

PHYSICAL SUBSPACE IDENTIFICATION FOR HELICOPTERS

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
OF  
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SEVIL AVCIOĞLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF DOCTOR OF PHILOSOPHY  
IN  
AEROSPACE ENGINEERING

MAY 2019



Approval of the thesis:

**PHYSICAL SUBSPACE IDENTIFICATION FOR HELICOPTERS**

submitted by **SEVIL AVCIOĞLU** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Aerospace Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar  
Dean, Graduate School of **Natural and Applied Sciences**

\_\_\_\_\_

Prof. Dr. İsmail Hakkı Tuncer  
Head of Department, **Aerospace Engineering**

\_\_\_\_\_

Assist. Prof. Dr. Ali Türker Kutay  
Supervisor, **Aerospace Engineering, METU**

\_\_\_\_\_

**Examining Committee Members:**

Assoc. Prof. Dr. İlkay Yavrucuk  
Aerospace Engineering, METU

\_\_\_\_\_

Assist. Prof. Dr. Ali Türker Kutay  
Aerospace Engineering, METU

\_\_\_\_\_

Prof. Dr. Kemal Leblebicioğlu  
Electrical&Electronics Engineering Dept., METU

\_\_\_\_\_

Assist. Prof. Dr. Yakup Özkazanç  
Electrical&Electronics Engineering Dept., Hacettepe Üni.

\_\_\_\_\_

Prof. Dr. Coşku Kasnakoğlu  
Electrical&Electronics Engineering Dept., TOBB ETÜ

\_\_\_\_\_

Date: 29.05.2019

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Surname: Sevil Avcioglu

Signature:

## **ABSTRACT**

### **PHYSICAL SUBSPACE IDENTIFICATION FOR HELICOPTERS**

Avcıođlu, Sevil  
Doctor of Philosophy, Aerospace Engineering  
Supervisor: Assist. Prof. Dr. Ali Türker Kutay

May 2019, 184 pages

Subspace identification is a powerful tool due to its well-understood techniques based on linear algebra (orthogonal projections and intersections of subspaces) and numerical methods like QR and singular value decomposition. However, the state space model matrices which are obtained from conventional subspace identification algorithms are not necessarily associated with the physical states. This can be an important deficiency when physical parameter estimation is essential. This holds for the area of helicopter flight dynamics where physical parameter estimation is mainly conducted for mathematical model improvement, aerodynamic parameter validation and flight controller tuning. The main objective of this study is to obtain helicopter physical parameters from subspace identification results. In order to achieve this objective, N4SID subspace identification algorithm is implemented for a multi-role helicopter using both FLIGHTLAB simulation and real flight test data. After obtaining state space matrices via subspace identification, constrained nonlinear optimization methodologies are utilized for extracting the physical parameters. The state space matrices are transformed into equivalent physical forms via both “Sequential Quadratic Programming” and “Interior Point” nonlinear optimization algorithms. The required objective function is generated by summing the square of similarity transformation

equations. The constraints are selected with physical insight. Many runs are conducted for randomly selected initial conditions. It can be concluded that all of the parameters with high significance can be obtained with a high level of accuracy for the data obtained from the linear model. This strongly supports the idea behind this study. Results for the data obtained from the nonlinear model are also evaluated to be satisfactory in the light of statistical error analysis. Results for the real flight test data are also evaluated to be good for the helicopter modes that are properly excited in the flight tests.

Keywords: Subspace Identification, Parameter Estimation, Similarity Transformation, Optimization, Helicopter Dynamics.

## ÖZ

### HELİKOPTERLER İÇİN FİZİKSEL ALT UZAY ESASLI SİSTEM TANIMLAMA

Avciođlu, Sevil  
Doktora, Havacılık ve Uzay Mühendisliđi  
Tez Danışmanı: Dr. Öğr. Üyesi Ali Türker Kutay

Mayıs 2019, 184 sayfa

Alt uzay esaslı sistem tanımlama yöntemi oldukça güçlü bir sistem tanımlama yöntemidir. Bu özelliđi kendini kanıtlamış dikey izdüşüm ve alt uzayların kesişimi gibi doğrusal cebir yöntemleri ile QR ayrıştırması ve tekil değerlerine ayrıştırma gibi sayısal açıdan sağlam sayısal yöntemlerin kullanılmasından gelir. Ancak, alt uzay esaslı sistem tanımlama yöntemi ile elde edilen durum uzay modeli her zaman fiziksel durum vektörleri ile eşleşmeyebilir. Bu sebeple; alt uzay esaslı sistem tanımlama yöntemi fiziksel parametre kestiriminin önemli olduđu çalışmalar için yetersiz kalmaktadır. Bu durum, fiziksel parametre kestiriminin matematik model iyileştirme, aerodinamik parametre doğrulama ve uçuş kontrolcüsü iyileştirme faaliyetlerinde kullanıldığı helikopter uçuş dinamiđi alanında da geçerlidir. Bu çalışmanın ana amacı helikopter fiziksel parameterlerini alt uzay esaslı sistem tanımlama yöntemi ile kestirebilmektir. Bu amacı gerçekleştirmek için, bir genel maksat taarruz helikopterine ait veriler kullanılarak N4SID alt uzay esaslı sistem tanımlama yöntemi uygulanmıştır. Gerekli veriler FLIGHTLAB uçuş benzetimlerinden ve gerçek uçuş testlerinden elde edilmiştir. Alt uzay esaslı sistem tanımlama yöntemi ile elde edilen sistem matris elemanları doğrusal olmayan optimizasyon yöntemleri kullanılarak fiziksel parametrelere çevrilmiştir. Bunun için “Sıralı Karesel Programlama” ve “İç Nokta” optimizasyon algoritmaları kullanılmıştır. Burada amaç fonksiyonu benzerlik

dönüşüm denklemlerinin karelerinin toplamı olarak ifade edilmiştir. Kısıtlar oluşturulurken parametrelerin fiziksel anlamlarından yararlanılmıştır. Kısıtlar dahilinde, rastgele seçilmiş değerler ile çok sayıda optimizasyon yapılmıştır. Sonuçlar göstermektedir ki; doğrusal helikopter modelinden yola çıkılarak yapılan sistem tanımlama faaliyeti kapsamında, fiziksel anlamda baskın olan parametreler oldukça yüksek doğrulukla elde edilebilmiştir. Bu durum; kullanılan yöntemin doğruluğunu kuvvetli bir biçimde desteklemektedir. Doğrusal olmayan helikopter modelinden yola çıkılarak tekrarlanan sistem tanımlama ve fiziksel parametre kestirimi faaliyeti de istatistiksel doğruluk analizi sonuçlarına göre başarılı olmuştur. Çalışma ayrıca gerçek uçuş test verileri kullanılarak tekrar edilmiş, düzgün olarak uyarılabilen uçuş modları ile ilişkili parameterler kestirilebilmiştir.

Anahtar Kelimeler: Alt Uzay Esaslı Sistem Tanımlama, Parametre Kestirimi, Benzerlik Dönüşümü, Optimizasyon, Helikopter Dinamiği



...for the end of speciesism

## ACKNOWLEDGEMENTS

The author would like to thank to her advisor, Assist. Prof. Dr. Ali Türker Kutay for his guidance throughout the Ph.D. study. She would like to express her deep gratitude to Prof. Dr. Kemal Leblebiciođlu for his comments, guidance and continuous support to the study and for his encouragement. The collaboration of the Ph.D. committee members Assist. Prof. Dr. Yakup Özkazanç and Assoc. Dr. İlkey Yavrucuk are also appreciated.

The author would like to acknowledge TUBİTAK (The Scientific and Technological Research Council of Turkey) for financially supporting the research activity. In the scope of 1501- Industrial Research and Design Projects, the study was supported by Project # 3130589. Simulation and real flight data of the study is provided by Turkish Aerospace Industries. The author would like to express their appreciation to Turkish Aerospace Industries for their comprehensive support on the study.

The author takes this opportunity to express gratitude to all of her colleagues Anıl Öztürk, Ilgaz Dođa Okçu, Özge Kapulu Gülbađ, Kadircan Kopşa, A. İlden Ak, Volkan Kargın, Mahmut Bilgin and Dođan Yıldız for their help and technical support. The author would also like to thank to her friends F. Demet Ülker, Duygu Barın and Suzan Esra Özkan for sharing expertise and giving their continuous encouragement.

Finally, the author would like to thanks to her mothers Ziyet and Şebnem, her fathers Hüseyin and Merter, sisters Fatoş and Demet, brother Serkan, nephews Sencer and Toprak for their love, support and encouragement. Most importantly, the author wishes to thank her loving husband, Tolga for his endless support and valuable technical guidance and her wonderful son, Tunç who provide unending inspiration.

## TABLE OF CONTENTS

ABSTRACT .....	v
ÖZ.....	vii
ACKNOWLEDGEMENTS .....	x
TABLE OF CONTENTS .....	xi
LIST OF TABLES .....	xiv
LIST OF FIGURES .....	xv
LIST OF ABBREVIATIONS .....	xix
LIST OF SYMBOLS .....	xx
CHAPTERS	
1. INTRODUCTION .....	1
1.1. Literature Survey .....	1
2. SUBSPACE IDENTIFICATION THEORY .....	11
2.1. Introduction to Linear System Identification .....	11
2.2. State Space Representation of Dynamic Systems .....	12
2.3. Brief Overview on Subspace Identification Theory [19] .....	13
2.4. Geometric Tools [19] .....	14
2.4.1. Orthogonal Projections .....	14
2.4.2. Oblique Projections.....	15
2.4.3. Singular Value Decomposition .....	16
2.4.4. Statistical Approaches.....	17
2.5. Subspace Identification for Deterministic Systems.....	19
2.5.1. Blok Hankel Matrices .....	19

2.5.2. State Sequence Matrix .....	21
2.5.3. System Related Matrices .....	22
2.5.4. Main Theorems for Deterministic Subspace Identification Algorithms ..	22
2.5.5. Algorithms for Deterministic Systems .....	25
2.5.5.1. Algorithm 1 ([19], Chapter 2).....	26
2.5.5.2. Algorithm 2 ([19], Chapter 2).....	27
2.6. Facts on Real World Applications .....	29
2.6.1. Algorithm 3 – Robust Subspace Identification ([19], Chapter 4) .....	32
2.7. Similarity Transformation.....	34
3. MODEL STRUCTURE .....	37
3.1. Derivative of Forces with respect to Translational Velocity Components ( $Xu, Xv, Xw, Yu, Yv, Yw, Zu, Zv, Zw$ ) .....	40
3.2. Derivative of Forces with respect to Translational Velocity Components ( $Xu, Xv, Xw, Yu, Yv, Yw, Zu, Zv, Zw$ ) .....	41
3.3. Derivative of Forces with respect to Translational Velocity Components ( $Xu, Xv, Xw, Yu, Yv, Yw, Zu, Zv, Zw$ ) .....	41
3.4. Derivative of Moments with respect to Angular Velocity Components ( $Lp, Lq, Lr, Mp, Mq, Mr, Np, Nq, Nr$ ) .....	41
3.5. General View on Stability Derivatives .....	42
3.6. Derivative of Forces with respect to Control Inputs ( $Z\delta col, Z\delta lon, Y\delta ped$ ) .....	43
3.7. Derivative of Moments with respect to Control Inputs ( $M\delta col, L\delta col,$ $N\delta ped, L\delta ped, L\delta lat, Llon, M\delta lat, M\delta lon$ ) .....	44
4. PARAMETER ESTIMATION WITH NONLINEAR CONSTRAINT OPTIMIZATION THEORY .....	45

5. IMPLEMENTATION ON HELICOPTER SYSTEMS .....	49
5.1. Implementation with Linear Model Data .....	52
6. NUMERICAL RESULTS .....	71
6.1. Case 1-Interior Point algorithm .....	71
6.2. Case 2-SQP algorithm .....	88
6.3. Case 3 – Interior Point vs. SQP under asymmetric constraint conditions.....	101
6.4. Case 4 Arbitrary initial value for T matrix .....	108
6.5. Case 5- Constraint with additional physical properties .....	133
6.6. Discussion on Results.....	147
6.6.1. Comparison of Algorithm Type: IP or SQP .....	147
6.6.2. Comparison of Constraint Levels ([-10% 10%] ... [-90% 90%])).....	148
6.6.3. Discussion on the Results Regarding the Initial Values of T Matrix .....	148
6.6.4. Discussion on the Case with Additional Physical Constraints .....	148
6.6.5. Discussion on Parameter Accuracy .....	149
6.7. Implementation for Nonlinear Simulation .....	150
6.8. Implementation to Real Flight Test Data .....	153
7. CONCLUSIONS AND FUTURE WORK.....	157
7.1. Conclusions .....	157
7.2. Future Work .....	159
REFERENCES.....	161
CURRICULUM VITAE .....	183

## LIST OF TABLES

### TABLES

Table 3.1. Derivatives with Expected Values to Ensure Stability.....	43
Table 5.1. Eigenvalue Comparison.....	65
Table 6.1. Summary of Analysis Conditions for Case 1 (Interior Point algorithm)..	72
Table 6.2. Percentage Estimation Errors of Stability Derivatives – Case 1 .....	82
Table 6.3. Percentage Estimation Errors of Control Derivatives – Case 1.....	83
Table 6.4. TIC Values for Case 1 (for Verification Signal) .....	86
Table 6.5. Summary of Analysis Conditions for Case 2 (SQP algorithm).....	88
Table 6.6. Percentage Estimation Errors of Stability Derivatives – Case 2 .....	96
Table 6.7. Percentage Estimation Errors of Control Derivatives – Case 2.....	97
Table 6.8. TIC Values for Case 2 (for Verification Signal) .....	99
Table 6.9. Summary of Analysis Conditions for Case 3 (Interior Point vs. SQP under asymmetric constraint conditions).....	102
Table 6.10. Summary of Analysis for Case 4 (Arbitrary Initial Point for T Matrix) .....	109
Table 6.11. Percentage Estimation Errors of Stability Derivatives – Case 4 .....	128
Table 6.12. Percentage Estimation Errors of Control Derivatives – Case 4.....	129
Table 6.13. TIC Values for Case 4 (for Verification Signal) .....	131
Table 6.14. Summary of Analysis Conditions for Case 5 (Constraint with additional physical properties).....	134
Table 6.15. Percentage Estimation Errors of Stability Derivatives – Case 5 .....	142
Table 6.16. Percentage Estimation Errors of Control Derivatives – Case 5.....	143
Table 6.17. TIC Values for Case 5 (for Verification Signal) .....	145

## LIST OF FIGURES

### FIGURES

Figure 1.1. Schematic Illustration of the Basic Differences between Classical Methods and Subspace Methods [19] .....	4
Figure 2.1. Discrete-time and time-invariant State Space Model of Dynamic Systems .....	13
Figure 2.2. Orthogonal Projection Representation in a Simple Form .....	15
Figure 2.3. Oblique Projection Representation in a Simple Form .....	16
Figure 2.4. Geometric Representation of $Y_f$ in terms of $X_f$ and $U_f$ .....	23
Figure 2.5. An overview of the deterministic subspace identification procedure [19] .....	26
Figure 5.1. FLME Interface of FLIGHTLAB .....	50
Figure 5.2. Xanalysis Interface of FLIGHTLAB for Trim Analysis .....	52
Figure 5.3. Xanalysis Interface of FLIGHTLAB for Linearization .....	53
Figure 5.4. Input Signals (3-2-1-1) .....	56
Figure 5.5. Single-Sided Amplitude Spectrum of Each Input Signal .....	57
Figure 5.6. Output Signals .....	58
Figure 5.7. Coherences between Lateral Cyclic Input and the System Outputs .....	61
Figure 5.8. Coherences between Longitudinal Cyclic Input and the System Outputs .....	62
Figure 5.9. Coherences between Pedal Input and the System Outputs .....	63
Figure 5.10. Coherences between Collective Input and the System Outputs .....	64
Figure 5.11. True Model Outputs vs. Outputs of the Model Generated by Subspace Identification .....	66
Figure 5.12. The Symbolic Representation of the Objective Function .....	67
Figure 5.13. Physical Subspace Identification Procedure with Simulation Model Generated Data .....	70

Figure 6.1. Convergence of Stability Derivative “Xu” (Case 1.1 – Case 1.3).....	73
Figure 6.2. Convergence of Stability Derivatives (Case 1.1 – Case 1.3) .....	74
Figure 6.3. Convergence of Control Derivatives (Case 1.1 – Case 1.3).....	75
Figure 6.4. Convergence of Stability Derivatives (Case 1.4 – Case 1.6) .....	76
Figure 6.5. Convergence of Control Derivatives (Case 1.4 – Case 1.6).....	77
Figure 6.6. Convergence of Stability Derivatives (Case 1.7 – Case 1.9) .....	78
Figure 6.7. Convergence of Control Derivatives (Case 1.7 – Case 1.9).....	79
Figure 6.8. Convergence of $N\delta_{ped}$ for One Optimization Run .....	81
Figure 6.9. Convergence of $N\delta_{ped}$ for Twenty Optimization Runs .....	81
Figure 6.10. Comparison of Outputs for 3-2-1-1 Excitation Signals (Identification Using Linear Model Simulation .....	84
Figure 6.11. Verification Input Signals (Doublet).....	86
Figure 6.12. Comparison of Outputs for Doublet Excitation Signals (Identification Using Linear Model Simulation) .....	87
Figure 6.13. Convergence of Stability Derivatives (Case 2.1 – Case 2.3) .....	90
Figure 6.14. Convergence of Control Derivatives (Case 2.1 – Case 2.3).....	91
Figure 6.15. Convergence of Stability Derivatives (Case 2.4 – Case 2.6) .....	92
Figure 6.16. Convergence of Control Derivatives (Case 2.4 – Case 2.6).....	93
Figure 6.17. Convergence of Stability Derivatives (Case 2.7 – Case 2.9) .....	94
Figure 6.18. Convergence of Control Derivatives (Case 2.7 – Case 2.9).....	95
Figure 6.19. Comparison of Outputs for 3-2-1-1 Excitation Signals (Identification Using Linear Model Simulation .....	98
Figure 6.20. Comparison of Outputs for Doublet Excitation Signals (Identification Using Linear Model Simulation) .....	100
Figure 6.21. Convergence of Stability Derivatives (Case 3.1 – Case 3.3) .....	103
Figure 6.22. Convergence of Control Derivatives (Case 3.1 – Case 3.3).....	104
Figure 6.23. Convergence of Stability Derivatives (Case 3.4 – Case 3.6) .....	105
Figure 6.24. Convergence of Control Derivatives (Case 3.4 – Case 3.6).....	106
Figure 6.25. Comparison in Convergence of Control Derivatives (Case3).....	107
Figure 6.26. Convergence of Stability Derivatives (Case 4.1 vs. Case 4.10).....	110



Figure 6.27. Convergence of Control Derivatives (Case 4.1 vs. Case 4.10) .....	111
Figure 6.28. Convergence of Stability Derivatives (Case 4.2 vs. Case 4.11) .....	112
Figure 6.29. Convergence of Control Derivatives (Case 4.1 vs. Case 4.11) .....	113
Figure 6.30. Convergence of Stability Derivatives (Case 4.3 vs. Case 4.12) .....	114
Figure 6.31. Convergence of Control Derivatives (Case 4.3 vs. Case 4.12) .....	115
Figure 6.32. Convergence of Stability Derivatives (Case 4.4 vs. Case 4.13) .....	116
Figure 6.33. Convergence of Control Derivatives (Case 4.4 vs. Case 4.13) .....	117
Figure 6.34. Convergence of Stability Derivatives (Case 4.5 vs. Case 4.14) .....	118
Figure 6.35. Convergence of Control Derivatives (Case 4.5 vs. Case 4.14) .....	119
Figure 6.36. Convergence of Stability Derivatives (Case 4.6 vs. Case 4.15) .....	120
Figure 6.37. Convergence of Control Derivatives (Case 4.6 vs. Case 4.15) .....	121
Figure 6.38. Convergence of Stability Derivatives (Case 4.7 vs. Case 4.16) .....	122
Figure 6.39. Convergence of Control Derivatives (Case 4.7 vs. Case 4.16) .....	123
Figure 6.40. Convergence of Stability Derivatives (Case 4.8 vs. Case 4.17) .....	124
Figure 6.41. Convergence of Control Derivatives (Case 4.8 vs. Case 4.17) .....	125
Figure 6.42. Convergence of Stability Derivatives (Case 4.9 vs. Case 4.18) .....	126
Figure 6.43. Convergence of Control Derivatives (Case 4.9 vs. Case 4.18) .....	127
Figure 6.44. Comparison of Outputs for 3-2-1-1 Excitation Signals (Identification Using Linear Model Simulation) .....	130
Figure 6.45. Comparison of Outputs for Doublet Excitation Signals (Identification Using Linear Model Simulation) .....	132
Figure 6.46. Convergence of Stability Derivatives (Case 5.1 vs. Case 5.4) .....	136
Figure 6.47. Convergence of Control Derivatives (Case 5.1 vs. Case 5.4) .....	137
Figure 6.48. Convergence of Stability Derivatives (Case 5.2 vs. Case 5.5) .....	138
Figure 6.49. Convergence of Control Derivatives (Case 5.2 vs. Case 5.5) .....	139
Figure 6.50. Convergence of Stability Derivatives (Case 5.3 vs. Case 5.6) .....	140
Figure 6.51. Convergence of Control Derivatives (Case 5.3 vs. Case 5.6) .....	141
Figure 6.52. Comparison of Outputs for 3-2-1-1 Excitation Signals (Identification Using Linear Model Simulation) .....	144

Figure 6.53. Comparison of Outputs for Doublet Excitation Signals (Identification Using Linear Model Simulation) .....	146
Figure 6.54. Comparison of Outputs for 3-2-1-1 Excitation Signals (Identification Using Nonlinear Model Simulation) .....	151
Figure 6.55. Comparison of Outputs for 3-2-1-1 Excitation Signals (Identification Using Nonlinear Model Simulation, for the order n is selected as 11).....	152
Figure 6.56. Input Signals of Real Flight Test.....	154
Figure 6.57. Comparison of Outputs for Longitudinal Cyclic Doublet Excitations (Identification Using Flight Test Data).....	155

## LIST OF ABBREVIATIONS

### ABBREVIATIONS

IP	: Interior-Point
LTI	: Linear Time Invariant
MOESP	: Multivariable Output-Error State Space
MIMO	: Multi Input Multi Output
N4SID	: Numerical Algorithms for Subspace State Space System Identification
NA	: Not Applicable
SQP	Sequential Quadratic Programming
TIC	Theil's inequality coefficient

## LIST OF SYMBOLS

### SYMBOLS

$u, v, w$	: translational velocity components (longitudinal, lateral, vertical), $\frac{ft}{s}$
$p, q, r$	: angular velocity components (roll, pitch, yaw), $\frac{deg}{s}$
$\phi, \theta$	: Euler angles (roll, pitch), $deg$
$A, B, C, D$	: state space matrices found by subspace identification
$A_{phy}, B_{phy}, \dots$	: state space matrices found by physical subspace identification
$X_u, X_v, X_w, \dots$	: force to translational velocity derivatives, $\frac{1}{s}$
$X_p, X_q, X_r, \dots$	: force to angular velocity derivatives, $\frac{ft}{deg.s}$
$L_u, L_v, L_w, \dots$	: moment to translational velocity derivatives, $\frac{deg}{ft.s}$
$L_p, L_q, L_r, \dots$	: moment to angular velocity derivatives, $\frac{1}{s}$
$X_{\delta_{lat}}, X_{\delta_{lon}}, \dots$	: force to control input derivatives, $\frac{ft}{s^2.\%}$
$L_{\delta_{lat}}, L_{\delta_{lon}}, \dots$	: force to control input derivatives, $\frac{deg}{s^2.\%}$
$g$	: gravity force, $\frac{ft}{s^2}$
$\rho$	: air density, $\frac{slug}{ft^3}$
$a_0$	: main rotor lift curve slope, $\frac{1}{rad}$
$\mu$	: advance ratio
$\Omega$	: main rotor speed

$R$	: main rotor radius, $ft$
$A_b$	: blade area, $ft^2$
$M_a$	: mass of helicopter, $lb$
$s$	: rotor solidity



# CHAPTER 1

## INTRODUCTION

The aim of this study is extraction of physical parameters from subspace identification results via optimization techniques. The implementation is realized on to a multi-role twin engine helicopter.

### 1.1. Literature Survey

The helicopters are composed of numerous interacting subsystems. These are main rotor, fuselage, engine, flight control system, empennage and tail rotor. The helicopter flight dynamics can be modeled as the combination of the inertial, aerodynamic and the flight control forces acting on corresponding interactive subsystems. The effects of these forces are varying with the flight conditions. The interactions between the helicopter subsystems cause nonlinearity in system dynamics and bring some difficulties in modeling of the helicopter flight. These nonlinearities may not be easily modeled by analytical or numerical ways. In that case, wind tunnel and the flight test practices may be necessary to predict the indeterminate dynamics. No matter wind tunnel testing is utilized or not, the final step for the validation of helicopter design is flight testing. Flight testing is also opportunity for the flight dynamist to predict the helicopter model more accurately. System identification which is the way of model determination from experimental data can be utilized in the scope of flight data analysis.

System identification is a foundational research field which is used to generate the dynamic model of a system by using input and output data set. System identification may be categorized mainly into three main groups: Output-Error Methods, Equation Error Methods and Subspace Identification methods.

The output error method is introduced in the 1960s. The objective of the Output Error Method is based on adjusting the values for the unknown parameters in the model to obtain the best possible fit between the measured output data,  $y_{meas}$  and the estimated model response,  $y_{est}$  (1). The best fit is obtained by iterating the model parameters. The measurement error noise matrix  $R$  is calculated as in (2).

$$e(t) = y_{meas}(t) - y_{est}(t) \quad (1)$$

$$R = \frac{1}{N} \sum [y_{meas}(t) - y_{est}(t)] \cdot [y_{meas}(t) - y_{est}(t)]^T \quad (2)$$

The minimum of the cost function  $J(\chi)$  with respect to the unknowns  $\chi$  is obtained by satisfying all first derivatives function  $\delta J(\chi)/\delta \chi$  are zero.

This leads to a set of nonlinear equations that can only be solved iteratively with the main steps:

- calculation of the cost function  $J$
- calculation of the matrix  $R$
- update of the unknowns  $\chi$  by minimizing the cost function  $J(\chi)$
- calculation of an output vector  $y_{est}$

Applications of output error method on helicopter systems are numerous. D. Banerjee and J.W. Harding [1] use flight test data to identify the AH-64 Apache attack helicopter by output error method. Kaletka [2] estimates the 6 DOF model of the BO 105 helicopter by utilizing Maximum Likelihood output error method. SA 330 Puma is another platform which is identified by output error method [3]

The equation error method is another widely used method in helicopter identification studies [4], [5] and [6]. In this method, the cost function which is defined directly in



terms of an input-output equation is minimized via least square techniques. For a system model given in (3), the equation error is defined as in (4) where the states, derivative of the states and the inputs are measurable ( $x_{meas}, \dot{x}_{meas}, u_{meas}$ ).

$$\dot{x} = Ax + Bu \quad (3)$$

$$e = \dot{x}_{meas} - Ax_{meas} + Bu_{meas} \quad (4)$$

There are many algorithms which utilize these methods. There are many system identification tools as well. CIPHER (Comprehensive Identification from Frequency Response) which is one of the a well-known one, has been studied on a wide range of helicopters like BO-105 [7] UH-60 [8], Yamaha R-50 [9], OH-58D [10], SH-2G [11] and AH-64 [12], [13] and on a quadrotor [14]. Higher order models for the Raptor 50 and Evolution EX small-scale UAV helicopters were identified in [15] and [16]. The frequency domain system identification method developed by Tischler and Remple [17] was implemented for R44 helicopter in [18].

As an alternative to these classical methods which are mentioned above, subspace identification [19] attracted attention in the helicopter design society recently. Subspace identification differs from the classical system identification methods in many aspects. In principle, the models of constructing sequences are different. In classical techniques, first the system matrices are obtained, and then the states are estimated. However, in case of subspace identification first the states are estimated directly from input-output data, then the system matrices are obtained.

The schematic illustration of these procedure differences is given in Figure 1.1. These differences bring some advantages such as computation accuracy and convergence.

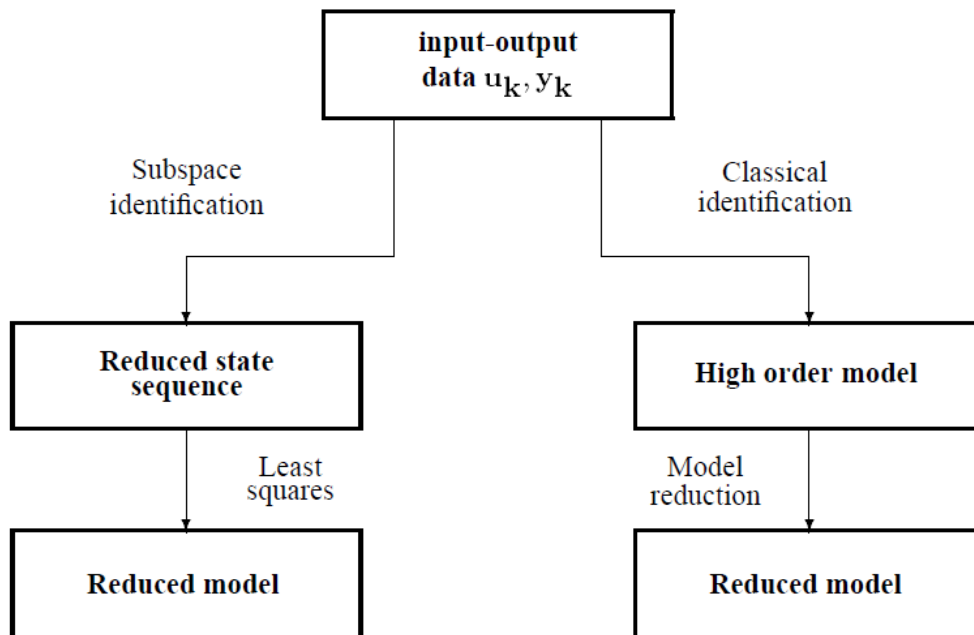


Figure 1.1. Schematic Illustration of the Basic Differences between Classical Methods and Subspace Methods [19]

Additionally, subspace identification algorithms utilize sound techniques based on linear algebra (orthogonal projections and intersections of subspaces) and numerical methods like QR and singular value decomposition. One of the main novelties of the subspace identification is to demonstrate how the Kalman filter states can be obtained from input-output data using linear algebra tools (QR and singular value decomposition). Hence, the identification problem becomes a linear least squares problem in the unknown system matrices. Another novelty is in the field of parameterization. Although classical system identification algorithms require a certain user-specified parameterization, subspace identification algorithms use full state space models and the only the order of the system is required as “parameter”. In fact, in subspace identification algorithms, the order of the system can be predicted by inspection of certain singular values.

By having the opportunity to obtain the reduced model directly from input-output data without having to compute the high order model, subspace identification algorithms always constitute models with as low order as possible.

The development of subspace identification algorithms is based on 1990s. The system model identification idea from the concepts like between subspaces or the singular value decomposition which are seemingly unrelated are started to be combined cleverly in 1970s ([20]-[29]). Finally the complete algorithms are dated to 1990s to 2000s ([19], [30] and [31]). The detailed historical progress of subspace identification algorithms is explained in [19].

The current subspace algorithms such as [19] and [31] have proven extremely successful in dealing with the estimation of discrete-time state space models. One class of subspace identification algorithm which is called as “Multivariable Output-Error State sPace” (MOESP) [30] based on the idea of estimating a basis of the observability subspace directly from data. The other class of algorithm called as “Numerical algorithms for Subspace State Space System IDentification” (N4SID) relies on the estimation of the state sequence for the system as an intermediate step for the estimation of the state space model. The details of these algorithms with the extended versions are explained in [19] and [31].

The interest of helicopter design society on subspace identification methods arise in the last decade. Until now, some variants of subspace identification algorithms like N4SID ([19]-[32]) and MOESP ([33]-[35]) were applied on a number of helicopter simulation data. As a further step, a real flight test application is performed both for EH101 helicopter [36] and ACT/FHS the DLR’s research helicopter [37]. In [38], subspace identification methods were used for the identification of a helicopter including rotor and engine dynamics. In this study where DLR’s research helicopter ACT/FHS is analyzed, the subspace identification is utilized to assure the maximum

likelihood system identification results regarding the system order and eigenvalues. The MOESP method was applied for a small-scale unmanned rotorcraft model in [39].

These studies showed that subspace identification method can be an alternative for helicopter systems due to having many advantages like parameterization, convergence and model reduction [19] as mentioned above. On the other hand, the state space model matrices which are obtained from conventional subspace identification algorithms are not necessarily associated with the physical parameters [19]. Physical parameter estimation based on subspace identification for helicopter systems is still being investigated.

The main objective of this study is to obtain helicopter physical parameters from subspace identification results. There are previous studies on this problem ([40]-[42]) utilizing Laguerre filters to convert the discrete time state space models into continuous models. Another approach for finding the physical parameters from subspace identification results is “optimization”. As it is mentioned before, the system matrices  $A, B, C, D$  which are found by subspace identification method do not necessarily have a direct physical interpretation. However, they have a conceptual relevance [19]. The similarity transformation of a discrete LTI system,  $\bar{x} = T^{-1}x$  leads to a new set of state space matrices in Eq. (14) - (16) [43].

$$\bar{A} = T^{-1}AT \quad (5)$$

$$\bar{B} = T^{-1}B \quad (6)$$

$$\bar{C} = CT \quad (7)$$

The aim is to find the  $\bar{A}, \bar{B}, \bar{C}$  matrices with the similarity transformation matrix  $T$  which lead us to the physical parameters.

The objective function (8) is defined as the sum squares of the difference between the right and left side of the similarity transformation equations Eq. (5)-(7) where the equality and the inequality constraints are determined with physical insight.

$$\begin{aligned} \min_x f(x) = \min_x (&\|\bar{A}(x) - T(x)^{-1}AT(x)\|_F + \|\bar{B}(x) - T(x)^{-1}B\|_F \\ &+ \|\bar{C}(x) - CT(x)\|_F) \end{aligned} \quad (8)$$

There are a limited number of studies in the literature which tackle with the identification problem by this approach ([44]-[49]). These studies propose simple cost functions (like least squares or quadratic) and most of them are applied for relatively simple systems like inverted pendulum or mass spring systems. However, helicopter identification may require far more variables to be solved. Therefore, it may require more advanced optimization algorithms. Due to the nonlinear characteristics of the objective function, this problem can be handled by NonLinear Programming (NLP) ([44]-[46]). In the literature, there exist a number of algorithms for solving NLP problems. In our case, we decided to concentrate on “large-scale” NLP algorithms where the total number of variables is greater than one hundred. In [50], Benson compares this type of algorithms in terms of efficiency. The preliminary study investigated in [51] is about estimating helicopter physical parameters from subspace identification. “Interior Point” algorithm was used in [51] to solve the aforementioned optimization problem. Regarding the “large-scale” NLP algorithm performance examination by Benson in [50], we selected both “Interior Point” algorithm and “Sequential Quadratic Programming” algorithm for our optimization problem.

Other improvements are made on constraint and initial value selections by introducing a variety of conditions. In addition, in [51], the required data is only obtained from linear model simulations. However, in this study, the methodology is extended for nonlinear model simulation data and real flight test data of a tactical helicopter (multi-role helicopter). The outcome will be critical in many aspects like model

improvement, wind tunnel data validation and flight controller design. The conducted study is summarized in the following paragraph.

As a first step, data gathering is studied. The excitation signal is selected as 3-2-1-1, which is commonly recommended for helicopter practices [17]. The input signal frequency content and amplitude are adjusted for exciting the helicopter body dynamics properly [17]. This will ensure the quality of system identification. As a second step N4SID algorithm [19] is applied to obtain the state-space model of the helicopter for a certain flight condition. The obtained state-space matrices, which are not necessarily associated with the physical states, are required to be transformed into the physical state-space matrices in order to obtain the stability and control derivatives. The similarity transformation theory [44] is utilized in conjunction with constrained nonlinear optimization for this purpose. The objective function, sum of the square of similarity transformation equations ([44]-[49]) is minimized to obtain the physical state-space matrices and the corresponding similarity transformation matrix.

In the above summarized optimization problem, two algorithms are experimented. These are “Sequential Quadratic Programming” and “Interior Point” ([52]-[54]) algorithms. They are selected considering both the size of the unknowns (over one hundred in this problem) and the nonlinear form of the objective function. The implementation of these algorithms for extracting physical parameters from subspace identification results is a candidate for being a prime in the field of helicopter flight dynamics.

Initial values of the parameters in optimization experiments are selected randomly. Constraints of the physical parameters may be selected considering typical error budgets of wind tunnel testing and aerodynamic prediction tools for helicopter systems or by using common practices of aerospace vehicle modeling. However, for convenience, constraints are selected considering linearized outputs of FLIGHTLAB with several error margins from [-10% 10%] to [-90% 90%] in this study. The physical

value for each parameter is estimated by minimizing the objective function via the above-mentioned algorithms under given constraints and initial values. The optimizations are repeated for different initial conditions in order to increase the confidence level. The percentage estimation errors are calculated for each parameter in every run. The physical parameters obtained at the end of the optimization are used for time domain verification of the model. The time domain output of the model is compared with the actual measurement using Theil's inequality coefficient (TIC) metric ([55], [56]). After obtaining promising results for the methodology, the study is repeated for the same helicopter by using nonlinear simulation data and real flight test data.

## **1.2 Main Contributions**

There are three main contributions of this study. The first one is successful implementation of subspace identification to a multi-role twin engine helicopter. The second contribution is the extraction of physical parameters from subspace identification results via optimization techniques. The third major contribution is the application of identification inputs in all four input channels of the helicopter for the same test case to extract both direct and cross coupling derivatives by utilizing subspace identification. With these contributions we are aiming to enhance the implementation of subspace identification technique for helicopters.

## **1.3 Organization of the Thesis**

The main objective of this study is presented in Chapter 1. Then the literature survey is introduced to define the problem properly. Background information on system identification techniques are presented here. Research is concentrated on the problem of physical parameter estimation from subspace identification results. The nonlinear optimization methodologies are searched for solving the problem. Then the main contributions are summarized in this chapter.

In Chapter 2, the theory of subspace identification is introduced. The main focus is given on a robust subspace algorithm which proved to work well on practical data. The problem of finding physical parameter estimation and the similarity transformation approach are mentioned here.

In Chapter 3, the model-structure of the helicopter is defined for our specific problem. The inputs, outputs and the states are introduced here. The main assumptions about the stability and control derivatives are also mentioned in this chapter.

The optimization procedure and the objective function are explained in Chapter 4. Both “Sequential Quadratic Programming” and “Interior Point” algorithms are briefly mentioned here. The formation of the objective function in accordance with the model structure is defined here. The basic assumptions about constraints, the initial values and the constants are also given in this chapter.

In Chapter 5, the implementation of the methodology is introduced on a multi-role helicopter. Data gathering, subspace identification and parameter estimation methodology is introduced for given specific flight conditions.

In Chapter 6, several conditions for the selected algorithm, boundary condition and initial values are examined, and the numerical results are presented. The implementation for nonlinear simulation output and real flight test data are presented here too.

Finally, the obtained results and the outcome of the research are discussed in Chapter 7.



## CHAPTER 2

### SUBSPACE IDENTIFICATION THEORY

#### 2.1. Introduction to Linear System Identification

Dynamic systems can be expressed as mathematical models. These models are used for design improvement, simulation, analysis and training. Typically, they provide advantageous in many circumstances where the real system testing is too expensive, too difficult or too time consuming. Especially for the rotorcrafts systems, it is almost impossible to validate a flight control system without mathematical model verifications on the ground since the challenging maneuvers performed in the air may contain in serious risks.

The mathematical model of the air vehicle systems can be derived from nonlinear equations-of-motion, typically by implementing several simplifying assumptions. However, for relatively more complex systems like rotorcrafts, a simplified model may not be sufficient for simulating the performance of the final implementation. In such a case the mathematical model improvement can be achieved via system identification techniques. In general practice, the collected input and output data is used to find parameters of predefined model structure. In this context, system identification is described as dynamic extension of curve fitting. In the final step, the model is validated with the experimental data which were not used in the system identification experiment.

Being an alternative to classical system identification methods subspace identification is based on finding the state space models by using only experimental input-output data set. The general overview of subspace identification is supplied in the following paragraph.

## 2.2. State Space Representation of Dynamic Systems

In subspace identification method, the dynamic model of the system is restricted to discrete time, linear, time-invariant, state space models. Mathematically, these models are defined by the following set of difference equations<sup>1</sup>:

$$x_{k+1} = Ax_k + Bu_k + w_k \quad (9)$$

$$y_k = Cx_k + Du_k + v_k \quad (10)$$

with

$$E \left[ \begin{pmatrix} w_p \\ v_p \end{pmatrix} \begin{pmatrix} w_q^T & v_q^T \end{pmatrix} \right] = \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \delta_{pq} \geq 0 \quad (11)$$

In this model, the vectors  $u_k \in \mathbb{R}^m$  and  $y_k \in \mathbb{R}^l$  are the input and outputs at time instant  $k$ . The size of the input and output are denoted  $m$  and  $l$  respectively.

The state vector of the process at discrete time instant  $k$  is represented as  $x_k \in \mathbb{R}^n$  where the size of it is equal to  $n$ .  $v_k \in \mathbb{R}^l$  and  $w_k \in \mathbb{R}^n$  are stochastic signals. It is assumed that they are zero mean, stationary, white noise vector sequences.

$A \in \mathbb{R}^{n \times n}$  matrix which is called as system matrix describes the dynamics of the system.  $B \in \mathbb{R}^{n \times m}$  is called as input matrix which represents the linear transformation by which the control inputs influence states in the next time step.  $C \in \mathbb{R}^{l \times n}$  is the output matrix which describes the linear relation between the states and the outputs (measurements)  $y_k$ . The  $D \in \mathbb{R}^{l \times m}$  matrix is the direct feedthrough term. In general practice, this term is most often 0 for continuous time systems. The covariance

---

<sup>1</sup> E denotes the expected value operator and  $\delta_{pq}$  the Kronecker delta.

matrices of the process and measurement noise sequences  $v_k \in \mathbb{R}^l$  and  $w_k \in \mathbb{R}^n$  are called as  $Q \in \mathbb{R}^{n \times n}$ ,  $S \in \mathbb{R}^{n \times l}$  and  $R \in \mathbb{R}^{l \times l}$ .

In subspace identification theory, the matrix pair  $\{A, C\}$  is assumed to be observable. It means that all modes in the system can be observed in the output,  $y_k$ . Also, the matrix pair  $\{A, [B \ Q^{\frac{1}{2}}]\}$  is required to be controllable for subspace identification algorithms. In other words, all modes of the system are excited by either the deterministic input  $u_k$  and/or the stochastic input  $w_k$  [19].

The graphical representation of discrete time and time-invariant state space model is given in Figure 2.1.

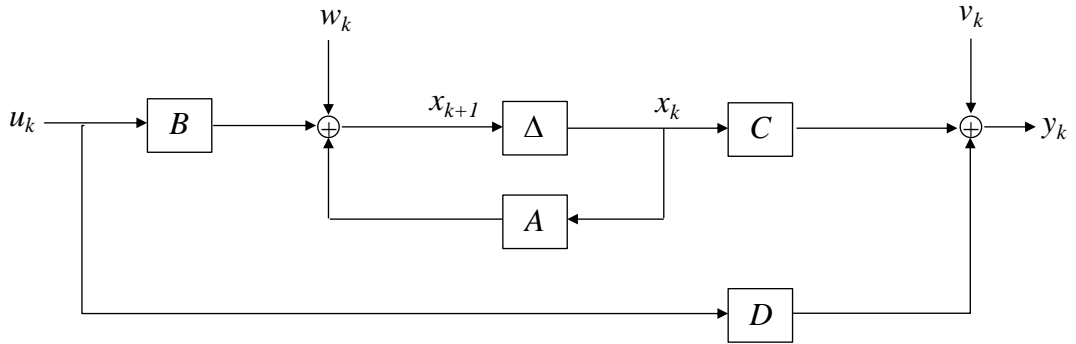


Figure 2.1. Discrete-time and time-invariant State Space Model of Dynamic Systems

### 2.3. Brief Overview on Subspace Identification Theory [19]

Subspace identification theory is based on system theory, linear algebra and statistics. The linear algebra tools (QR and singular value decomposition) are utilized to find the system states. Once these states are known, the problem turns into a linear least squares problem. Since the problem converted into a linear form, it can be solved easier when compared to “classical” prediction error methods [57].

In subspace identification algorithms, only the order of the system requires to be selected as a parameter. However, for classical methods, there has been an extensive amount of research to determine so-called canonical models, i.e. models with a minimum number of parameters. The problems arise from with these minimal parameterizations are listed in detail in [19].

Due to the order reduction and consequently the matrix size reduction accomplished by using QR and singular value decompositions, the subspace identification algorithms are quite fast. Also, they are faster than (again) the “classical” identification methods, such as Prediction Error Methods, because they are not iterative. By extension, there are no convergence problems. In addition to all of these, numerical robustness is guaranteed since subspace identification algorithms are proven by numerical linear algebra.

## 2.4. Geometric Tools [19]

Subspace identification algorithms use of several geometric tools. These geometric tools are defined in the following paragraphs. The matrices  $A \in \mathbb{R}^{p \times j}$ ,  $B \in \mathbb{R}^{q \times j}$  and  $C \in \mathbb{R}^{r \times j}$  are assumed to be given here below.

### 2.4.1. Orthogonal Projections

The projection of the row space of a matrix  $B \in \mathbb{R}^{q \times j}$  onto the row space of  $B \in \mathbb{R}^{q \times j}$  is denoted as  $\Pi_B$  where the related equation is given (12).

$$\Pi_B \stackrel{\text{def}}{=} B^T \cdot (BB^T)^\dagger \cdot B, \quad (12)$$

The operator “ $( )^\dagger$ ” denotes the Moore-Penrose pseudo-inverse of the matrix. The projection of the row space of the matrix  $A \in \mathbb{R}^{p \times j}$  on the row space of the matrix  $B \in \mathbb{R}^{q \times j}$  is given in (13). Here the boldface notation denotes the row space onto which one projects.

$$A / \mathbf{B} \stackrel{\text{def}}{=} A \cdot \prod_B = A \cdot B^T \cdot (BB^T)^\dagger \cdot B \quad (13)$$

The orthogonal projection of the row space of the matrix  $A \in \mathbb{R}^{p \times j}$  on the row space of the matrix  $B \in \mathbb{R}^{q \times j}$  is illustrated in a simplest way in Figure 2.2 to make it more understandable.

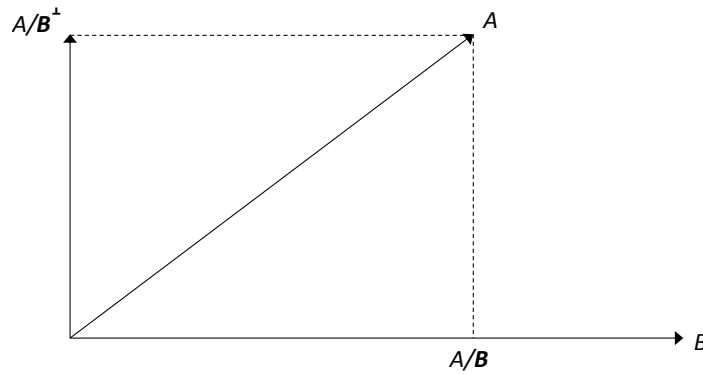


Figure 2.2. Orthogonal Projection Representation in a Simple Form2

#### 2.4.2. Oblique Projections

Oblique projection is projection of a matrix onto the linear combinations of two non-orthogonal matrices. In other words, project the row space of  $A$  orthogonally on the joint row space of  $B$  and  $C$ ; and decompose the result along the row space of  $C$ . Mathematically, the orthogonal projection of the row space of  $A$  on the joint row space of  $B$  and  $C$  can be given in (14).

