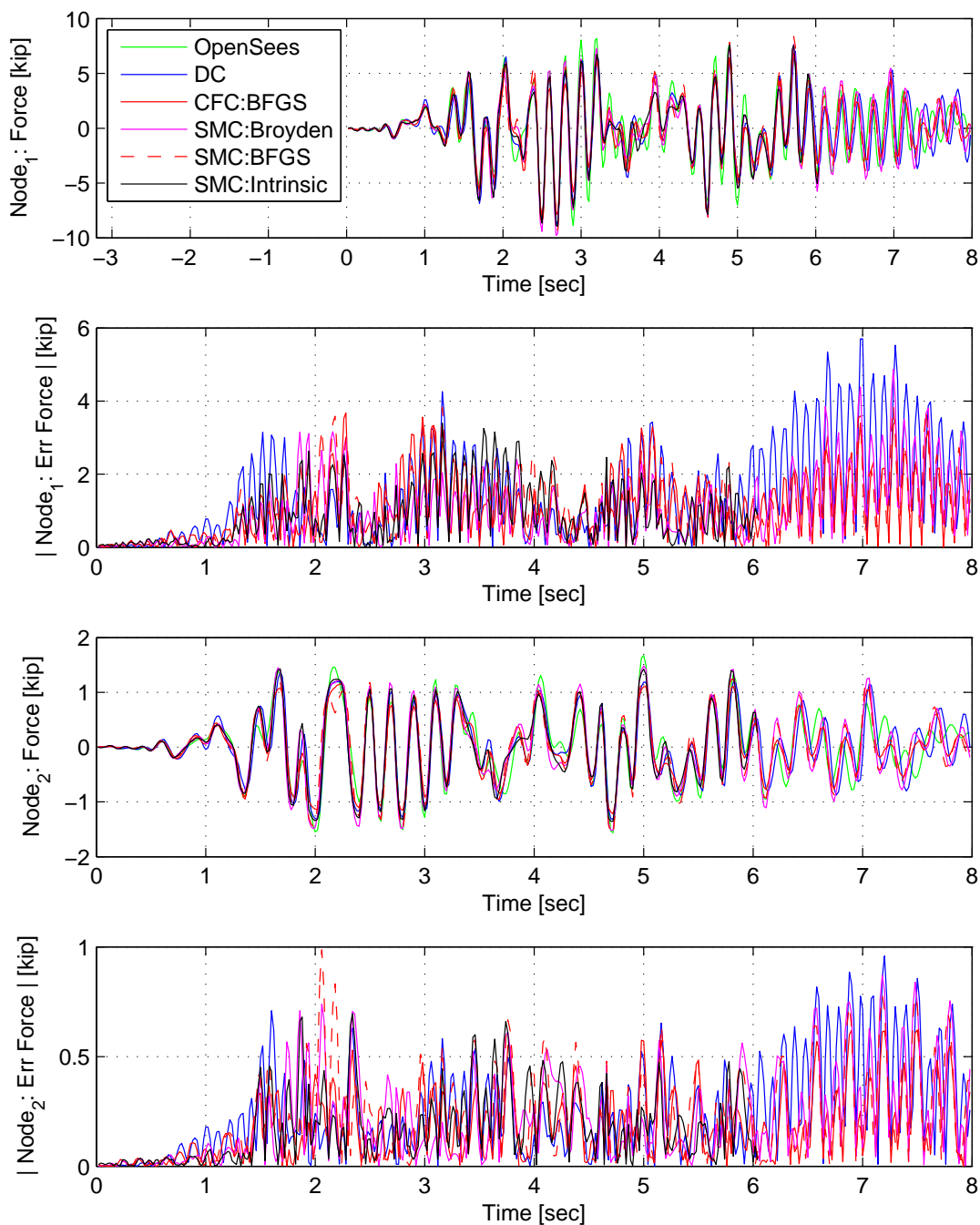


Figure E.56: Node 1 force-time history plot (top), Node 1 absolute force error (middle-top), Node 2 force-time history plot (middle-bottom) and Node 2 absolute force error (bottom) for SMC hybrid simulation. The ground motion is scaled to 50% to keep the simulation in the nonlinear range. The 2-DOF setup uses NMR with DC, CFC:BFGS, SMC:Broyden, SMC: BFGS, SMC:Intrinsic. All experimental result errors are calculated using the NMI numerical result.



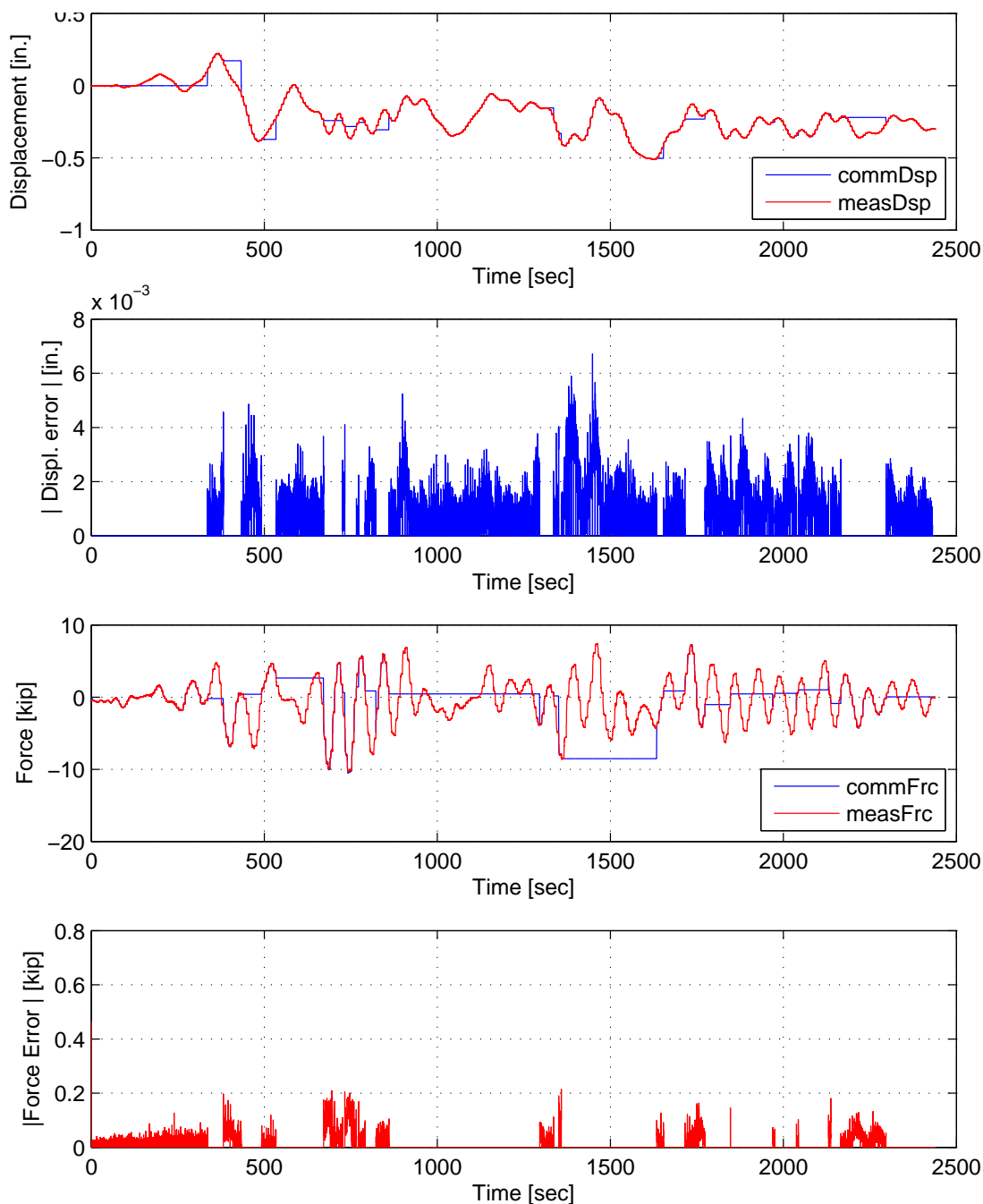


Figure E.57: STS controller recorded values for bottom actuator. Top: Command and measured displacement plot from the STS controller for SMC:BFGS hybrid simulation in DC. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for SMC:Broyden hybrid simulation in FC. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NMR with SMC:Broyden.