---

<sup>2</sup>  $( )^\perp$  denotes orthogonal complement of the row space of  $( )$ .

$$A / \begin{pmatrix} C \\ B \end{pmatrix} \stackrel{\text{def}}{=} A \cdot (C^T \quad B^T) \cdot \begin{pmatrix} CC^T & CB^T \\ BC^T & BB^T \end{pmatrix}^\dagger \cdot \begin{pmatrix} C \\ B \end{pmatrix} \quad (14)$$

The oblique projection the row space of the matrix  $A$  along the row space of the matrix  $B$  onto the row space of the matrix  $C$  is illustrated in a simplest way in Figure 2.3 to make it more understandable.

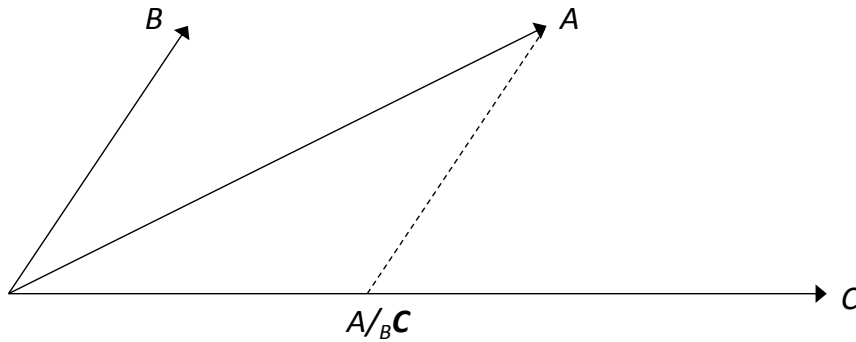


Figure 2.3. Oblique Projection Representation in a Simple Form3

### 2.4.3. Singular Value Decomposition

The order of the system is determined according to the results of Singular Value Decomposition (SVD). Singular values are another expression of the principal angles and directions.

#### *Definition 1 Principal angles and directions*

For given two matrices  $A \in \mathbb{R}^{p \times j}$ ,  $B \in \mathbb{R}^{q \times j}$ , the singular value decomposition is expressed as in (15).

$$A^T \cdot (AA^T)^\dagger \cdot AB^T \cdot (BB^T)^\dagger \cdot B = USV^T \quad (15)$$

---

<sup>3</sup>  $(\cdot)^\perp$  denotes orthogonal complement of the row space of  $(\cdot)$ .

Here the principal directions between the row spaces of  $A$  and  $B$  matrices are equivalent with the rows of  $U^T$  and the rows of  $V^T$ . Also the row spaces of  $A$  and  $B$  matrices are related with the singular values (the diagonal of  $S$ ) which is equal to the cosines of the principal angles in between them. The formulas are as given in (16).

$$\begin{aligned}
[A \nabla B] &\stackrel{\text{def}}{=} U^T \\
[A \nabla B] &\stackrel{\text{def}}{=} V^T \\
[A \nabla B] &\stackrel{\text{def}}{=} S^T
\end{aligned} \tag{16}$$

#### 2.4.4. Statistical Approaches

In subspace identification algorithms, some statistical assumptions work very well if large number of data is available. In subspace identification it is assumed that there are infinitely many data sets available ( $j \rightarrow \infty$ ) and that the data is “ergodic”.

To make it more illustrative, consider that there are two given sequences  $a_k \in \mathbb{R}^{n_a}$  and  $e_k \in \mathbb{R}^{n_e}$ ,  $k = 0, 1, \dots, j$ . The sequence  $e_k$  is a zero mean sequence as given in Eq.(17) and independent from  $a_k$  (Eq. (18))

$$E[e_k] = 0 \tag{17}$$

$$E[a_k e_k] = 0 \tag{18}$$

Due to the assumptions of ergodicity and the infinite number of data can be driven: The expectation operator  $E$  which is the average over an infinite number of experiments can be replaced with the operator  $E_j$  which is applied to the sum of variables. For instance, for the correlation between  $a_k$  and  $e_k$  we get:

$$E[a_k e_k] = \lim_{j \rightarrow \infty} \left[ \frac{1}{j} \sum_{i=0}^j a_i e_i^T \right] \quad (19)$$

$$= E_j \left[ \sum_{i=0}^j a_i e_i^T \right] \quad (20)$$

Here the operator  $E_j$  is defined as in (21).

$$E_j[\blacksquare] \stackrel{\text{def}}{=} \lim_{j \rightarrow \infty} \frac{1}{j} [\blacksquare] \quad (21)$$

In the light of this information, consider two given sequences of input  $u_k$  and noise  $e_k$ :

$$u \stackrel{\text{def}}{=} (u_0 \ u_1 \ \cdots \ u_j) \quad (22)$$

$$e \stackrel{\text{def}}{=} (e_0 \ e_1 \ \cdots \ e_j) \quad (23)$$

It can be found that the expected value of the sum of the vectors  $u$  and  $e$  becomes zero as it is seen in Eq. (24) under the assumptions that an infinite number of data available (a large set of data samples) and the data are ergodic.

$$E_j \left[ \sum_{i=0}^j u_i e_i^T \right] = 0 \quad (24)$$

Eq. (24) leads to

$$E_j[u \cdot e^T] = 0 \quad (25)$$

Geometric interpretation of this result is that the input vector  $u$  is perpendicular to the noise vector  $e$ . This is a precious inference because it is used in subspace identification algorithms to subtract the noise effects.



$$E_j[\|e/u\|] = 0 \quad (26)$$

## 2.5. Subspace Identification for Deterministic Systems

The subspace identification problem is formulated for deterministic LTI systems which are not disturbed by noise. Let such a system be given by

$$x_{k+1} = Ax_k + Bu_k, \quad (27)$$

$$y_k = Cx_k + Du_k, \quad (28)$$

where  $x_k \in \mathbb{R}^n$ ,  $u_k \in \mathbb{R}^m$  and  $y_k \in \mathbb{R}^l$ . The process noise  $w_k$  and the measurement noise  $v_k$  do not exist for deterministic systems. The goal of subspace identification is to find the state space matrices  $A, B, C, D$  and the state vector  $x_k \in \mathbb{R}^n$  up to a similarity transformation in the presence of the input  $u_k \in \mathbb{R}^m$  and output data set  $y_k \in \mathbb{R}^l$ .

It is definite that most of the real systems contain noise. However, we start to explain the theory of subspace identification from a deterministic system which is easier to understand. Subspace identification starts with the construction of Block Hankel matrices.

### 2.5.1. Blok Hankel Matrices

Block Hankel matrices are constituted from the input-output data set. The form of the input Block Hankel matrix is given in (29).

$$U_{0|2i-1} \stackrel{\text{def}}{=} \begin{bmatrix} u_0 & u_1 & u_2 & u_3 & \cdot & \cdot & \cdot & u_{j-1} \\ u_1 & u_2 & u_3 & u_4 & \cdot & \cdot & \cdot & u_j \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ u_i & u_{i+1} & u_{i+2} & u_{i+3} & \cdot & \cdot & \cdot & u_{j+i-1} \\ u_{i+1} & u_{i+2} & u_{i+3} & u_{i+4} & \cdot & \cdot & \cdot & u_{j+i} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ u_{2i-1} & u_{2i} & u_{2i+1} & u_{2i+2} & \cdot & \cdot & \cdot & u_{2i+j-2} \end{bmatrix} \quad (29)$$

Here the number of block index  $i$  defines the approximate order of the system being identified. The only requirement for  $i$  being larger than the estimated order of the system. The number of columns ( $j$ ) is formulated as in (30) where  $s$  is the total number of data samples.

$$j = s - 2i + 1 \quad (30)$$

The subscript of  $U_{0|2i-1}$  denote the first and last element of the first column in the block Hankel matrix of inputs.  $U_{0|2i-1}$  is combination of two equally sized matrices. By definition, the first part of the matrix  $U_{0|i}$  is called as Block Hankel matrices of past inputs. Other representation of this matrix is  $U_p$  where the subscript “ $p$ ” stands for “past”. The second part of the matrix  $U_{i|2i-1}$  is called as Block Hankel matrices of future inputs. Similarly, it is symbolized as  $U_f$  where the subscript “ $f$ ” stands for “future”. Another Block Hankel matrices used in subspace identification theory are  $U_p^+$  and  $U_f^-$  are obtained by shifting the borders of  $U_p$  and  $U_f$ .

This notation which is adopted from [64] and [19] also holds for the other Block Hankel matrices of the outputs:  $Y_{0|2i-1}$ ,  $Y_{0|i}$ ,  $Y_{i|2i-1}$ ,  $Y_p$ ,  $Y_f$ ,  $Y_p^+$  and  $Y_f^-$ .  $Y_{0|i}$  ( $Y_p$ ) is called as Block Hankel matrices of past inputs.  $Y_{i|2i-1}$  ( $Y_f$ ) is called as Block Hankel

matrices of future inputs. Also,  $Y_p^+$  and  $Y_f^-$  are obtained by shifting the borders of  $Y_p$  and  $Y_f$ .

$$Y_{0|2i-1} = \begin{bmatrix} y_0 & y_1 & y_2 & y_3 & \cdot & \cdot & \cdot & y_{j-1} \\ y_1 & y_2 & y_3 & y_4 & \cdot & \cdot & \cdot & y_j \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ y_i & y_{i+1} & y_{i+2} & y_{i+3} & \cdot & \cdot & \cdot & y_{j+i-1} \\ y_{i+1} & y_{i+2} & y_{i+3} & y_{i+4} & \cdot & \cdot & \cdot & y_{j+i} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ y_{2i-1} & y_{2i} & y_{2i+1} & y_{2i+2} & \cdot & \cdot & \cdot & y_{2i+j-2} \end{bmatrix} \quad (31)$$

Another Block Hankel matrix which is formulated in (32) is  $W_p$  obtained by combining the past inputs and outputs.

$$W_{0|i-1} \stackrel{\text{def}}{=} \begin{pmatrix} U_{0|i-1} \\ Y_{0|i-1} \end{pmatrix} = \begin{pmatrix} U_p \\ Y_p \end{pmatrix} = W_p \quad (32)$$

$W_p^+$  matrix is obtained by shifting the borders of  $U_p$  and  $Y_p$  as defined in Eq. (33).

$$W_p^+ = \begin{pmatrix} U_p^+ \\ Y_p^+ \end{pmatrix} \quad (33)$$

### 2.5.2. State Sequence Matrix

Another important matrix is "state sequence matrix"  $X_i$  for subspace identification algorithms. The state sequence matrix,  $X_i$  is shown in Eq. (34) as more explicit form. In this equation,  $i$  stands for the subscript of the first element of the state sequence.

$$X_i = (x_i \quad x_{i+1} \quad x_{i+2} \quad x_{i+3} \quad \dots \quad x_{i+j-1}) \in \mathbb{R}^{n \times j} \quad (34)$$

### 2.5.3. System Related Matrices

The extended observability matrix  $\Gamma_i$  is extensively used in subspace identification algorithms. Explicit representation of the observability matrix  $\Gamma_i$  is shown as in Eq.(35). Here it is assumed that the system is observable.

$$\Gamma_i \stackrel{\text{def}}{=} \begin{pmatrix} C \\ CA \\ CA^2 \\ \dots \\ CA^{i-1} \end{pmatrix} \in \mathbb{R}^{li \times n} \quad (35)$$

The reversed extended controllability matrix  $\Delta_i$  (where the subscript  $i$  denotes the number of block columns) represented as:

$$\Delta_i \stackrel{\text{def}}{=} (A^{i-1}B \quad A^{i-2}B \quad A^{i-3}B \quad A^{i-4}B \quad \dots \quad B) \in \mathbb{R}^{n \times mi} \quad (36)$$

The lower block triangular Toeplitz matrix  $H_i$  is defined as:

$$H_i \stackrel{\text{def}}{=} \begin{pmatrix} D & 0 & 0 & \dots & 0 \\ CB & D & 0 & \dots & 0 \\ CAB & CB & D & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ CA^{i-2}B & CA^{i-3}B & CA^{i-4}B & \dots & D \end{pmatrix} \in \mathbb{R}^{li \times mi} \quad (37)$$

Here it is assumed that the system is controllable.

### 2.5.4. Main Theorems for Deterministic Subspace Identification Algorithms

#### **Theorem 1 Matrix input-output equations ([19], Chapter 2)**

The following Theorem which is widely used in subspace identification algorithms states how the linear state space relations of formula (27) - (28) can be reformulated

in a matrix form. The proof of (38)-(40) follows directly from state space equations given in (27) -(28).

$$Y_p = \Gamma_i X_p + H_i U_p \quad (38)$$

$$Y_f = \Gamma_i X_f + H_i U_f \quad (39)$$

$$X_f = A^i X_p + \Delta_i U_p \quad (40)$$

The geometric representation of the equation (39) is given in Figure 2.4. Here the vectors in the row space of  $Y_f$  are obtained as a sum of linear combinations of vectors in the row space of  $X_f$  and linear combinations of vectors in the row space of  $U_f$ .

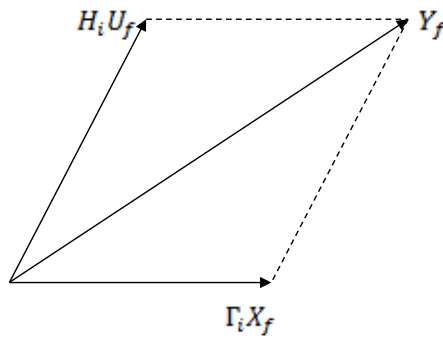


Figure 2.4. Geometric Representation of  $Y_f$  in terms of  $X_f$  and  $U_f$

### **Main Theorem ([19], Chapter 2)**

This theorem indicates that the state sequence  $X_i$  and the extended observability matrix  $\Gamma_i$  can be extracted from the given input-output data set  $(U_p, U_f, Y_p, Y_f)$ .

This can be expressed in two ways:

- The state sequence  $X_f$  can be determined directly from the given input-output data, without knowledge of the system matrices  $A, B, C$  and  $D$ .
- The extended observability matrix  $\Gamma_i$  can be determined directly from the given input-output data.

**Definition 2: Persistency of excitation ([19], Chapter 2)**

The input sequence  $u_k \in \mathbb{R}^{l_i \times n}$  is persistently exciting of order  $2i$  if the input covariance matrix (41) has full rank.

$$R_{uu} \stackrel{\text{def}}{=} \Phi[U_{0|2i-1}, U_{0|2i-1}] \quad (41)$$

**Theorem 2: Deterministic Identification ([19], Chapter 2)**

Under the assumptions that:

1. The input  $u_k$  is persistently exciting of order  $2i$  (Definition 2).
2. The intersection of the row space of  $U_f$  (the future inputs) and the row space of  $X_p$  (the past states) is empty.
3. The user-defined weighting matrices  $W_1 \in \mathbb{R}^{l_i \times l_i}$  and  $W_2 \in \mathbb{R}^{j \times j}$  are such that  $W_1$  is full rank and  $W_2$  obeys:  $\text{rank}(W_p) = \text{rank}(W_p \cdot W_2)$  where  $W_p$  is the block Hankel matrix containing the past inputs and outputs.

and  $O_i$  is defined as the oblique projection:

$$O_i \stackrel{\text{def}}{=} Y_f /_{U_f} W_p \quad (42)$$

and the singular value decomposition:

$$W_1 O_i W_2 = (U_1 \quad U_2) \begin{pmatrix} S_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} \quad (43)$$

$$= U_1 S_1 V_1^T \quad (44)$$

We have

1. The matrix  $O_i$  is equal to the product of the extended observability matrix and the states:

$$O_i = \Gamma_i X_i \quad (45)$$

2. The order of the system (27)- (28) is equal to the number of singular values in equation (43) different from zero.

3. The extended observability matrix  $\Gamma_i$  is equal to:

$$\Gamma_i = W_1^{-1} U_1 S_1^{1/2} T \quad (46)$$

where  $T \in \mathbb{R}^{n \times n}$  is an arbitrary non-singular similarity transformation.

4. The part of the state sequence  $X_i$  that lies in the column space of  $W_2$  can be recovered from:

$$X_i W_2 = T^{-1} S_1^{1/2} V_1^T \quad (47)$$

5. The state sequence  $X_i$  is equal to:

$$X_i = \Gamma_i^\dagger O_i \quad (48)$$

The proof of the Theorem which provides some insight in how subspace identification results are typically derived is given in Appendix A.

### 2.5.5. Algorithms for Deterministic Systems

According to [19], the system matrices  $A, B, C$  and  $D$  can be computed in two different ways:

- finding the state sequence matrix  $X_i$

- finding the extended observability matrix  $\Gamma_i$

The schematic illustration deterministic identification procedure is presented in Figure 2.5. The system matrices can be computed by using any of the two ways.

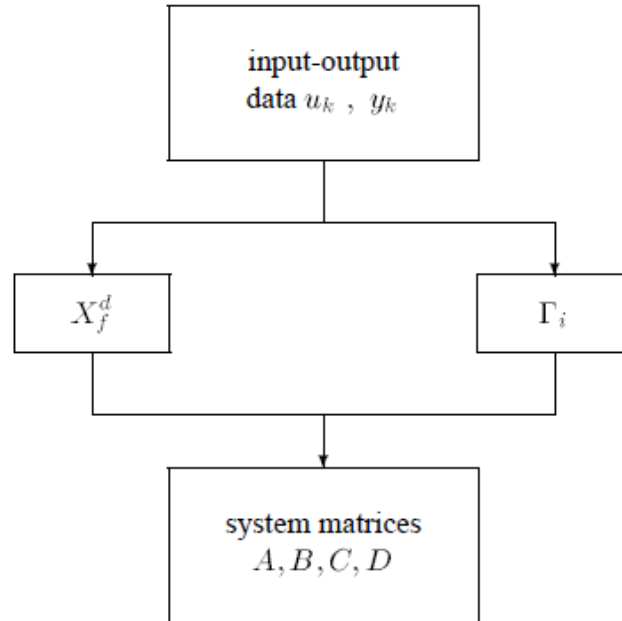


Figure 2.5. An overview of the deterministic subspace identification procedure [19]

The algorithms which are based on Theorem 2 will be outlined in the following paragraphs.

### 2.5.5.1. Algorithm 1 ([19], Chapter 2)

The algorithm starts with oblique projection  $O_i$  calculations.

$$O_{i-1} \stackrel{\text{def}}{=} Y_f^- / U_f^- W_p^+ \quad (49)$$

$$= \Gamma_{i-1} X_{i+1} \quad (50)$$

Then  $X_{i+1}$  can be calculated from (50) as below:



$$X_{i+1} = \Gamma_{i-1}^\dagger O_{i-1} \quad (51)$$

Then with  $U_{i|i}$  and  $Y_{i|i}$  matrices which are obtained from input-output data and the state sequence matrices calculated by the equations (48) and (51) we can estimate the system matrices A, B, C and D according to least square approach (52).

$$\begin{pmatrix} X_{i+1} \\ Y_{i|i} \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} X_i \\ U_{i|i} \end{pmatrix} \quad (52)$$

With this approximation, it is possible to solve all of the system matrices in one step. However, in the following algorithm these matrices are estimated in two separate steps. First A, C matrices are determined then the rest of the system matrices (B, D) are estimated. The algorithm steps are explained in the following paragraph.

#### 2.5.5.2. Algorithm 2 ([19], Chapter 2)

After calculating the extended observability matrix  $\Gamma_i$ , the system matrices are determined in two separate steps.

##### *Determination of A and C matrices*

The matrices A and C can be determined from the extended observability matrix  $\Gamma_i$ . The shift structure of the matrix  $\Gamma_i$  is used for this purpose [66].

$$\underline{\Gamma_i} A = \overline{\Gamma_i} \quad (53)$$

Where  $\overline{\Gamma_i}$  denotes  $\Gamma_i$  without the first  $l$  (number of outputs) rows. Then the matrix A can be determined by equation (54). Then the C matrix is determined as the first  $l$  rows of  $\Gamma_i$

$$A = \underline{\Gamma_i}^\dagger \cdot \overline{\Gamma_i} \quad (54)$$

##### *Determination of B and D matrices*

In the next stage,  $B$  and  $D$  matrices are computed. When we rearrange the input-output equation (39), we find that,

$$\Gamma_i^\perp Y_f = \Gamma_i^\perp H_i U_f \quad (55)$$

Where  $\Gamma_i^\perp \in \mathbb{R}^{(li-n) \times li}$  is a full row rank matrix satisfying  $\Gamma_i^\perp \cdot \Gamma_i = 0$ .

If we multiply the equation (55) with  $U_f^\dagger$  we obtain (56) where the linear least square approach is applicable. Here the matrices  $\Gamma_i^\perp, Y_f, U_f^\dagger$  are all known matrices. The only unknown is  $H_i$  matrix which is the combination of known matrices  $A, C$  and the unknown matrices  $B, D$ .

$$\Gamma_i^\perp Y_f U_f^\dagger = \Gamma_i^\perp H_i \quad (56)$$

For simplicity, the left-hand side of the equation is symbolized with  $\mathcal{M}$  and  $\Gamma_i^\perp$  with  $\mathcal{L}$ . Then the equation (56) can be written as;

$$\begin{aligned} & (\mathcal{M}_1 \quad \mathcal{M}_2 \quad \dots \quad \mathcal{M}_i) \\ & = (\mathcal{L}_1 \quad \mathcal{L}_2 \quad \dots \quad \mathcal{L}_i) \begin{pmatrix} D & 0 & 0 & \dots & 0 \\ CB & D & 0 & \dots & 0 \\ CAB & CB & D & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ CA^{i-2}B & CA^{i-3}B & CA^{i-4}B & \dots & D \end{pmatrix} \end{aligned} \quad (57)$$

Where  $\mathcal{M}_k \in \mathbb{R}^{(li-n) \times m}$  and  $\mathcal{L}_k \in \mathbb{R}^{(li-n) \times 1}$ . Then the equation (57) is rewritten as

$$(\mathcal{M}_1 \quad \mathcal{M}_2 \quad \dots \quad \mathcal{M}_i) = \begin{pmatrix} \mathcal{L}_1 & \mathcal{L}_2 & \dots & \mathcal{L}_{i-1} & \mathcal{L}_i \\ \mathcal{L}_2 & \mathcal{L}_3 & \dots & \mathcal{L}_i & 0 \\ \mathcal{L}_3 & \mathcal{L}_4 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \mathcal{L}_i & 0 & \dots & 0 & 0 \end{pmatrix} \begin{pmatrix} I_l & 0 \\ 0 & \underline{\Gamma_i} \end{pmatrix} \begin{pmatrix} D \\ B \end{pmatrix} \quad (58)$$

Then  $B$  and  $D$  matrices formulated in (58) are computed with least square approach.

## 2.6. Facts on Real World Applications

So far, we discussed subspace identification algorithms for deterministic systems. However, the real system measurements generally contain noise terms. Therefore, for real-life applications, the systems (Eq. (9) - (10)) are modeled with the process and measurement noise  $w_k$  and  $v_k$ .

From many available subspace identification methods in literature [19] the ‘‘Robust Subspace Algorithm’’ is utilized on this study. This method is proved with many industrial data and it is suggested for practical applications ([19], Chapter 4, Algorithm 3). The required information about theory and the algorithm is given in the following paragraphs.

### Theorem 3 Matrix input-output equations ([19], Chapter 4)

The matrix input-output equations for the combined system (similar to the matrix input output equations (38) - (40)) are defined in the following Theorem.

$$Y_p = \Gamma_i X_p + H_i U_p + Y_p^s \quad (59)$$

$$Y_f = \Gamma_i X_f + H_i U_f + Y_f^s \quad (60)$$

$$X_f = A^i X_p + \Delta_i U_p \quad (61)$$

Where  $Y_p^s$  and  $Y_f^s$  are the stochastic terms.

#### Theorem 4 Orthogonal projection

Under the assumptions that:

1. The deterministic input  $u_k$  is uncorrelated with the process noise  $w_k$  and measurement noise  $v_k$
2. The input  $u_k$  is persistently exciting of order  $2i$  (Definition 2)
3. The number of measurements goes to infinity  $j \rightarrow \infty$
4. The process noise  $w_k$  and the measurement noise  $v_k$  are not identically zero.

Then

$$Z_i \stackrel{\text{def}}{=} Y_f / \begin{pmatrix} W_p \\ U_f \end{pmatrix} \quad (62)$$

$$= \Gamma_i \hat{X}_i + H_i U_f \quad (63)$$

With Kalman filter state sequence

$$\hat{X}_i \stackrel{\text{def}}{=} \hat{X}_{i[\hat{x}_0, P_0]} \quad (64)$$

Where  $\hat{X}_0$  is the initial state sequence matrix and  $P_0$  is initial state covariance matrix.

#### Optimal prediction:

Another projection matrix in the theory of robust subspace identification is the prediction matrix  $Z_i$  which can be computed from the input output data, without having the system matrices. The prediction matrix ( $Z_i$ ) is considered as an optimal prediction of the future output ( $Y_f$ ) on the subspace formed by Block Hankel matrices of past

inputs and outputs ( $W_p$ ) and the Block Hankel matrices of the future input ( $U_f$ ). The corresponding formulation is shown in Eq. (65).

$$Z_i = Y_f / \begin{pmatrix} W_p \\ U_f \end{pmatrix} \quad (65)$$

Here, it is proposed to combine the past ( $W_p$ ) and the future inputs ( $U_f$ ) linearly to predict the future outputs ( $Y_f$ ) [19].

### **Theorem 5 Combined Identification ([19], Chapter 4)**

*Under the assumptions that:*

1. *The deterministic input  $u_k$  is uncorrelated with the process noise  $w_k$  and measurement noise  $v_k$*
2. *The input  $u_k$  is persistently exciting of order  $2i$  (Definition 2)*
3. *The number of measurements goes to infinity  $j \rightarrow \infty$*
4. *The process noise  $w_k$  and the measurement noise  $v_k$  are not identically zero.*
5. *The user-defined weighting matrices  $W_1 \in \mathbb{R}^{li \times li}$  and  $W_2 \in \mathbb{R}^{j \times j}$  are such that  $W_1$  is full rank and  $W_2$  obeys:  $\text{rank}(W_p) = \text{rank}(W_p \cdot W_2)$  where  $W_p$  is the block Hankel matrix containing the past inputs and outputs.*

*and  $O_i$  is defined as the oblique projection:*

$$O_i \stackrel{\text{def}}{=} Y_f /_{U_f} W_p \quad (66)$$

*and the singular value decomposition:*

$$W_1 O_i W_2 = (U_1 \quad U_2) \begin{pmatrix} S_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} \quad (67)$$

We have

1. The matrix  $O_i$  is equal to the product of the extended observability matrix and the states:

$$O_i = \Gamma_i X_i \quad (68)$$

2. The order of the system (9)-(10) is equal to the number of singular values in equation (43) different from zero.
3. The extended observability matrix  $\Gamma_i$  is equal to:

$$\Gamma_i = W_1^{-1} U_1 S_1^{1/2} T \quad (69)$$

where  $T \in \mathbb{R}^{n \times n}$  is an arbitrary non-singular similarity transformation.

4. The part of the state sequence  $X_i$  that lies in the column space of  $W_2$  can be recovered from:

$$X_i W_2 = T^{-1} S_1^{1/2} V_1^T \quad (70)$$

5. The state sequence  $\tilde{X}_i$  is equal to:

$$\tilde{X}_i = \Gamma_i^\dagger O_i \quad (71)$$

### 2.6.1. Algorithm 3 – Robust Subspace Identification ([19], Chapter 4)

The algorithm starts with oblique projection  $O_i$  (66) and orthogonal projection  $Z_i$  (62),  $Z_{i+1}$  (72) calculations.

$$Z_{i+1} = Y_f^- / \begin{pmatrix} W_p^+ \\ U_f^- \end{pmatrix} \quad (72)$$

$$= \Gamma_{i-1} \hat{X}_{i+1} + H_{i-1} U_f^- \quad (73)$$

$$\hat{X}_{i+1} \stackrel{\text{def}}{=} \hat{X}_{i+1}[\hat{x}_0, P_0] \quad (74)$$

Reduction of the size of oblique projection matrix  $O_i$  will simplify the rest of the matrix operations. Thus, SVD is calculated for  $O_i$  (67). By inspecting the singular values the  $U_1$  and  $S_1$  matrices are calculated to determine extended observability matrix  $\Gamma_i$  (69). Here the weight matrices  $W_1$  and  $W_2$  are assumed as identity which is compatible with Theorem 5.

Then some set of linear equations which are quite complicated are solved to find  $A$  and  $C$  matrices. The equations (75)-(78) are used to generate the linear least square problem given in (79). Here  $\rho_v, \rho_w$  are the covariances of the process and measurement noise of the residuals and an intermediate matrix  $\kappa$  is given as in (80).

$$\hat{X}_{i+1} = A\hat{X}_i + BU_{i|i} + K_i(Y_{i|i} - C\hat{X}_i - DU_{i|i}) \quad (75)$$

$$Y_{i|i} = C\hat{X}_i + DU_{i|i} + (Y_{i|i} - C\hat{X}_i - DU_{i|i}) \quad (76)$$

$$\hat{X}_i = \Gamma_i^\dagger \cdot [Z_i - H_i U_f] \quad (77)$$

$$\hat{X}_{i+1} = \Gamma_{i-1}^\dagger \cdot [Z_{i+1} - H_{i-1} U_f^-] \quad (78)$$

$$\begin{pmatrix} \Gamma_{i-1}^\dagger Z_{i+1} \\ Y_{(i|i)} \end{pmatrix} = \begin{pmatrix} A \\ C \end{pmatrix} \Gamma_i^\dagger Z_i + \kappa U_f + \begin{pmatrix} \rho_w \\ \rho_v \end{pmatrix} \quad (79)$$

$$\kappa \stackrel{\text{def}}{=} \begin{pmatrix} B|\Gamma_{i-1}^\dagger H_{i-1} - A\Gamma_i^\dagger H_i \\ D|0 - C\Gamma_i^\dagger H_i \end{pmatrix} \quad (80)$$

After finding  $A$  and  $C$  matrices from (79), the  $\Gamma_i$  and  $\Gamma_{i-1}$  matrices are recomputed to get better estimation on the remaining system matrices  $B$  and  $D$  matrices where both  $\{A, C\}$  and  $\{\Gamma_i, \Gamma_{i-1}\}$  are used.

$$B, D = \underset{B, D}{\operatorname{argmin}} \left\| \begin{pmatrix} \Gamma_{i-1}^\dagger Z_{i+1} \\ Y(i|i) \end{pmatrix} - \begin{pmatrix} A \\ C \end{pmatrix} \Gamma_i^\dagger Z_i - \kappa(B, D) U_f \right\|_F \quad (81)$$

The matrices  $B$  and  $D$  are calculated by solving the minimization problem shown in Eq. (81). The intermediate steps are explained in [19] in more detail.

## 2.7. Similarity Transformation

The system matrices  $A, B, C, D$  found through the above given formulation does not necessarily have a direct physical interpretation but they have a conceptual relevance [19]. According to the similarity transformation theory [43] the state vector of a discrete LTI system can be transformed into another state vector. This is shown in Eq. (82).

$$x_{phy} = T^{-1}x \quad (82)$$

Such an operation leads to a new set of state space matrices as shown in Eq. (83)-(85). However,  $D$  and  $D_{phy}$  matrix are not state dependent. Therefore, these matrices are not included in the similarity transformation equations.

$$A_{phy} = T^{-1}AT \quad (83)$$

$$B_{phy} = T^{-1}B \quad (84)$$

$$C_{phy} = CT \quad (85)$$



In this study, our primary aim is to obtain the  $A_{phy}, B_{phy}, C_{phy}$  matrices with the corresponding similarity transformation matrix,  $T$  which is invertible. This will hopefully lead us to the physical parameters. Since both the physical system matrices and the similarity transformation matrix is lacking, a candidate solution is the minimization of the difference between the left hand side and the right hand side of Eq. (83)-(85).

$$TA_{phy} = AT \quad (86)$$

$$TB_{phy} = B \quad (87)$$

This can be achieved by an optimization that makes use of the lower bound of the sum squares of the difference between the right and left sides, (83)-(85). It may be important to mention that Eq. (83) and Eq. (84) are highly nonlinear and difficult to solve it [47]. The forms shown in Eq. (86) and Eq. (87) are chosen instead of Eq. (83) and Eq. (84) to reduce the difficulty.

This optimization problem may have infinitely many solutions if we do not define well-founded constraints. At this point, a good model structure proposition becomes crucial for the estimation of the state space model which is constructed from the physical parameters.



## CHAPTER 3

### MODEL STRUCTURE

The model structure determination is highly related with the dynamics concerned for identification. The model structure of an aerospace vehicle is usually obtained from the governing 6-DOF flight-dynamics equations. These equations inherently contain a substantial number of parameters required for validating mathematical models, wind tunnel test results and for tuning the flight controller gains. The 6-DOF nonlinear equations of motion for a helicopter can be written as shown in Eq. (88) - Eq. (93) [58] with forces and moments represented by the small perturbation theory. In these equations, the force derivatives are normalized by mass, and the moment derivatives are normalized by the corresponding moments of inertia. Moreover for the moment derivatives, a pre-multiplication by the inertia tensor has been carried out so that they implicitly include products of inertia terms (i.e.,  $I_{xy}$ ,  $I_{xz}$ , etc.) [58].

$$\dot{u} = X_u u + X_v v + X_w w + X_p p + (X_q - w_0)q + (X_r + v_0)r - g \cos \theta_0 \theta \quad (88)$$

$$+ X_{\delta_{lat}} \delta_{lat} + X_{\delta_{lon}} \delta_{lon} + X_{\delta_{ped}} \delta_{ped} + X_{\delta_{col}} \delta_{col}$$

$$\dot{v} = Y_u u + Y_v v + Y_w w + (Y_p + w_0)p + Y_q q + (Y_r - u_0)r \\ + g \cos \phi_0 \cos \theta_0 \phi - g \sin \phi_0 \sin \theta_0 \theta + Y_{\delta_{lat}} \delta_{lat} \quad (89)$$

$$+ Y_{\delta_{lon}} \delta_{lon} + Y_{\delta_{ped}} \delta_{ped} + Y_{\delta_{col}} \delta_{col}$$

$$\dot{w} = Z_u u + Z_v v + Z_w w + (Z_p - v_0)p + (Z_q + u_0)q + Z_r r \quad (90)$$

$$- g \sin \phi_0 \cos \theta_0 \phi - g \cos \phi_0 \sin \theta_0 \theta$$

$$+Z_{\delta_{lat}}\delta_{lat} + Z_{\delta_{lon}}\delta_{lon} + Z_{\delta_{ped}}\delta_{ped} + Z_{\delta_{col}}\delta_{col}$$

$$\begin{aligned} \dot{p} = & L_u u + L_v v + L_w w + L_p p + L_q q \\ & + L_r r + L_{\delta_{lat}}\delta_{lat} + L_{\delta_{lon}}\delta_{lon} + L_{\delta_{ped}}\delta_{ped} + L_{\delta_{col}}\delta_{col} \end{aligned} \quad (91)$$

$$\begin{aligned} \dot{q} = & M_u u + M_v v + M_w w + M_p p + M_q q + M_r r + M_{\delta_{lat}}\delta_{lat} + M_{\delta_{lon}}\delta_{lon} \\ & + M_{\delta_{ped}}\delta_{ped} + M_{\delta_{col}}\delta_{col} \end{aligned} \quad (92)$$

$$\begin{aligned} \dot{r} = & N_u u + N_v v + N_w w + N_p p + N_q q + N_r r + N_{\delta_{lat}}\delta_{lat} + N_{\delta_{lon}}\delta_{lon} \\ & + N_{\delta_{ped}}\delta_{ped} + N_{\delta_{col}}\delta_{col} \end{aligned} \quad (93)$$

$$\dot{\phi} = p + q \sin \phi_0 \tan \theta_0 + r \cos \phi_0 \tan \theta_0 \quad (94)$$

$$\dot{\theta} = q \cos \phi_0 - r \sin \theta_0 \quad (95)$$

These equations can be represented in the state space form as shown in Eq. (96) with the motion states and the controls inputs given in Eq. (97) and Eq. (98) respectively. The proposed model structure has 8 states and 4 inputs.

$$\dot{x}_{phy} = A_{phy} x_{phy} + B_{phy} u \quad (96)$$

$$x_{phy} = [u \quad v \quad w \quad p \quad q \quad r \quad \phi \quad \theta] \quad (97)$$

$$u = [\delta_{lat} \quad \delta_{lon} \quad \delta_{ped} \quad \delta_{col}] \quad (98)$$

The physical system matrices  $A_{phy}$  and  $B_{phy}$  are shown in Eq. (99) and Eq. (100) respectively.

$A_{phy}$

$$= \begin{bmatrix} X_u & X_v & X_w & X_p & X_q - w_0 & X_r + v_0 & 0 & -gc\theta_0 \\ Y_u & Y_v & Y_w & Y_p + w_0 & Y_q & Y_r - u_0 & gc\phi_0c\theta_0 & -gs\phi_0s\theta_0 \\ Z_u & Z_v & Z_w & Z_p - v_0 & Z_q + u_0 & Z_r & -gs\phi_0c\theta_0 & -gc\phi_0s\theta_0 \\ L_u & L_v & L_w & L_p & L_q & L_r & 0 & 0 \\ M_u & M_v & M_w & M_p & M_q & M_r & 0 & 0 \\ N_u & N_v & N_w & N_p & N_q & N_r & 0 & 0 \\ 0 & 0 & 0 & 1 & s\phi_0t\theta_0 & c\phi_0t\theta_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c\theta_0 & -s\theta_0 & 0 & 0 \end{bmatrix} \quad (99)$$

$$B_{phy} = \begin{bmatrix} X_{lat} & X_{lon} & X_{ped} & X_{col} \\ Y_{lat} & Y_{lon} & Y_{ped} & Y_{col} \\ Z_{lat} & Z_{lon} & Z_{ped} & Z_{col} \\ L_{lat} & L_{lon} & L_{ped} & L_{col} \\ M_{lat} & M_{lon} & L_{ped} & M_{col} \\ N_{lat} & N_{lon} & N_{ped} & N_{col} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (100)$$

When the rest of the state space matrices are concerned, with the assumption that all of the system states are perfectly measurable, the associated  $C_{phy}$  is an identity matrix and according to our problem formulation  $D_{phy}$  is equal to zero.

Since all of the states are assumed to be perfectly measurable, the total number of parameters to be estimated in  $A_{phy}$ ,  $B_{phy}$  and  $T$  matrices are 36, 24 and 64 respectively. Therefore, altogether there are 124 unknowns. Such a problem can be classified as an optimization problem with large number of variables. The solution methodology is explained in the following paragraph.

It is important to understand the physical behavior of these parameters prior to estimation. In other words, initial value assessment and constraint value selection with physical intuition is important to get a solution [45].

Each stability and control derivative are made up of a contribution from different components of a helicopter such as main rotor, tail rotor, fuselage, stabilizers etc. The significant stability and control derivatives are defined briefly by referencing [58] and [59] in the following paragraphs. The detail explanations with formulations and the illustrations are given in [58] and [59].

### 3.1. Derivative of Forces with respect to Translational Velocity Components

$(X_u, X_v, X_w, Y_u, Y_v, Y_w, Z_u, Z_v, Z_w)$

Perturbation in translational velocity changes the rotor flapping which causes change in forces and moments around rotor, fuselage and empennage. The derivatives  $X_u, Y_v, X_v, Y_u$  which are coupled at low speeds, becomes independent from each other with an increasing forward velocity [58]. Direct force damping  $X_u$  and  $Y_v$  which reflect the drag and side force on the rotor–fuselage combination respectively changes linearly with speed [58]. The coupling derivatives  $X_v$  and  $Y_u$  are less significant compared to the direct derivatives as it is expected.

The approximation for heave damping derivative can be written as in Eq. (18) for forward flight condition, [58].

$$Z_w = -\frac{\rho a_0 \mu (\Omega R) A_b}{2M_a} \left( \frac{4}{8\mu + a_0 s} \right) \quad (101)$$

### 3.2. Derivative of Forces with respect to Translational Velocity Components

$$(X_u, X_v, X_w, Y_u, Y_v, Y_w, Z_u, Z_v, Z_w)$$

The speed and incidence stability  $M_u, M_w$  have major effect on longitudinal stability. Although the main rotor moments do not significantly change with forward velocity, the pitching moment contributions of the fuselage and empennage become significant due to aerodynamic loads. Positive  $M_u$  indicates speed stability whereas negative  $M_w$  refers incidence stability [58].

Pitching moment due to sideslip  $M_v$  is also another important parameter. The changes in sideslip cause significant variations in downwash at the horizontal stabilizer [58].

The derivatives  $L_u, L_w, N_u, N_w$  couple with each other at the low-frequency longitudinal and lateral motions of the helicopter.

Dihedral effect ( $L_v$ ) and weathercock stability ( $N_v$ ) parameters are significant sideslip derivatives. A positive value for  $N_v$  implies stability. A negative value for  $L_v$  implies stability [58].

### 3.3. Derivative of Forces with respect to Translational Velocity Components

$$(X_u, X_v, X_w, Y_u, Y_v, Y_w, Z_u, Z_v, Z_w)$$

The derivatives  $X_q, Y_p$  change significantly by main rotor contributions. These derivatives contribute significantly to the pitch and roll damping characteristics [58].

### 3.4. Derivative of Moments with respect to Angular Velocity Components

$$(L_p, L_q, L_r, M_p, M_q, M_r, N_p, N_q, N_r)$$

The direct and coupled damping derivatives  $L_p, L_q, M_p, M_q$  are significantly important derivatives in system dynamics. The direct damping derivatives  $L_p, M_q$

indicate short-term, small and moderate amplitude, handling characteristics, while the cross-damping derivatives  $L_q$ ,  $M_p$  characterize the level of pitch–roll and roll–pitch couplings [58].

The derivatives  $L_r$ ,  $N_p$ ,  $N_r$  have influence on the character of the lateral / directional stability and control characteristics of the helicopter [58]. In general, the derivatives  $L_r$ ,  $N_p$  are presumed to be less significant compare to their primary counterparts  $L_p$ ,  $N_r$  [59].

### 3.5. General View on Stability Derivatives

The parameters which implicate powerful information about helicopter flight stability are tabulated in Table 3.1 with the expected values for satisfying stability [58]. Prior knowledge of the helicopter flight stability under examination may give hint about the sign of these derivatives.

As it is mentioned above, there exists 60 parameters to be estimated (36 parameters in  $A_{phy}$  and 24 parameters in  $B_{phy}$ ) in our problem. However, they are not all the same in the sense of significance. Some of them are quite insignificant compared to the others. In fact, these “insignificant” parameters vary from helicopter to helicopter due to their dynamic characteristics. Flight region is another factor determining the set of “insignificant” parameters. In common practice, the stability parameters  $X_v$ ,  $X_w$ ,  $Y_u$ ,  $Y_w$ ,  $Z_u$ ,  $Z_v$ ,  $M_w$ ,  $M_r$ ,  $N_w$  and the control parameters  $X_{\delta_{ped}}$ ,  $Y_{\delta_{ped}}$ ,  $M_{\delta_{ped}}$ ,  $Y_{\delta_{ped}}$  are assumed as “insignificant”. Therefore, these may set to zero. Moreover for flight conditions with high forward velocity (where the inertial velocities are so dominant) the aerodynamic effects may be negligible (e.g.,  $Z_q$ ,  $Y_r$ ) [58].



Table 3.1. Derivatives with Expected Values to Ensure Stability

Stability Criteria	Expected Value
Dihedral effect	$L_v < 0$
Roll damping	$L_p < 0$
Yaw to roll coupling	$L_r > 0$
Static speed stability	$M_u > 0$
Incidence stability	$M_w < 0$
Pitch damping	$M_q < 0$
Weathercock stability	$N_v > 0$
Adverse yaw	$N_p < 0$
Yaw damping	$N_r < 0$
Drag damping	$X_u < 0$
Side Force Damping	$Y_v < 0$
Heave Damping	$Z_w < 0$

### 3.6. Derivative of Forces with respect to Control Inputs ( $Z_{\delta_{col}}, Z_{\delta_{lon}}, Y_{\delta_{ped}}$ )

Heave control sensitivity ( $Z_{\delta_{col}}$ ) mainly affected by the blade loading and tip speed. The control sensitivity increases with forward speed [58]. The derivative of thrust with respect to longitudinal cyclic ( $Z_{\delta_{lon}}$ ) increases almost linearly with increasing speed.

The derivative of thrust with respect to the main rotor collective ( $Z_{\delta_{col}}$ ) and with respect to the longitudinal cyclic ( $Z_{\delta_{lon}}$ ) which can be obtained from the thrust and uniform inflow equations can be formulated as in Eq. (102) and in Eq. (103) [58].

$$Z_{\delta_{col}} = -\frac{4 a_0 A_b \rho (\Omega R)^2 \mu (1 + \mu^2)}{3 (8\mu + a_0 s) M_a} \quad (102)$$

$$Z_{\delta_{lon}} = -2 \frac{a_0 A_b \rho (\Omega R)^2 \mu^2}{(8\mu + a_0 s) M_a} \quad (103)$$

The side force is mainly affected by tail rotor thrust which changes directly with the pedal input ( $Y_{\delta_{ped}}$ ).

### 3.7. Derivative of Moments with respect to Control Inputs ( $M_{\delta_{col}}, L_{\delta_{col}}, N_{\delta_{ped}}, L_{\delta_{ped}}, L_{\delta_{lat}}, L_{lon}, M_{\delta_{lat}}, M_{\delta_{lon}}$ )

Changes in collective control may cause pitching and rolling moment ( $M_{\delta_{col}}, L_{\delta_{col}}$ ). In fact, the changes in rotor thrust may generate a moment if there exists a thrust offset. Moreover, the changes in flapping due to the collective input generate hub moment proportional to the flap angle.

The pedal input is directly related with tail rotor thrust which has significant impact on the yawing moment ( $N_{\delta_{ped}}$ ). The cross-coupling derivative ( $L_{\delta_{ped}}$ ) is also significant in rotor dynamics. Both  $N_{\delta_{ped}}, L_{\delta_{ped}}$  derivatives increase with forward speed [58].

The direct and coupled flap responses to cyclic control inputs ( $L_{\delta_{lat}}, L_{lon}, M_{\delta_{lat}}, M_{\delta_{lon}}$ ) change with the stiffness number; and they are practically independent of forward speed [58].

Understanding the behavior of these parameters under certain flight conditions will be useful in the estimation of their values. The initial value assessment and the constraint selections of the optimization problem whose explanation is presented in the following chapters are performed under the guidance of the information given in this chapter.

## CHAPTER 4

### PARAMETER ESTIMATION WITH NONLINEAR CONSTRAINT OPTIMIZATION THEORY

Consider an optimization problem;

$$\begin{aligned} & \text{minimize} && \min f(\chi) \\ & \text{subject to} && \chi \in \Omega \end{aligned} \tag{104}$$

The real-valued function  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  which is desired to be minimized is named as the objective function. The vector  $\chi$  is a vector with  $n$  independent variables:  $\chi = [x_1, x_2, x_3, \dots, x_n]^T \in \mathbb{R}^n$ . The set  $\Omega$  is a subset of  $\mathbb{R}^n$  called the constraint set or feasible set. In our problem, the objective function shown in Eq. (105) is the sum squares of the difference between the right and left side of the similarity transformation equations shown in Eq. (86), (87) and Eq. (85) ([44]-[46])

$$\begin{aligned} \min_{\chi} f(\chi) = \min_{\chi} & \left( \|T(\chi)A_{phy}(\chi) - AT(\chi)\|_F + \|T(\chi)B_{phy}(\chi) - B\|_F \right. \\ & \left. + \|C_{phy}(\chi) - CT(\chi)\|_F \right) \end{aligned} \tag{105}$$

In the literature, there exist a number of algorithms for solving NonLinear Programming (NLP) problems ([44]-[46]). In our case, we decided to concentrate on “large-scale” NLP algorithms where the total number of variables is greater than one hundred. In [50] Benson compares these types of algorithms in terms of efficiency. In this respect we utilized both the IP (Interior-Point) method and the SQP (Sequential Quadratic Programming) for our optimization problem.