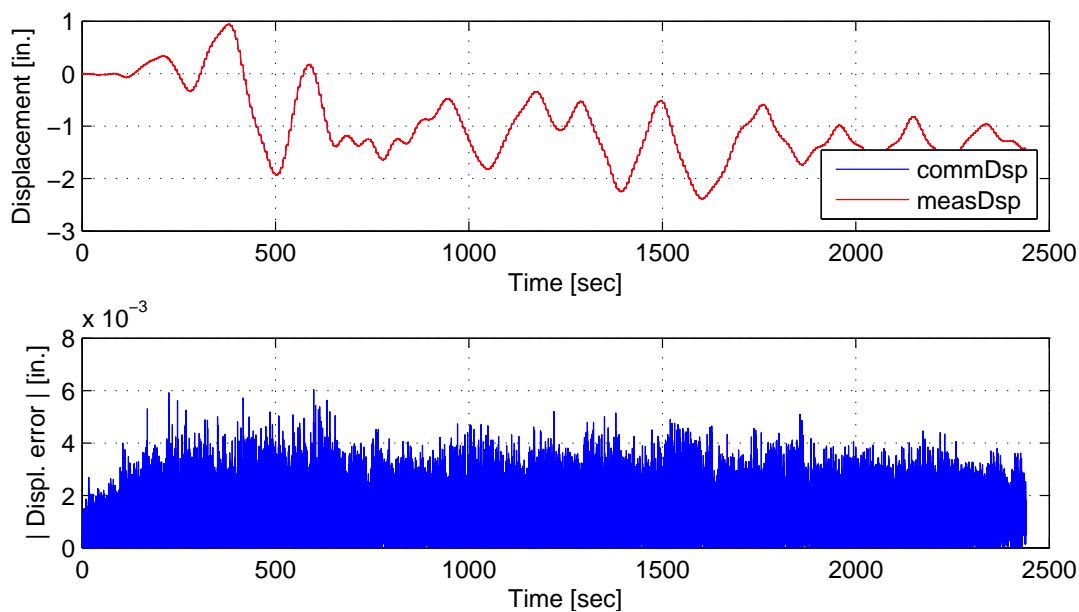


Figure E.58: STS controller recorded values for top actuator. Top: Command and measured displacement plot from the STS controller for SMC:Broyden hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NMR with SMC:Broyden.

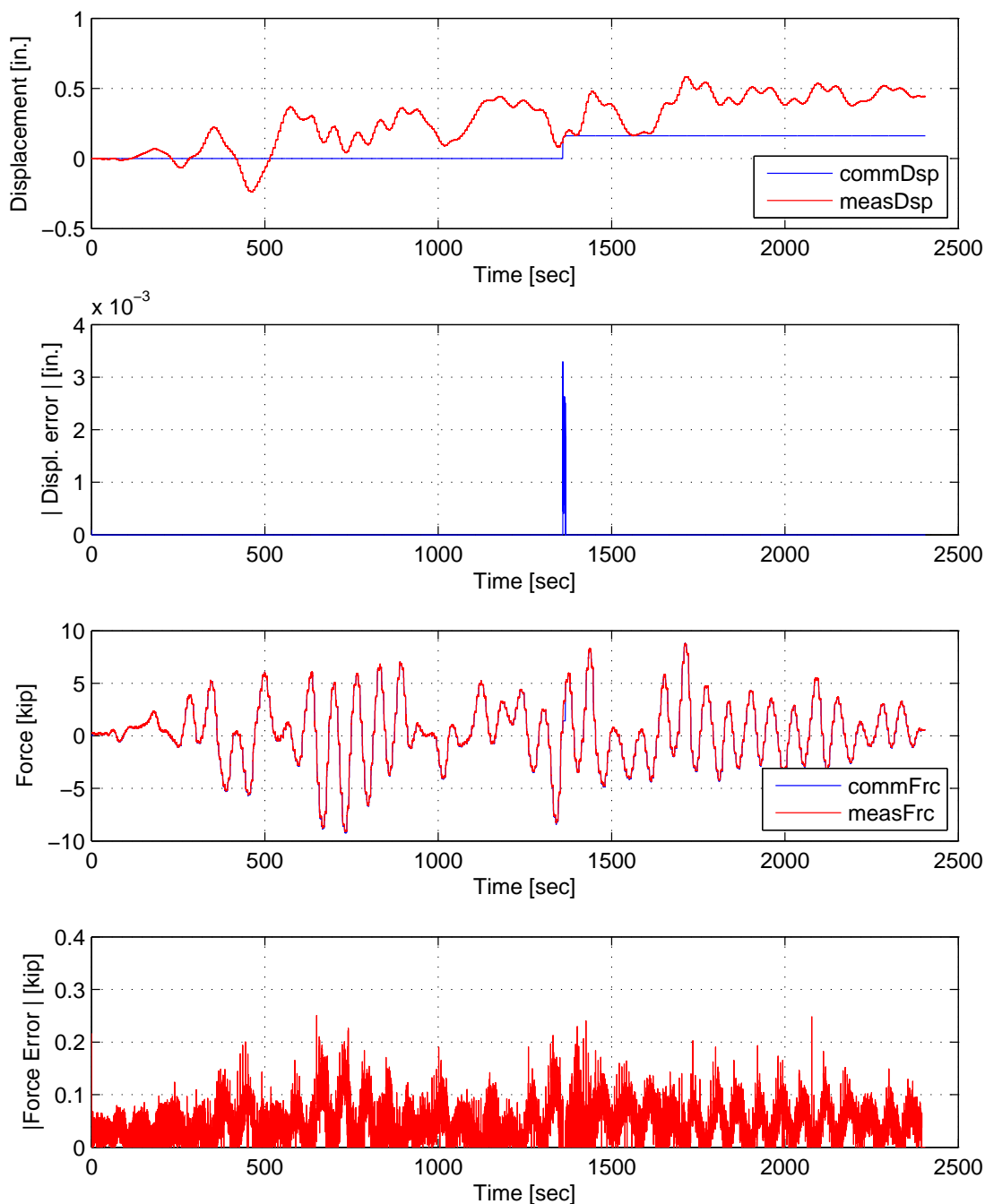


Figure E.59: STS controller recorded values for bottom actuator. Top: Command and measured displacement plot from the STS controller for SMC:BFGS hybrid simulation in DC. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for SMC:BFGS hybrid simulation in FC. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NMR with SMC:BFGS.