The interior methods, which are also called as barrier methods, are used to transform a constrained problem into an unconstrained problem or into a sequence of unconstrained problems [35]. Interior Point Algorithms, in general sense, are based on searching the optimum solution by starting from an available point and continuing gradually to get better ones which lie in the interior points of the available area. Consider that our objective function  $f(\chi)$ , which is aimed to be minimized, is subjected to the constraint function,  $g(\chi) \leq 0$ . The barrier problem aims to find infimum of a function  $f(\chi) + \mu B(\chi)$  where  $g(\chi) < 0$ . Here  $B(\chi)$  is a barrier function that is nonnegative and continuous over the region  $\{\chi : g(\chi) < 0\}$  and approaches  $\infty$  as the boundary of the region  $\{\chi : g(\chi) \leq 0\}$  is approached from the interior. More specifically, problem is reformulated in Eq. (106) with *Frisch's logarithmic barrier function* [60] for each barrier parameter  $\mu > 0$ , and nonnegative slack variables,  $s_i$ .

$$\begin{aligned} \text{minimize} \quad & \min_{\chi, s} f(\chi) - \mu \sum_i \ln(s_i), \\ \text{subject to} \quad & g(\chi) + s = 0 \end{aligned} \tag{106}$$

As  $\mu$  converges to zero, the approximate problem (Eq. (106)) becomes a sequence of equality constrained problems which are easier to solve than the original inequality constrained problem.

SQP is also one of the most effective methods for nonlinearly constrained optimization problems [52]. It provides successful results for both small and large-scale problems. For SQP, we can express our optimization problem as given in (107).

$$\begin{aligned} \text{minimize} \quad & \min f(\chi) \\ \text{over} \quad & \chi \in \mathbb{R}^n \\ \text{subject to} \quad & h(\chi) = 0 \\ & g(\chi) \leq 0 \end{aligned} \tag{107}$$

The idea behind the SQP is to model the objective function at the current iterate  $\chi_k$  by a quadratic programming sub problem. Then a new iterate  $\chi_{k+1}$  is defined by minimizing the sub problem [52]. In general practice, the SQP methods are executed in two stages. These are step computation and the Hessian approximation. Merit function is used to ensure that the SQP method converges from remote starting points.

Both IP and SQP algorithms are readily available in the optimization toolbox of MATLAB with a wide variety options for the user. The *fmincon* solver of MATLAB is utilized as the optimization tool for our problem. The IP and SQP algorithms are utilized here under a variety of initial conditions and constraints.



## CHAPTER 5

### IMPLEMENTATION ON HELICOPTER SYSTEMS

The method described on the previous sections is implemented first on simulated flight data. For this purpose, we used a nonlinear model of a multi-role helicopter which was developed in FLIGHTLAB environment. The FLIGHTLAB Model Editor (FLME) is used for data entry. FLME structure (Figure 5.1) allowed us to allocate the data according to hierarchical modules that correspond to a physical or a logical subsystem of the helicopter. The model is composed of main rotor, tail rotor, airframe and flight control modules.

The “Main Rotor” is modeled with “Blade Element” approach. The number of blades, rotor radius, rotational speed with direction, rotor hub location, shaft tilt, swashplate phase angle and blade tip loss factor properties are supplied. The blade structure is selected as “Articulated”. The rigid blade model includes both flapping and lead-lag dynamics. The damper of the lead-lag dynamics is modeled as linear. The physical parameters like torque offset, rotor precone angle, precone / flapping / feathering/lead-lag hinge offset, flapping hinge / lag damper spring stiffness, flapping hinge / linear lag damper damping coefficient, flap / lag spring undeformed angle and effective delta-3 angle are modeled. The geometric / inertial blade is generated in many equally spaced segments. The aerodynamic data is generated by wind tunnel tests and the FLUENT analysis results are used as a complementary source. The main rotor air loads are represented with a quasi-unsteady aerodynamics model featuring stall delay due to rotation empirical corrections. The air load entry is performed according to the blade segments which are consistent with airfoil radial station positions. Peters-He Finite State model is selected for induced velocity model. The inflow harmonics are

selected as three. The inflow correction data is modeled from empirical data regarding the ground effect and wake distortion effect. Peters-He 3-state interference model is generated to simulate the main rotor interference on fuselage, tail rotor and tail surfaces.

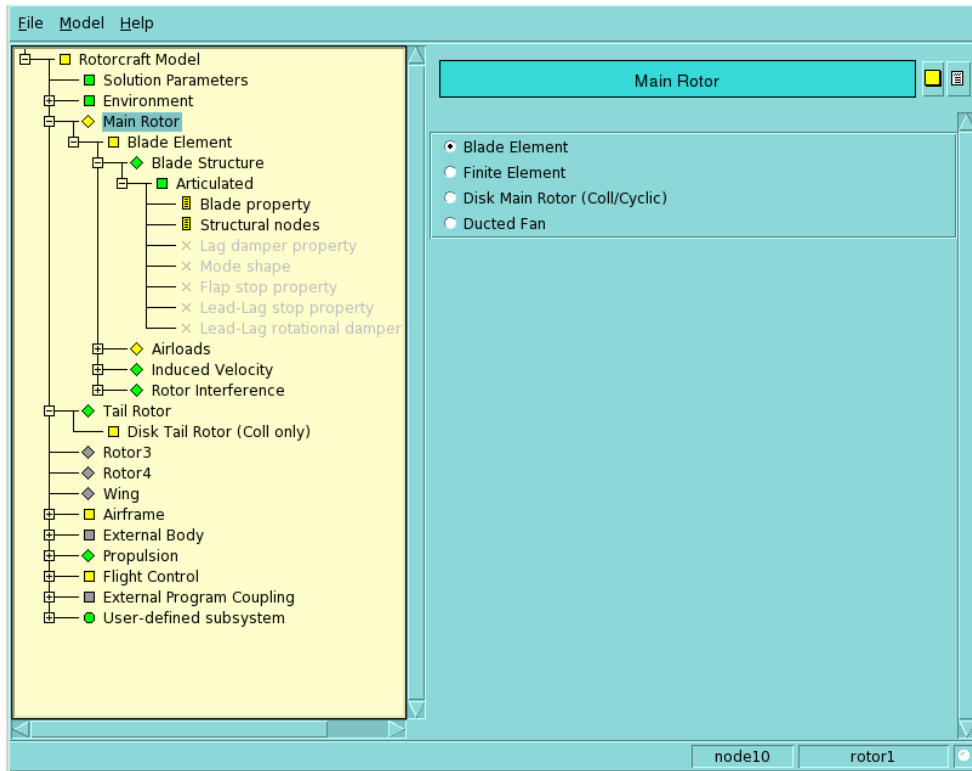


Figure 5.1. FLME Interface of FLIGHTLAB

“Actuator Disk Model” is used for “Tail Rotor” module. The blade properties like number of tail rotor blades, rotor radius, rotational speed with direction, hub location, cant angle, blade tip loss factor, lift curve slope, rotor head drag coefficient, effective rotor head drag area, airfoil constant drag coefficient, solidity weighted blade chord, linear blade twist, delta 3 angle, partial of coning with respect to thrust, blockage effect properties, inflow / profile drag correction, induced inflow / coning time constants are modeled in this module.



Airframe model is comprised of fuselage, horizontal tail and vertical tail components. The “Rigid Fuselage” model is generated with vehicle center of gravity, mass, moment of inertia in pitch, roll, yaw axis and the total product of inertias. All of the associated aerodynamic data belongs to wind tunnel test results and numerical analysis performed in FLUENT environment.

The flight stability augmentation system model is embedded in the flight control module. Rate feedback stabilization systems for roll, pitch and yaw channels are enabled to increase system stability. These stabilization systems also include the main rotor and tail rotor actuator models. Actuator models are linear.

The “ideal engine” model is selected for propulsion system. The number of engines, nominal engine torque and main rotor to engine gear ratio properties are produced as engine properties.

The FLIGHTLAB model is trimmed (Figure 5.2) and flight simulations are performed in several flight maneuvering conditions like hovering, forward flight, climb. The model is verified by comparing the simulation results with the related flight test data. Both simulations and the flight test efforts are repeated with several environmental conditions for verification.

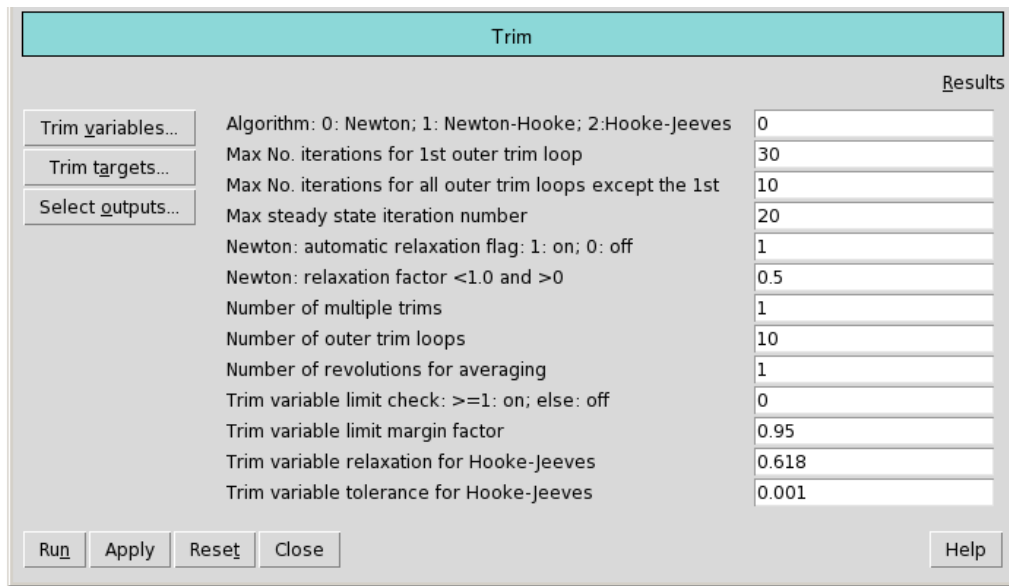


Figure 5.2. Xanalysis Interface of FLIGHTLAB for Trim Analysis

## 5.1. Implementation with Linear Model Data

Two linearization approaches are available in FLIGHTLAB [67]. These are “*averaged genq*” and “*steady perturbation*” (Figure 5.3). The “*averaged genq*” method estimates the stability and control matrices by perturbing the system model at each rotor azimuth. Then the resultant derivatives are by averaging the resulting partial derivatives over one rotor revolution. On the other hand, the “*steady perturbation*” method obtains the derivatives by perturbing the state or control, running the model to steady state, and then averaging the resulting partial derivatives over one rotor revolution [67].

Both methods are practiced in this study and the method selection is performed by comparing the linear model responses with the nonlinear model simulations. Since the “*averaged genq*” method gives better results, it is selected for our problem.

The nonlinear FLIGHTLAB model is linearized around a specific trim point (level flight at 70 knots, 2000 ft MSL). Linearization is performed again using the linearization feature of FLIGHTLAB Xanalysis interface.

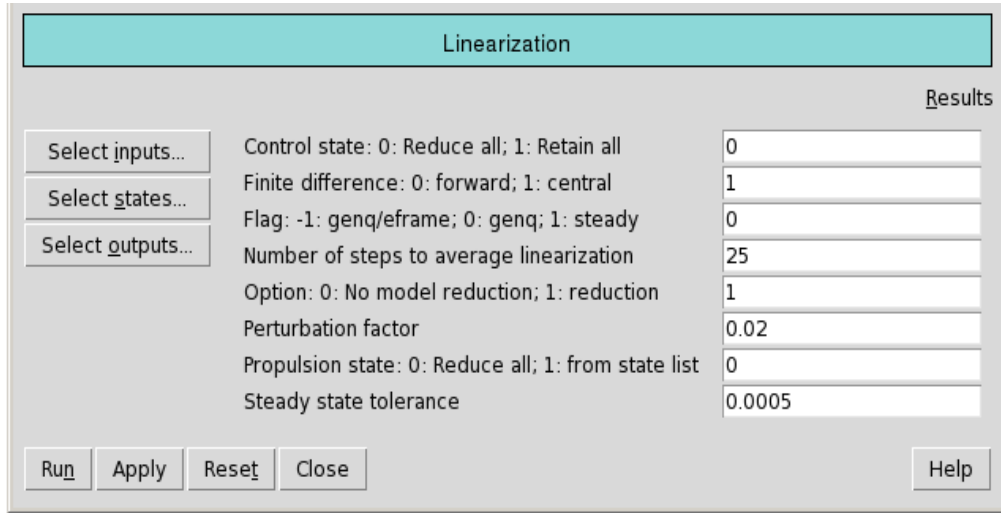


Figure 5.3. Xanalysis Interface of FLIGHTLAB for Linearization

Linear model configuration is selected from “Xanalysis” interface [67]. In this process, inputs (Eq. (109)) are selected as longitudinal cyclic, lateral cyclic, collective and pedal; the states are selected as (Eq. (110)) roll angle, pitch angle; translational velocity components (longitudinal, lateral and vertical velocity), and angular velocity components (roll, pitch and yaw rates). The outputs are assigned to the states (Eq. (111)).

$$\dot{x} = Ax + Bu \quad (108)$$

$$inputs \quad : \quad \delta_{lon}, \delta_{lat}, \delta_{col}, \delta_{ped} \quad (109)$$

$$states \quad : \quad \phi, \theta, u, v, w, p, q, r \quad (110)$$

$$outputs : \phi, \theta, u, v, w, p, q, r \quad (111)$$

The expanded form on the basis of the input-state-output configuration is given in Appendix B.

Model linearization is applicable around the trim point. Trim condition is determined according to the model verification status. Trim condition at 70 knot forward velocity where the model data coincide with the flight test results is selected as the initial condition for linearization. Pressure altitude is selected as 2000 ft and the ambient temperature is chosen as 15 °C. Appropriate trim targets and trim variables are set before trim analysis. For forward velocity condition body accelerations are set as trim targets. Pilot control inputs and the Euler angles are defined as trim variables which are released as free to solve trim equations. After trim analysis, linearization analysis is performed to calculate the linear model in FLIGHTLAB environment. Then the model structure is converted to the form given in Eq. (112). The transformation equations are defined in Appendix B.

$$\begin{bmatrix} \dot{u} \\ v \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \\ \dot{\phi} \\ \dot{\theta} \end{bmatrix} = A \cdot \begin{bmatrix} \phi \\ \theta \\ u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} + B \cdot \begin{bmatrix} \delta_{lat} \\ \delta_{lon} \\ \delta_{ped} \\ \delta_{col} \end{bmatrix} \quad (112)$$

where

A

$$= \begin{bmatrix} X_u & X_v & X_w & X_p & X_q - w_0 & X_r + v_0 & 0 & -gc\theta_0 \\ Y_u & Y_v & Y_w & Y_p + w_0 & Y_q & Y_r - u_0 & gc\phi_0 c\theta_0 & -gs\phi_0 s\theta_0 \\ Z_u & Z_v & Z_w & Z_p - v_0 & Z_q + u_0 & Z_r & -gs\phi_0 c\theta_0 & -gc\phi_0 s\theta_0 \\ L_u & L_v & L_w & L_p & L_q & L_r & 0 & 0 \\ M_u & M_v & M_w & M_p & M_q & M_r & 0 & 0 \\ N_u & N_v & N_w & N_p & N_q & N_r & 0 & 0 \\ 0 & 0 & 0 & 1 & s\phi_0 t\theta_0 & c\phi_0 t\theta_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c\theta_0 & -s\theta_0 & 0 & 0 \end{bmatrix} \quad (113)$$

$$B = \begin{bmatrix} X_{lat} & X_{lon} & X_{ped} & X_{col} \\ Y_{lat} & Y_{lon} & Y_{ped} & Y_{col} \\ Z_{lat} & Z_{lon} & Z_{ped} & Z_{col} \\ L_{lat} & L_{lon} & L_{ped} & L_{col} \\ M_{lat} & M_{lon} & M_{ped} & M_{col} \\ N_{lat} & N_{lon} & N_{ped} & N_{col} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (114)$$

The obtained linear model is transferred to MATLAB environment, which will be utilized for the rest of the analysis.

Now we shall proceed by generating input and output data required for identification. One of the most optimal input signal types which meet the well-known requirement of persistently exciting ([17], [37]) is 3-2-1-1. This input signal is sequentially applied for each channel during the same identification test. The input signal frequency content and amplitude shall be well adjusted for exciting the helicopter body dynamics properly ([17]). In the light of this, frequency content of the input signal is adjusted to cover a frequency range of 0.1-1 Hz. Signal to noise ratio is also taken into account while selecting the amplitudes of the input signals. Moreover, helicopter is not allowed to drift away from the trim condition too much ([17]). For this purpose, input signal amplitudes are limited in such a way that the helicopter attitude angles stay in the

range of  $\pm 10$  deg around the specific trim point and the helicopter angular velocity components shall not exceed  $\pm 10$  deg/s. These considerations about the input design are expected to ensure the quality of identification. Constructed signals for each input channel are shown in Figure 5.4. MATLAB ‘fft’ command is used to check the frequency content of the input signal. Single-sided amplitude spectrum of the input signals is shown in Figure 5.5.

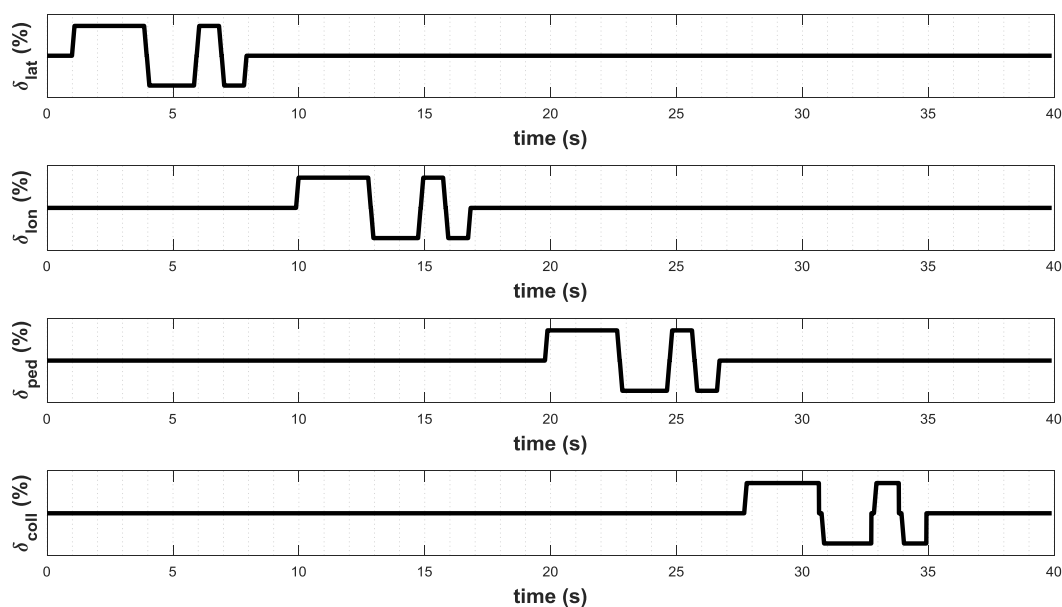


Figure 5.4. Input Signals (3-2-1-1)

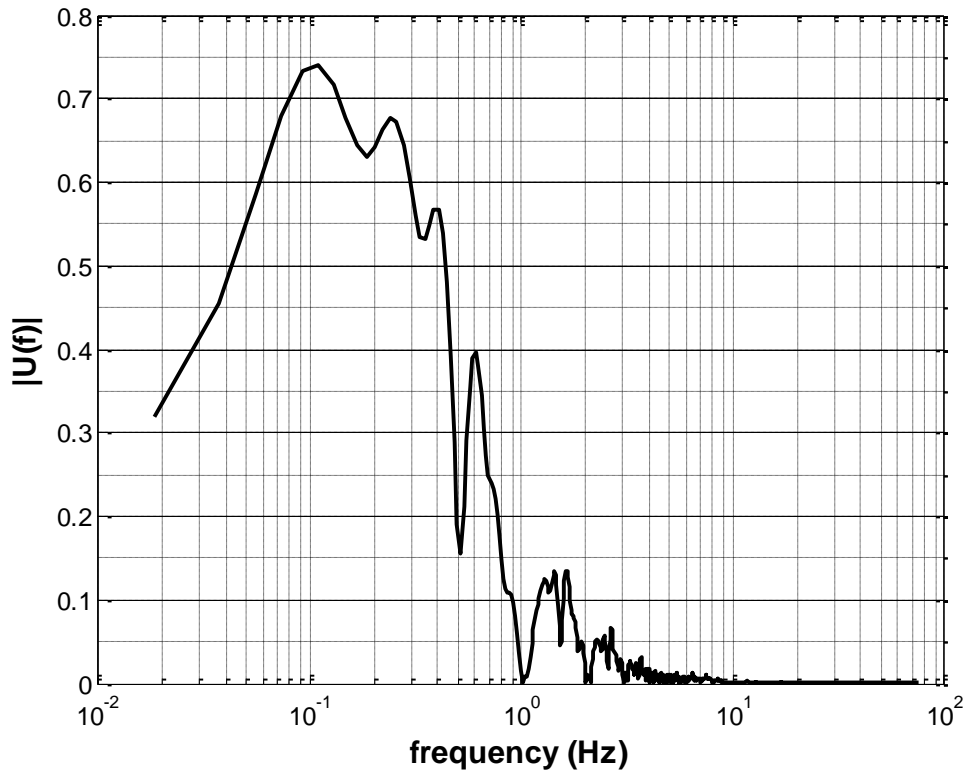


Figure 5.5. Single-Sided Amplitude Spectrum of Each Input Signal

The input signals are fed to the linear model in MATLAB environment to generate the outputs required for identification. The outputs were selected as roll angle, pitch angle; translational velocity components (longitudinal, lateral and vertical velocity), and angular velocity components (roll, pitch and yaw rates). Time domain responses of the system to the inputs given in Figure 5.4 are illustrated in Figure 5.6.

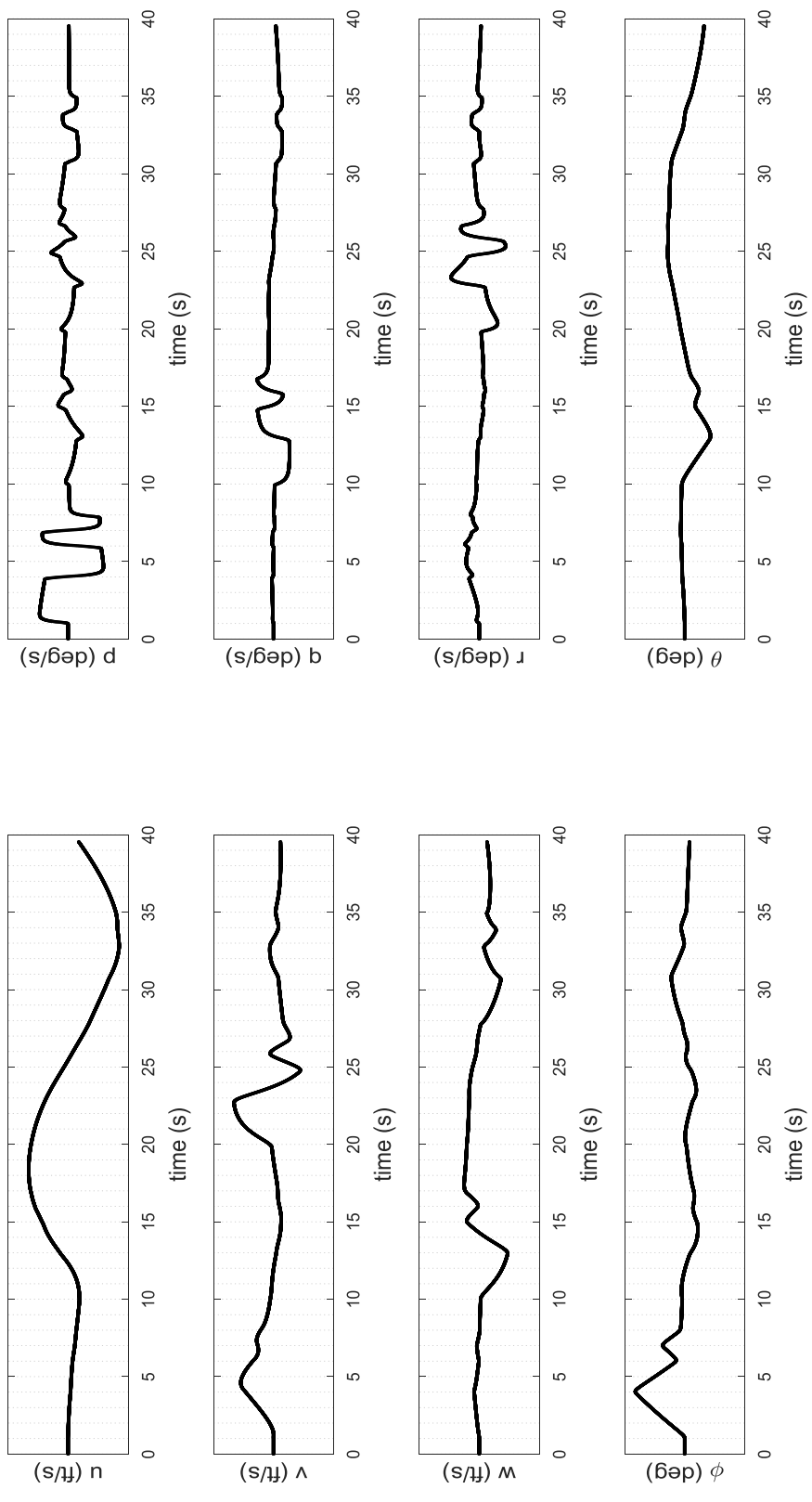


Figure 5.6. Output Signals



One important issue for input output data set is linearity. During the parametric estimation phase, the linearity between input output couples will give information about the accuracy of the corresponding parameters ([17]). A “*coherence*” value which is higher than 0.6 implies that there is satisfying system response linearity for the corresponding input frequencies ([17]). The equation of “*coherence function*” is given in (115).

$$\hat{\gamma}_{xy}^2(f) = \frac{|\hat{G}_{xy}(f)|^2}{|\hat{G}_{xx}(f)||\hat{G}_{yy}(f)|} \quad (115)$$

According to this formulation, the coherence values between inputs and the outputs for each signal are illustrated separately in Figure 5.7 - Figure 5.10. To clarify, if we look at Figure 5.7, we observe that the relation between lateral cyclic input and roll rate is satisfying for the desired frequency range (0.1-1 Hz). Same comment is valid for the lateral cyclic input and roll angle pair. These coherence plots given in Figure 5.7 - Figure 5.10 are associated with the physical parameters to be estimated in the following chapter.

Since we have an input and output set, we can initiate the identification process. The sampling time is selected as 1/150 to be compatible with real system sensor properties (see Paragraph 6.8). First the Block Hankel Matrices,  $(Y_f, U_f, W_p)$  which have significant importance in subspace identification, are constructed by using the available input and output data set. In our problem, the row number Block Hankel Matrices will be 16 ( $2n$  where  $n$  is selected as 8) and the column number will be 5985<sup>4</sup> (See Eq. (30)). The obtained Block Hankel Matrices will be used to compute the oblique projection matrix  $O_i$ . At this stage the RQ decomposition algorithm is utilized to reduce the computational load and to avoid rank deficiency during the projection operations. After obtaining the oblique projection matrix, SVD method is used to inspect and reduce the order of the system. The U and S matrices of SVD are used to compute the extended observability matrix,  $\Gamma_i$ . Moreover, the prediction matrix  $Z_i$  is computed again by using the Block Hankel Matrices. Finally,  $A$  and  $C$  matrices are computed by using Eq. (79);  $B$  and  $D$  matrices are computed by using Eq.

---

<sup>4</sup> For 40 s simulation data with 150 hz

(81). These discrete time linear model matrices are then converted into their continuous time counterparts using “d2c” command of MATLAB. “*tustin*” discretization method is used for this purpose. This “inverse-discretization” operation is required since we are seeking for the continuous time form of the system model in order to obtain the physical parameters.

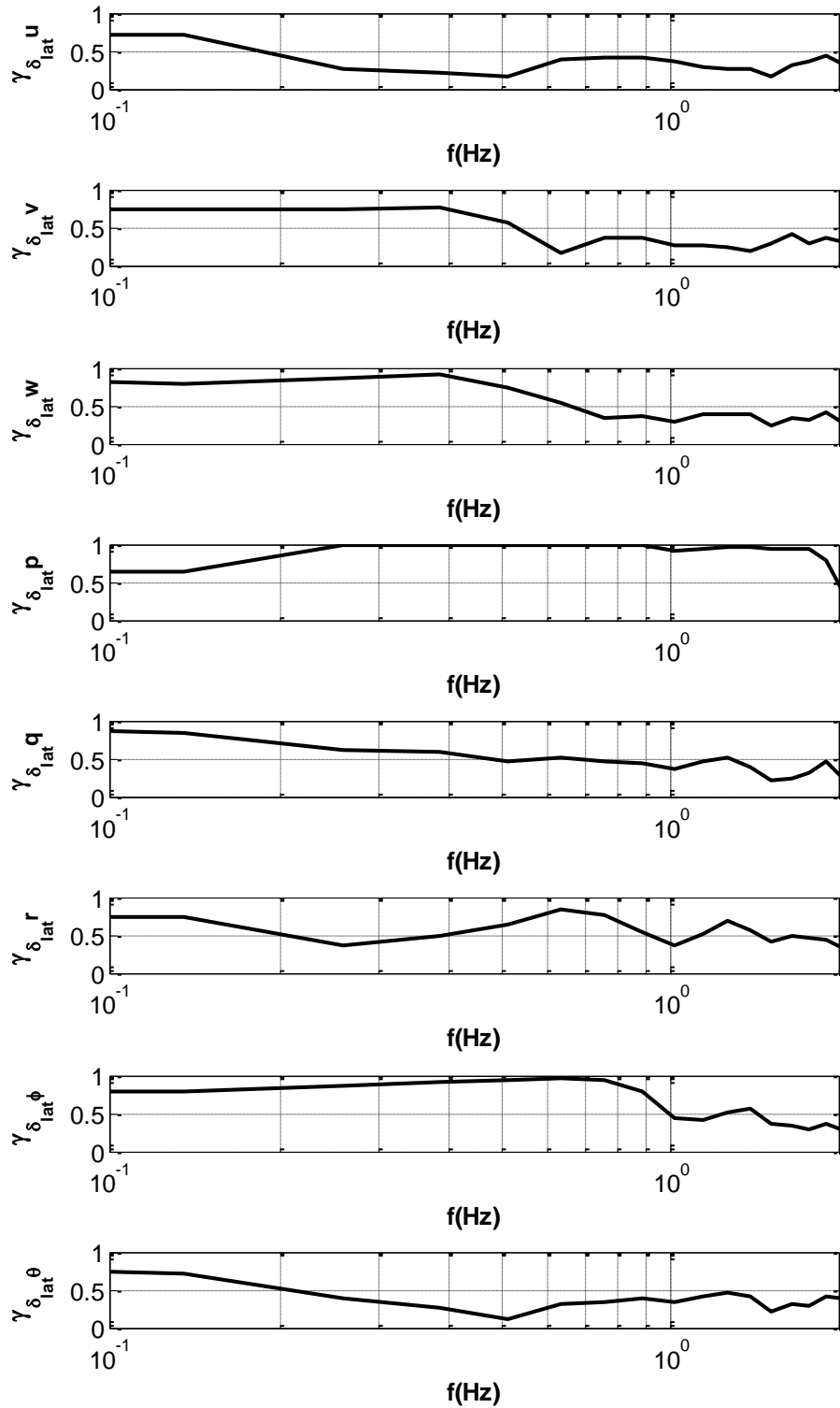


Figure 5.7. Coherences between Lateral Cyclic Input and the System Outputs

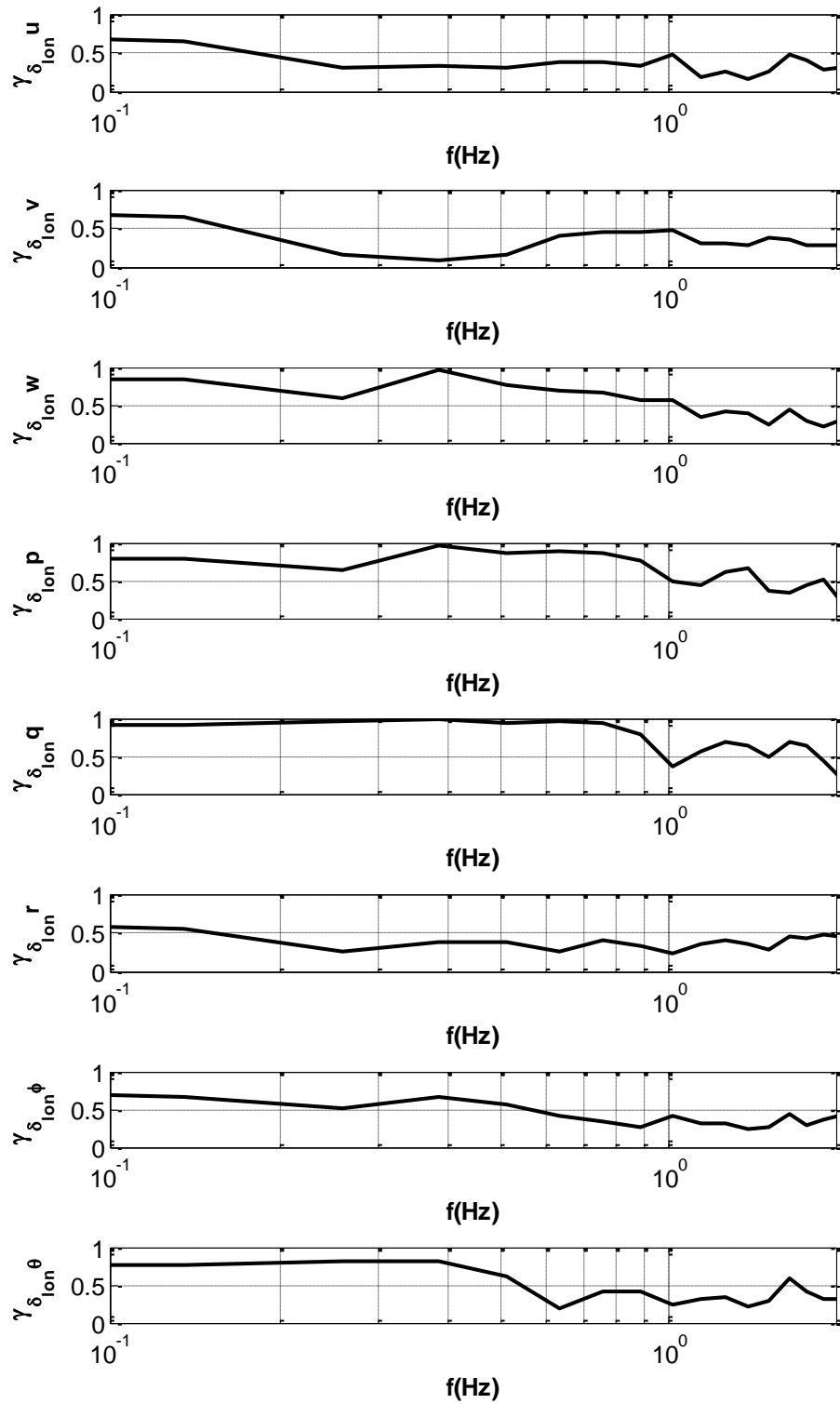


Figure 5.8. Coherences between Longitudinal Cyclic Input and the System Outputs

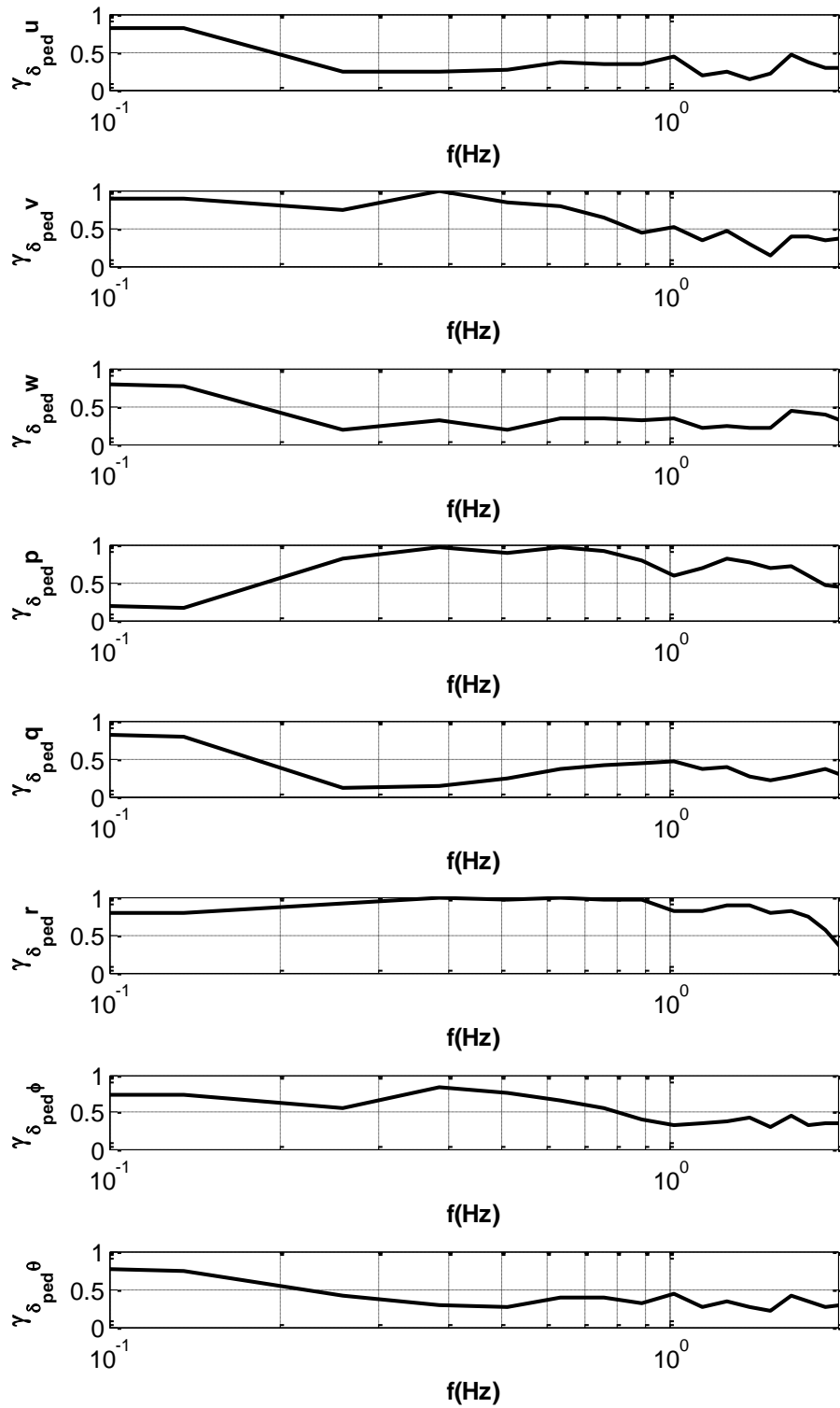


Figure 5.9. Coherences between Pedal Input and the System Outputs

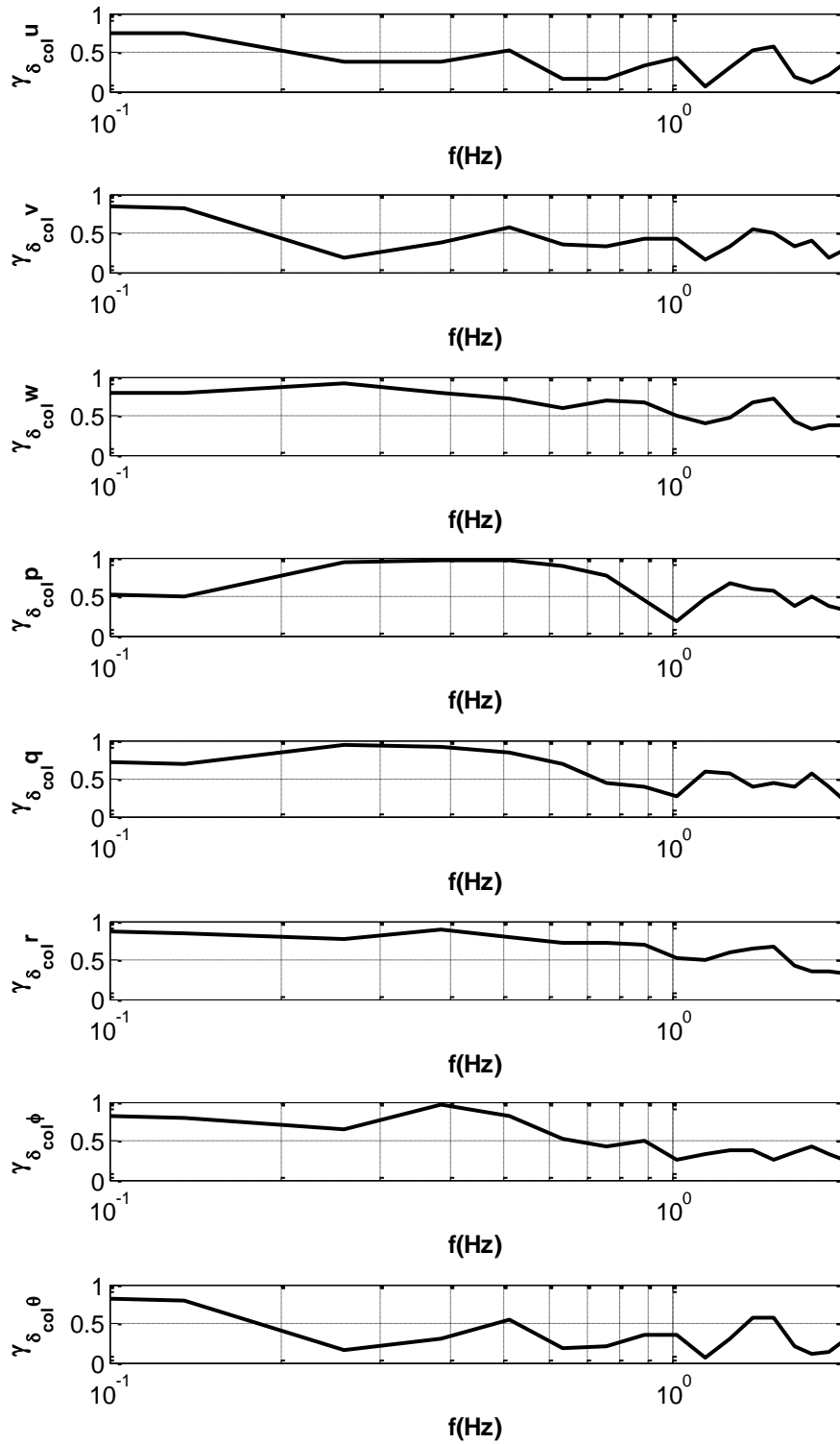


Figure 5.10. Coherences between Collective Input and the System Outputs

The true system matrices and their counterparts that are obtained by identification are shown in Appendix C. Although a direct comparison between these two systems is not correct and not a recommended practice, we would like to emphasize the following for clarity. When we inspect these two systems (the true one and the identified counterpart), we can see that they have nearly identical poles (Table 5.1). However, the numerical values of the matrix elements are quite different. This is an expected phenomenon as it was mentioned before.