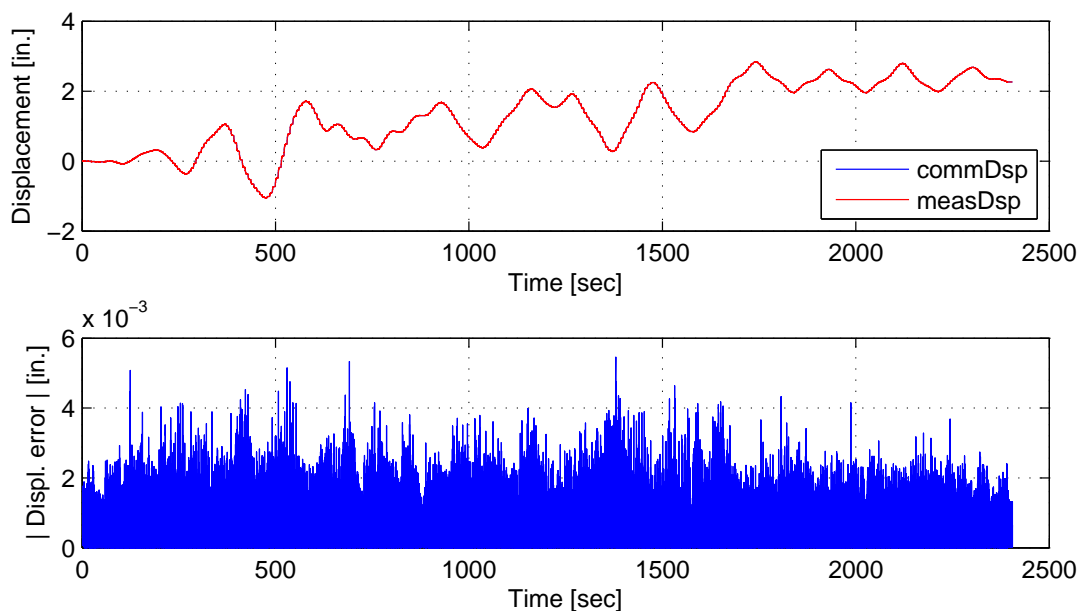


Figure E.60: STS controller recorded values for top actuator. Top: Command and measured displacement plot from the STS controller for SMC:BFGS hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NMR with SMC:BFGS.