Table 5.1. *Eigenvalue Comparison*

<b>True Model</b>	<b>Subspace Id. Results</b>
$-7.7321$	$-7.7319$
$-3.6525 \pm 0.2764i$	$-3.6525 \pm 0.2763i$
$-0.6125 \pm 0.0779i$	$-0.6124 \pm 0.0782i$
$0.0095 \pm 0.2070i$	$0.0095 \pm 0.2070i$
$-0.1377$	$-0.1377$

The outputs of the true model and the model obtained through subspace identification under the same 3-2-1-1 control input set are presented in Figure 5.11

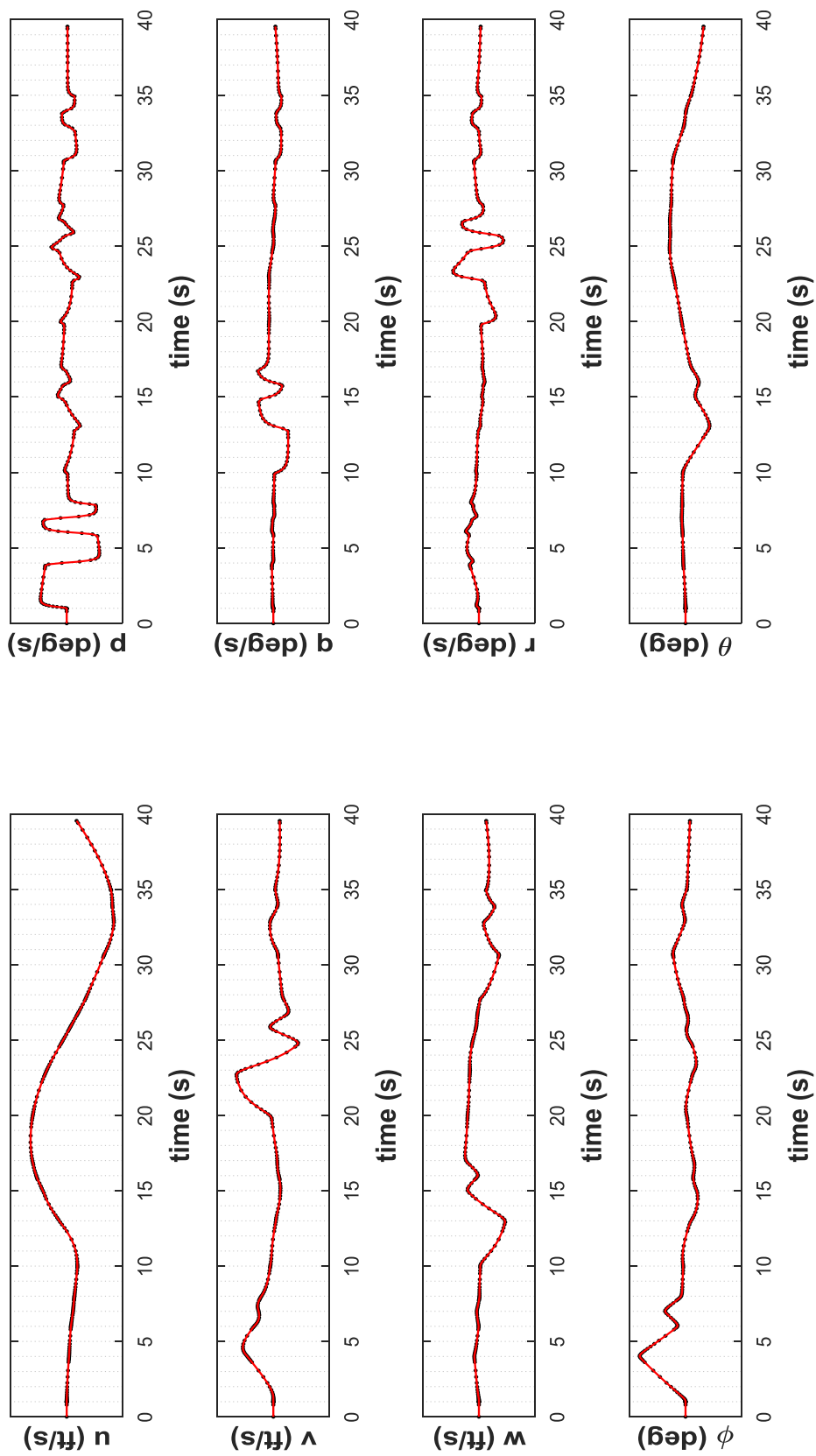


Figure 5.11. True Model Outputs vs. Outputs of the Model Generated by Subspace Identification



The subspace identification process worked well as expected. Now we can continue our route for the physical parameter estimation. Based on the equations of the similarity transformation theory (Eq.(85) - Eq. (87)), the objective function presented in Eq. (105)) is symbolically generated for our problem. MATLAB symbolic toolbox is used for this purpose. The obtained symbolic objective function is shown in Figure 5.12.

$$\begin{aligned}
 & \min_f(\chi) \\
 & = \min_x \left( \begin{array}{c} \begin{matrix} x_{125} & x_{126} & x_{127} & x_{128} & x_{129} & x_{130} & x_{131} & x_{132} \\ x_{133} & x_{134} & x_{135} & x_{136} & x_{137} & x_{138} & x_{139} & x_{140} \\ x_{141} & x_{142} & x_{143} & x_{144} & x_{145} & x_{146} & x_{147} & x_{148} \\ x_{149} & x_{150} & x_{151} & x_{152} & x_{153} & x_{154} & x_{155} & x_{156} \\ x_{157} & x_{158} & x_{159} & x_{160} & x_{161} & x_{162} & x_{163} & x_{164} \\ x_{165} & x_{166} & x_{167} & x_{168} & x_{169} & x_{170} & x_{171} & x_{172} \\ x_{173} & x_{174} & x_{175} & x_{176} & x_{177} & x_{178} & x_{179} & x_{180} \\ x_{181} & x_{182} & x_{183} & x_{184} & x_{185} & x_{186} & x_{187} & x_{188} \end{matrix} \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & 0 & -gc\theta_0 \\ x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12} & gc\phi_0c\theta_0 & -gs\phi_0s\theta_0 \\ x_{13} & x_{14} & x_{15} & x_{16} & x_{17} & x_{18} & -gs\phi_0c\theta_0 & -gc\phi_0s\theta_0 \\ x_{19} & x_{20} & x_{21} & x_{22} & x_{23} & x_{24} & 0 & 0 \\ x_{25} & x_{26} & x_{27} & x_{28} & x_{29} & x_{30} & 0 & 0 \\ x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \right)^2 \\
 & + \left( \begin{array}{c} \begin{matrix} x_{125} & x_{126} & x_{127} & x_{128} & x_{129} & x_{130} & x_{131} & x_{132} \\ x_{133} & x_{134} & x_{135} & x_{136} & x_{137} & x_{138} & x_{139} & x_{140} \\ x_{141} & x_{142} & x_{143} & x_{144} & x_{145} & x_{146} & x_{147} & x_{148} \\ x_{149} & x_{150} & x_{151} & x_{152} & x_{153} & x_{154} & x_{155} & x_{156} \\ x_{157} & x_{158} & x_{159} & x_{160} & x_{161} & x_{162} & x_{163} & x_{164} \\ x_{165} & x_{166} & x_{167} & x_{168} & x_{169} & x_{170} & x_{171} & x_{172} \\ x_{173} & x_{174} & x_{175} & x_{176} & x_{177} & x_{178} & x_{179} & x_{180} \\ x_{181} & x_{182} & x_{183} & x_{184} & x_{185} & x_{186} & x_{187} & x_{188} \end{matrix} \begin{bmatrix} x_{37} & x_{38} & x_{39} & x_{40} \\ x_{41} & x_{42} & x_{43} & x_{44} \\ x_{45} & x_{46} & x_{47} & x_{48} \\ x_{49} & x_{50} & x_{51} & x_{52} \\ x_{53} & x_{54} & x_{55} & x_{56} \\ x_{57} & x_{58} & x_{59} & x_{60} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{array}{c} 0.005 \\ -0.014 \\ -0.029 \\ 0.080 \\ -0.874 \\ -0.807 \\ 0.687 \\ -0.050 \\ -1.114 \\ 1.176 \\ 0.030 \\ 2.654 \\ 0.585 \\ 0.854 \\ 0.548 \end{array} \end{array} \right)^2 \\
 & + \left( \begin{array}{c} \begin{matrix} x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & x_{66} & x_{67} & x_{68} \\ x_{69} & x_{70} & x_{71} & x_{72} & x_{73} & x_{74} & x_{75} & x_{76} \\ x_{77} & x_{78} & x_{79} & x_{80} & x_{81} & x_{82} & x_{83} & x_{84} \\ x_{85} & x_{86} & x_{87} & x_{88} & x_{89} & x_{90} & x_{91} & x_{92} \\ x_{93} & x_{94} & x_{95} & x_{96} & x_{97} & x_{98} & x_{99} & x_{100} \\ x_{101} & x_{102} & x_{103} & x_{104} & x_{105} & x_{106} & x_{107} & x_{108} \\ x_{109} & x_{110} & x_{111} & x_{112} & x_{113} & x_{114} & x_{115} & x_{116} \\ x_{117} & x_{118} & x_{119} & x_{120} & x_{121} & x_{122} & x_{123} & x_{124} \end{matrix} \begin{array}{c} -1.254 \\ -0.874 \\ -0.003 \\ -0.032 \\ -0.001 \end{array} \begin{matrix} -0.134 \\ 0.051 \\ 0.014 \\ -0.001 \\ -0.075 \end{matrix} \begin{matrix} 0.006 \\ 0.014 \\ -0.001 \\ 0.0027 \\ -0.005 \end{matrix} \begin{matrix} x_{125} & x_{126} & x_{127} & x_{128} & x_{129} & x_{130} & x_{131} & x_{132} \\ x_{133} & x_{134} & x_{135} & x_{136} & x_{137} & x_{138} & x_{139} & x_{140} \\ x_{141} & x_{142} & x_{143} & x_{144} & x_{145} & x_{146} & x_{147} & x_{148} \\ x_{149} & x_{150} & x_{151} & x_{152} & x_{153} & x_{154} & x_{155} & x_{156} \\ x_{157} & x_{158} & x_{159} & x_{160} & x_{161} & x_{162} & x_{163} & x_{164} \\ x_{165} & x_{166} & x_{167} & x_{168} & x_{169} & x_{170} & x_{171} & x_{172} \\ x_{173} & x_{174} & x_{175} & x_{176} & x_{177} & x_{178} & x_{179} & x_{180} \\ x_{181} & x_{182} & x_{183} & x_{184} & x_{185} & x_{186} & x_{187} & x_{188} \end{matrix} \end{array} \right)^2
 \end{aligned}$$

Figure 5.12. The Symbolic Representation of the Objective Function <sup>5</sup>

<sup>5</sup> Some elements of the state space matrices are intentionally hidden.

There are infinitely many solutions for this minimization. However, our aim is to reach to a solution that corresponds to the physical domain. Thus, constraint determination and initial value assessment becomes an important issue. Constraints of the physical parameters may be selected considering typical error budgets of wind tunnel testing and aerodynamic prediction tools for helicopter systems or by using common practices of aerospace vehicle modeling. However, for convenience we conducted the analysis for different symmetric and asymmetric constraint levels. Here, we assume that the constraints for each parameter are equal to the true parameter values with  $[-10\% 10\%]$  error in regular intervals up to  $[-90\% 90\%]$  error. These nine set of analysis are performed to investigate the effect of the constraints on the convergence of the solution. The initial values are randomly selected between these lower and upper bounds (constraints). In order to assure that the solution is consistent, the optimizations are repeated for 180 times starting from randomly selected initial conditions (20 run for each set of constraints). It is expected that the analysis will converge almost to the same results for each set of initial conditions; and they will hopefully be close to the true values.

Differently from SQP algorithms, the middle point of the lower and upper bounds plays very important role for IP algorithm [52]. The common practice is setting the initial iteration to the midpoint of finite bounds. Therefore, starting from the symmetric error bound (i.e.  $[-10\% 10\%]$ ). However, in real application this is infeasible. Therefore, the optimizations are repeated for the different asymmetric constraint levels (i.e.  $[-10\% 20\%]$ ,  $[-10\% 30\%]$ ) to better understand the performance of the IP algorithm.

Another method for constraint selection is contained in the knowledge helicopter flight stability. With this additional knowledge, we proceed by reshaping the constraints according to the physical contents of the relevant parameters. For instance, if we have a priori information about the stability characteristics then we can estimate the sign of the stability characteristic. More specifically, if we have an idea about the

pitch damping characteristics of the system for a predefined flight condition, we can set a sign constraint for the relevant derivative ( $M_q$ ). All of the stability related parameters which are listed in Table 3.1 are bounded according to the stability characteristics of our system under examination. Then, we can eliminate the insignificant parameters which are mentioned in Chapter 3. In addition to these, the constraints for some derivatives ( $Z_w$ ,  $Z_{\delta_{col}}$  and  $Z_{\delta_{lon}}$ ) are set by calculating the approximations equations (Eq. (101) - Eq. (103) for this study. The constraints for all of the stability and control derivatives defined in the first step (true parameter values with [-10% 10%] error in regular intervals up to [-90% 90%] error) are combined with the constraint values for the ones specified here. Then the optimization simulations are repeated for these confined space constraints set which is acquired by intersecting of these values.

Until now we discussed the constraints for stability and control parameters (36 parameters in  $A_{phy}$  and 24 parameters in  $B_{phy}$ ). However, the above-mentioned methods of selection for the constraints do not hold for the similarity transformation matrix,  $T$ . The  $T$  matrix whose elements are not physical do not have any constraints in our problem. Besides, considering Eq. (85), the initial value of  $T$  matrix is set to the inverse of the  $C$  matrix (obtained by subspace identification). This is not a compulsory practice; however, a clever initialization of  $T$  that complies with the equations of the similarity transformation theory speeds up the computations. Verification is performed by repeating the optimizations for randomly selected initial values for  $T$  matrix.

After constructing the objective function, setting up of the constraints and the initial conditions, we can now proceed to the optimization process. As mentioned before the *fmincon* solver of MATLAB is utilized. The IP and SQP algorithms are tested with the above-mentioned constraint and initial condition settings. Moreover, the *fmincon* solver is externally assisted by the symbolic gradient of the objective function during

the optimization process. This symbolic gradient is computed via the “*gradient*” command of MATLAB.

The process of physical subspace identification can be summarized as in Figure 5.13

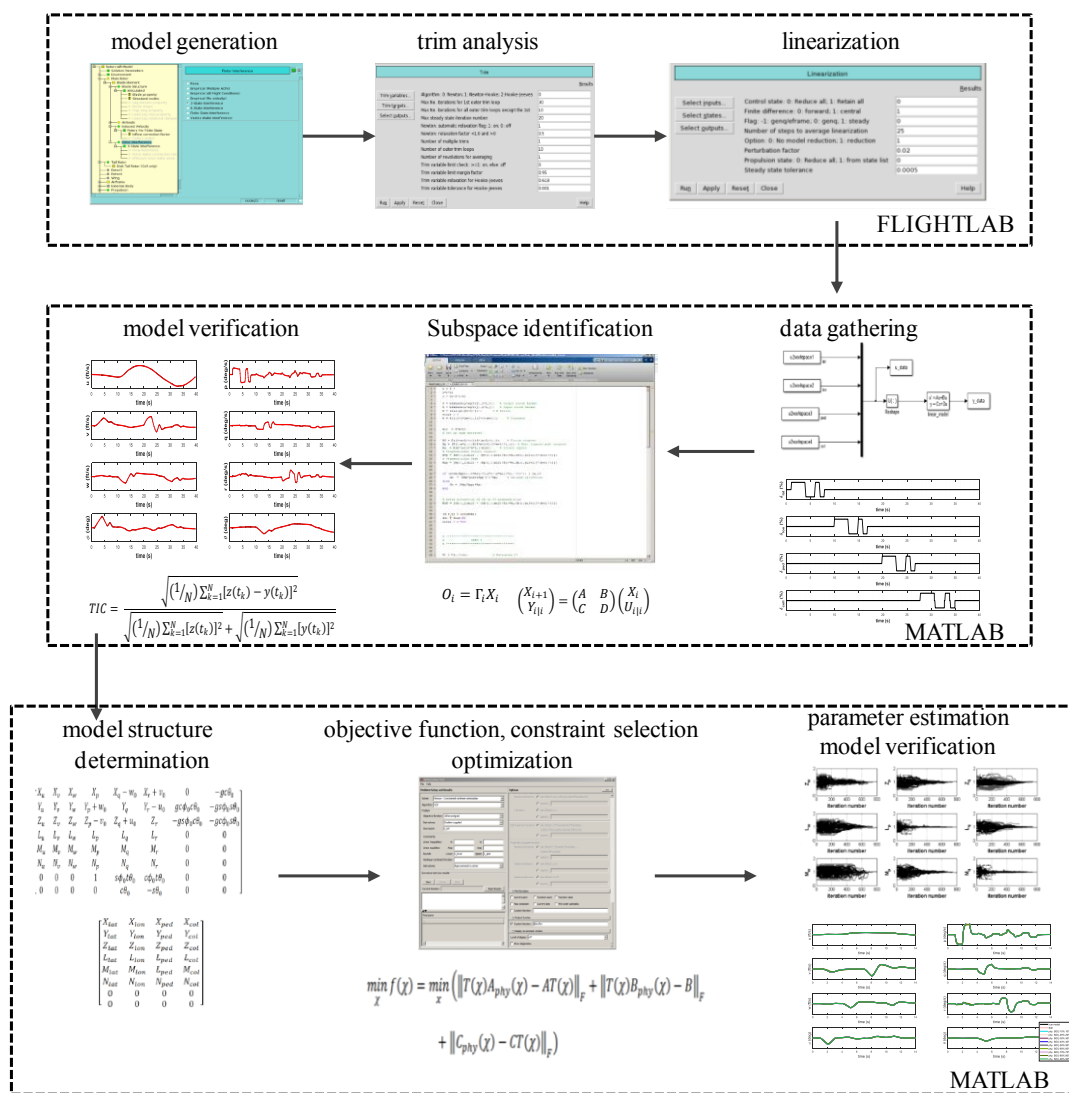


Figure 5.13. Physical Subspace Identification Procedure with Simulation Model Generated Data

## CHAPTER 6

### NUMERICAL RESULTS

In this chapter, the numerical results are presented for different optimization problems. These are tackled in five case studies which are categorized according to the utilized optimization algorithm, parameter constraint determination and the initial condition settings.

#### 6.1. Case 1-Interior Point algorithm

In this case study, IP algorithm is used for optimization. The analyses are constructed for different symmetric constraint levels. Here, we examined nine different constraint set for the parameters of  $A_{phy}$  and  $B_{phy}$  matrices. We assume that the constraints are equal to the true parameter values with [-10% 10%] error in regular intervals up to [-90% 90%] error. The parameters of the  $T$  matrix elements do not have a physical meaning; so, they are not constrained. The initial values of the parameters which belong to  $A_{phy}$  and  $B_{phy}$  matrices are selected randomly between these constraints. With the assumption that all of the system states are perfectly measurable, the associated  $C_{phy}$  is an identity matrix and according to our problem formulation  $D_{phy}$  is equal to zero. These assumptions for  $C_{phy}$  and  $D_{phy}$  matrices are hold for all of the case studies presented in this thesis. All of the conditions examined in Case 1 are summarized in Table 6.1.

Table 6.1. Summary of Analysis Conditions for Case 1 (Interior Point algorithm)

	alg.	constraint (error bound) for the parameters of			initial condition for the parameters of			number of run
		$A_{phy}$	$B_{phy}$	$T$	$A_{phy}$	$B_{phy}$	$T$	
		Case 1.1	IP <sup>6</sup>	[-10% 10%]		NA	random sel. <sup>7</sup>	
Case 1.2	IP	[-20% 20%]		NA	random sel.	const	20	
Case 1.3	IP	[-30% 30%]		NA	random sel.	const	20	
Case 1.4	IP	[-40% 40%]		NA	random sel.	const	20	
Case 1.5	IP	[-50% 50%]		NA	random sel.	const	20	
Case 1.6	IP	[-60% 60%]		NA	random sel.	const	20	
Case 1.7	IP	[-70% 70%]		NA	random sel.	const	20	
Case 1.8	IP	[-80% 80%]		NA	random sel.	const	20	
Case 1.9	IP	[-90% 90%]		NA	random sel.	const	20	

According to Table 6.1, 180 (20x9) optimization runs are performed in total. Iteration index versus minimization output curves of each parameter (each stability and control derivatives) are gathered from all of the optimization runs. To make it more understandable, the results are presented in three groups. In the first group of figures (Figure 6.2, Figure 6.3) the optimization runs are presented for Case 1.1 – Case 1.3. In the second group (Figure 6.4, Figure 6.5), the results of Case 1.4 – Case 1.6 are illustrated. In the last group the results of Case 1.7 – Case 1.9 are presented in Figure 6.6 and Figure 6.7. Each figure contains 60 (20x3) optimization runs that are initiated for 3 different constraint sets and initiated from 60 (20x3) different initial values. Due to simplicity, the initial values and the convergence results are illustrated in normalized

<sup>6</sup> “IP” denotes “Interior-Point”

<sup>7</sup> “random sel.” denotes “random selection”

<sup>8</sup> “const.” denotes “constant”

form. For this purpose, the y axis of each subplot is normalized with the true value of corresponding parameter. In other words, if optimization outputs converge to 1.0 this would mean that the estimation is perfect for that parameter. If we look at Figure 6.1 which is one of the 36 subplots of Figure 6.2, y axis shows the convergence results of the parameter  $X_u$  in normalized form. The initial values which are selected randomly are within the bounds defined in the legend of the plot. For the problem defined in Case 1.1 – Case 1.3, it is seen that the parameter  $X_u$  converges to the true value perfectly after about 80<sup>th</sup> iteration.

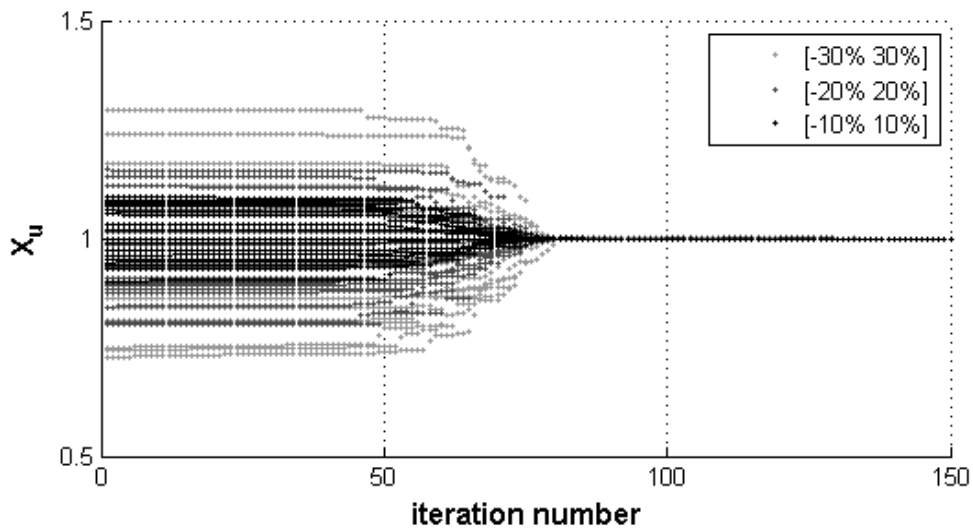


Figure 6.1. Convergence of Stability Derivative “ $X_u$ ” (Case 1.1 – Case 1.3)

For Case 1.1 – Case 1.3, it is seen that all of the parameters converge to the same value independently of the constraint levels and the initial values (Figure 6.2, Figure 6.3). In addition, the optimization results converge to “1” which is the normalized true value. However, according to Figure 6.4 and Figure 6.5, the convergence performance slightly deteriorates with larger constraint ranges (Case 1.3 – Case 1.6). In the last group, deteriorations become more visible where the constraint ranges are largest compare to the other three groups. Convergence results of this group (Case 1.7 – Case 1.9) are given in Figure 6.6 and Figure 6.7.

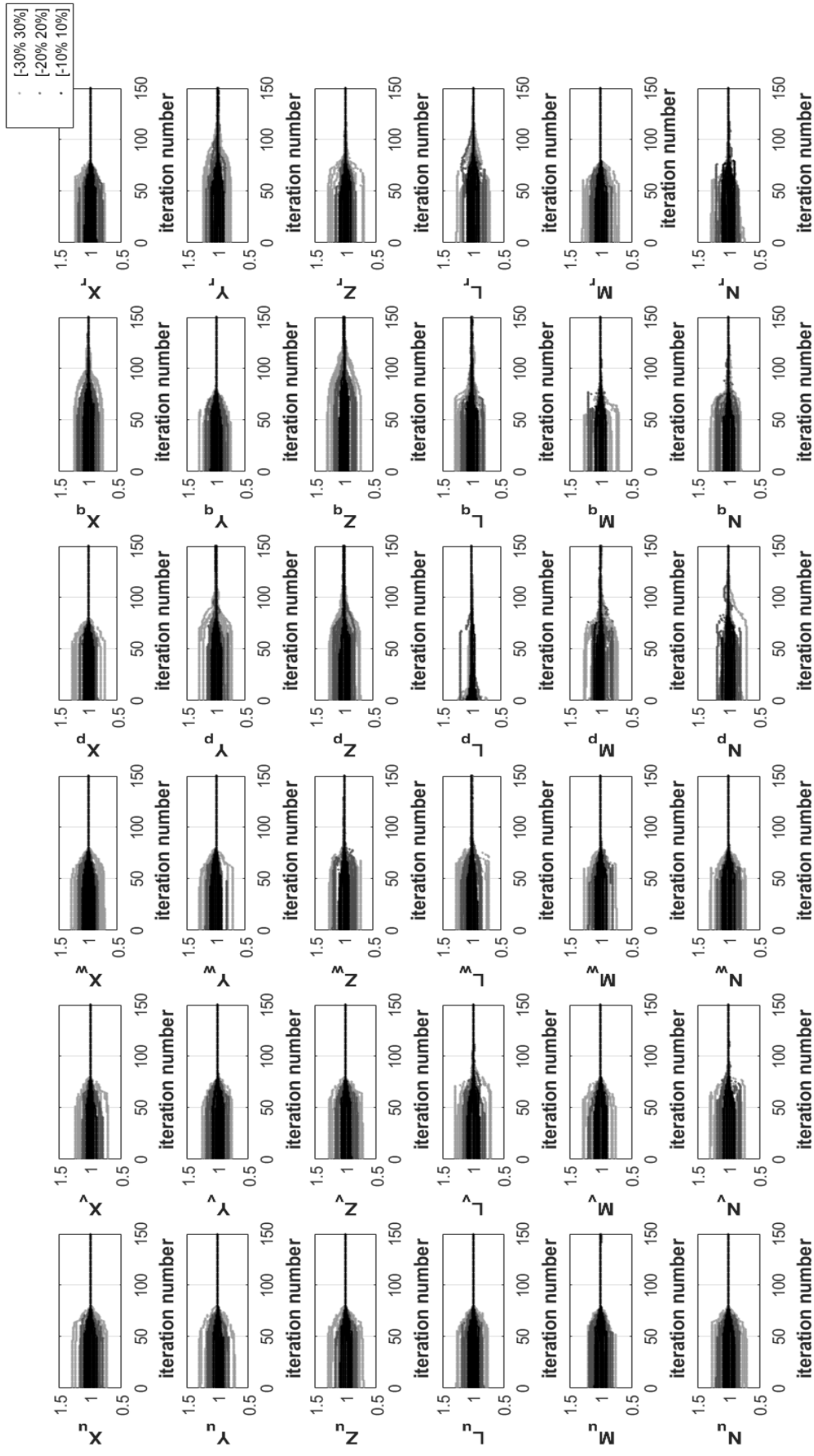


Figure 6.2. Convergence of Stability Derivatives (Case 1.1 – Case 1.3)



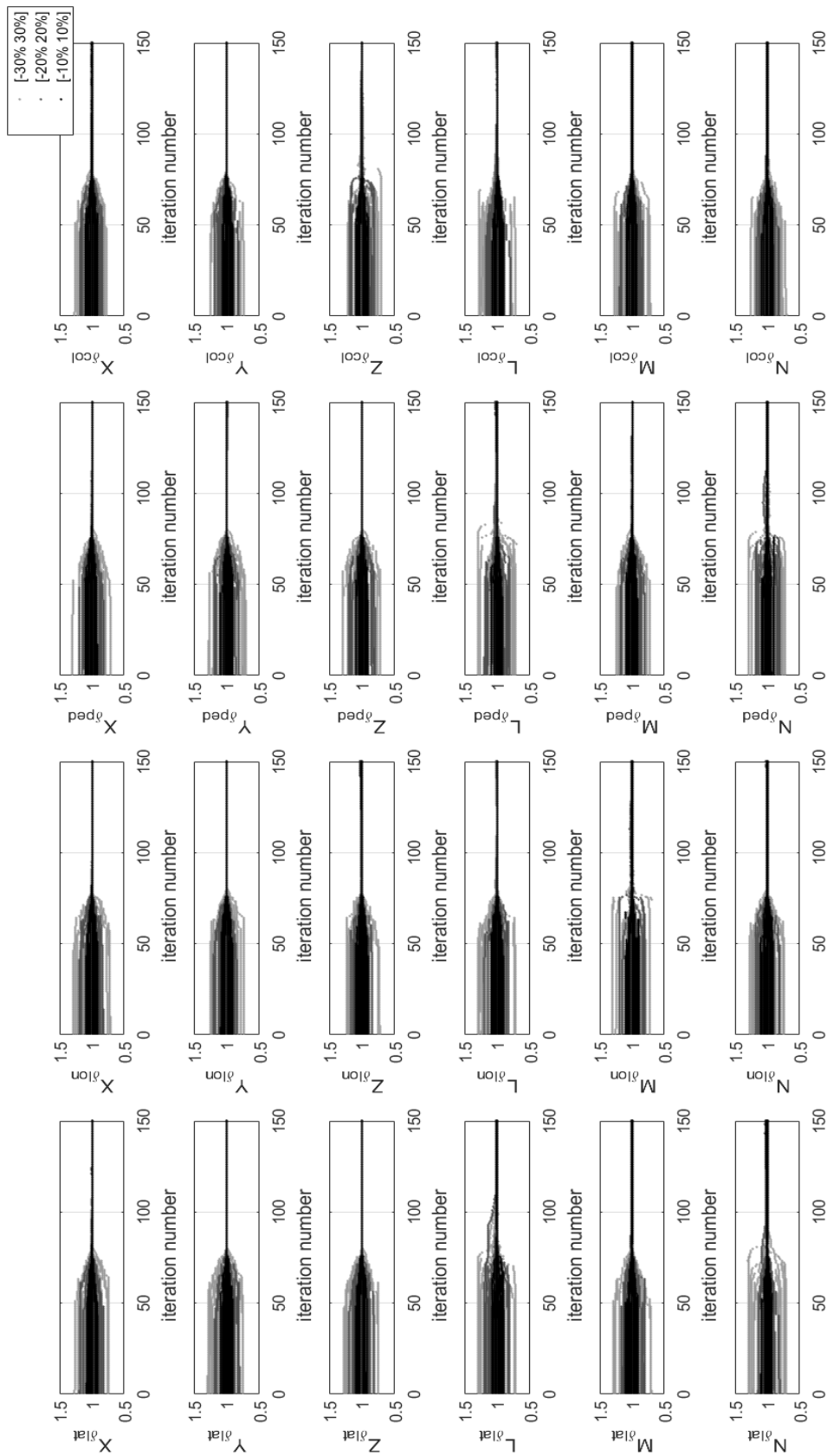


Figure 6.3. Convergence of Control Derivatives (Case 1.1 – Case 1.3)

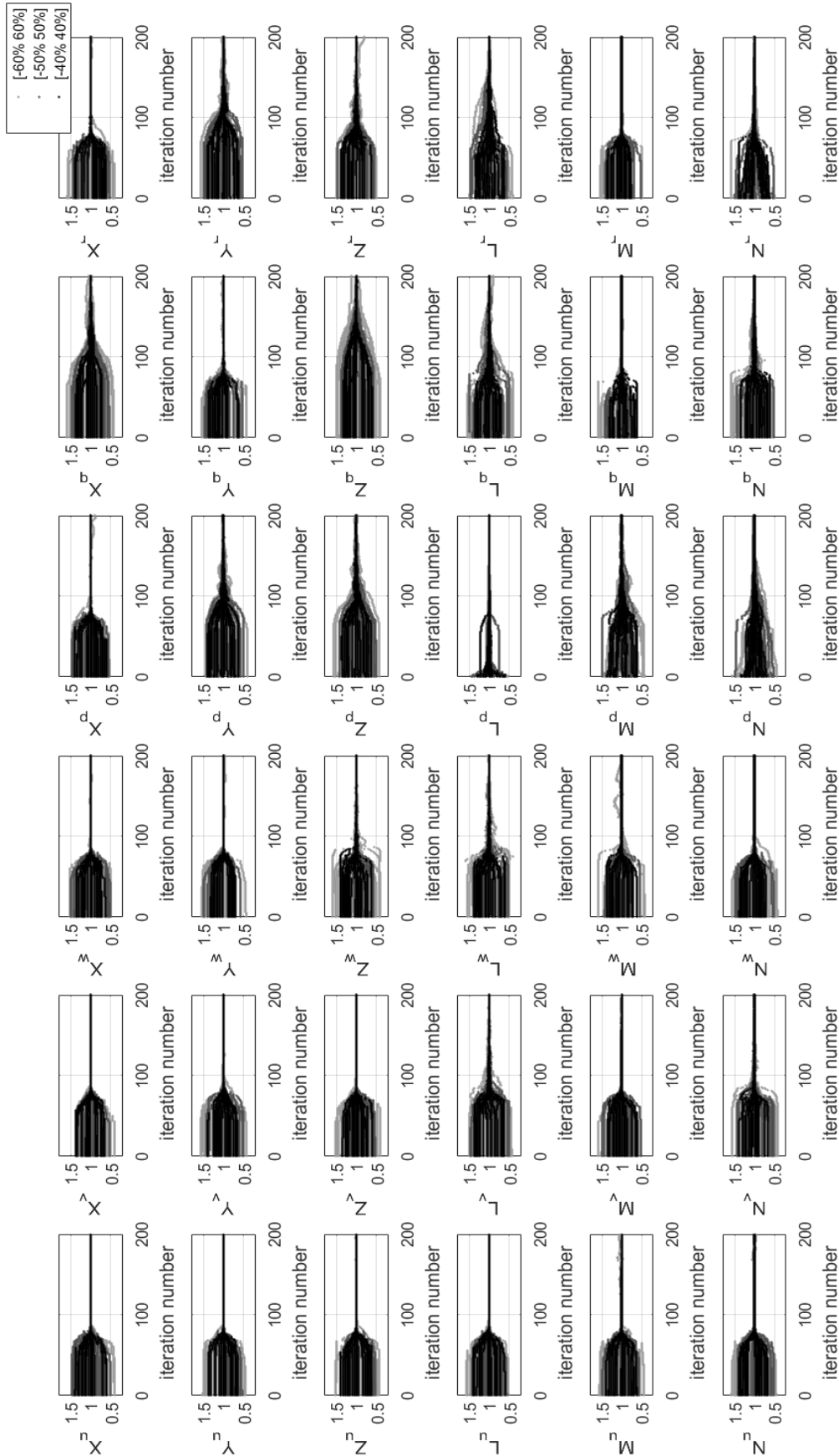


Figure 6.4. Convergence of Stability Derivatives (Case 1.4 – Case 1.6)

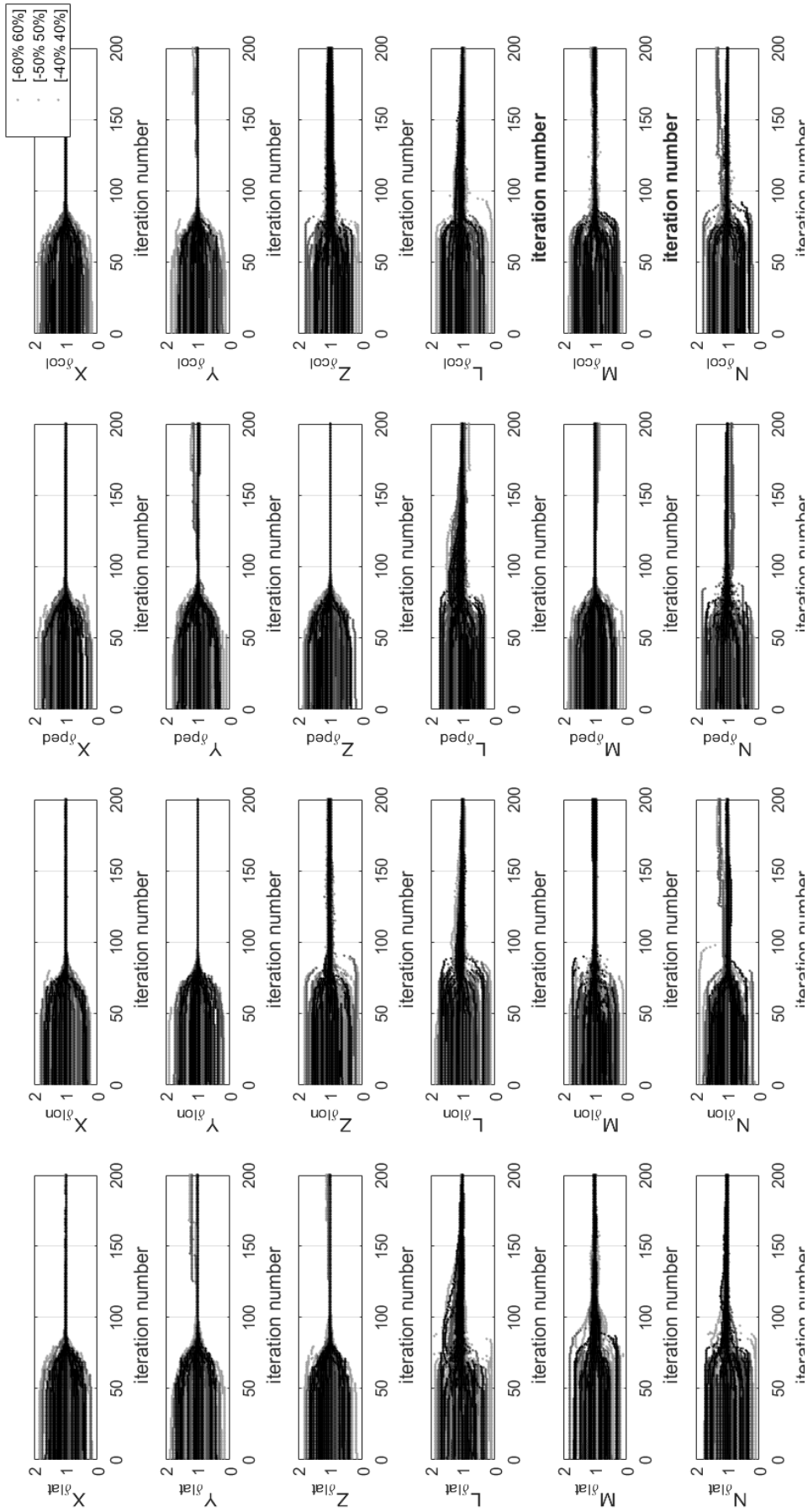


Figure 6.5. Convergence of Control Derivatives (Case 1.4 – Case 1.6)

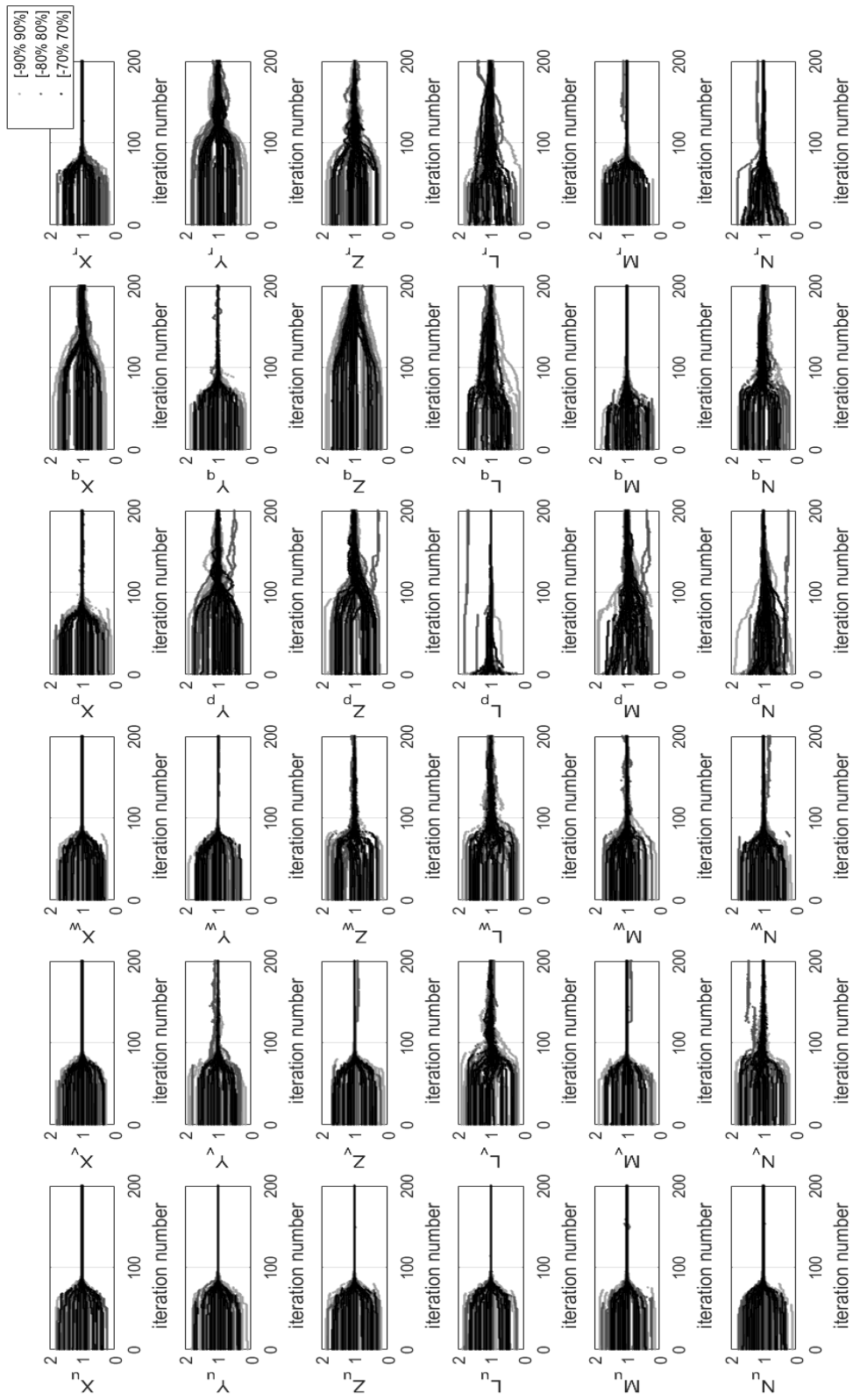


Figure 6.6. Convergence of Stability Derivatives (Case 1.7 – Case 1.9)

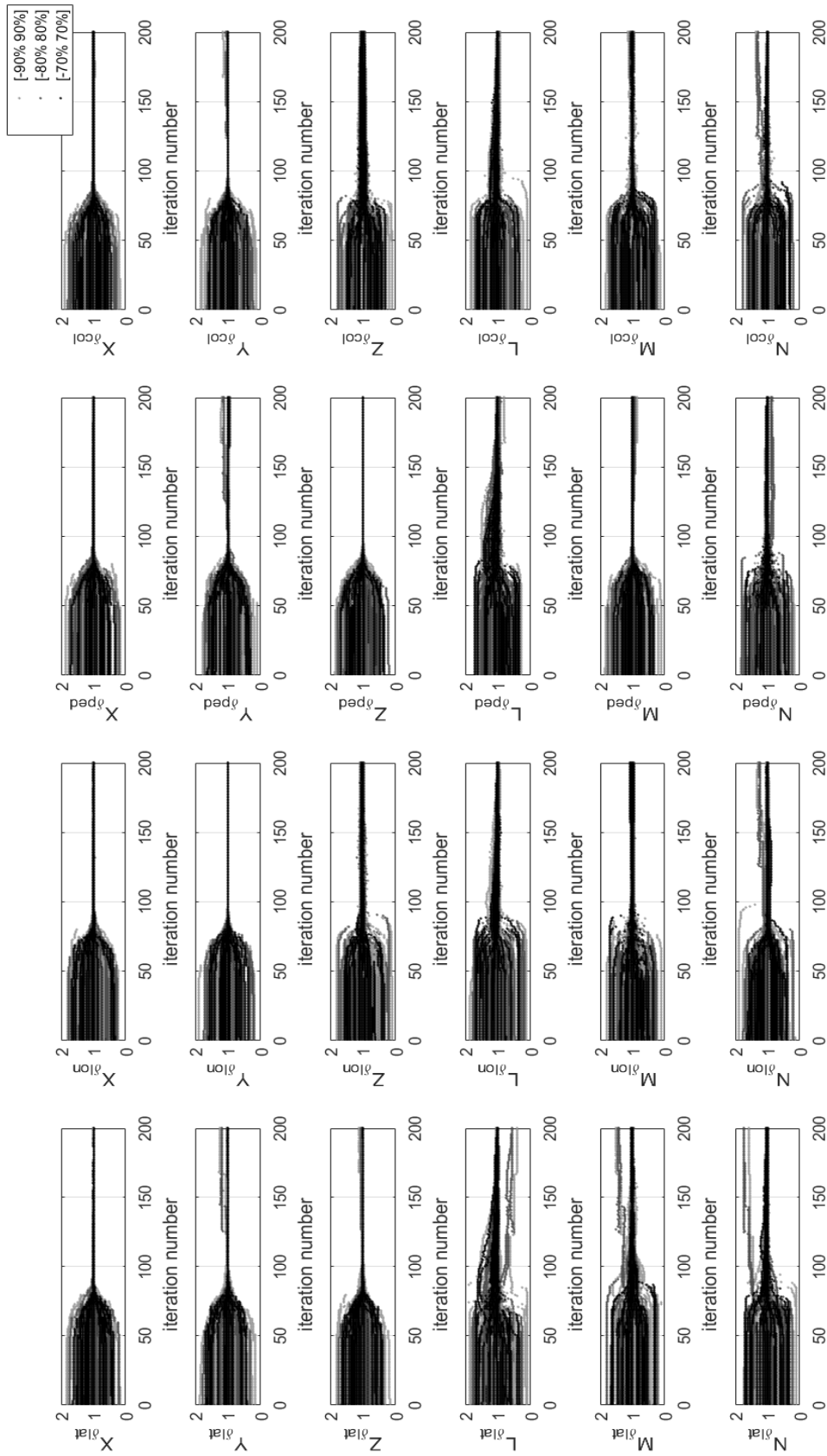


Figure 6.7. Convergence of Control Derivatives (Case 1.7 – Case 1.9)

The results convergence to the normalized true value for the majority of the optimization runs. However, a metric is required for quantitative analysis. For this purpose, percent estimation errors are calculated for each parameter. The formulation of the percent estimation error is given in Eq. (116).

$$estimation\ error\ (\%) = \frac{abs(estimated\ value - true\ value)}{abs(true\ value)} \times 100 \quad (116)$$

However, the question is how we define the “estimated value”. In other words, there are many results for many iteration indexes. For this purpose, some averaging calculations are performed. But it should be noticed that averaging is meaningful only when convergence is achieved. In our problem, for one optimization run, the mean value is calculated for a set of points where the convergence is ensured. For this purpose, “mean” function of MATLAB is utilized here. For example, the mean value of last 50 results is set as the convergence result of  $N_{\delta_{ped}}$  for the corresponding optimization run given Figure 6.6. The points for which the mean value is calculated are roughly shown with red dashed line in Figure 6.6. However, considering the number of runs for one specific condition (Figure 6.7) there will be 20 different mean values. At this point “*K-Means*” algorithm is used to refine the result. “*kmeans*” function of MATLAB is performed to get the “estimated value” of each parameter which used for “estimation error” calculations as given in Eq. (116).

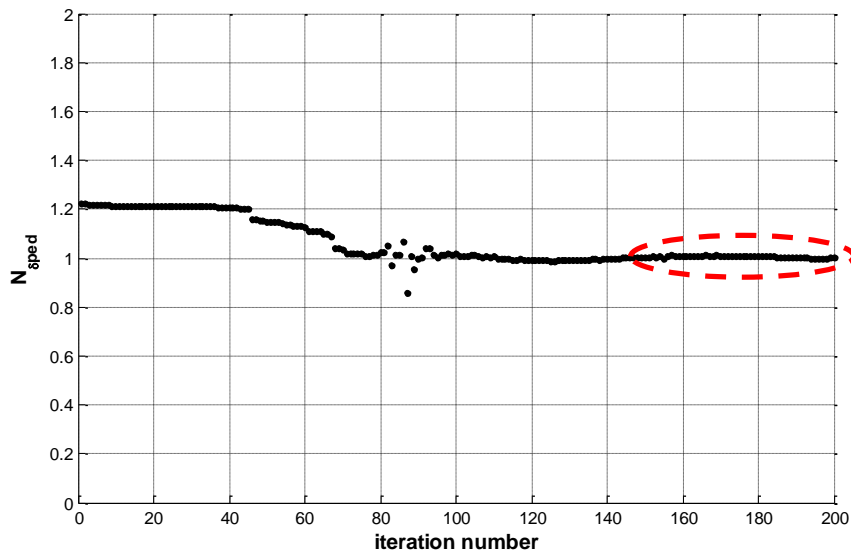


Figure 6.8. Convergence of  $N_{\delta_{ped}}$  for One Optimization Run

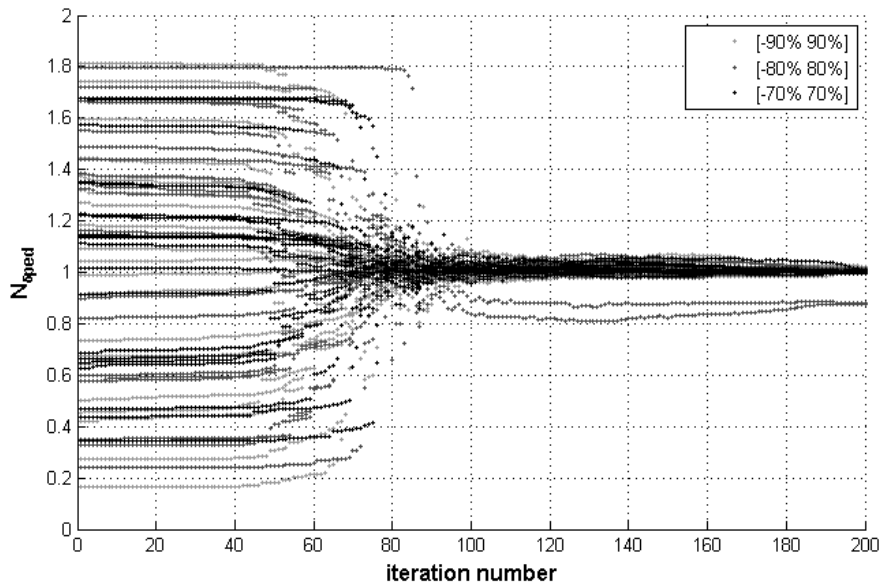


Figure 6.9. Convergence of  $N_{\delta_{ped}}$  for Twenty Optimization Runs

The estimation error values obtained from the results are presented through Figure 6.2 - Figure 6.7 are tabulated in Table 6.2 and Table 6.3. These results are close to perfect.

Table 6.2. Percentage Estimation Errors of Stability Derivatives – Case 1

parameter	symbol	estimation error %									
		Case 1.1	Case 1.2	Case 1.3	Case 1.4	Case 1.5	Case 1.6	Case 1.7	Case 1.8	Case 1.9	
		[-10% 10%]	[-20% 20%]	[-30% 30%]	[-40% 40%]	[-50% 50%]	[-60% 60%]	[-70% 70%]	[-80% 80%]	[-90% 90%]	
x <sub>1</sub>	X <sub>u</sub>	0.003	0.007	0.012	0.011	0.023	0.001	0.028	0.054	0.028	
x <sub>2</sub>	X <sub>v</sub>	0.000	0.000	0.000	0.000	0.000	0.130	0.001	0.392	0.001	
x <sub>3</sub>	X <sub>w</sub>	0.004	0.008	0.013	0.013	0.019	0.308	0.008	0.007	0.007	
x <sub>4</sub>	X <sub>p</sub>	0.003	0.004	0.003	0.004	0.018	0.966	0.044	2.410	0.079	
x <sub>5</sub>	X <sub>q</sub>	0.036	0.068	0.075	0.108	0.257	1.039	0.075	0.059	0.254	
x <sub>6</sub>	X <sub>r</sub>	0.000	0.000	0.001	0.001	0.006	0.240	0.012	0.570	0.021	
x <sub>7</sub>	Y <sub>u</sub>	0.000	0.001	0.002	0.002	0.005	0.003	0.008	0.038	0.010	
x <sub>8</sub>	Y <sub>v</sub>	0.001	0.006	0.016	0.027	0.078	1.861	0.092	0.712	0.133	
x <sub>9</sub>	Y <sub>w</sub>	0.000	0.001	0.002	0.001	0.002	0.431	0.001	1.756	0.000	
x <sub>10</sub>	Y <sub>p</sub>	0.495	0.821	1.106	1.025	1.390	2.987	1.506	6.240	1.438	
x <sub>11</sub>	Y <sub>q</sub>	0.003	0.008	0.014	0.012	0.041	0.443	0.066	1.744	0.078	
x <sub>12</sub>	Y <sub>r</sub>	0.007	0.013	0.017	0.017	0.017	0.165	0.031	0.462	0.035	
x <sub>13</sub>	Z <sub>u</sub>	0.008	0.017	0.028	0.026	0.058	0.043	0.077	0.077	0.085	
x <sub>14</sub>	Z <sub>v</sub>	0.002	0.003	0.001	0.001	0.011	2.105	0.019	5.667	0.027	
x <sub>15</sub>	Z <sub>w</sub>	0.002	0.014	0.028	0.036	0.081	0.238	0.082	1.866	0.130	
x <sub>16</sub>	Z <sub>p</sub>	0.311	0.499	0.611	0.621	0.679	2.189	0.613	7.060	0.630	
x <sub>17</sub>	Z <sub>q</sub>	0.012	0.015	0.008	0.013	0.009	0.036	0.035	0.019	0.019	
x <sub>18</sub>	Z <sub>r</sub>	0.011	0.003	0.020	0.003	0.101	2.243	0.237	0.281	0.292	
x <sub>19</sub>	L <sub>u</sub>	0.002	0.003	0.005	0.005	0.016	0.044	0.031	0.056	0.033	
x <sub>20</sub>	L <sub>v</sub>	0.044	0.087	0.142	0.139	0.350	2.424	0.543	5.736	0.729	
x <sub>21</sub>	L <sub>w</sub>	0.009	0.007	0.011	0.001	0.013	2.017	0.030	4.702	0.037	
x <sub>22</sub>	L <sub>p</sub>	0.013	0.012	0.003	0.004	0.009	2.985	0.019	7.756	0.023	
x <sub>23</sub>	L <sub>q</sub>	0.001	0.048	0.061	0.147	0.118	0.688	0.016	1.867	0.048	
x <sub>24</sub>	L <sub>r</sub>	0.031	0.034	0.069	0.040	0.046	2.826	0.114	7.329	0.112	
x <sub>25</sub>	M <sub>u</sub>	0.049	0.108	0.179	0.170	0.375	0.068	0.491	0.151	0.547	
x <sub>26</sub>	M <sub>v</sub>	0.008	0.024	0.051	0.041	0.177	2.220	0.338	6.229	0.434	
x <sub>27</sub>	M <sub>w</sub>	0.013	0.028	0.040	0.040	0.110	0.069	0.138	4.165	0.228	
x <sub>28</sub>	M <sub>p</sub>	0.114	0.034	0.107	0.070	0.342	3.295	0.330	6.540	0.316	
x <sub>29</sub>	M <sub>q</sub>	0.013	0.010	0.008	0.014	0.011	0.018	0.007	0.218	0.016	
x <sub>30</sub>	M <sub>r</sub>	0.000	0.004	0.016	0.011	0.071	2.224	0.137	6.005	0.180	
x <sub>31</sub>	N <sub>u</sub>	0.008	0.016	0.029	0.027	0.095	0.265	0.161	0.422	0.187	
x <sub>32</sub>	N <sub>v</sub>	0.059	0.085	0.078	0.081	0.060	2.784	0.191	7.215	0.294	
x <sub>33</sub>	N <sub>w</sub>	0.003	0.007	0.010	0.006	0.019	2.187	0.013	6.112	0.024	
x <sub>34</sub>	N <sub>p</sub>	0.384	0.381	0.331	0.283	0.137	2.876	0.076	7.881	0.032	
x <sub>35</sub>	N <sub>q</sub>	0.066	0.112	0.139	0.133	0.118	1.057	0.027	2.525	0.047	
x <sub>36</sub>	N <sub>r</sub>	0.087	0.088	0.072	0.073	0.050	0.615	0.027	1.937	0.021	



Table 6.3. Percentage Estimation Errors of Control Derivatives – Case 1

parameter	symbol	estimation error %									
		Case 1.1	Case 1.2	Case 1.3	Case 1.4	Case 1.5	Case 1.6	Case 1.7	Case 1.8	Case 1.9	
		[-10% 10%]	[-20% 20%]	[-30% 30%]	[-40% 40%]	[-50% 50%]	[-60% 60%]	[-70% 70%]	[-80% 80%]	[-90% 90%]	
x <sub>37</sub>	X <sub>δlat</sub>	0.003	0.011	0.023	0.019	0.092	0.259	0.184	0.736	0.245	
x <sub>38</sub>	X <sub>δlon</sub>	0.053	0.112	0.172	0.164	0.276	0.312	0.309	0.302	0.302	
x <sub>39</sub>	X <sub>δped</sub>	0.000	0.000	0.001	0.000	0.001	0.046	0.002	0.161	0.003	
x <sub>40</sub>	X <sub>δcol</sub>	0.001	0.003	0.006	0.004	0.030	0.101	0.054	0.431	0.093	
x <sub>41</sub>	Y <sub>δlat</sub>	0.056	0.189	0.400	0.339	1.430	3.534	2.582	9.091	3.191	
x <sub>42</sub>	Y <sub>δlon</sub>	0.001	0.005	0.015	0.011	0.099	0.451	0.253	1.614	0.386	
x <sub>43</sub>	Y <sub>δped</sub>	0.269	0.699	1.300	1.158	3.802	2.254	6.244	1.910	7.470	
x <sub>44</sub>	Y <sub>δcol</sub>	0.012	0.038	0.079	0.065	0.308	2.069	0.652	4.920	0.890	
x <sub>45</sub>	Z <sub>δlat</sub>	0.049	0.143	0.290	0.247	1.069	2.327	2.098	5.754	2.762	
x <sub>46</sub>	Z <sub>δlon</sub>	0.634	1.345	2.092	1.995	3.640	4.061	4.397	4.808	4.660	
x <sub>47</sub>	Z <sub>δped</sub>	0.001	0.002	0.003	0.003	0.010	0.046	0.015	0.101	0.017	
x <sub>48</sub>	Z <sub>δcol</sub>	0.030	0.142	0.274	0.265	0.506	1.024	0.641	1.258	0.709	
x <sub>49</sub>	L <sub>δlat</sub>	0.987	1.347	1.613	1.554	2.022	0.982	2.204	2.509	2.261	
x <sub>50</sub>	L <sub>δlon</sub>	0.287	0.627	1.006	0.926	1.822	1.864	2.270	3.795	2.412	
x <sub>51</sub>	L <sub>δped</sub>	0.758	1.343	1.868	1.766	2.671	1.092	3.047	0.187	3.180	
x <sub>52</sub>	L <sub>δcol</sub>	0.249	0.548	0.853	0.804	1.472	1.288	1.754	4.092	1.833	
x <sub>53</sub>	M <sub>δlat</sub>	0.258	0.558	0.874	0.819	1.577	4.393	2.049	8.354	2.215	
x <sub>54</sub>	M <sub>δlon</sub>	0.269	0.480	0.652	0.619	0.923	1.207	1.095	1.213	1.153	
x <sub>55</sub>	M <sub>δped</sub>	0.003	0.007	0.015	0.012	0.052	1.818	0.103	5.003	0.139	
x <sub>56</sub>	M <sub>δcol</sub>	0.033	0.114	0.206	0.185	0.452	1.272	0.652	1.153	0.708	
x <sub>57</sub>	N <sub>δlat</sub>	1.101	1.872	2.452	2.295	3.146	6.035	3.438	11.035	3.459	
x <sub>58</sub>	N <sub>δlon</sub>	0.234	0.552	0.915	0.860	1.866	3.924	2.454	6.882	2.749	
x <sub>59</sub>	N <sub>δped</sub>	0.069	0.205	0.350	0.262	0.533	0.188	0.682	0.589	0.659	
x <sub>60</sub>	N <sub>δcol</sub>	0.274	0.562	0.855	0.766	1.328	3.222	1.565	5.198	1.563	

The physical system matrices are constructed using the above presented optimization results. Then the true model and the estimated one are simulated with the same 3-2-1-1 excitation signals (Figure 6.10). The results show that simulation responses of “true model”, “SID model” and “Physical SID model” to 3-2-1-1 excitation signals are almost the same for Case 1.1 to Case 1.9.

In order to verify that our identification is still valid under different inputs, a doublet input is applied in four channels sequentially in another single test case. The inputs and the outputs of this test case are given in Figure 6.11 and Figure 6.12 and respectively. After that, TIC values are calculated to make quantitative analysis.

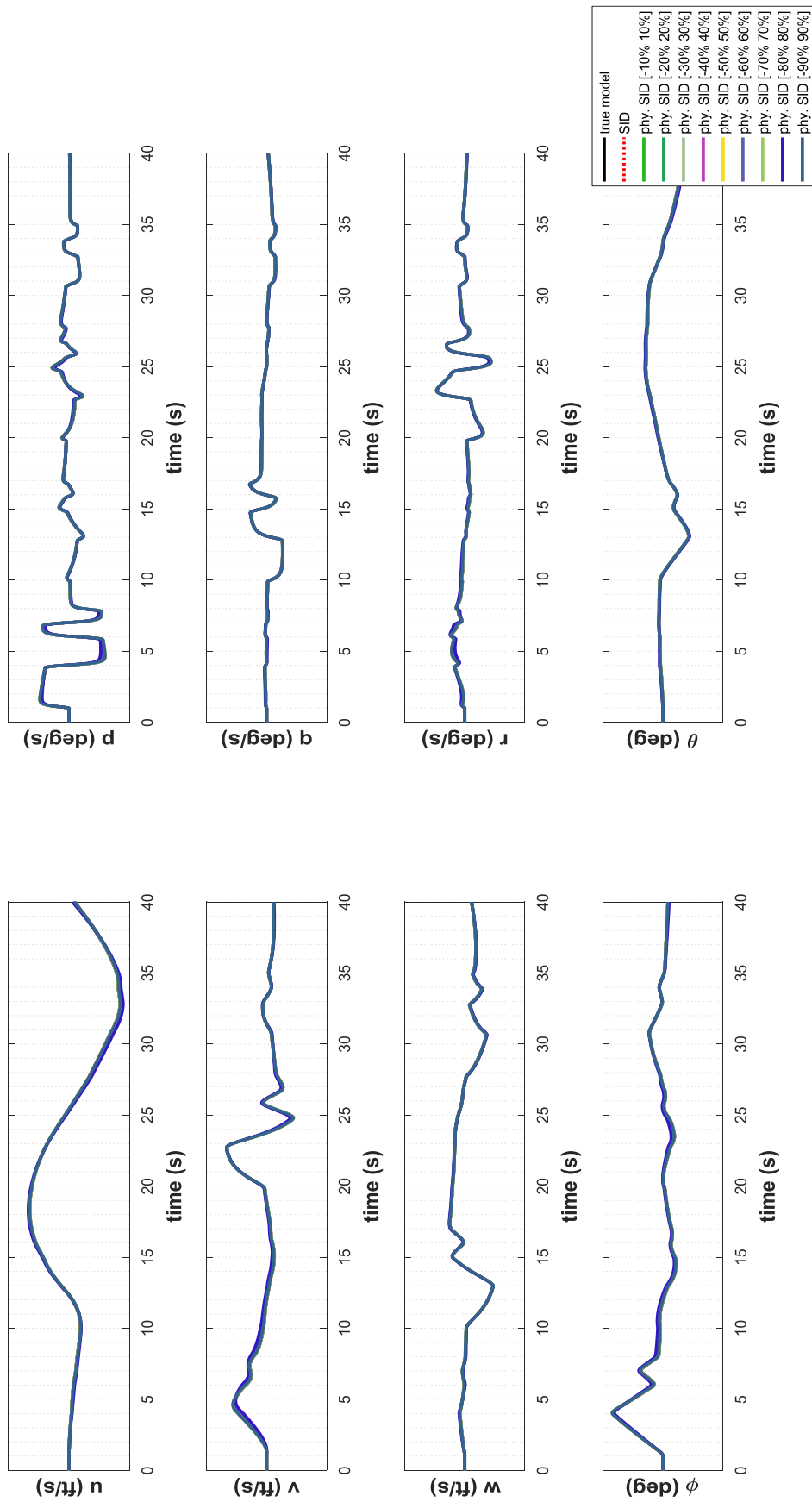


Figure 6.10. Comparison of Outputs for 3-2-1-1 Excitation Signals (Identification Using Linear Model Simulation)

TIC is a tool of verification used for the comparison of time domain outputs of the true model and the estimated counterpart. TIC is calculated using the two time domain outputs according to Eq.(117) ([61]-[63]).

$$TIC = \frac{\sqrt{(1/N) \sum_{k=1}^N [z(t_k) - y(t_k)]^2}}{\sqrt{(1/N) \sum_{k=1}^N [z(t_k)]^2 + (1/N) \sum_{k=1}^N [y(t_k)]^2}} \quad (117)$$

TIC is defined as the ratio of the RMS value of the residuals to the summation of RMS value of the true model outputs and RMS value of the estimated model outputs. TIC value may vary between 0 and 1 where 0 corresponds to a perfect estimation. As a rule of thumb, TIC values under 0.3 mean that the two models are complying [61].

The calculated TIC values for the verification signal outputs (Figure 6.12) are tabulated in Table 6.4. The maximum TIC value is less than 0.055 which is quite smaller than 0.3. This verifies that the physical subspace identification results are highly complying with the true model and indicates that our estimation is quite accurate. Also the comparison of the true model outputs with the outputs of the model obtained through subspace identification and the outputs of the model obtained through physical subspace identification which are shown in Figure 6.10 and Figure 6.12 support our quantitative outcome.

Table 6.4. TIC Values for Case 1 (for Verification Signal)

optimization batch for constraint set		TIC value
Case 1.1	[-10% 10%]	0.0042
Case 1.2	[-20% 20%]	0.0055
Case 1.3	[-30% 30%]	0.0066
Case 1.4	[-40% 40%]	0.0064
Case 1.5	[-50% 50%]	0.0079
Case 1.6	[-60% 60%]	0.0230
Case 1.7	[-70% 70%]	0.0083
Case 1.8	[-80% 80%]	0.0548
Case 1.9	[-90% 90%]	0.0083

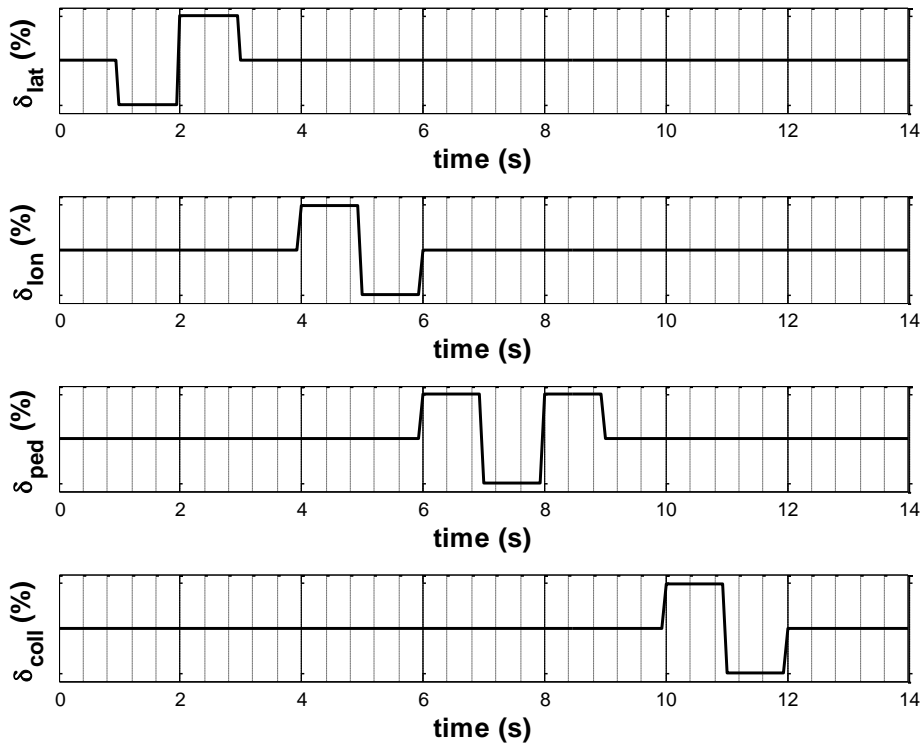


Figure 6.11. Verification Input Signals (Doublet)

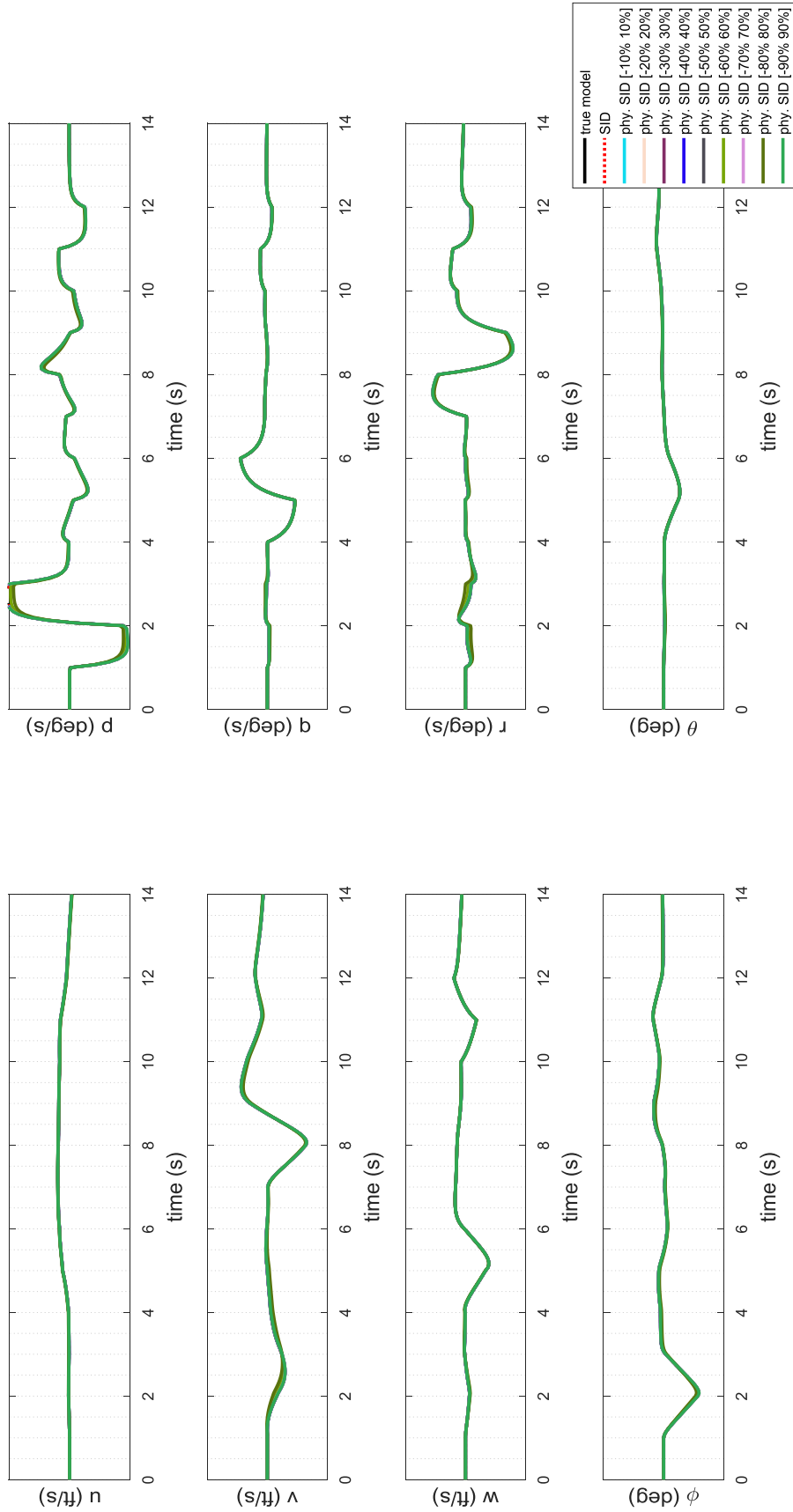


Figure 6.12. Comparison of Outputs for Doublet Excitation Signals (Identification Using Linear Model Simulation)

## 6.2. Case 2-SQP algorithm

In this case study, SQP algorithm is used for optimization. The analyses are constructed for similar conditions as given in “Case 1”. All of the assumptions on initial conditions and the constraints are also valid for this case. The only difference from the previous case study is the optimization algorithm type. The summary of the conditions for this case study are given in Table 6.5.

Table 6.5. Summary of Analysis Conditions for Case 2 (SQP algorithm)

	alg.	constraint (error bound) for the parameters of			initial condition for the parameters of			number of run
		$A_{phy}$	$B_{phy}$	$T$	$A_{phy}$	$B_{phy}$	$T$	
Case 2.1	SQP	[-10% 10%]		NA	random sel.	const	20	
Case 2.2	SQP	[-20% 20%]		NA	random sel.	const	20	
Case 2.3	SQP	[-30% 30%]		NA	random sel.	const	20	
Case 2.4	SQP	[-40% 40%]		NA	random sel.	const	20	
Case 2.5	SQP	[-50% 50%]		NA	random sel.	const	20	
Case 2.6	SQP	[-60% 60%]		NA	random sel.	const	20	
Case 2.7	SQP	[-70% 70%]		NA	random sel.	const	20	
Case 2.8	SQP	[-80% 80%]		NA	random sel.	const	20	
Case 2.9	SQP	[-90% 90%]		NA	random sel.	const	20	

According to Table 6.5, 180 (20x9) optimization runs are performed in total. Iteration index versus minimization output curves of each parameter (each stability and control derivatives) are gathered from all of the optimization runs. To make it more understandable, the results are grouped as in Case 1. In the first group of figures (Figure 6.13, Figure 6.14), the optimization runs are presented for Case 2.1 – Case 2.3. In the second group (Figure 6.15, Figure 6.16), the results of Case 2.4 – Case 2.6 are illustrated. In the last group (Figure 6.17, Figure 6.18), the convergence results are

presented for Case 2.7 – Case 2.9. As in Case 1, each figure contains 60 (20x3) optimization runs that are initiated for 3 different constraint sets and 60 (20x3) different initial values.

For Case 2.1 – Case 2.3, we can see that all of the derivatives except  $Y_w, Z_v, M_{\delta ped}$  converge almost to the same value for all of the optimization run. The derivatives  $Y_w, Z_v, M_{\delta ped}$  have relatively low level of significance in helicopter dynamics. Therefore, this is an expected phenomenon. The results are complying with the fundamentals of the estimation theory. It is not possible to estimate a parameter which is not significant or even which does not exist. These findings provide us valuable physical insight about the model of the system being identified. The results show that, rest of the parameters converge to the same value independently from the constraint levels and the initial values. In addition, the optimization results converge to the 1 which is the normalized true value.

For Case 2.4 – Case 2.6 where the constraints are enlarged to [-60% 60%] error bounds, all of the derivatives except  $Y_w, Y_q, M_r, M_{\delta ped}$  converge to true value (Figure 6.15 and Figure 6.16). Again, these derivatives which do not converged have low level of significance. In this group, more iterations are required for the optimizations to converge when compared to the first group (Case 2.1 – Case 2.3). However, the differences in iteration numbers are not so big.

In the last group (Case 2.7 – Case 2.9), all of the derivatives except  $Y_u, Y_w, Y_q, M_r, M_{\delta ped}$  converge to true values (Figure 6.17, Figure 6.18). Again, the derivatives which do not converged have low level of significance. Compared to the first two groups, we observe that, some of the optimization runs cannot converge to a solution. However, considering the total number of runs these are a few. Also, more iteration is required for optimization to converge when compared to the first and second groups.

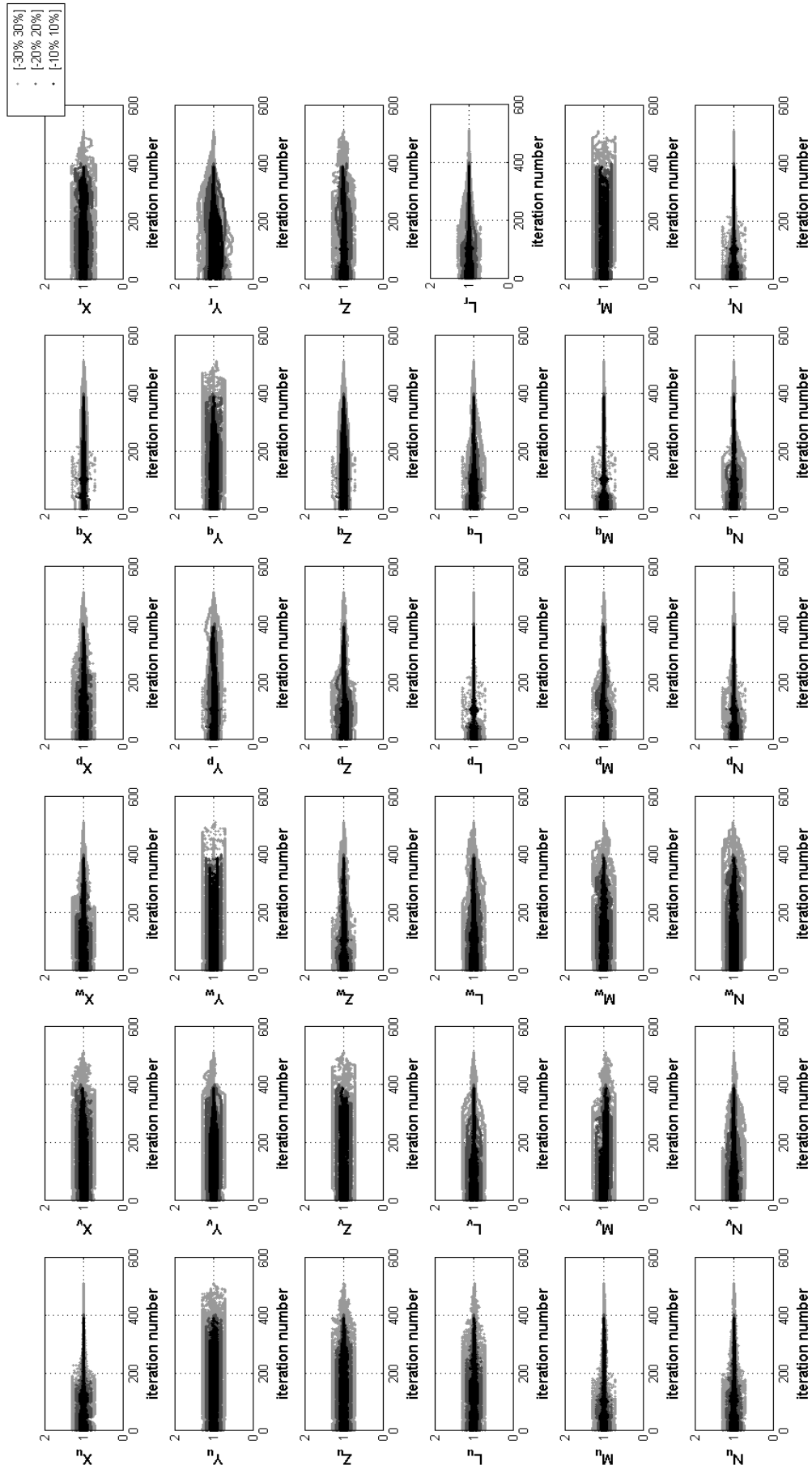


Figure 6.13. Convergence of Stability Derivatives (Case 2.1 – Case 2.3)



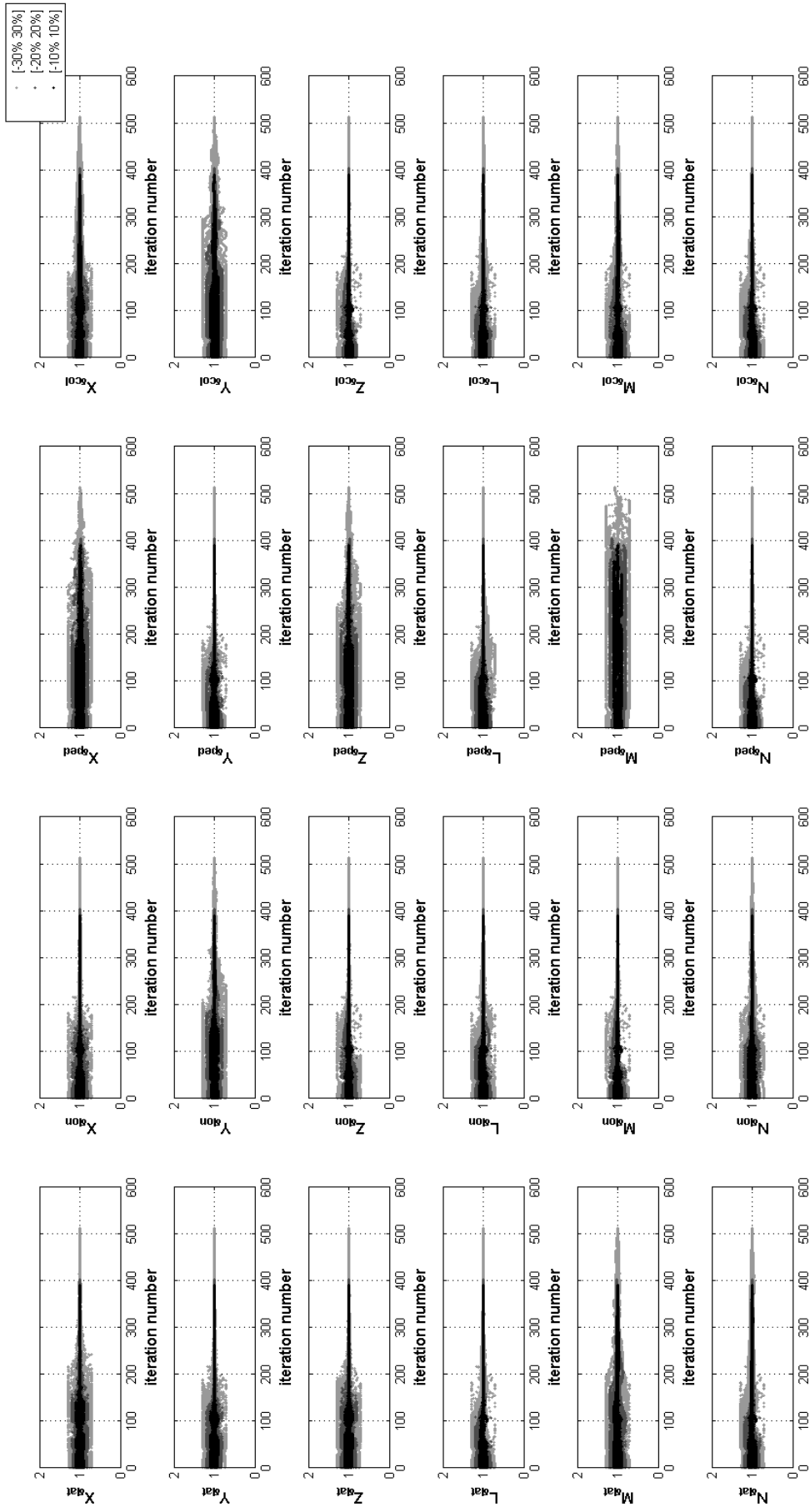


Figure 6.14. Convergence of Control Derivatives (Case 2.1 – Case 2.3)

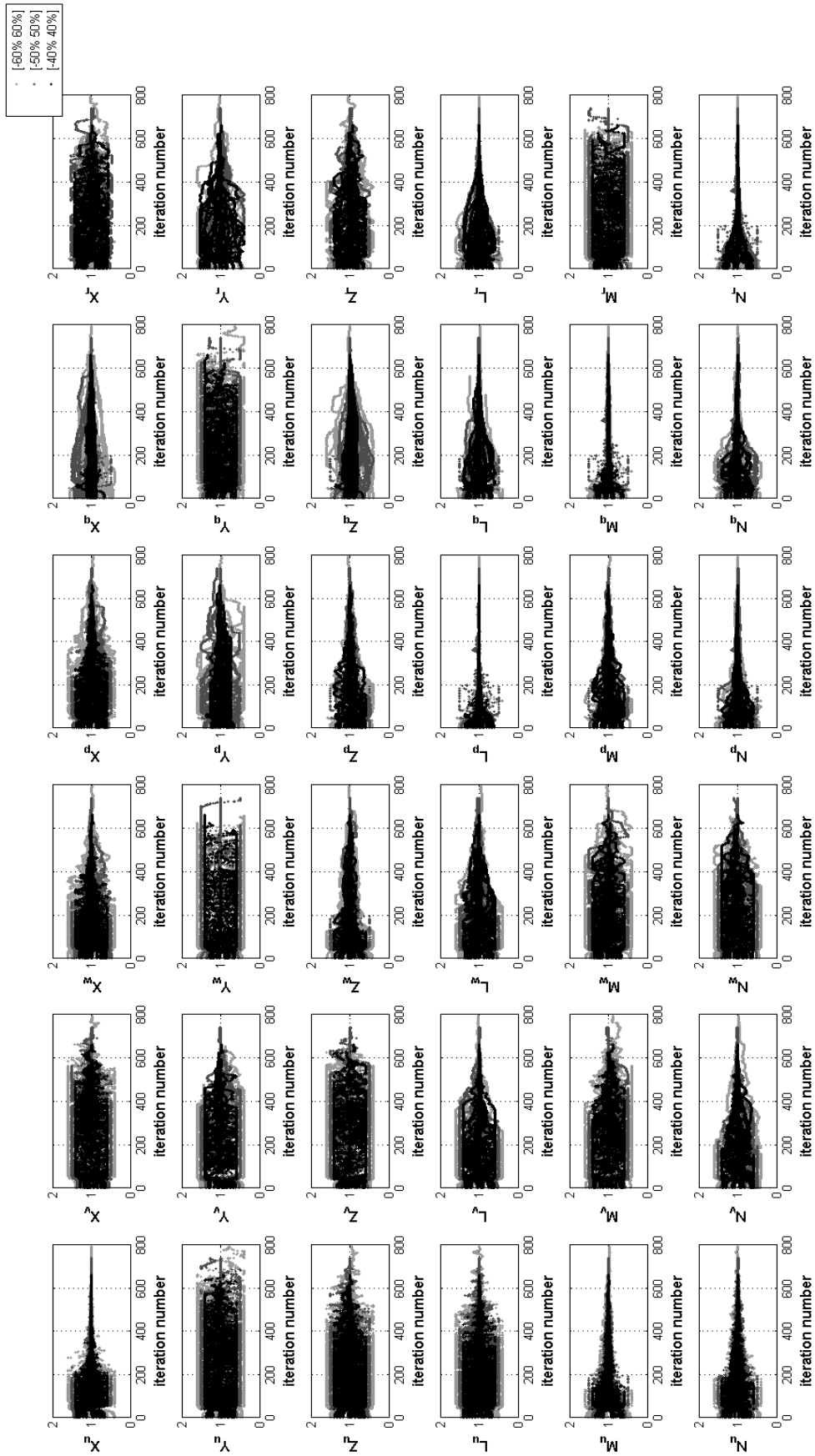


Figure 6.15. Convergence of Stability Derivatives (Case 2.4 – Case 2.6)

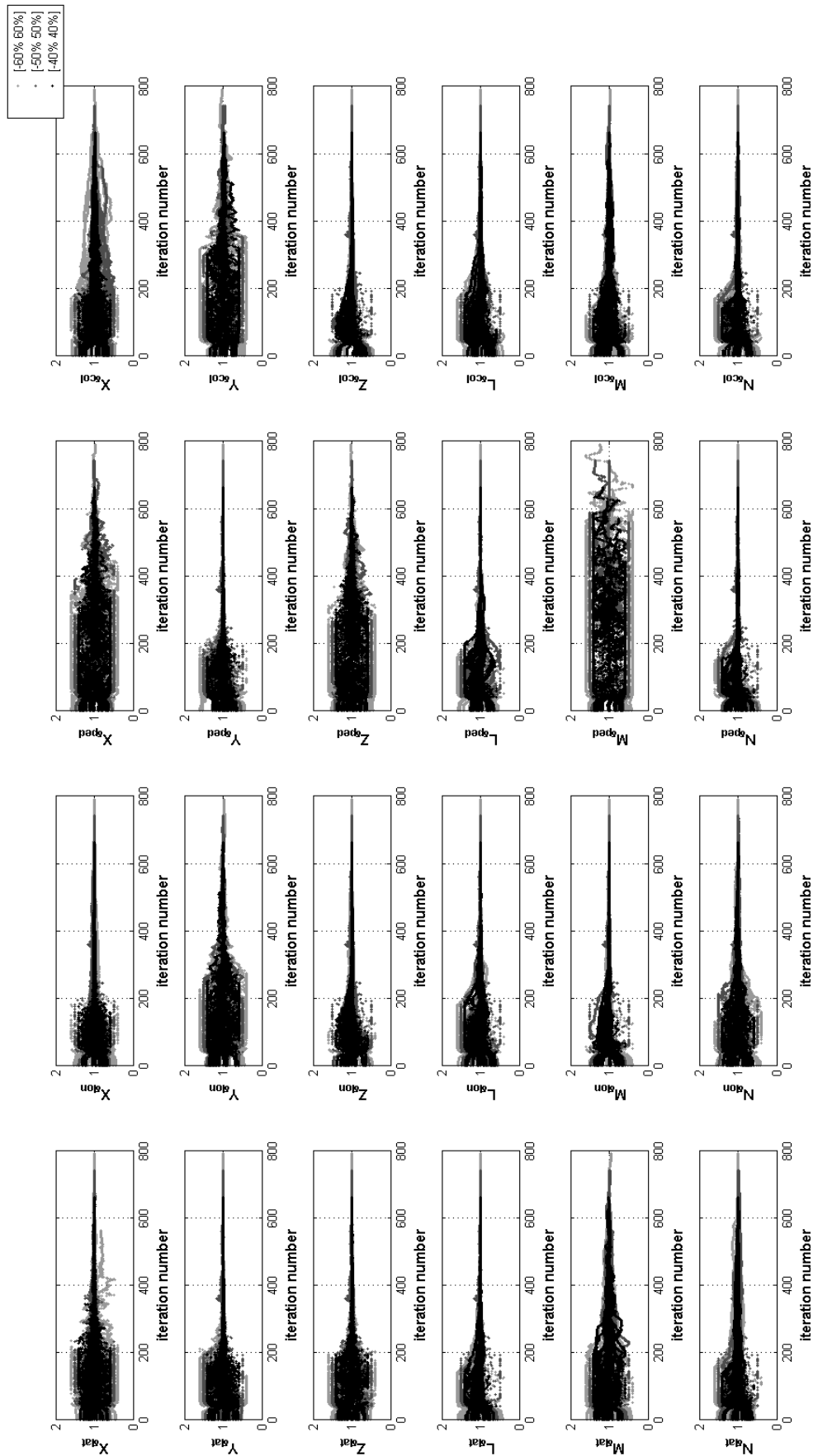


Figure 6.16. Convergence of Control Derivatives (Case 2.4 – Case 2.6)

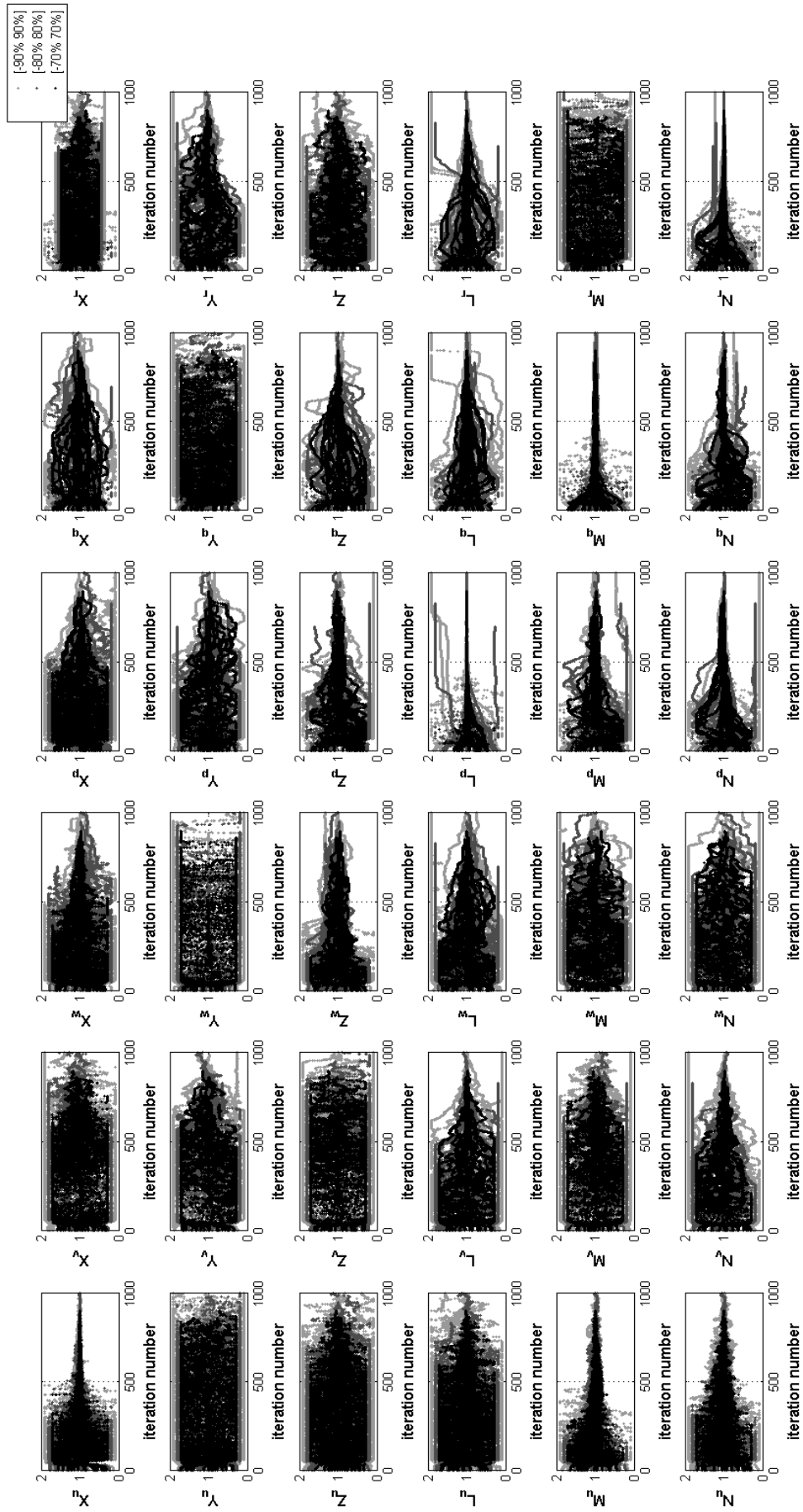


Figure 6.17. Convergence of Stability Derivatives (Case 2.7 – Case 2.9)

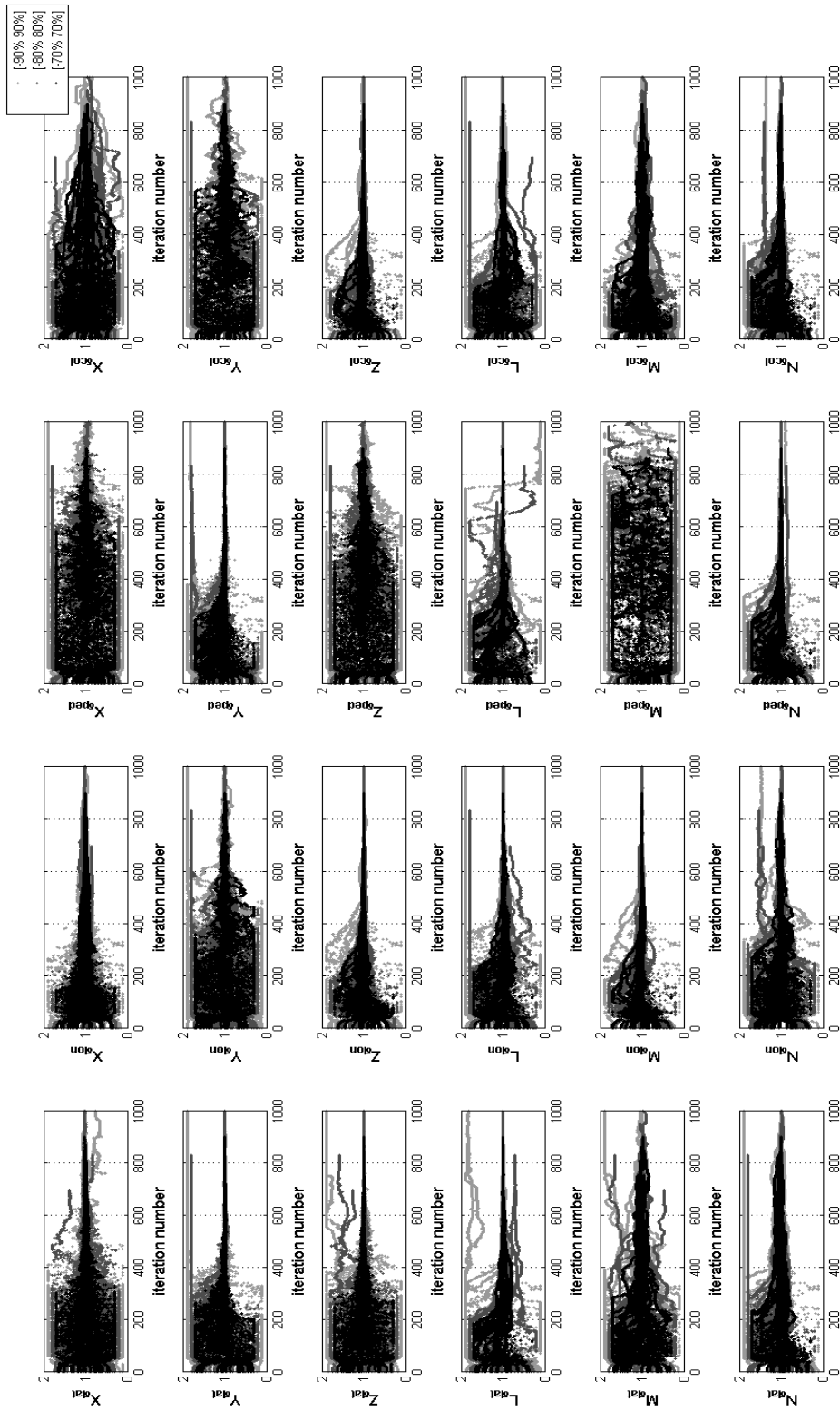


Figure 6.18. Convergence of Control Derivatives (Case 2.7 – Case 2.9)

The estimation error values are calculated according to Eq. (116). The results are tabulated in Table 6.6 and Table 6.7.