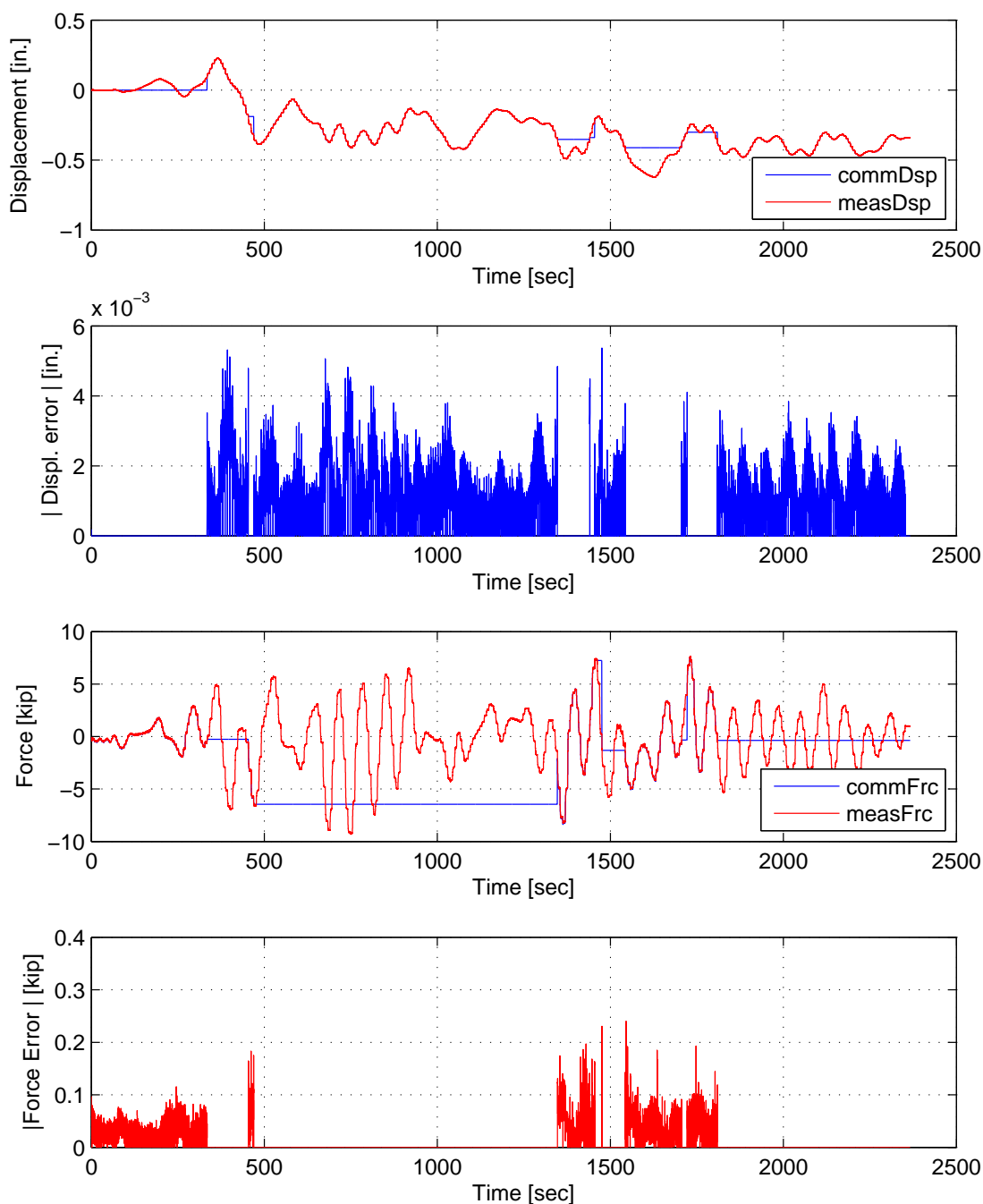


Figure E.61: STS controller recorded values for bottom actuator. Top: Command and measured displacement plot from the STS controller for SMC:Intrinsic hybrid simulation in DC. Middle-Top: Absolute error between command and measured displacement from the STS controller. Middle-Bottom: Command and measured force plot from the STS controller for SMC:Intrinsic hybrid simulation in FC. Bottom: Absolute error between command and measured force from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NMR with SMC:Intrinsic.

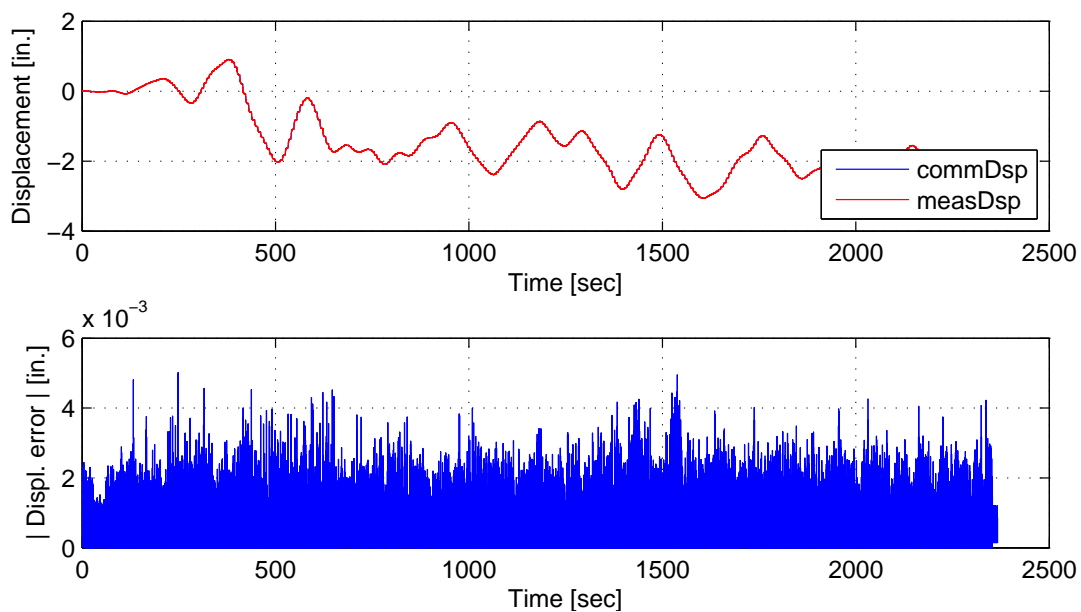


Figure E.62: STS controller recorded values for top actuator. Top: Command and measured displacement plot from the STS controller for SMC:Intrinsic hybrid simulation. Bottom: Absolute error between command and measured displacement from the STS controller. The ground motion is scaled to 50% to push the specimen into the nonlinear range. The 2-DOF setup is used with NMR with SMC:Intrinsic.