Table 6.6. *Percentage Estimation Errors of Stability Derivatives – Case 2*

parameter	symbol	estimation error %									
		Case 2.1 [-10% 10%]	Case 2.2 [-20% 20%]	Case 2.3 [-30% 30%]	Case 2.4 [-40% 40%]	Case 2.5 [-50% 50%]	Case 2.6 [-60% 60%]	Case 2.7 [-70% 70%]	Case 2.8 [-80% 80%]	Case 2.9 [-90% 90%]	
x <sub>1</sub>	X <sub>u</sub>	0.075	0.170	0.135	0.197	0.286	0.336	0.187	0.106	0.099	
x <sub>2</sub>	X <sub>v</sub>	3.364	4.233	2.799	1.428	1.875	4.598	6.397	10.913	11.927	
x <sub>3</sub>	X <sub>w</sub>	0.007	0.406	0.474	0.191	0.467	0.948	0.646	1.085	2.808	
x <sub>4</sub>	X <sub>p</sub>	0.148	0.258	0.642	0.802	2.336	7.298	0.691	9.825	10.038	
x <sub>5</sub>	X <sub>q</sub>	0.019	0.043	1.478	0.841	4.551	4.168	2.234	7.352	5.343	
x <sub>6</sub>	X <sub>r</sub>	1.158	1.775	0.534	0.912	1.020	3.305	0.444	0.795	3.617	
x <sub>7</sub>	Y <sub>u</sub>	0.058	2.107	2.937	1.540	0.402	9.326	8.580	12.546	6.529	
x <sub>8</sub>	Y <sub>v</sub>	0.054	0.167	0.183	1.233	3.025	0.596	0.813	1.960	8.220	
x <sub>9</sub>	Y <sub>w</sub>	0.156	0.927	1.077	7.139	1.420	2.151	2.829	0.329	6.252	
x <sub>10</sub>	Y <sub>p</sub>	1.624	0.263	1.677	0.951	4.735	8.421	3.955	2.274	2.162	
x <sub>11</sub>	Y <sub>q</sub>	0.716	1.202	7.008	4.515	1.938	13.995	0.663	5.164	18.951	
x <sub>12</sub>	Y <sub>r</sub>	0.031	0.015	0.020	0.072	0.096	0.296	0.077	0.547	0.569	
x <sub>13</sub>	Z <sub>u</sub>	0.528	0.826	0.625	1.553	1.408	1.099	0.246	7.513	10.223	
x <sub>14</sub>	Z <sub>v</sub>	1.004	5.680	0.421	4.010	2.200	12.865	11.616	12.844	10.145	
x <sub>15</sub>	Z <sub>w</sub>	0.082	0.111	0.433	0.329	0.505	0.598	0.398	4.077	5.003	
x <sub>16</sub>	Z <sub>p</sub>	0.370	0.365	0.168	0.484	0.845	5.382	0.305	0.886	8.812	
x <sub>17</sub>	Z <sub>q</sub>	0.014	0.172	0.059	0.143	0.212	0.095	0.078	0.247	0.226	
x <sub>18</sub>	Z <sub>r</sub>	1.008	1.838	0.053	1.719	0.949	8.009	4.358	5.908	4.211	
x <sub>19</sub>	L <sub>u</sub>	1.074	0.955	1.112	1.033	0.497	4.622	1.516	1.969	5.016	
x <sub>20</sub>	L <sub>v</sub>	0.230	0.191	0.076	0.597	0.472	5.432	0.979	3.587	9.574	
x <sub>21</sub>	L <sub>w</sub>	0.027	0.042	0.212	0.672	0.172	6.036	1.373	1.873	6.128	
x <sub>22</sub>	L <sub>p</sub>	0.041	0.024	0.008	0.039	0.075	5.971	0.010	0.379	8.984	
x <sub>23</sub>	L <sub>q</sub>	0.084	0.208	0.162	0.004	0.209	0.349	0.521	1.573	8.092	
x <sub>24</sub>	L <sub>r</sub>	0.064	0.070	0.048	0.179	0.102	6.094	0.222	0.214	8.852	
x <sub>25</sub>	M <sub>u</sub>	0.875	0.648	0.931	0.557	0.509	0.061	0.747	1.091	1.818	
x <sub>26</sub>	M <sub>v</sub>	6.859	5.478	6.949	5.966	5.740	14.365	2.615	17.520	14.930	
x <sub>27</sub>	M <sub>w</sub>	0.196	0.041	1.309	0.852	1.565	2.838	1.468	9.680	12.945	
x <sub>28</sub>	M <sub>p</sub>	0.345	0.105	0.009	0.258	0.461	6.266	0.159	7.499	5.358	
x <sub>29</sub>	M <sub>q</sub>	0.007	0.011	0.089	0.068	0.107	0.222	0.068	0.387	0.883	
x <sub>30</sub>	M <sub>r</sub>	3.514	2.413	1.330	6.740	18.109	7.485	6.253	4.209	3.177	
x <sub>31</sub>	N <sub>u</sub>	0.800	0.655	0.629	0.802	0.805	1.346	0.971	0.050	0.620	
x <sub>32</sub>	N <sub>v</sub>	0.319	0.226	0.035	0.199	0.840	6.664	0.128	7.434	9.516	
x <sub>33</sub>	N <sub>w</sub>	0.114	0.270	0.371	1.091	0.372	6.191	5.204	11.506	15.605	
x <sub>34</sub>	N <sub>p</sub>	0.185	0.057	0.095	0.099	0.523	6.141	0.317	8.086	9.084	
x <sub>35</sub>	N <sub>q</sub>	0.196	0.139	0.090	0.239	0.581	2.981	0.564	4.518	2.684	
x <sub>36</sub>	N <sub>r</sub>	0.030	0.012	0.031	0.007	0.172	1.454	0.100	2.387	2.361	

Table 6.7. Percentage Estimation Errors of Control Derivatives – Case 2

parameter	symbol	estimation error %								
		Case 2.1 [-10% 10%]	Case 2.2 [-20% 20%]	Case 2.3 [-30% 30%]	Case 2.4 [-40% 40%]	Case 2.5 [-50% 50%]	Case 2.6 [-60% 60%]	Case 2.7 [-70% 70%]	Case 2.8 [-80% 80%]	Case 2.9 [-90% 90%]
x <sub>37</sub>	X <sub>δlat</sub>	0.115	0.076	0.241	0.184	0.374	1.545	0.204	1.382	2.302
x <sub>38</sub>	X <sub>δlon</sub>	0.016	0.014	0.079	0.040	0.277	0.269	0.130	0.485	0.393
x <sub>39</sub>	X <sub>δped</sub>	1.637	1.339	1.209	0.976	1.634	5.231	1.501	6.682	7.735
x <sub>40</sub>	X <sub>δcol</sub>	0.038	0.037	0.469	0.236	1.456	2.286	0.710	1.739	2.518
x <sub>41</sub>	Y <sub>δlat</sub>	0.282	0.269	0.279	0.303	0.226	5.812	0.313	7.748	8.742
x <sub>42</sub>	Y <sub>δlon</sub>	0.148	0.260	0.027	0.049	0.121	6.229	0.096	7.718	9.359
x <sub>43</sub>	Y <sub>δped</sub>	0.078	0.086	0.090	0.060	0.144	6.158	0.066	7.999	8.354
x <sub>44</sub>	Y <sub>δcol</sub>	0.599	0.324	0.892	0.907	0.433	4.897	0.823	7.789	7.444
x <sub>45</sub>	Z <sub>δlat</sub>	0.347	0.443	0.385	0.466	0.467	3.444	0.373	3.940	8.611
x <sub>46</sub>	Z <sub>δlon</sub>	0.015	0.120	0.055	0.109	0.131	0.172	0.071	0.086	0.106
x <sub>47</sub>	Z <sub>δped</sub>	0.589	0.176	0.138	0.921	0.572	6.808	0.138	8.433	9.316
x <sub>48</sub>	Z <sub>δcol</sub>	0.012	0.093	0.028	0.083	0.107	1.030	0.041	0.264	0.351
x <sub>49</sub>	L <sub>δlat</sub>	0.077	0.081	0.072	0.086	0.066	6.056	0.075	3.020	8.275
x <sub>50</sub>	L <sub>δlon</sub>	0.250	0.262	0.246	0.241	0.234	5.775	0.276	2.811	8.667
x <sub>51</sub>	L <sub>δped</sub>	0.069	0.063	0.062	0.072	0.072	6.052	0.052	1.876	8.909
x <sub>52</sub>	L <sub>δcol</sub>	0.191	0.180	0.171	0.255	0.169	5.590	0.348	0.630	9.126
x <sub>53</sub>	M <sub>δlat</sub>	0.093	0.471	0.363	0.110	0.383	5.930	0.253	0.687	9.394
x <sub>54</sub>	M <sub>δlon</sub>	0.193	0.234	0.164	0.241	0.306	0.458	0.181	0.460	0.045
x <sub>55</sub>	M <sub>δped</sub>	1.763	2.787	3.004	10.859	12.893	5.708	10.096	4.225	1.421
x <sub>56</sub>	M <sub>δcol</sub>	0.634	0.422	0.672	0.390	0.339	1.595	0.719	2.038	1.227
x <sub>57</sub>	N <sub>δlat</sub>	0.198	0.006	0.300	0.073	0.728	5.932	0.667	7.993	8.996
x <sub>58</sub>	N <sub>δlon</sub>	0.272	0.214	0.091	0.242	0.615	5.547	0.238	4.658	4.499
x <sub>59</sub>	N <sub>δped</sub>	0.058	0.023	0.015	0.070	0.066	1.291	0.029	0.412	1.113
x <sub>60</sub>	N <sub>δcol</sub>	0.021	0.086	0.177	0.101	0.179	4.722	0.281	4.236	3.759

The physical system matrices are constructed using the above presented optimization results. Then the true model and the estimated one are simulated with the same 3-2-1-1 excitation signals (Figure 5.4). The results show that simulation responses of “true model”, “SID model” and “Physical SID model” to 3-2-1-1 excitation signals are almost the same for Case 2.1 to Case 2.9. The simulation results are illustrated in Figure 6.19.

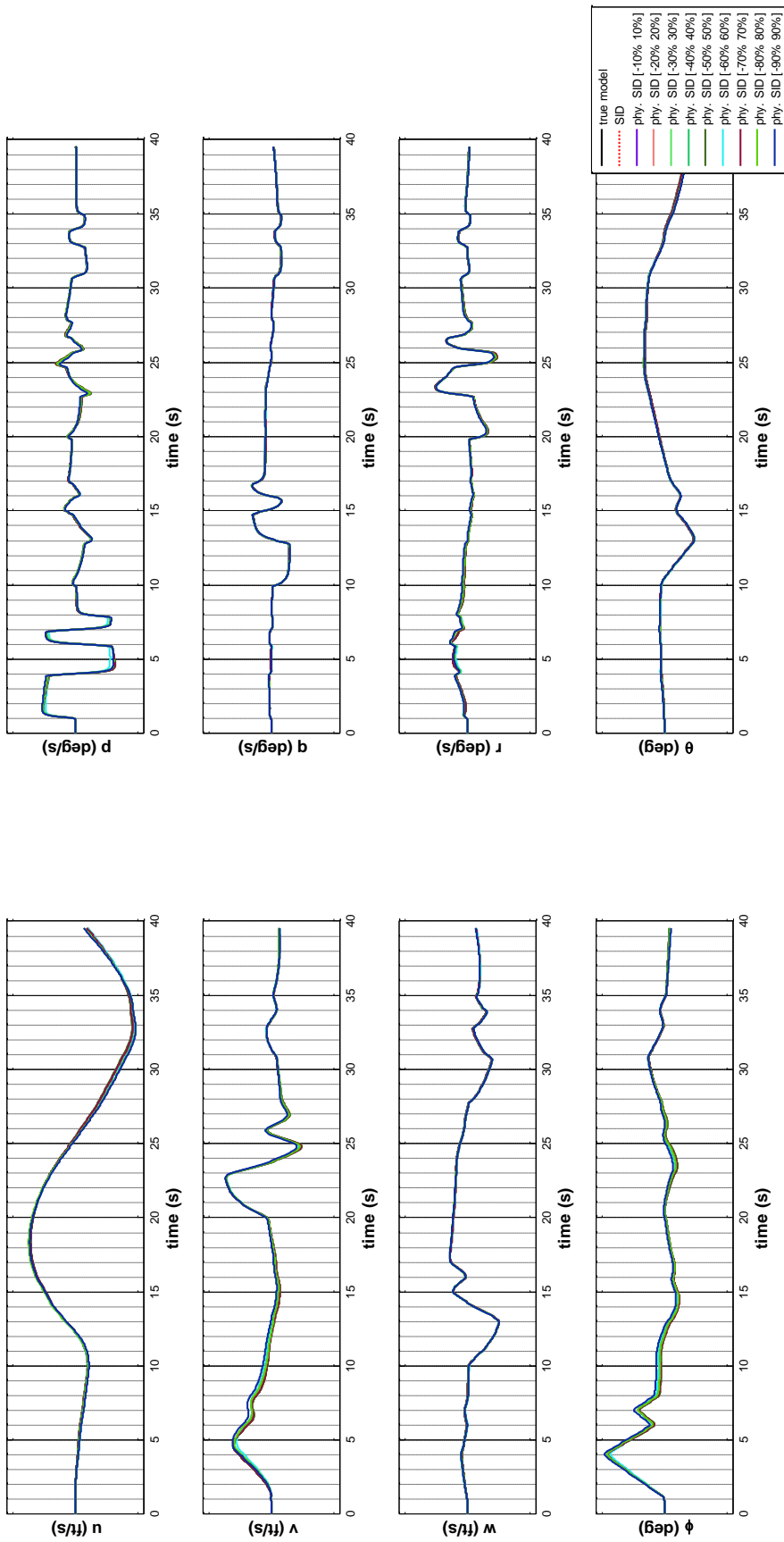


Figure 6.19. Comparison of Outputs for 3-2-1-1 Excitation Signals (Identification Using Linear Model Simulation)



The doublet input is applied in four channels sequentially for verification. The inputs and the outputs of this test case are given in Figure 6.11 and Figure 6.20 respectively. TIC values are calculated according to Eq.(117) and presented in Table 6.8.

Table 6.8. *TIC Values for Case 2 (for Verification Signal)*

optimization batch for constraint set		TIC value
Case 2.1	[-10% 10%]	0.0021
Case 2.2	[-20% 20%]	0.0022
Case 2.3	[-30% 30%]	0.0023
Case 2.4	[-40% 40%]	0.0022
Case 2.5	[-50% 50%]	0.0029
Case 2.6	[-60% 60%]	0.0109
Case 2.7	[-70% 70%]	0.0024
Case 2.8	[-80% 80%]	0.0181
Case 2.9	[-90% 90%]	0.0179

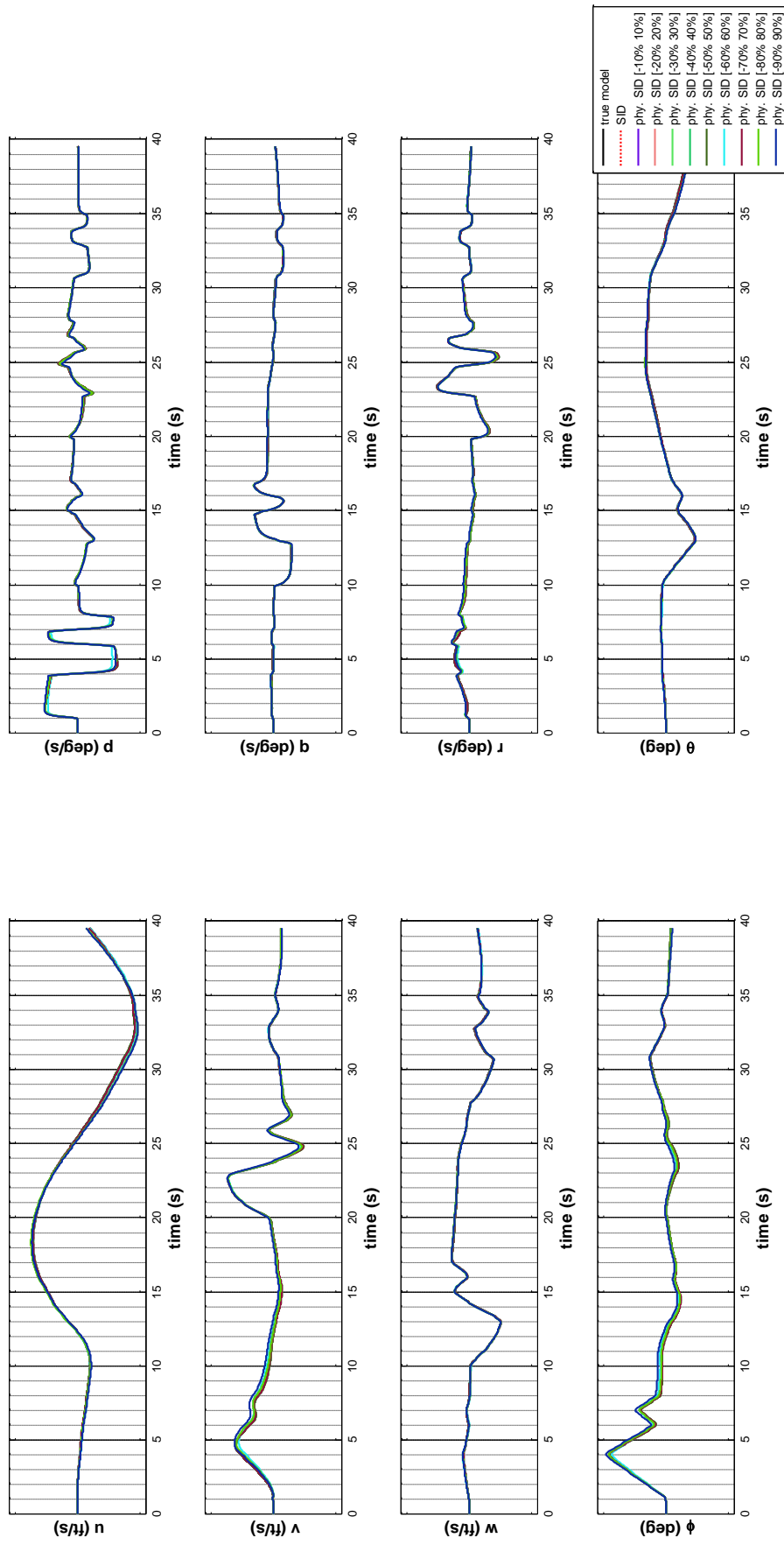


Figure 6.20. Comparison of Outputs for Doublet Excitation Signals (Identification Using Linear Model Simulation)

Now we can compare the results of the IP algorithm with the results of the SQP algorithm in the sense of convergence performance. When we compare the results of Case 1 with the results of Case 2, it seems that IP algorithm is superior than SQP. Although all of the derivatives are converged in Case 1, there are some deficiencies in Case 2. In addition, Convergence is faster in Case 1 compared to Case 2 regarding the required iteration numbers. Also, the convergence seems better in Case 1 compared to Case 2. In other words, the convergence characteristics are more obvious in the results of Case 1. Considering the “estimation error” values and the TIC values, again Case 1 results are superior to those of Case 2. However, we are suspicious about the results due to one feature of the IP algorithm. It is known that the middle point of the lower and upper bounds plays very important role for IP algorithm [52]. The common practice is setting the initial iteration to the midpoint of finite bounds. Selecting a symmetric error bound (i.e. [-10% 10%]) will provide a solution in a few steps. However, in a real application it is quite unfeasible. Therefore, the optimizations are repeated for the different asymmetric constraint levels (i.e. [-10% 20%], [-10% 30%]) to better understand the performance of the IP algorithm. For this reason, Case 3 analysis is performed to examine the effect of different asymmetric constraint levels on the optimization results.

### **6.3. Case 3 – Interior Point vs. SQP under asymmetric constraint conditions**

The aim of this study is to compare the performance of the IP algorithm with the results of SQP algorithm under asymmetric constraint levels. The summary of the conditions for this case study are given in Table 6.9.

Table 6.9. Summary of Analysis Conditions for Case 3 (Interior Point vs. SQP under asymmetric constraint conditions)

	alg.	constraint (error bound) for the parameters of			initial condition for the parameters of			number of run
		$A_{phy}$	$B_{phy}$	$T$	$A_{phy}$	$B_{phy}$	$T$	
Case 3.1	IP	[-10% 10%]		NA	random sel.	const	20	
Case 3.2	IP	[-10% 20%]		NA	random sel.	const	20	
Case 3.3	IP	[-10% 30%]		NA	random sel.	const	20	
Case 3.4	SQP	[-10% 10%]		NA	random sel.	const	20	
Case 3.5	SQP	[-10% 20%]		NA	random sel.	const	20	
Case 3.6	SQP	[-10% 30%]		NA	random sel.	const	20	

According to Table 6.9, 120 (20x6) optimization runs are performed in total. Iteration index versus minimization output curves of each parameter (each stability and control derivatives) are gathered from all of the optimization runs. The results are analyzed in two groups. In the first group, the “Interior Point” algorithm is examined under symmetric and asymmetric constraint levels ([-10% 10%], [-10% 20%], [-10% 30%]). The convergence results are illustrated in Figure 6.21 and Figure 6.22. It is observed that some certain derivatives do not converge to the same value for different constraint levels. Also, for those parameters, the asymmetric constraint conditions ([-10% 20%], [-10% 30%]) do not converge to the true value whereas true values are achieved for symmetric constraint levels ([-10% 10%]). These results support our suspicion.

However, for Case 3.4- Case 3.6 where the SQP algorithm is utilized, the results are not much affected due to symmetric and asymmetric constraint levels (Figure 6.23 and Figure 6.24). It is observed that the parameters with low level of significance do not converge for SQP algorithm whereas they converge to different values for different / unrepeatable constraint levels in case of IP algorithm. Figure 6.25 summarizes these findings.

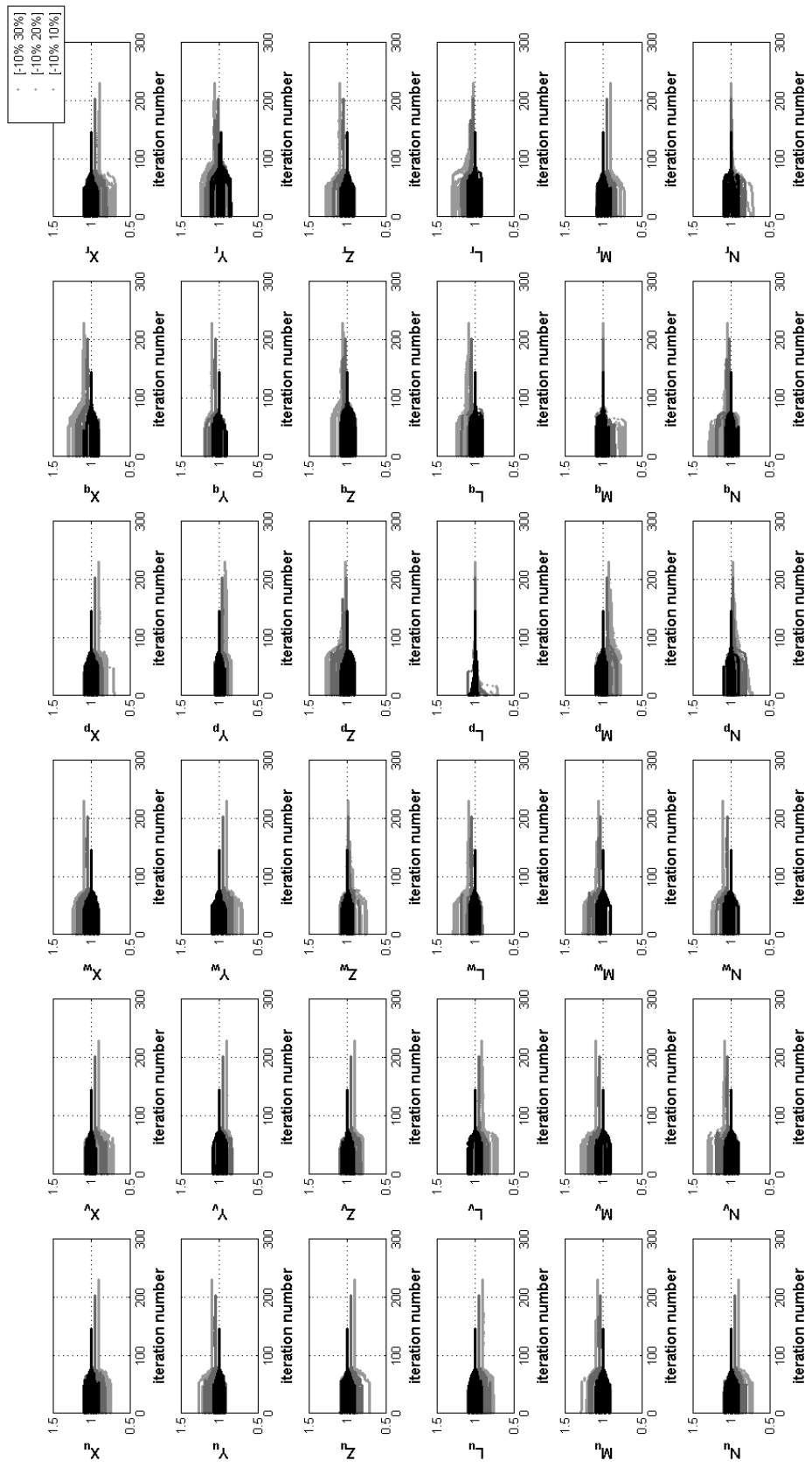


Figure 6.2.1. Convergence of Stability Derivatives (Case 3.1 – Case 3.3)

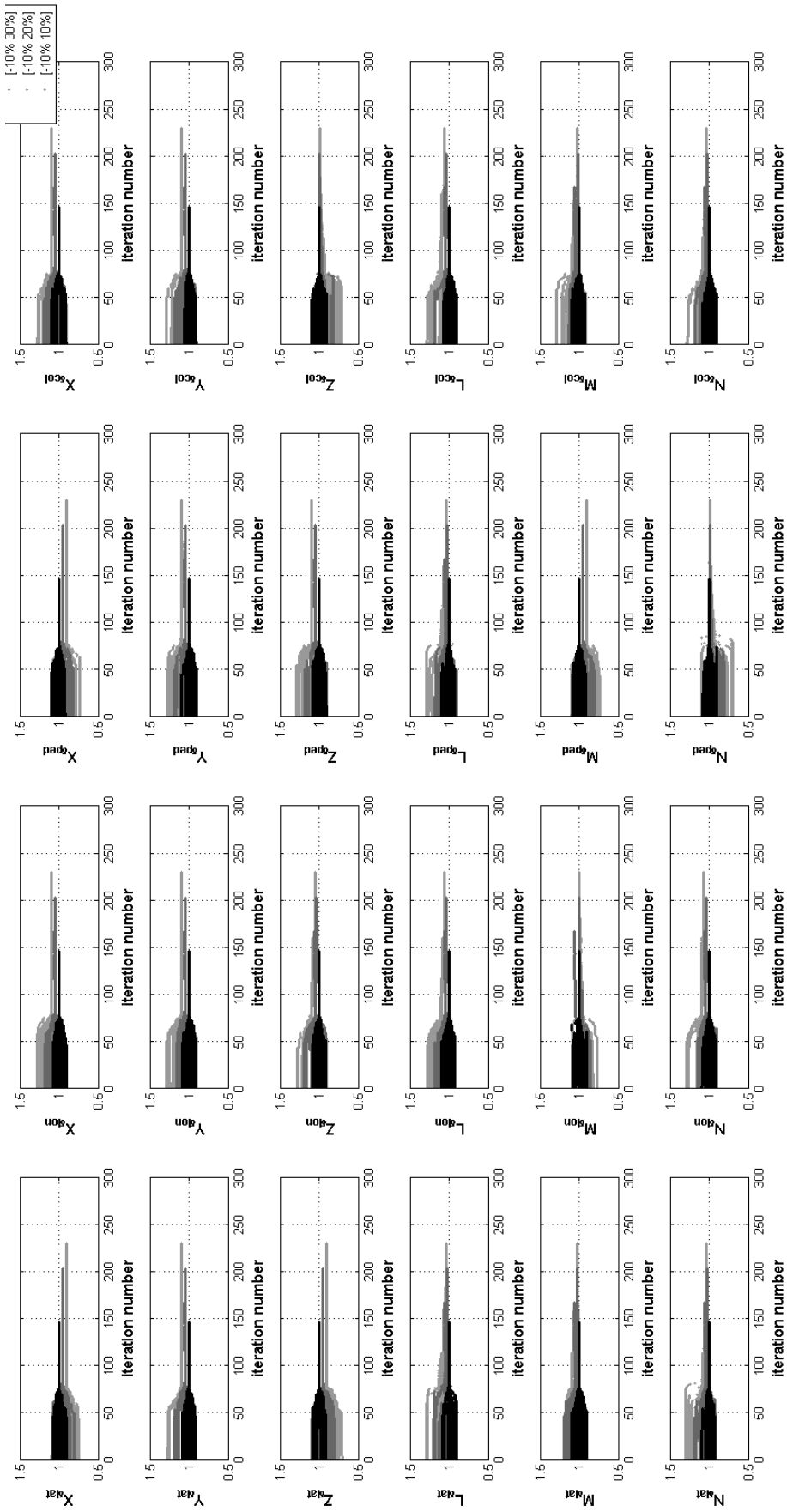


Figure 4.15: Convergence metrics for various parameters (Case 3.1 - Case 3.2)

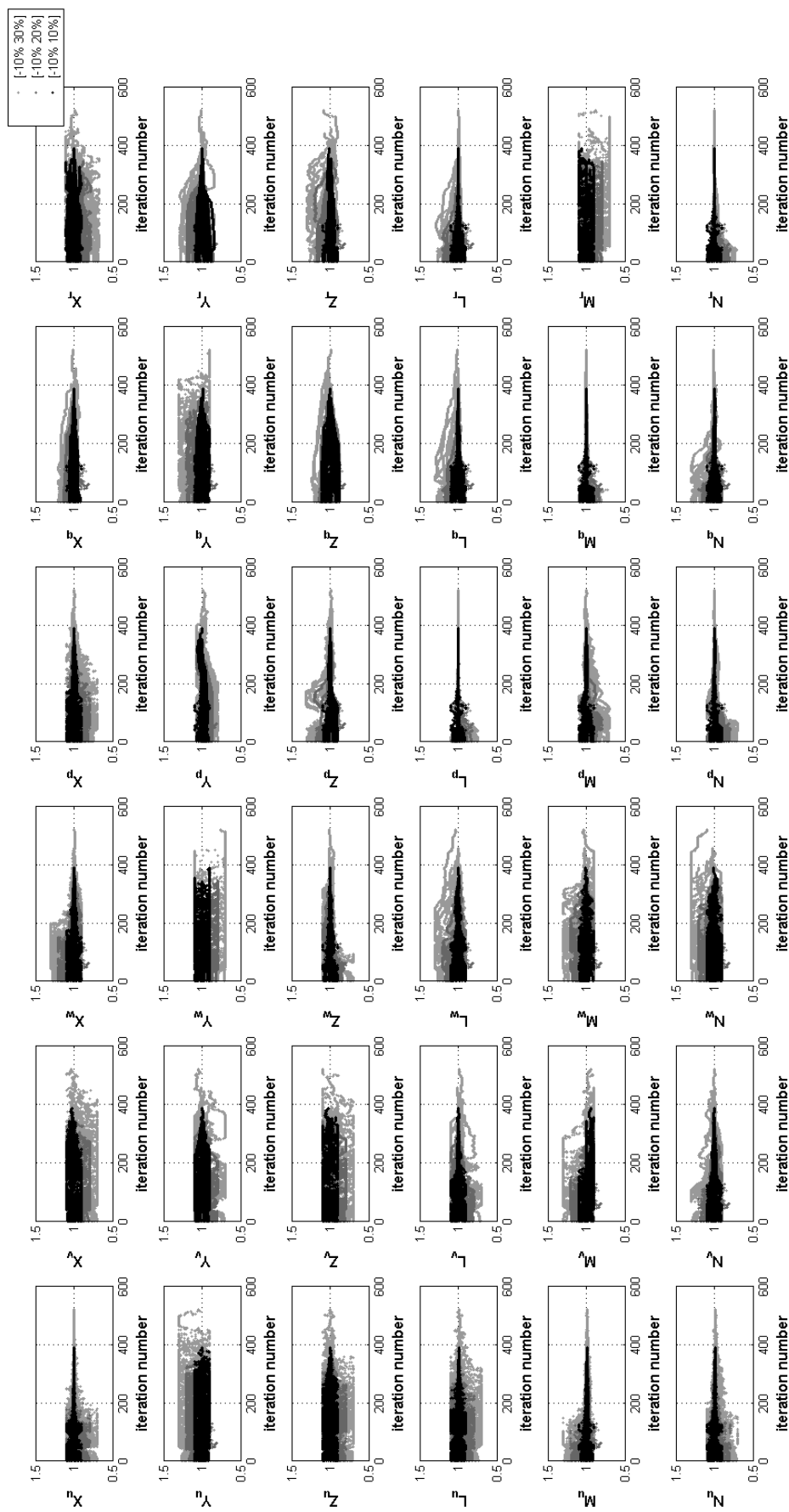


Figure 6.2.3. Convergence of Stability Derivatives (Case 3.4 – Case 3.6)

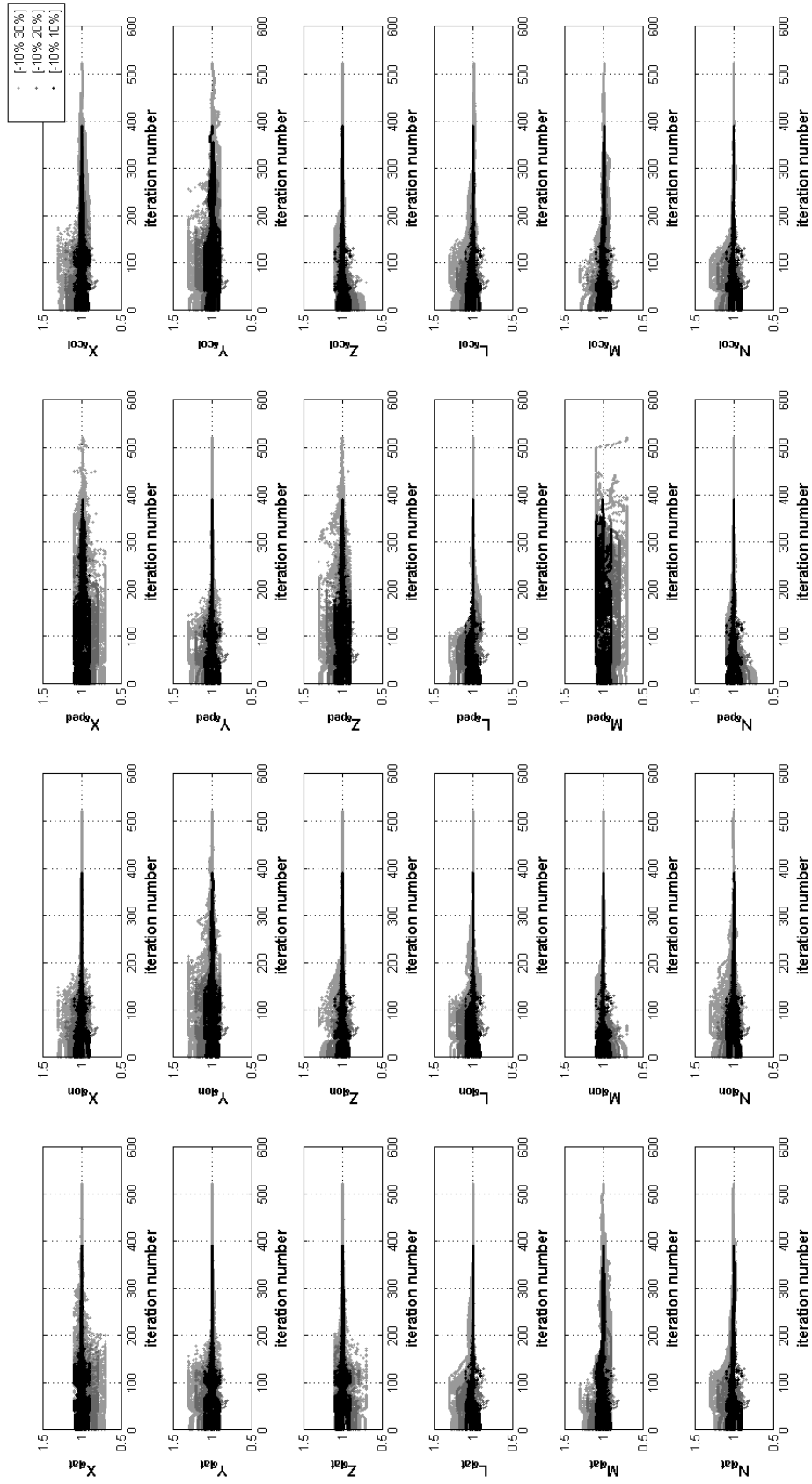


Figure 6.24. Convergence of Control Derivatives (Case 3.4 – Case 3.6)



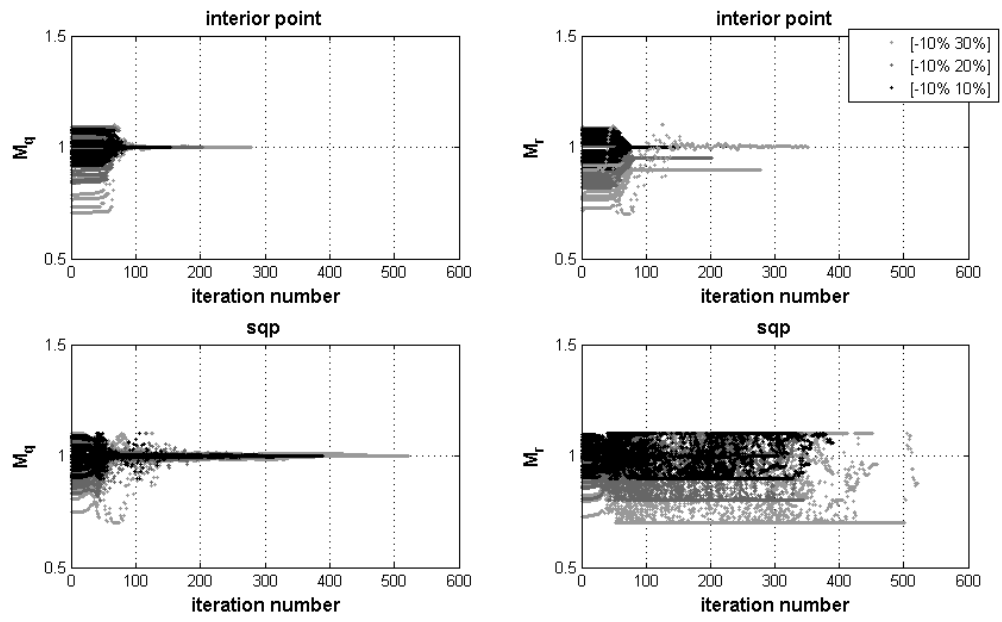


Figure 6.25. Comparison in Convergence of Control Derivatives (Case3)

#### 6.4. Case 4 Arbitrary initial value for T matrix

As we mentioned before we set the initial value of  $T$  matrix to the inverse of the  $C$  matrix (obtained by subspace identification). In this way, we aim to speed up the computations if  $T$  complies with the Eq. (85). We aim to show that, our results which are found by the previous optimizations are not achieved by chance. Thus, we examine the performance of our solution method for arbitrary initialization of  $T$ . Then we compare the results with the previous analysis given in Case2. The summary of the conditions for this case study are given in Table 6.10.

According to Table 6.10, 360 (20x9x2) optimization runs are performed in total. In addition to 180 optimization runs which are generated in Case 2, additional 180 runs are generated with arbitrary initialization of  $T$  (Case 4.10-Case 4.18). Then the results are compared with each other. For example, the results of Case 2.1 are compared with Case 4.10 where the only difference is initialization of  $T$ . This comparison is continued until to analysis the last pair (Case 4.9 and Case 4.18). The results are compared by minimization output curves of each parameter with respect to iteration index. The findings are illustrated in Figure 6.26 - Figure 6.43. The results show that obtaining a solution is possible with the arbitrary initialization of  $T$ . The only degradation is that the optimization takes a little bit more time with arbitrary initialization.

The percentage estimation error results given in Table 6.11 and Table 6.12 support the idea that the results found by the methodology cannot be by chance.

The physical system matrices are constructed using the above presented optimization results (Case 4.10 - Case 4.18). Then the true model and the estimated one are simulated with the same 3-2-1-1 excitation signals (Figure 5.4). The results show that simulation responses of “true model”, “SID model” and “Physical SID model” to 3-2-1-1 excitation signals are almost the same for Case 4.10 to Case 4.18. The simulation results are illustrated in Figure 6.44.

Table 6.10. Summary of Analysis for Case 4 (Arbitrary Initial Point for T Matrix)

	alg.	constraint (error bound) for the parameters of			initial condition for the parameters of			number of run
		$A_{phy}$	$B_{phy}$	$T$	$A_{phy}$	$B_{phy}$	$T$	
Case 4.1	SQP	[-10% 10%]		NA <sup>9</sup>	random sel.		const	20
Case 4.2	SQP	[-20% 20%]		NA	random sel.		const	20
Case 4.3	SQP	[-30% 30%]		NA	random sel.		const	20
Case 4.4	SQP	[-40% 40%]		NA	random sel.		const	20
Case 4.5	SQP	[-50% 50%]		NA	random sel.		const	20
Case 4.6	SQP	[-60% 60%]		NA	random sel.		const	20
Case 4.7	SQP	[-70% 70%]		NA	random sel.		const	20
Case 4.8	SQP	[-80% 80%]		NA	random sel.		const	20
Case 4.9	SQP	[-90% 90%]		NA	random sel.		const	20
Case 4.10	SQP	[-10% 10%]		NA	random sel.			20
Case 4.11	SQP	[-20% 20%]		NA	random sel.			20
Case 4.12	SQP	[-30% 30%]		NA	random sel.			20
Case 4.13	SQP	[-40% 40%]		NA	random sel.			20
Case 4.14	SQP	[-50% 50%]		NA	random sel.			20
Case 4.15	SQP	[-60% 60%]		NA	random sel.			20
Case 4.16	SQP	[-70% 70%]		NA	random sel.			20
Case 4.17	SQP	[-80% 80%]		NA	random sel.			20
Case 4.18	SQP	[-90% 90%]		NA	random sel.			20

<sup>9</sup> NA means “Not Applicable”

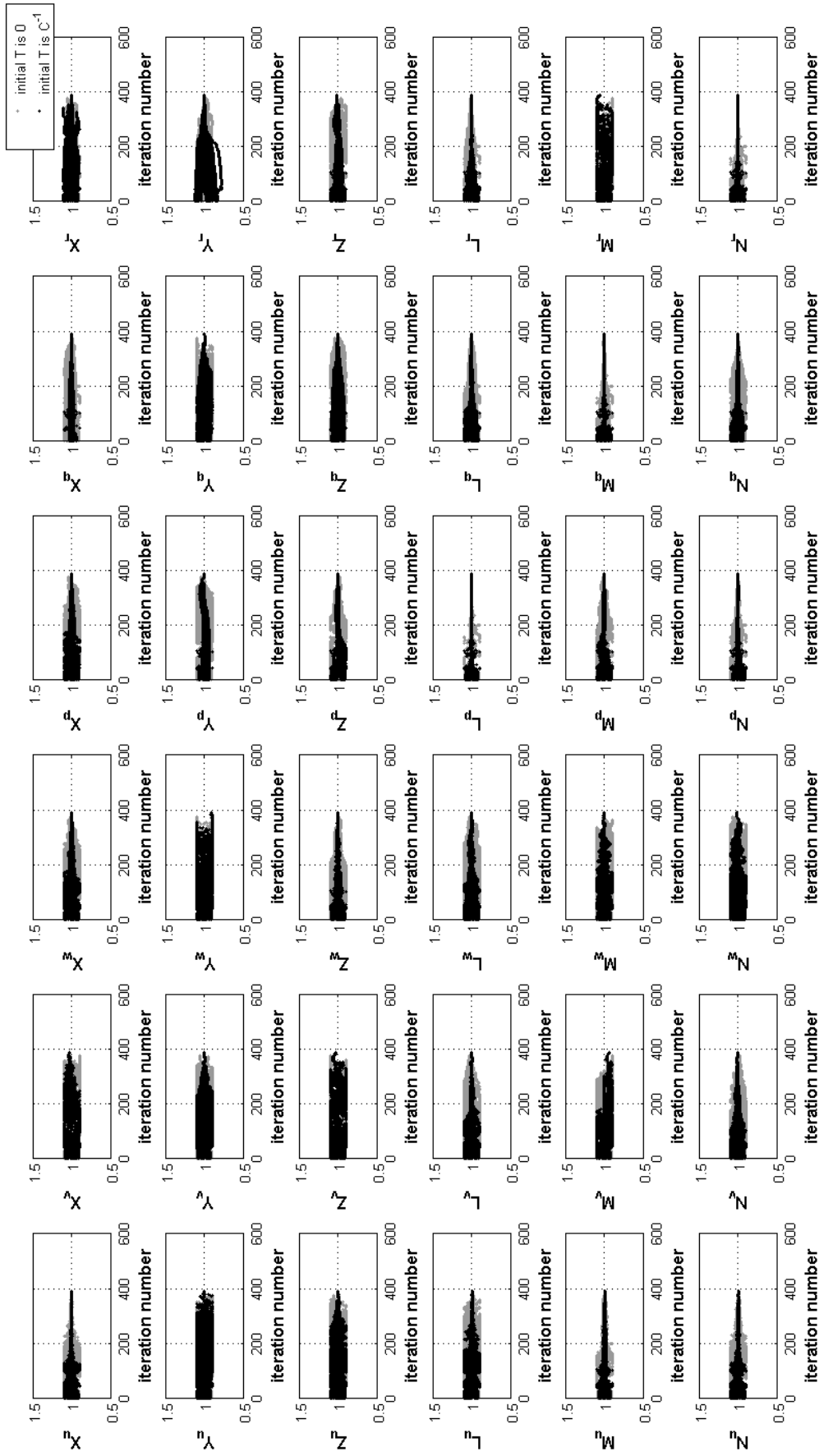


Figure 6.26. Convergence of Stability Derivatives (Case 4.1 vs. Case 4.10)

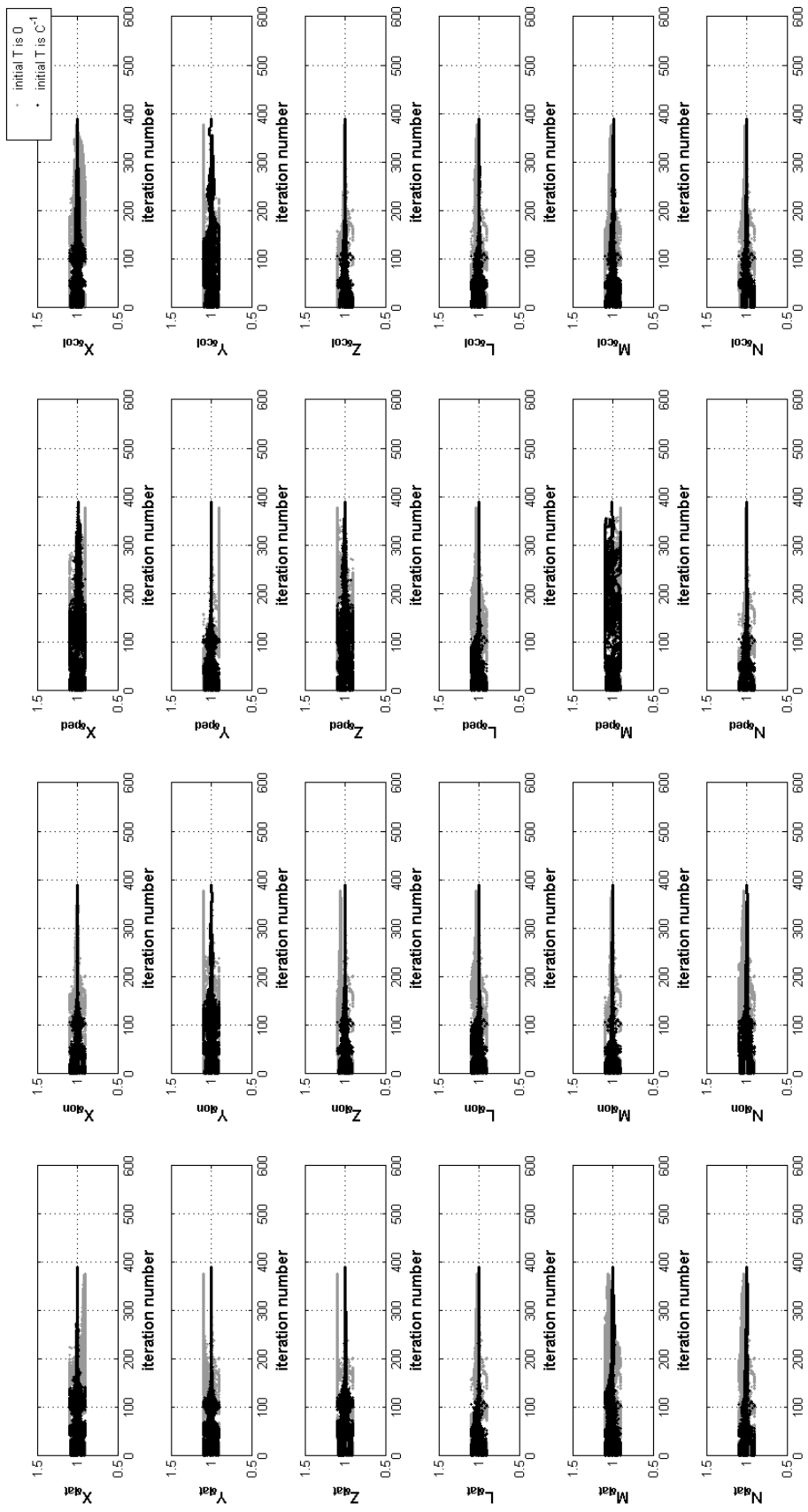


Figure 6.27. Convergence of Control Derivatives (Case 4.1 vs. Case 4.10)

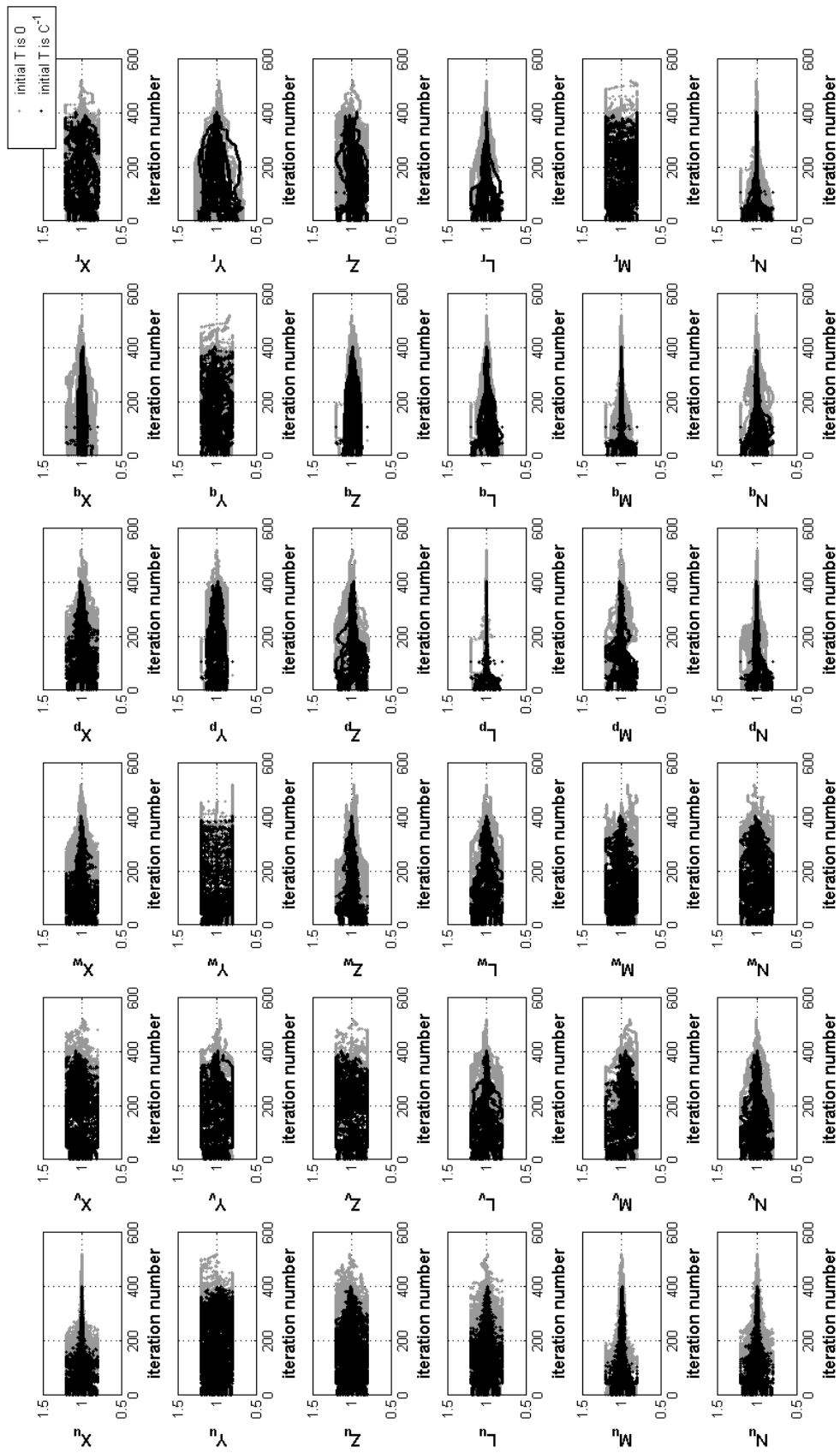


Figure 6.28. Convergence of Stability Derivatives (Case 4.2 vs. Case 4.11)

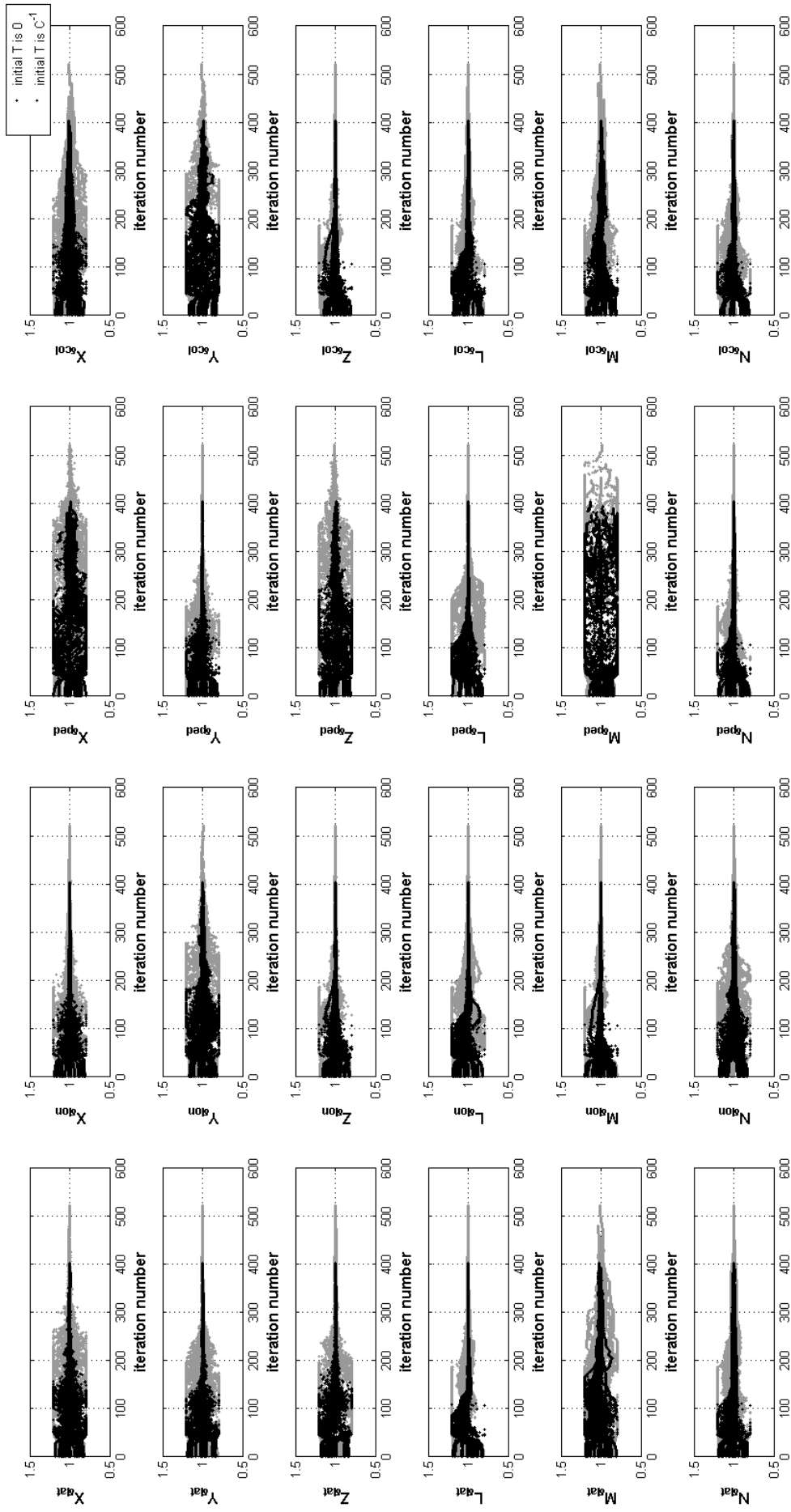


Figure 6.29. Convergence of Control Derivatives (Case 4.1 vs. Case 4.11)

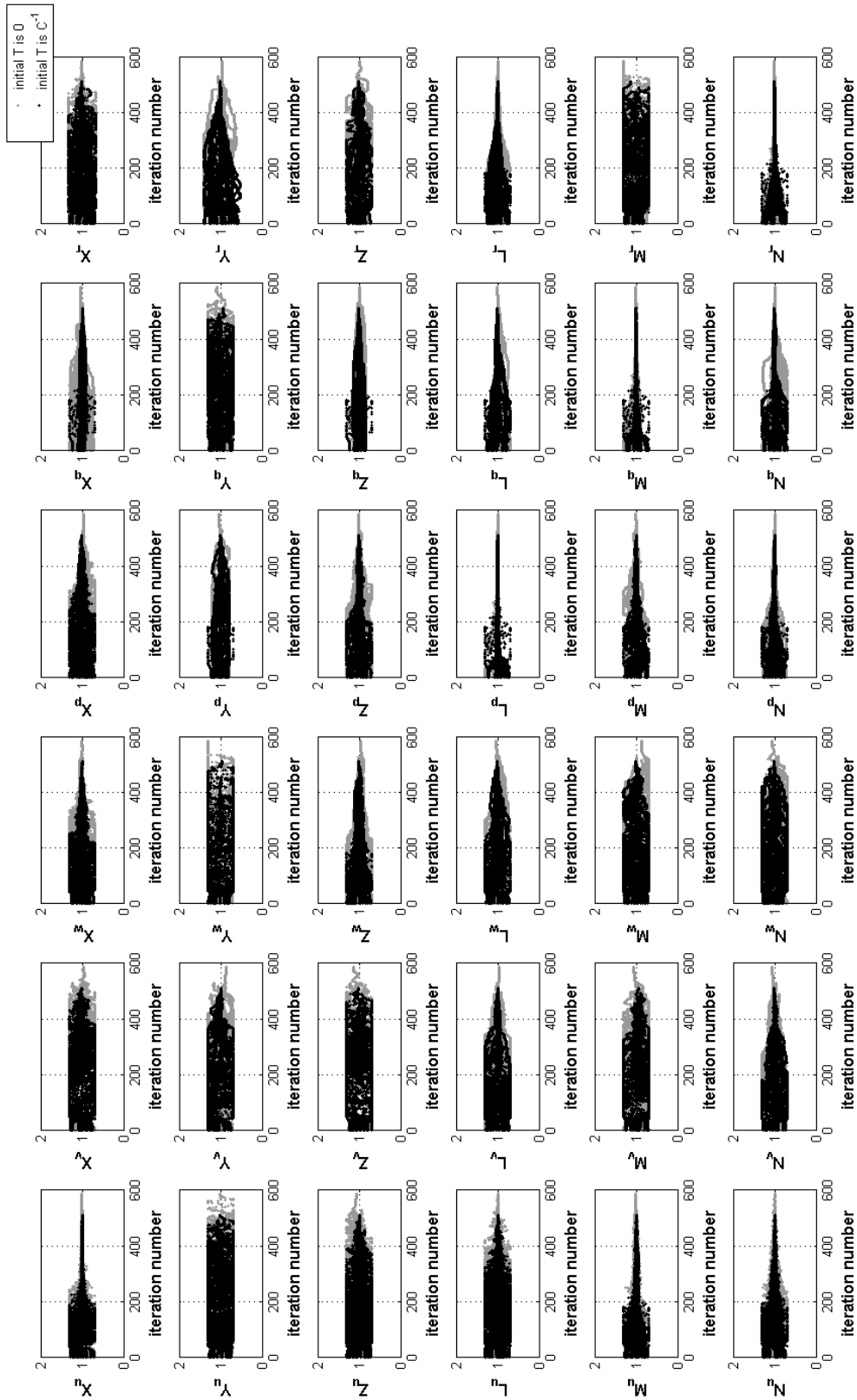


Figure 6.30. Convergence of Stability Derivatives (Case 4.3 vs. Case 4.12)



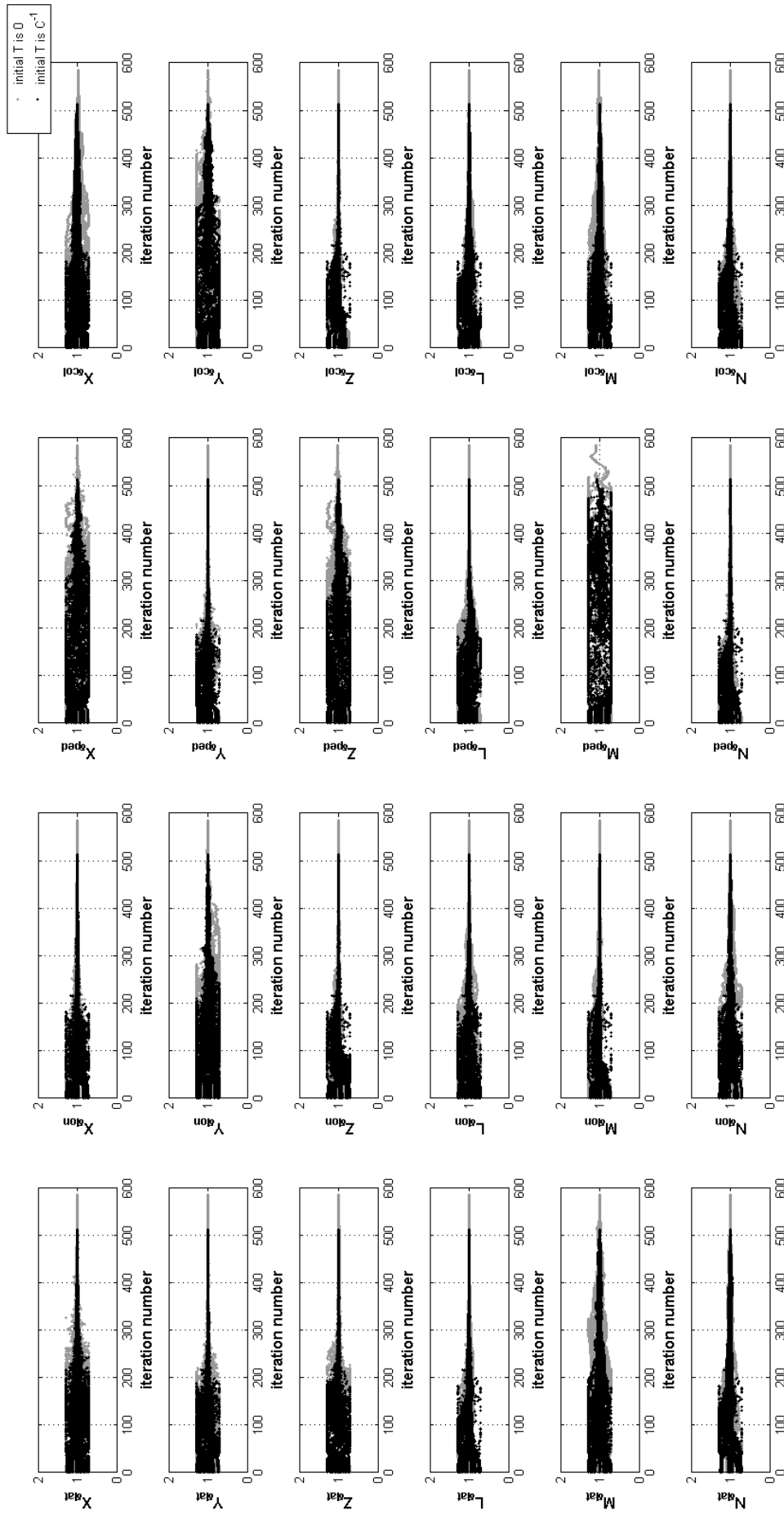


Figure 6.31. Convergence of Control Derivatives (Case 4.3 vs. Case 4.12)

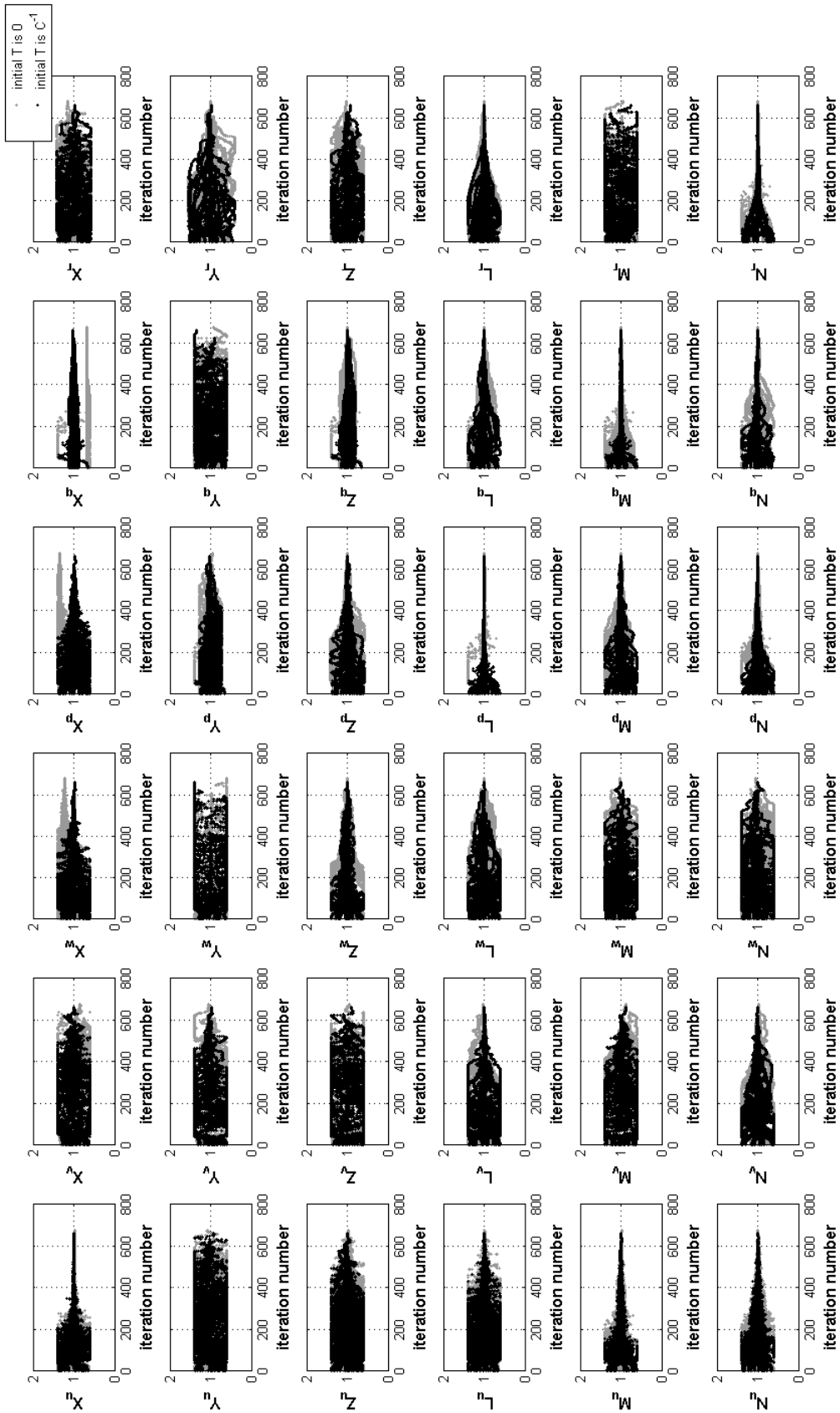


Figure 6.32. Convergence of Stability Derivatives (Case 4.4 vs. Case 4.13)

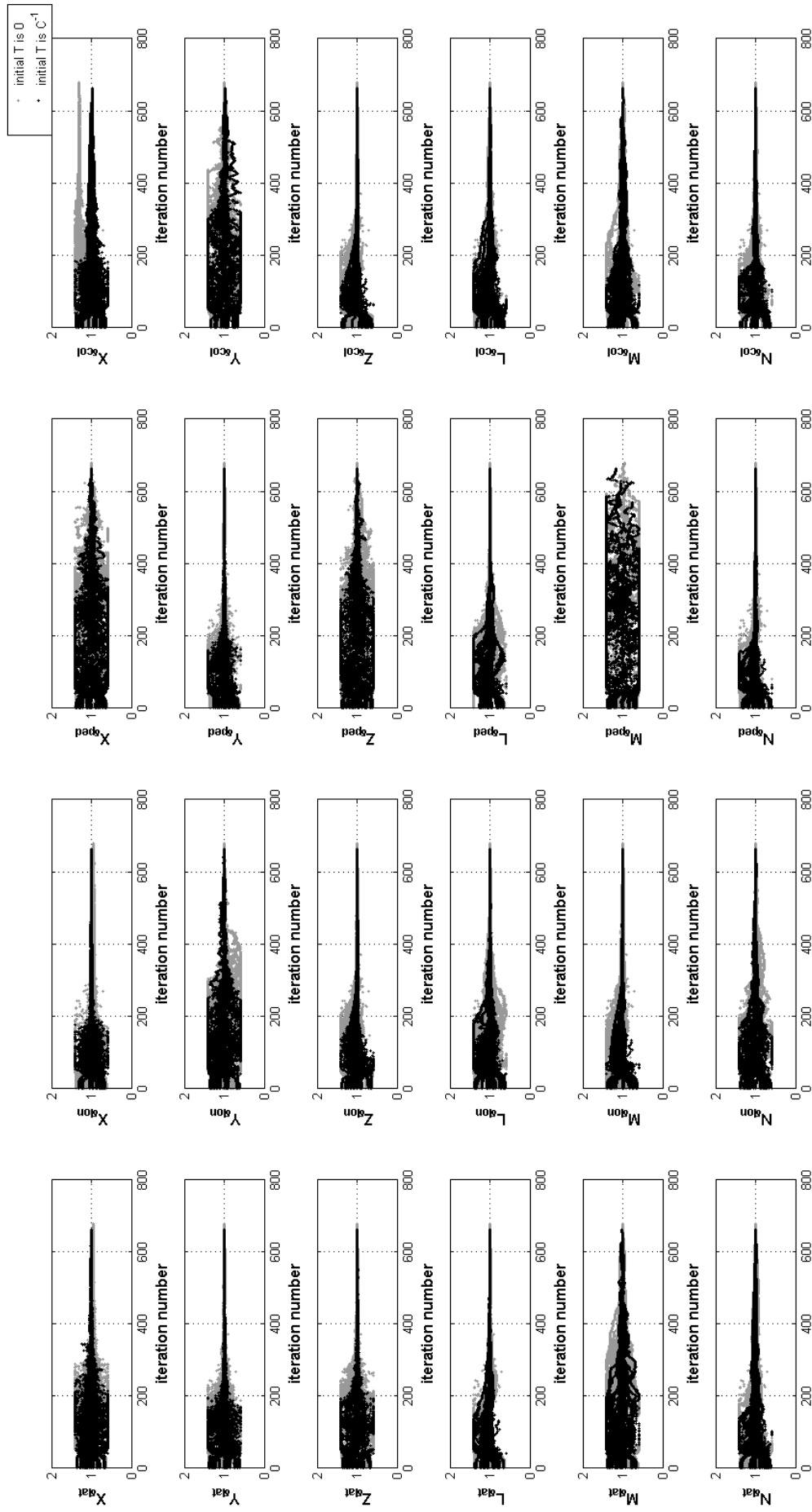


Figure 6.33. Convergence of Control Derivatives (Case 4.4 vs. Case 4.13)

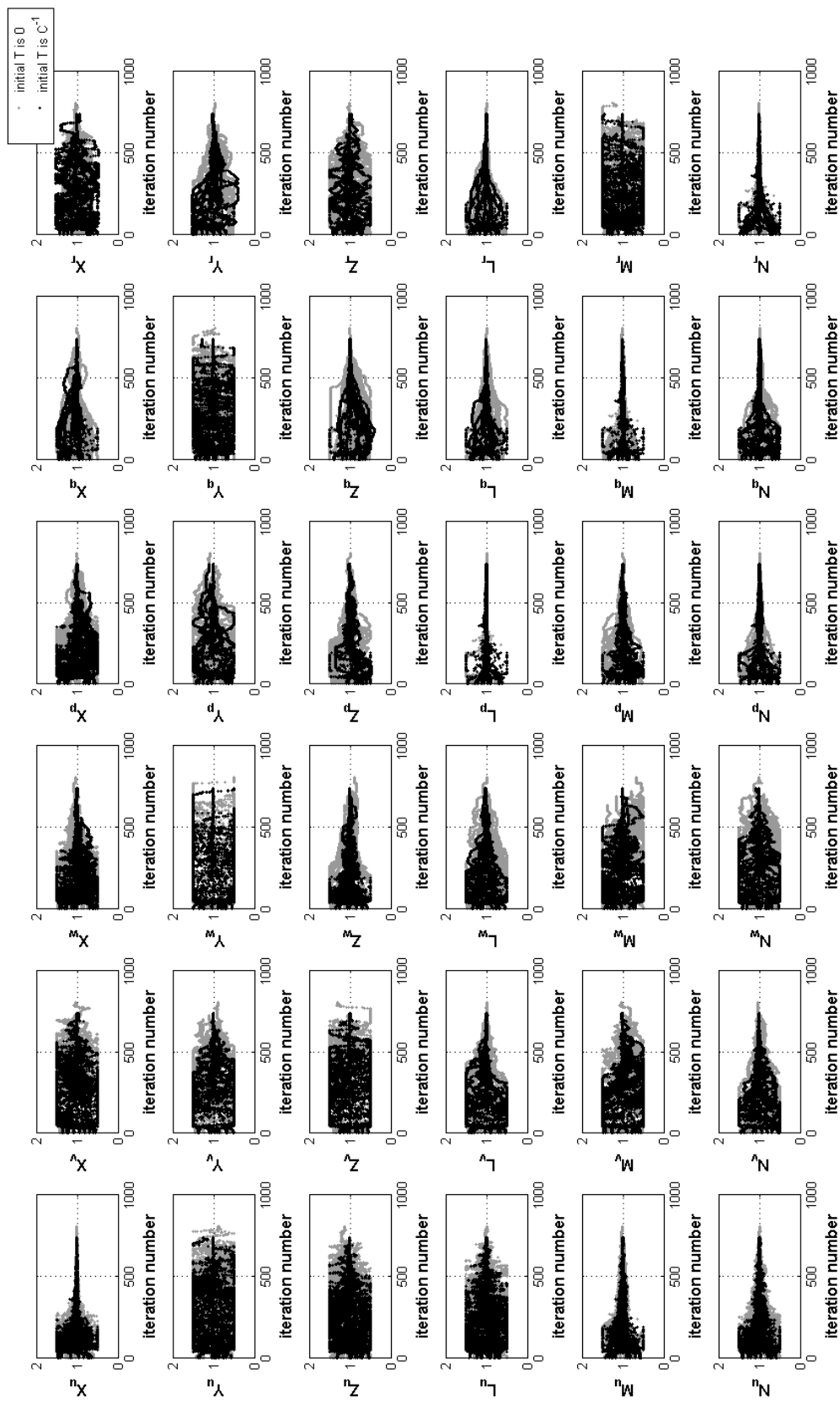


Figure 6.34. Convergence of Stability Derivatives (Case 4.5 vs. Case 4.14)

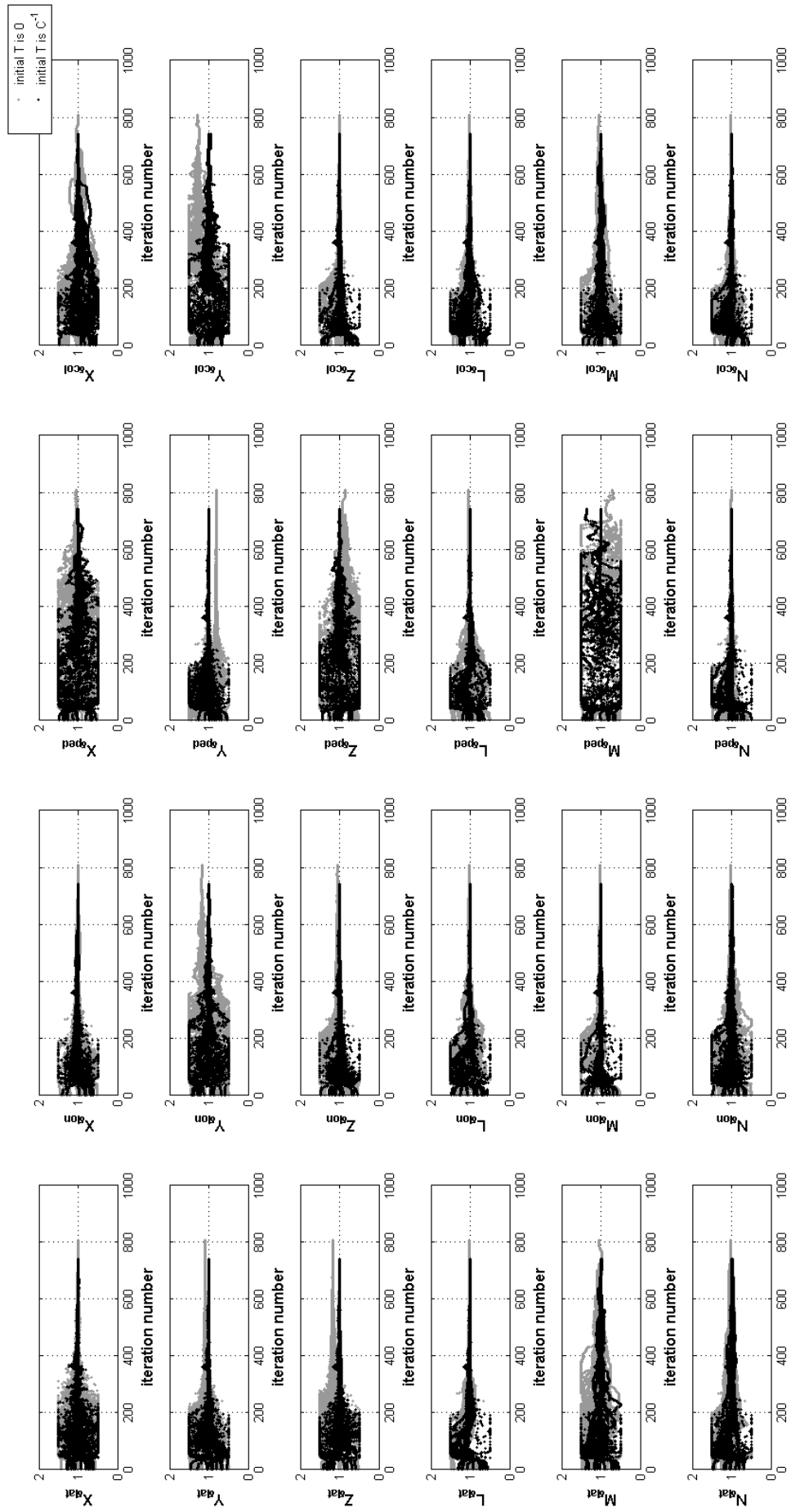


Figure 6.35. Convergence of Control Derivatives (Case 4.5 vs. Case 4.14)

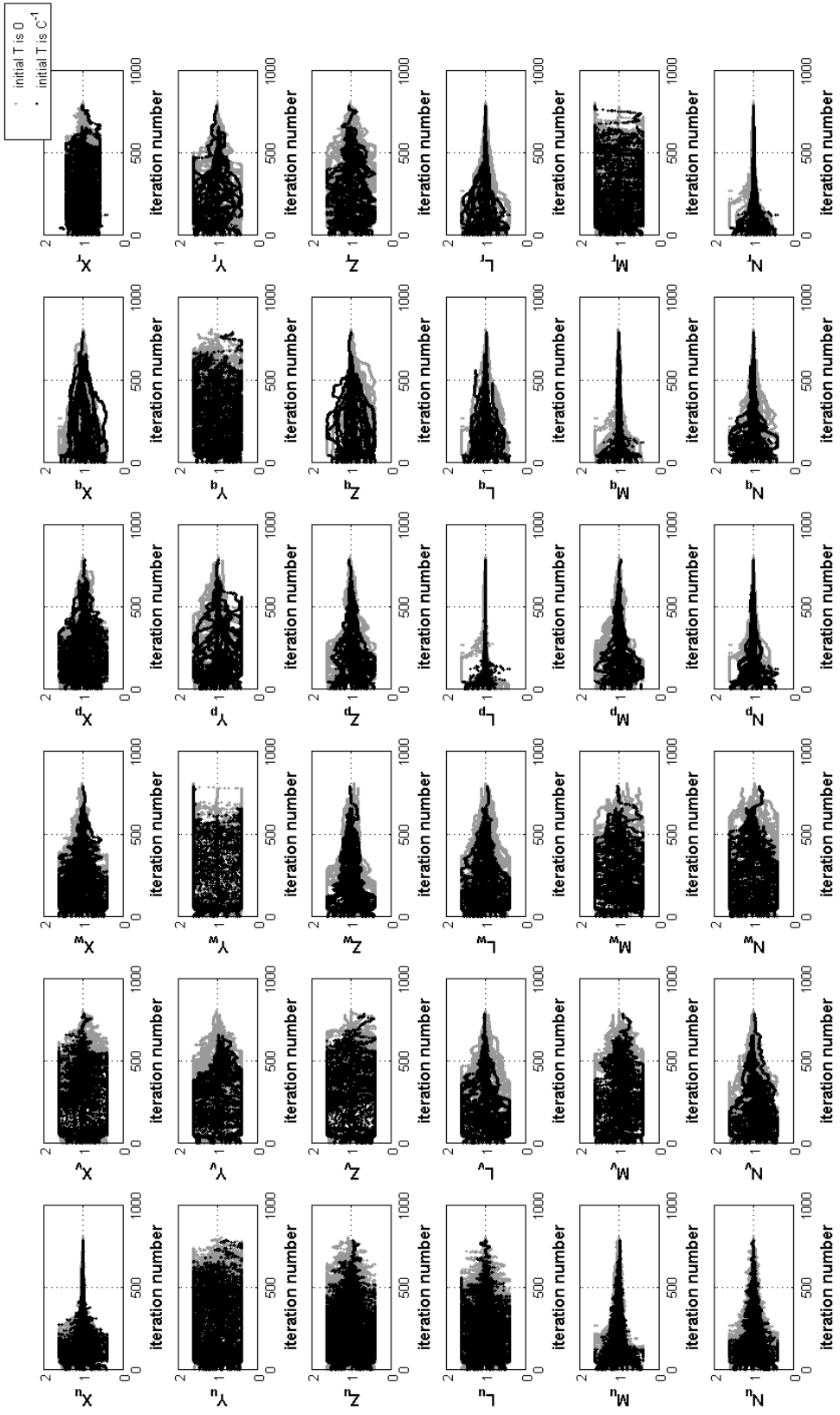


Figure 6.36. Convergence of Stability Derivatives (Case 4.6 vs. Case 4.15)

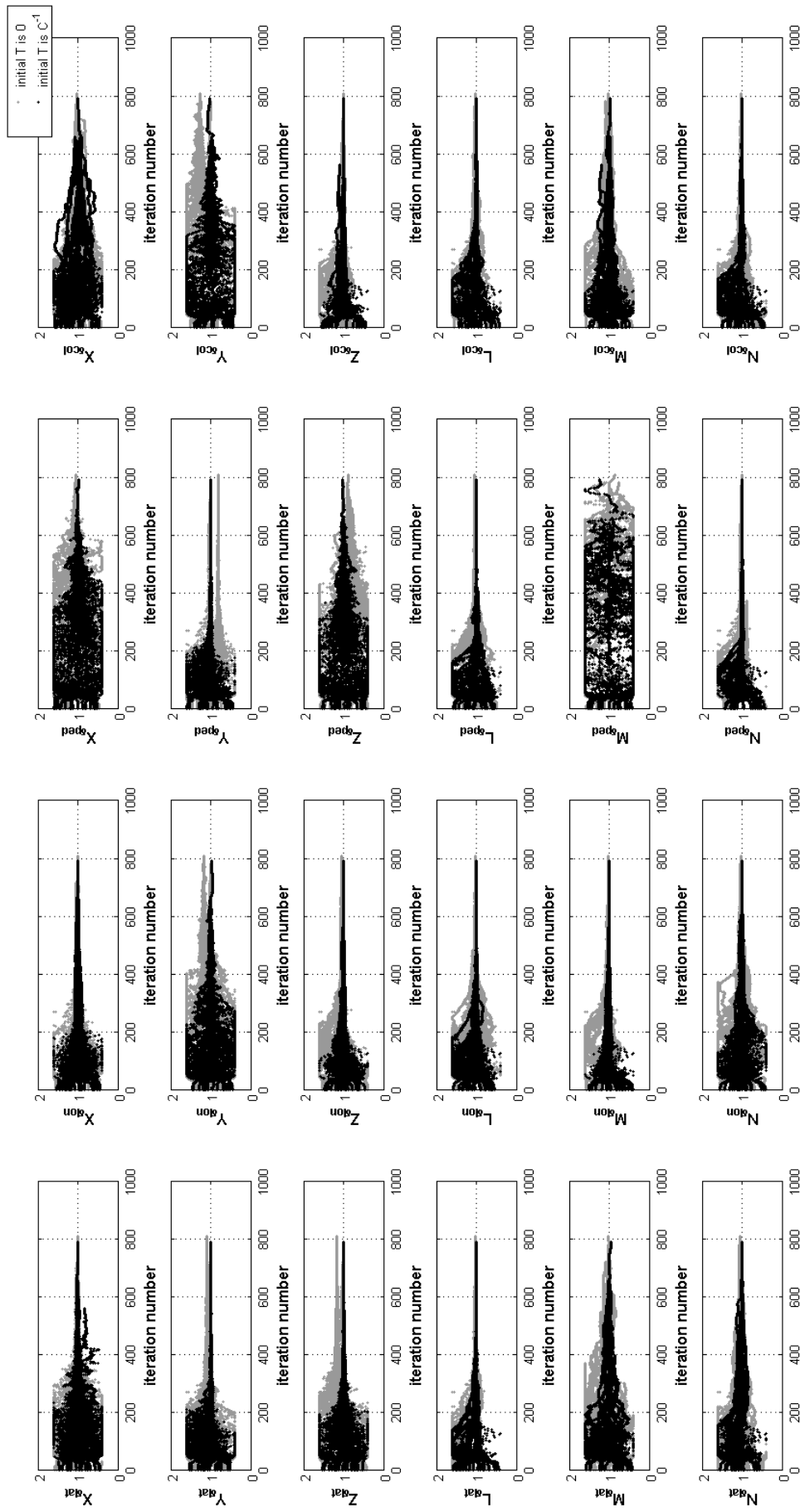


Figure 6.37. Convergence of Control Derivatives (Case 4.6 vs. Case 4.15)

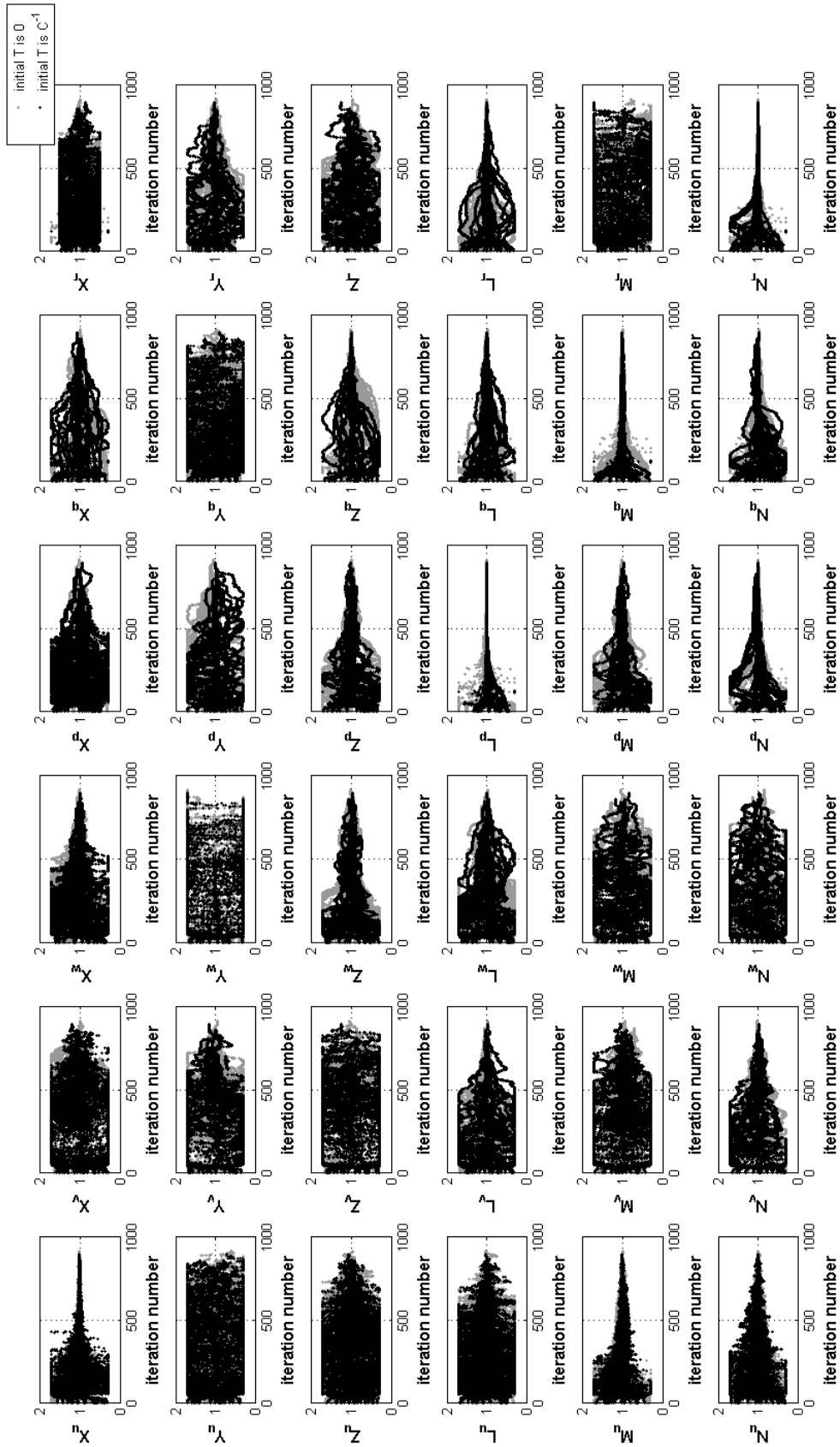


Figure 6.38. Convergence of Stability Derivatives (Case 4.7 vs. Case 4.16)



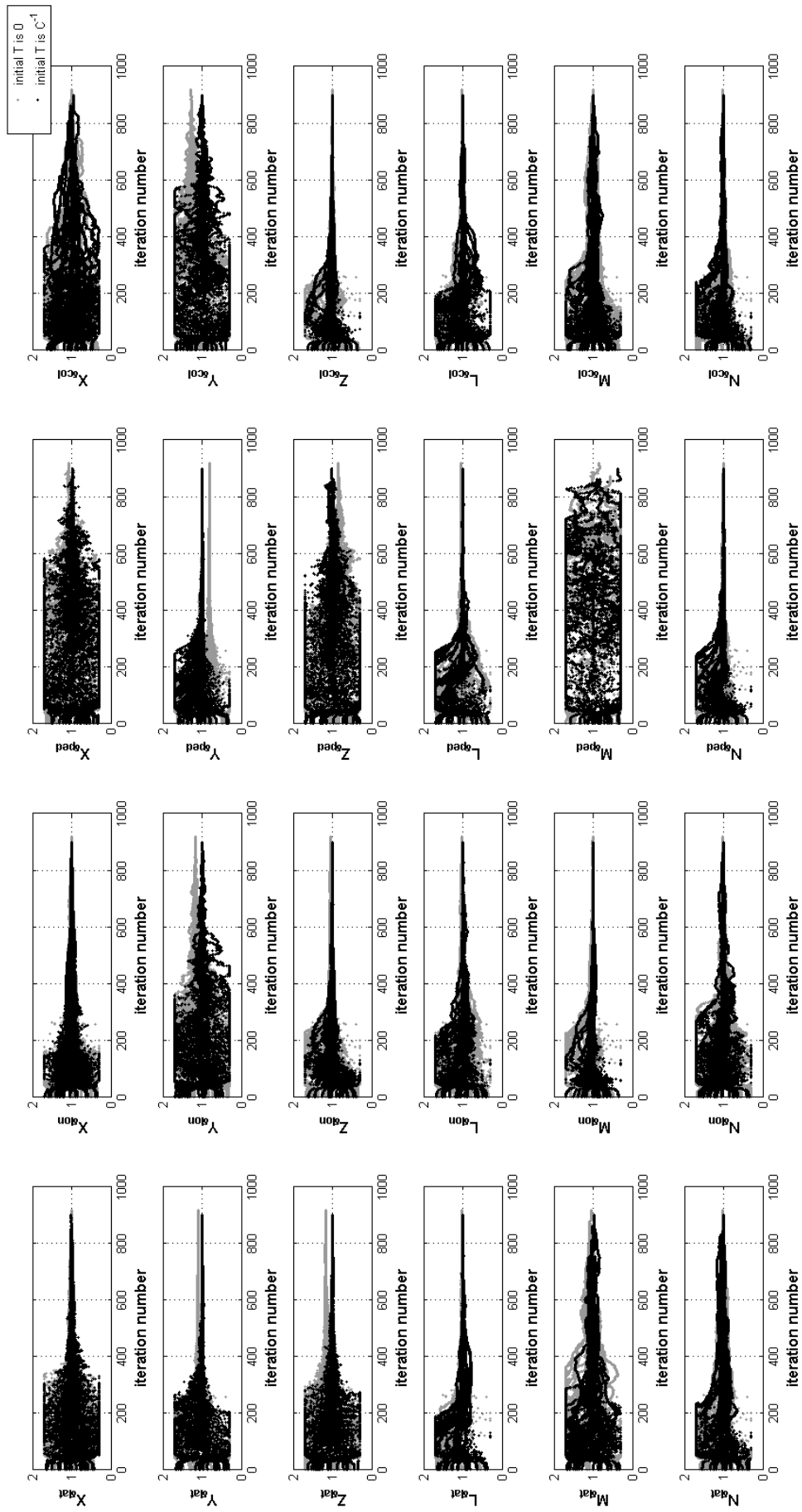


Figure 6.39. Convergence of Control Derivatives (Case 4.7 vs. Case 4.16)

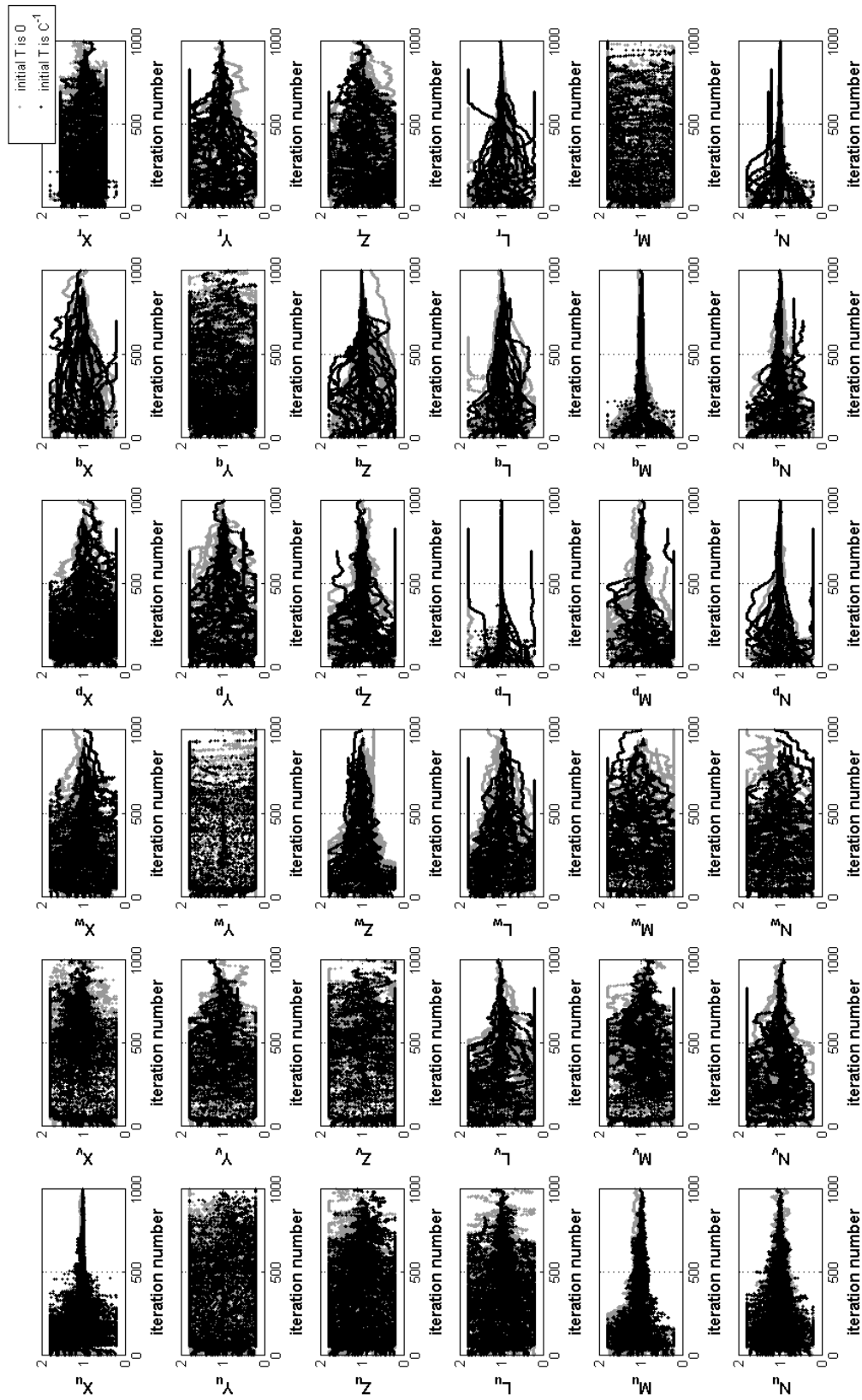


Figure 6.40. Convergence of Stability Derivatives (Case 4.8 vs. Case 4.17)

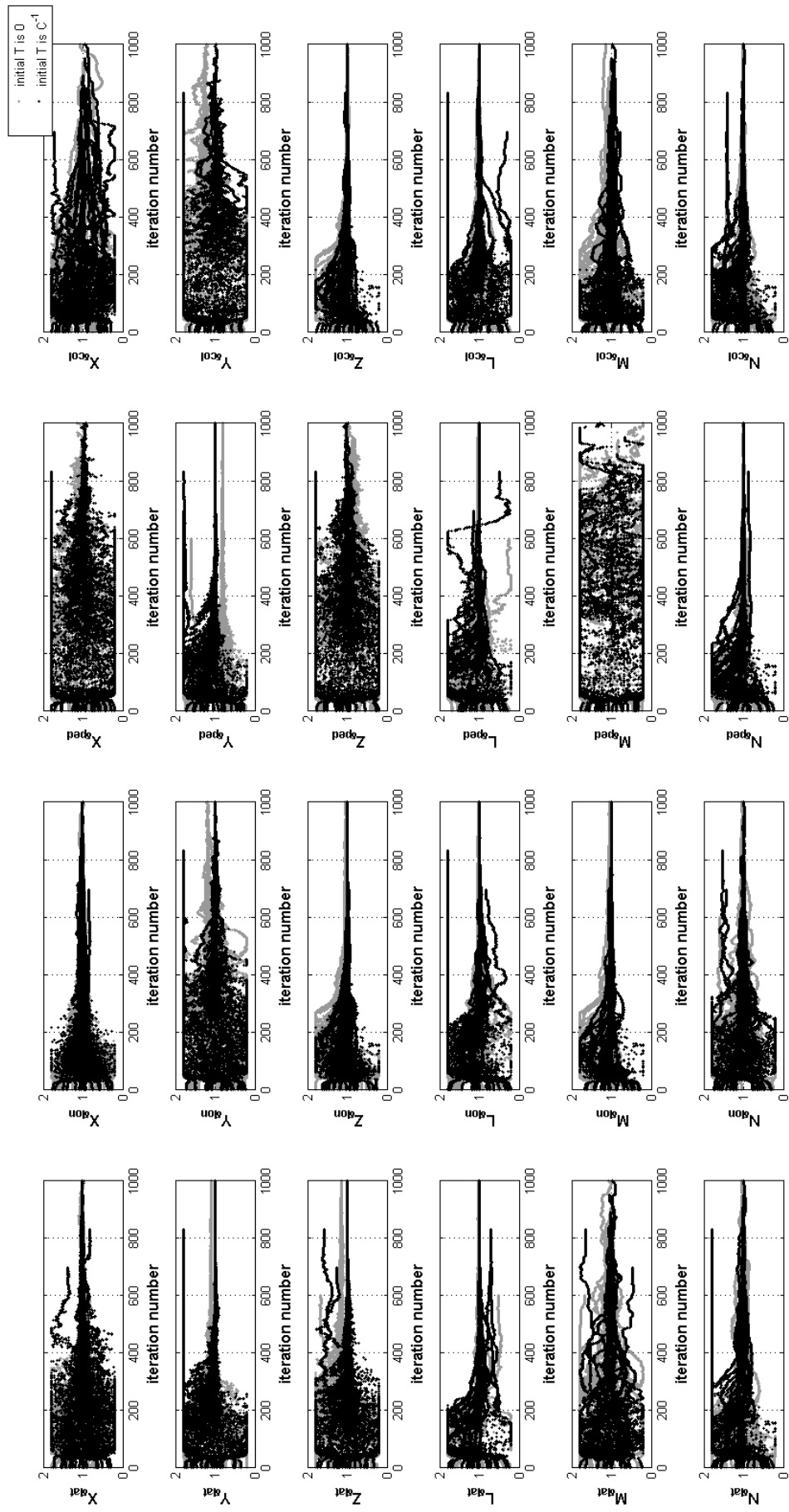


Figure 6.41. Convergence of Control Derivatives (Case 4.8 vs. Case 4.17)

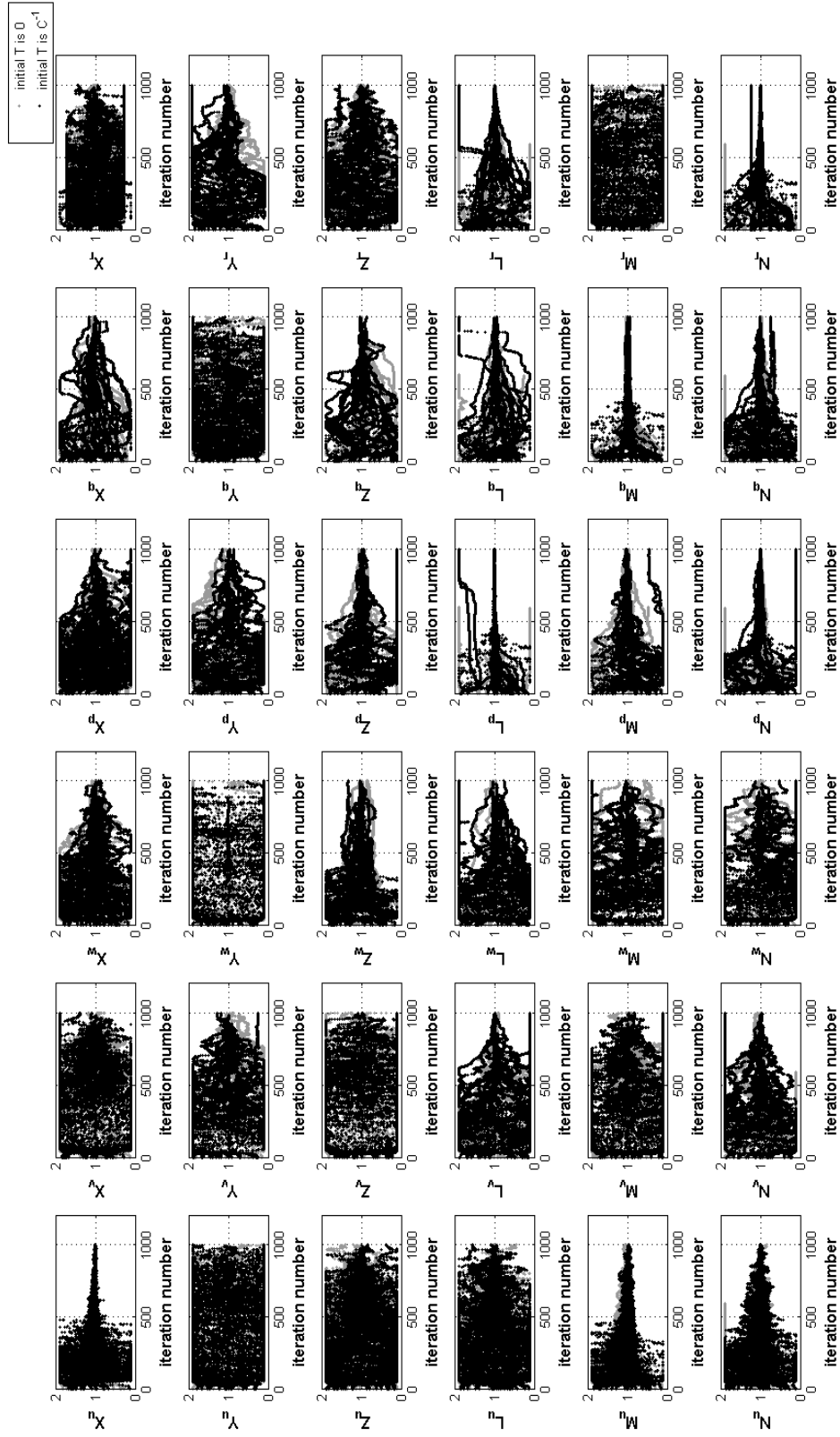


Figure 6.42. Convergence of Stability Derivatives (Case 4.9 vs. Case 4.18)

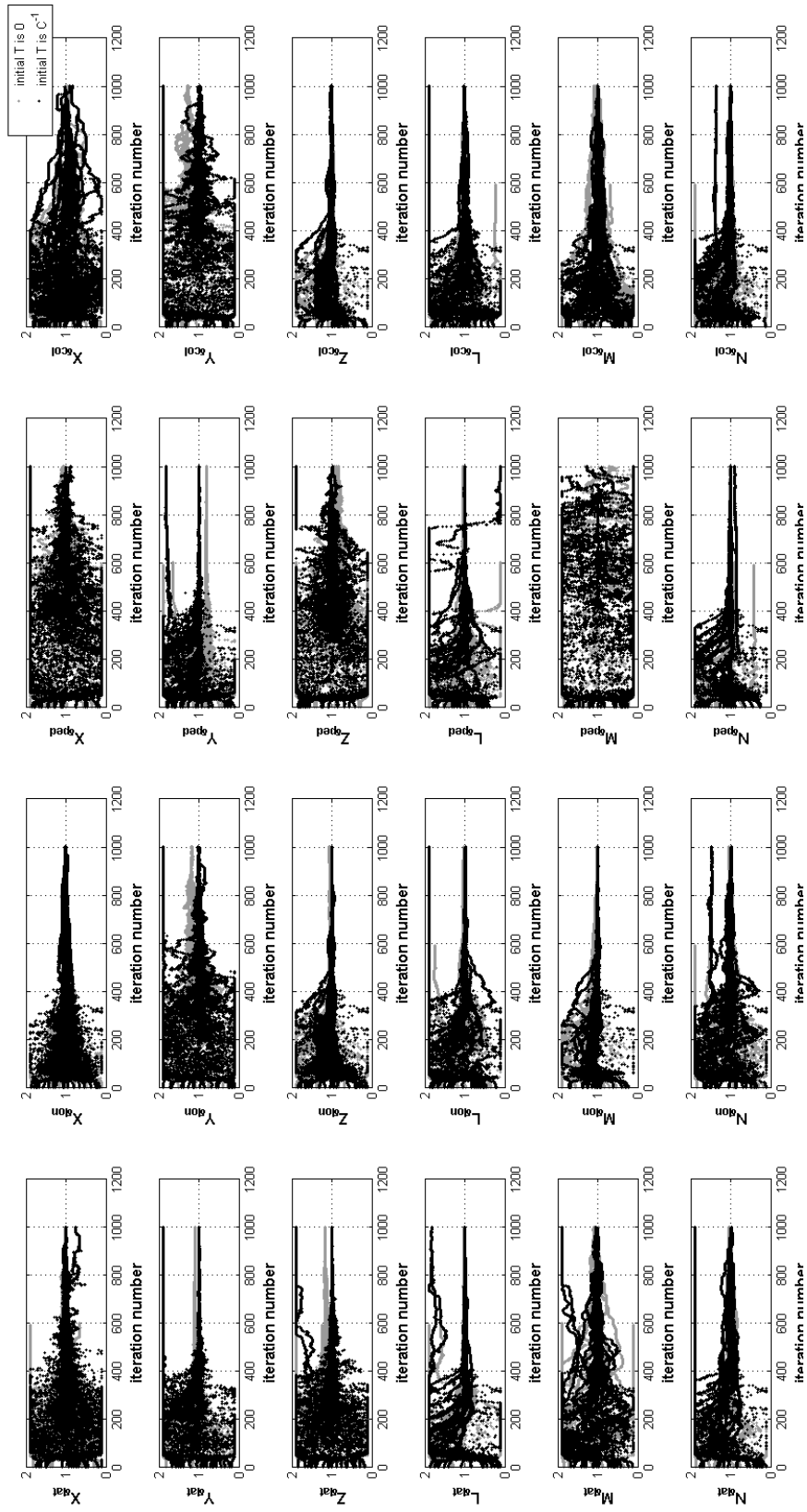


Figure 6.43. Convergence of Control Derivatives (Case 4.9 vs. Case 4.18)

Table 6.11. Percentage Estimation Errors of Stability Derivatives – Case 4

parameter	symbol	estimation error %								
		Case 4.10	Case 4.11	Case 4.12	Case 4.13	Case 4.14	Case 4.15	Case 4.16	Case 4.17	Case 4.18
		[-10% 10%]	[-20% 20%]	[-30% 30%]	[-40% 40%]	[-50% 50%]	[-60% 60%]	[-70% 70%]	[-80% 80%]	[-90% 90%]
x <sub>1</sub>	X <sub>u</sub>	0.351	0.262	0.388	2.101	0.326	0.296	0.321	0.408	1.551
x <sub>2</sub>	X <sub>v</sub>	0.623	4.271	2.339	1.154	0.968	1.347	2.108	2.439	3.479
x <sub>3</sub>	X <sub>w</sub>	0.221	1.029	0.051	2.438	1.940	2.598	2.316	3.431	2.195
x <sub>4</sub>	X <sub>p</sub>	1.306	2.191	2.064	2.856	1.377	2.225	2.201	5.057	8.006
x <sub>5</sub>	X <sub>q</sub>	4.184	5.487	5.643	3.411	1.476	1.369	1.078	1.269	2.305
x <sub>6</sub>	X <sub>r</sub>	0.381	0.786	0.739	4.573	0.155	0.243	1.646	1.910	0.821
x <sub>7</sub>	Y <sub>u</sub>	4.400	4.126	2.108	0.623	2.834	2.485	9.578	8.434	11.822
x <sub>8</sub>	Y <sub>v</sub>	0.646	1.584	2.827	0.216	1.452	4.333	1.316	4.068	5.274
x <sub>9</sub>	Y <sub>w</sub>	0.297	0.172	3.345	5.231	6.495	2.463	2.238	1.713	13.789
x <sub>10</sub>	Y <sub>p</sub>	2.035	0.257	0.971	0.824	5.285	1.624	8.550	1.582	15.855
x <sub>11</sub>	Y <sub>q</sub>	4.983	0.678	0.421	2.165	5.373	11.098	5.606	6.163	8.605
x <sub>12</sub>	Y <sub>r</sub>	0.057	0.138	0.067	0.046	0.153	0.139	0.086	0.274	1.054
x <sub>13</sub>	Z <sub>u</sub>	2.310	0.868	3.360	1.247	6.277	6.032	5.713	5.954	11.994
x <sub>14</sub>	Z <sub>v</sub>	0.080	1.656	3.123	1.015	10.479	6.853	4.674	8.701	12.330
x <sub>15</sub>	Z <sub>w</sub>	0.355	0.334	1.158	0.321	3.744	2.390	3.531	4.360	6.895
x <sub>16</sub>	Z <sub>p</sub>	0.428	0.229	0.214	0.138	1.850	2.440	1.373	2.906	9.642
x <sub>17</sub>	Z <sub>q</sub>	0.270	0.102	0.322	0.197	0.139	0.005	0.231	0.375	0.706
x <sub>18</sub>	Z <sub>r</sub>	3.181	4.237	1.886	0.185	1.510	2.261	2.124	3.099	10.719
x <sub>19</sub>	L <sub>u</sub>	0.754	1.065	1.542	1.002	0.275	2.198	2.206	1.323	5.070
x <sub>20</sub>	L <sub>v</sub>	1.545	1.346	0.392	0.450	0.866	3.117	0.649	3.170	11.862
x <sub>21</sub>	L <sub>w</sub>	0.753	0.746	2.079	0.481	0.848	4.913	0.969	6.219	15.504
x <sub>22</sub>	L <sub>p</sub>	0.029	0.013	0.010	0.034	0.049	2.913	0.088	3.957	0.080
x <sub>23</sub>	L <sub>q</sub>	0.057	0.192	1.104	0.272	0.379	3.117	0.401	3.860	0.964
x <sub>24</sub>	L <sub>r</sub>	0.357	0.544	0.095	0.127	0.227	3.027	0.196	4.292	0.842
x <sub>25</sub>	M <sub>u</sub>	0.460	0.590	0.112	1.139	0.073	0.417	0.162	0.839	1.264
x <sub>26</sub>	M <sub>v</sub>	6.904	8.645	5.598	8.073	7.418	10.423	6.904	12.657	15.945
x <sub>27</sub>	M <sub>w</sub>	0.612	1.133	3.718	0.504	10.520	7.856	10.062	13.984	14.819
x <sub>28</sub>	M <sub>p</sub>	0.557	0.548	0.104	0.095	1.737	3.106	1.478	3.257	12.698
x <sub>29</sub>	M <sub>q</sub>	0.042	0.079	0.194	0.081	0.626	0.492	0.603	0.878	0.923
x <sub>30</sub>	M <sub>r</sub>	3.465	5.174	3.114	1.910	9.086	13.767	12.391	12.102	16.112
x <sub>31</sub>	N <sub>u</sub>	0.638	0.991	0.716	0.869	0.606	1.072	0.910	0.656	7.648
x <sub>32</sub>	N <sub>v</sub>	0.866	0.342	0.873	0.059	0.572	2.947	0.495	2.260	3.366
x <sub>33</sub>	N <sub>w</sub>	1.137	2.269	3.367	1.579	1.009	0.290	1.086	2.219	5.995
x <sub>34</sub>	N <sub>p</sub>	0.217	0.178	0.006	0.195	0.482	3.580	0.682	4.155	0.396
x <sub>35</sub>	N <sub>q</sub>	0.350	0.081	0.259	0.239	0.914	1.589	1.054	1.784	4.452
x <sub>36</sub>	N <sub>r</sub>	0.007	0.000	0.058	0.028	0.068	0.728	0.141	0.875	2.621

Table 6.12. Percentage Estimation Errors of Control Derivatives – Case 4

parameter	symbol	Case 4.10	Case 4.11	Case 4.12	Case 4.13	Case 4.14	Case 4.15	Case 4.16	Case 4.17	Case 4.18
		[-10% 10%]	[-20% 20%]	[-30% 30%]	[-40% 40%]	[-50% 50%]	[-60% 60%]	[-70% 70%]	[-80% 80%]	[-90% 90%]
x <sub>37</sub>	X <sub>δlat</sub>	8.574	0.436	0.388	4.307	0.530	0.057	0.443	0.208	5.852
x <sub>38</sub>	X <sub>δlon</sub>	0.332	0.334	0.351	5.929	0.304	0.123	0.280	0.288	0.171
x <sub>39</sub>	X <sub>δped</sub>	9.811	0.587	0.944	0.973	5.978	8.341	6.130	7.899	5.443
x <sub>40</sub>	X <sub>δcol</sub>	1.701	1.753	1.813	1.245	0.151	0.275	0.024	0.479	0.329
x <sub>41</sub>	Y <sub>δlat</sub>	6.992	0.230	0.249	0.269	10.046	12.483	10.007	13.453	24.588
x <sub>42</sub>	Y <sub>δlon</sub>	10.000	0.132	0.244	0.257	17.302	19.758	17.509	20.387	30.517
x <sub>43</sub>	Y <sub>δped</sub>	10.000	0.142	0.121	0.102	18.624	15.742	18.650	14.728	1.215
x <sub>44</sub>	Y <sub>δcol</sub>	10.000	0.400	0.147	0.870	27.795	28.473	27.122	30.131	39.063
x <sub>45</sub>	Z <sub>δlat</sub>	10.000	0.415	0.521	0.253	17.412	19.443	17.300	19.784	24.604
x <sub>46</sub>	Z <sub>δlon</sub>	5.608	0.089	0.166	0.062	5.764	5.915	5.719	5.701	5.721
x <sub>47</sub>	Z <sub>δped</sub>	9.820	1.045	1.070	0.075	15.183	9.474	14.153	8.271	11.556
x <sub>48</sub>	Z <sub>δcol</sub>	0.719	0.065	0.141	0.140	0.780	1.323	0.732	0.978	0.819
x <sub>49</sub>	L <sub>δlat</sub>	2.550	0.013	0.087	0.048	2.526	0.590	2.464	0.059	15.648
x <sub>50</sub>	L <sub>δlon</sub>	3.099	0.176	0.337	0.183	3.059	5.924	3.007	6.832	17.300
x <sub>51</sub>	L <sub>δped</sub>	3.692	0.135	0.065	0.050	3.838	0.703	3.833	0.160	3.174
x <sub>52</sub>	L <sub>δcol</sub>	2.200	0.151	0.091	0.252	2.193	3.521	2.100	5.955	2.958
x <sub>53</sub>	M <sub>δlat</sub>	4.118	0.245	0.384	0.199	5.573	6.094	5.189	6.228	3.212
x <sub>54</sub>	M <sub>δlon</sub>	1.407	0.233	0.385	0.106	1.445	1.558	1.504	1.785	1.998
x <sub>55</sub>	M <sub>δped</sub>	2.666	4.553	3.495	8.280	21.044	20.305	20.457	27.302	24.054
x <sub>56</sub>	M <sub>δcol</sub>	1.546	0.526	0.023	0.400	3.128	2.937	3.002	3.632	1.781
x <sub>57</sub>	N <sub>δlat</sub>	3.878	0.017	0.144	0.138	3.369	6.074	2.892	7.697	18.934
x <sub>58</sub>	N <sub>δlon</sub>	3.680	0.207	0.157	0.185	2.980	5.851	2.840	6.213	15.510
x <sub>59</sub>	N <sub>δped</sub>	0.800	0.083	0.048	0.006	0.917	0.296	1.032	0.346	5.417
x <sub>60</sub>	N <sub>δcol</sub>	1.930	0.004	0.104	0.017	1.952	4.026	1.791	4.184	13.350

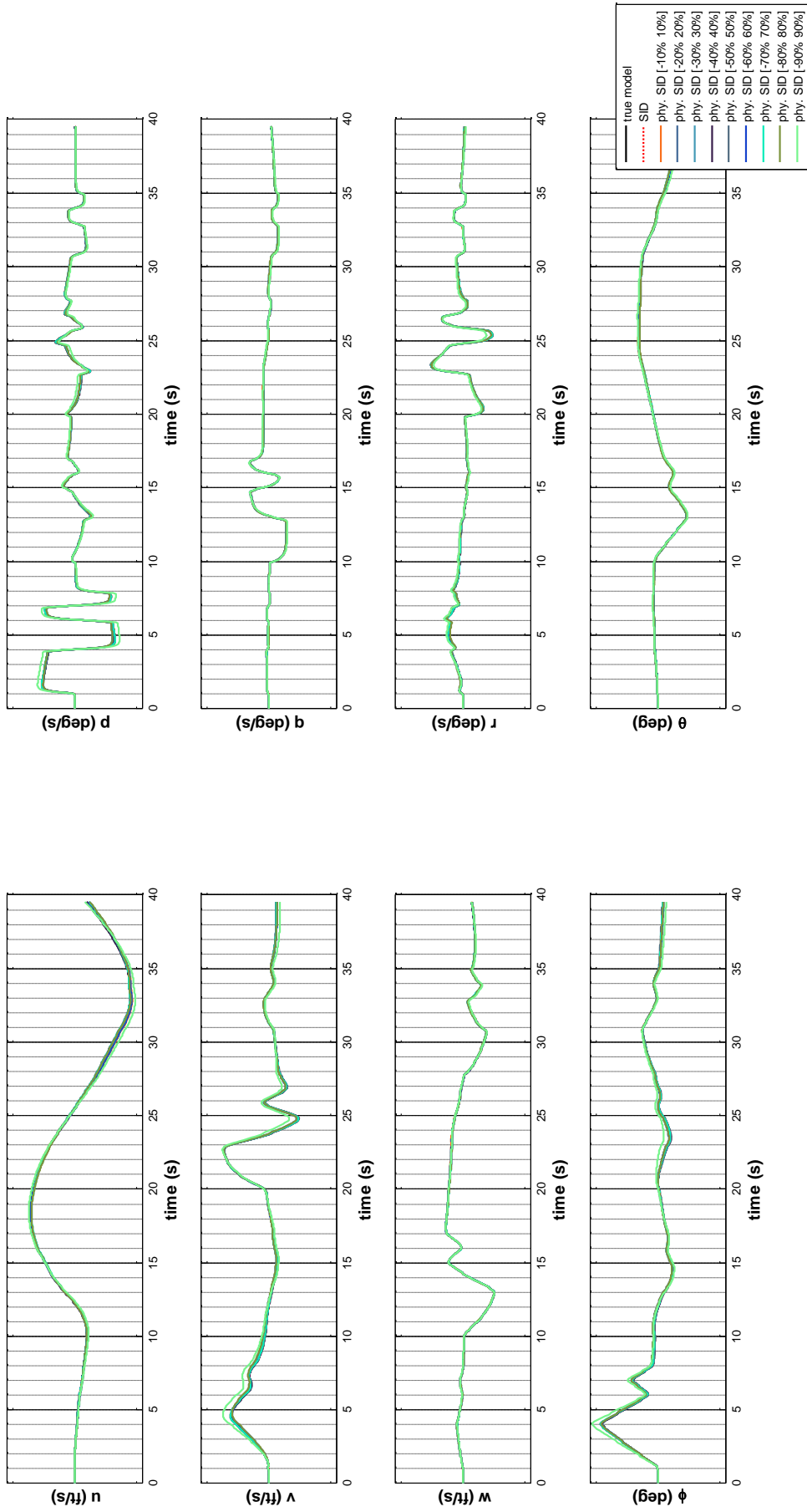


Figure 6.44. Comparison of Outputs for 3-2-1-1 Excitation Signals (Identification Using Linear Model Simulation)



The doublet input is applied in four channels sequentially for verification. The inputs and the outputs of this test case are given in Figure 6.11 and Figure 6.45 respectively. TIC values are calculated according to Eq.(117). Results are listed in Table 6.13.

Table 6.13. TIC Values for Case 4 (for Verification Signal)<sup>10</sup>

optimization batch for constraint set		TIC value (Case 4.1 to Case 4.9)	TIC value (Case 4.10 to Case 4.18)
Case 4.1 vs. Case 4.10	[-10% 10%]	0.0021	0.0103
Case 4.2 vs. Case 4.11	[-20% 20%]	0.0022	0.0036
Case 4.3 vs. Case 4.12	[-30% 30%]	0.0023	0.0038
Case 4.4 vs. Case 4.13	[-40% 40%]	0.0022	0.0066
Case 4.5 vs. Case 4.14	[-50% 50%]	0.0029	0.0041
Case 4.6 vs. Case 4.15	[-60% 60%]	0.0109	0.0192
Case 4.7 vs. Case 4.16	[-70% 70%]	0.0024	0.0040
Case 4.8 vs. Case 4.17	[-80% 80%]	0.0181	0.0241
Case 4.9 vs. Case 4.18	[-90% 90%]	0.0179	0.0883

---

<sup>10</sup> Remember that the scenarios Case 2.1 to Case 2.9 are the same with Case 4.1 to Case 4.9.

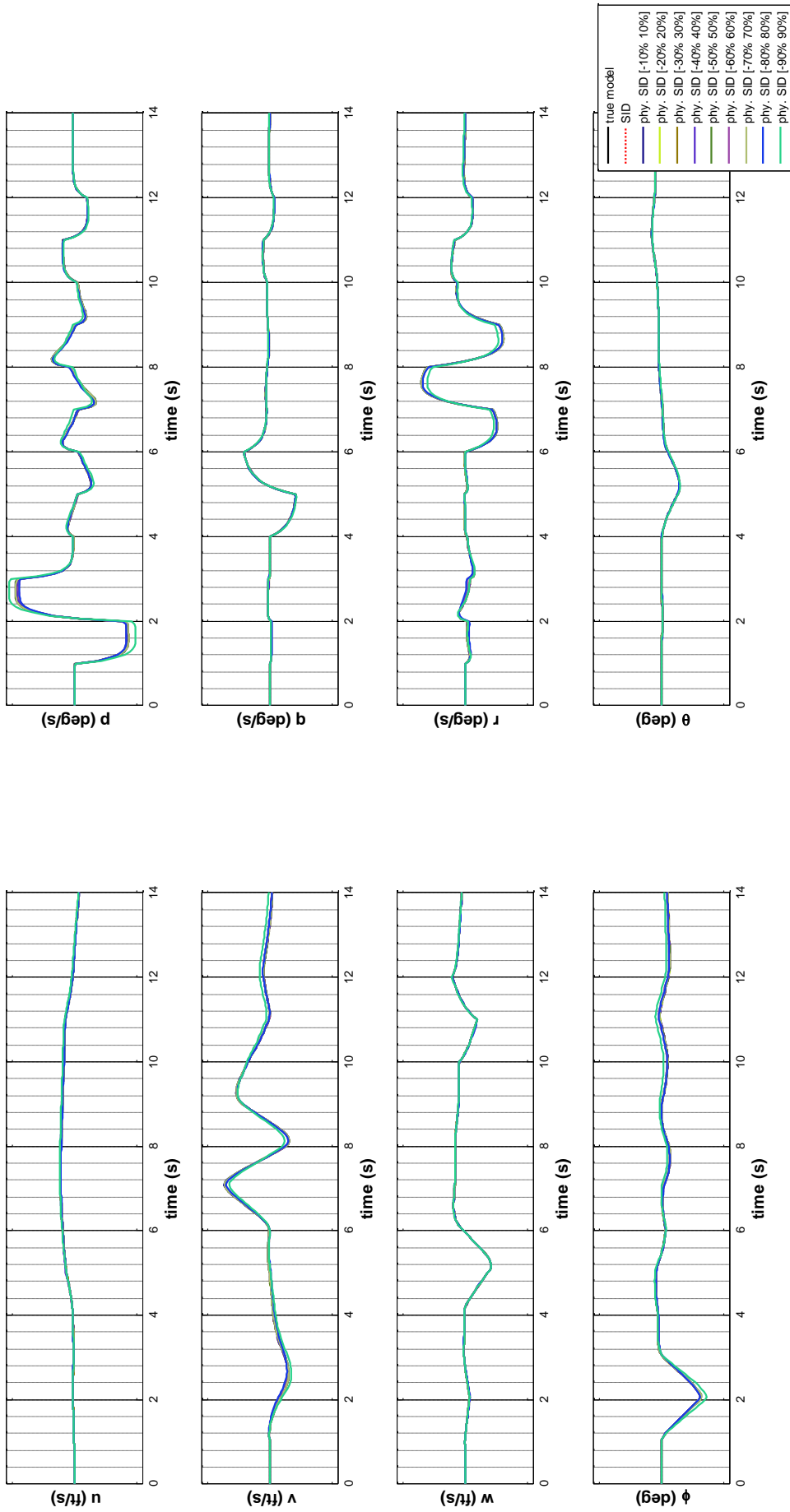


Figure 6.45. Comparison of Outputs for Doublet Excitation Signals (Identification Using Linear Model Simulation)

## 6.5. Case 5- Constraint with additional physical properties

Until now we assumed that we have priori information about all of the parameters within an uncertainty range (from [-10% 10%] to [-90% 90%]). However, we may not have information about some of the parameters and even about their error boundaries in some real-life applications. In this case study, we forced well known parameters to their “known” values and relaxed the constraints (increased the boundaries) for the other ones. First it is assumed that all of the parameters are known with some uncertainty (from [-10% 10%] to [-30% 30%]). Then we constrained certain parameters to their priori values as it is explained below.

- The insignificant parameters are set to zero as a common practice. As it is mentioned in Chapter 3, the stability parameters  $X_v, X_w, Y_u, Y_w, Z_u, Z_v, M_w, M_r, N_w, Z_q, Y_r$  and the control parameters  $X_{\delta_{ped}}, Y_{\delta_{ped}}, M_{\delta_{ped}}, Y_{\delta_{ped}}$  are assumed as “insignificant” and they are set to zero.
- Assuming that we have priory information on the helicopter stability characteristics, the sign information of the stability derivatives (as presented in Chapter 3, Table 3.1) are assigned as constraints.
- For some parameters such as  $Z_w, Z_{\delta_{col}}$  or  $Z_{\delta_{lon}}$  it is possible to assign values by only considering the physical characteristics of the helicopter. We set  $Z_w$  to the value obtained from Eq.(100),  $Z_{\delta_{col}}$  to the value obtained from Eq.(102) and  $Z_{\delta_{lon}}$  to the value obtained from Eq. (103).

The conditions analyzed in this case study are summarized in Table 6.14. According to Table 6.14, 120 (20x3x2) optimization runs are analyzed in total. In addition to the 60 optimization runs which are generated in Case 2, additional 60 runs are created with additional constraints (Case 5.4-Case 5.6). Then the results are compared with each other. For example, the results of Case 5.1 (or Case2.1, they are the same) are compared with results of Case 5.4. Comparisons are continued until to the last pair Case 5.3 and Case 5.6. The results are compared in terms of minimization output

curves. The findings are illustrated in Figure 6.46 to Figure 6.51. The percentage estimation error results are given in Table 6.15 and Table 6.16.

Table 6.14. Summary of Analysis Conditions for Case 5 (Constraint with additional physical properties)

	alg.	constraint (error bound) for the parameters of			initial condition for the parameters of			number of run
		$A_{phy}$	$B_{phy}$	$T$	$A_{phy}$	$B_{phy}$	$T$	
Case 5.1	SQP	[-10% 10%]		NA	random sel.	const	20	
Case 5.2	SQP	[-20% 20%]		NA	random sel.	const	20	
Case 5.3	SQP	[-30% 30%]		NA	random sel.	const	20	
Case 5.4	SQP	[-10% 10%] + additional physical constraints		NA	random sel.	const	20	
Case 5.5	SQP	[-20% 20%] + additional physical constraints		NA	random sel.	const	20	
Case 5.6	SQP	[-30% 30%] + additional physical constraints		NA	random sel.	const	20	

By examining Figure 6.46 - Figure 6.51, Table 6.15 and Table 6.16 we can see that most of the insignificant parameters such as  $X_r, Y_q, Z_p$  do not converge to their true values. However most of the significant parameters that are related with the dynamics being excited such as  $X_q, Y_r, L_p, M_q, N_r, L_{lat}, M_{lon}, N_{ped}$  converge. It is good that the methodology still works for the significant parameters in such scenarios.

The physical system matrices are constructed using the above presented optimization results (Case 5.4 - Case 5.6). Then the true model and the estimated one are simulated with the same 3-2-1-1 excitation signals (Figure 5.4). The results show that simulation responses of “true model”, “SID model” and “Physical SID model” to 3-2-1-1

excitation signals are satisfactory for Case 5.4 to Case 5.6. The simulation results are illustrated in Figure 6.52.

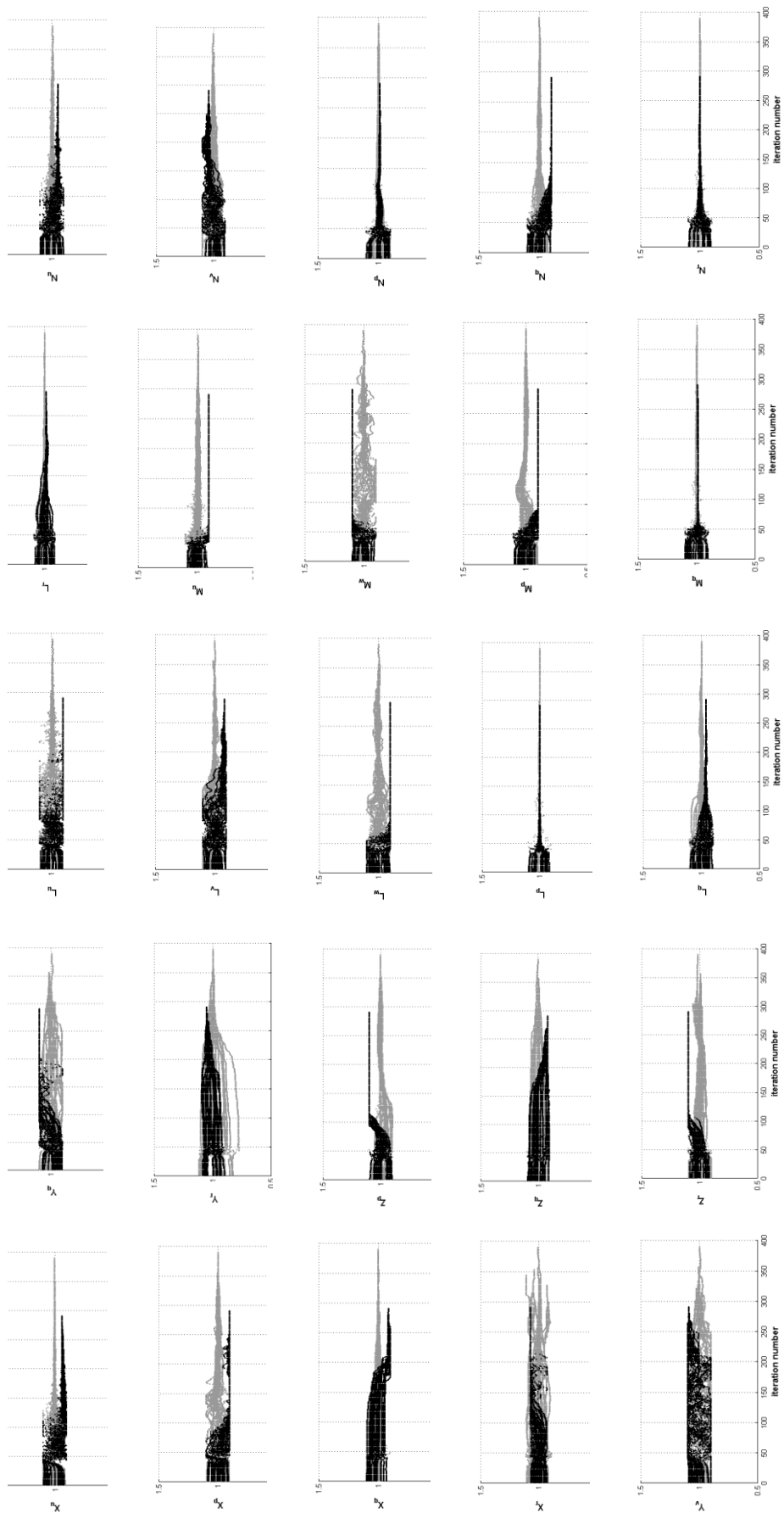


Figure 6.46. Convergence of Stability Derivatives (Case 5.1 vs. Case 5.4)

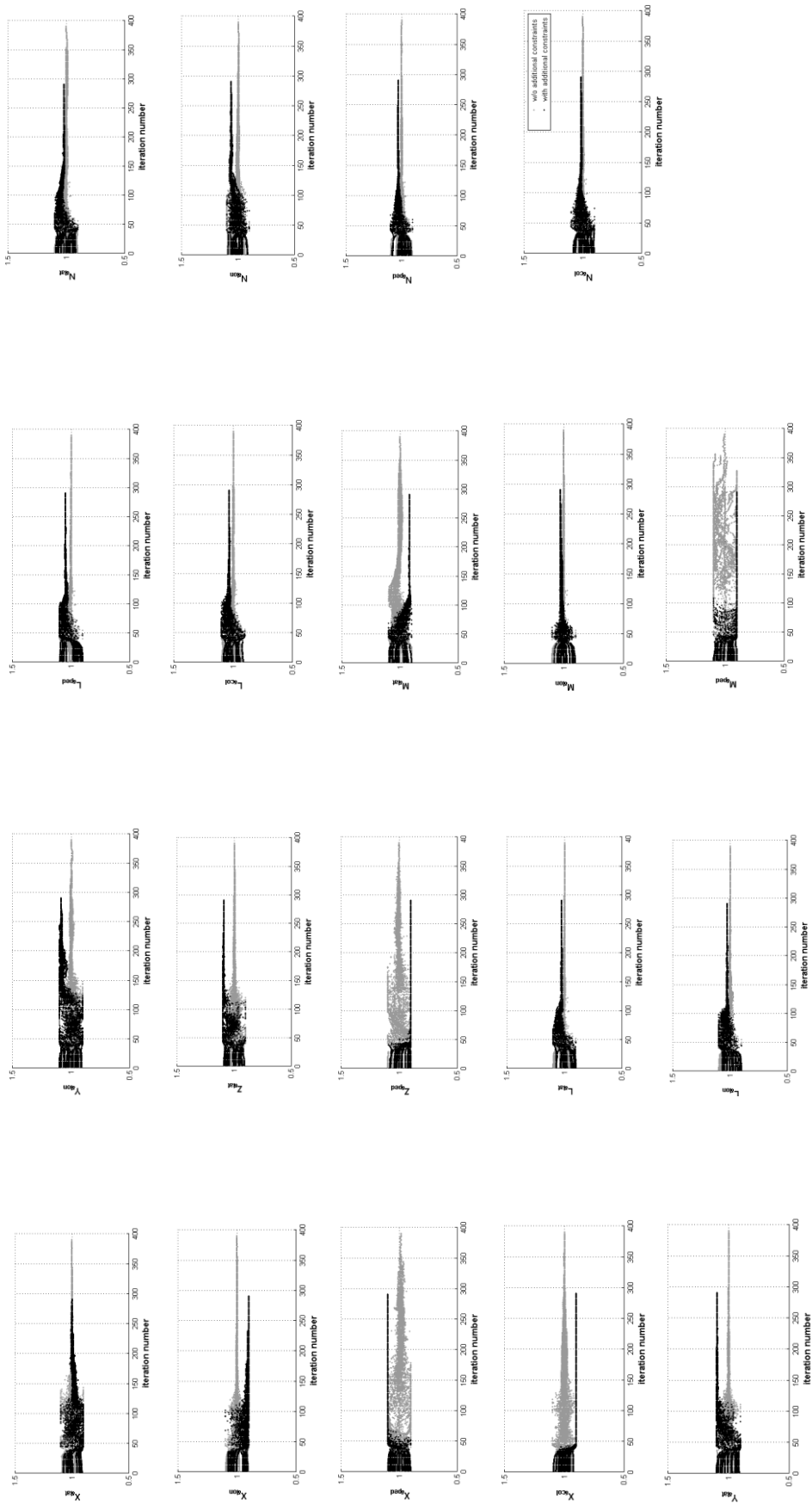


Figure 6.47. Convergence of Control Derivatives (Case 5.1 vs. Case 5.4)

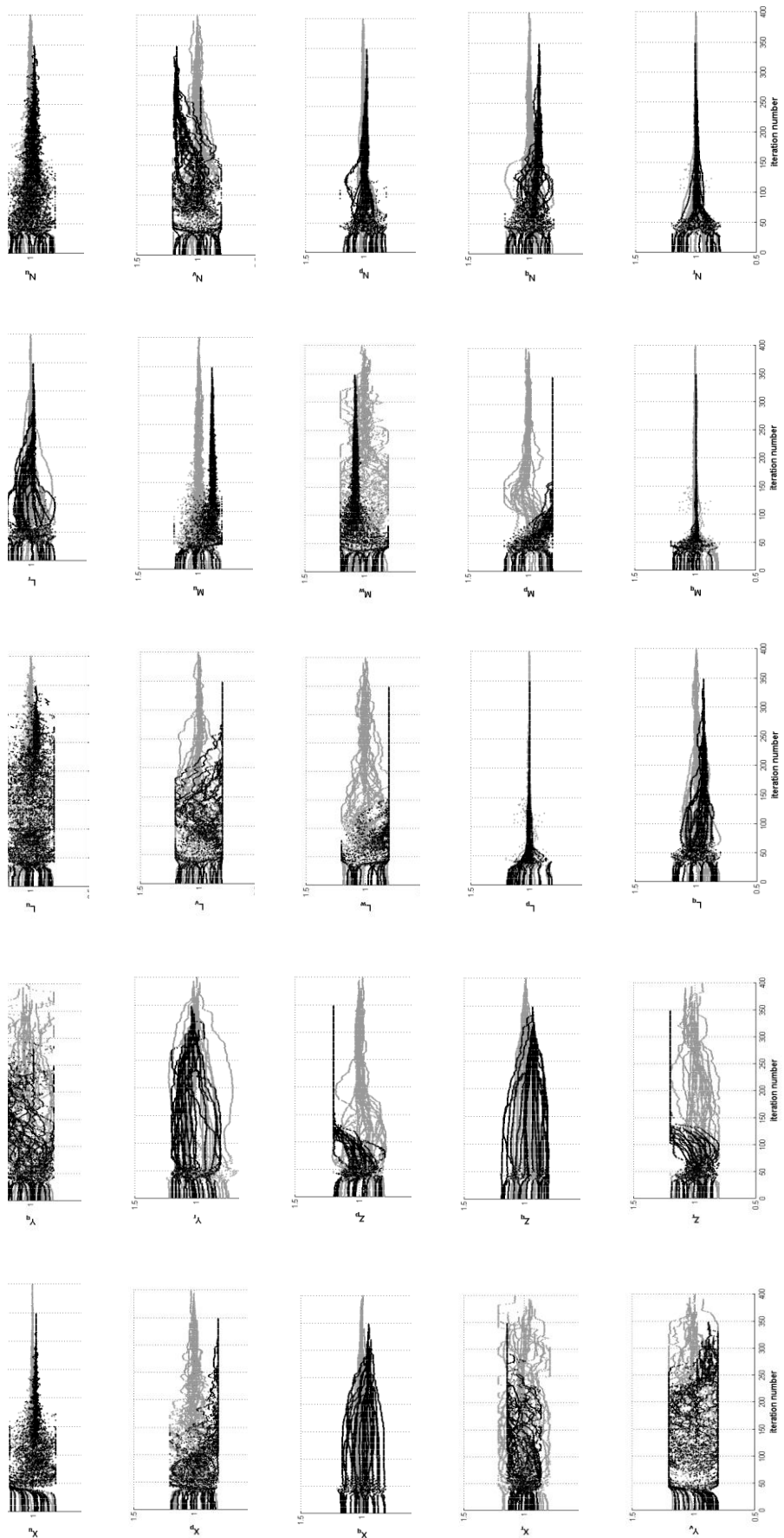


Figure 6.48. Convergence of Stability Derivatives (Case 5.2 vs. Case 5.5)



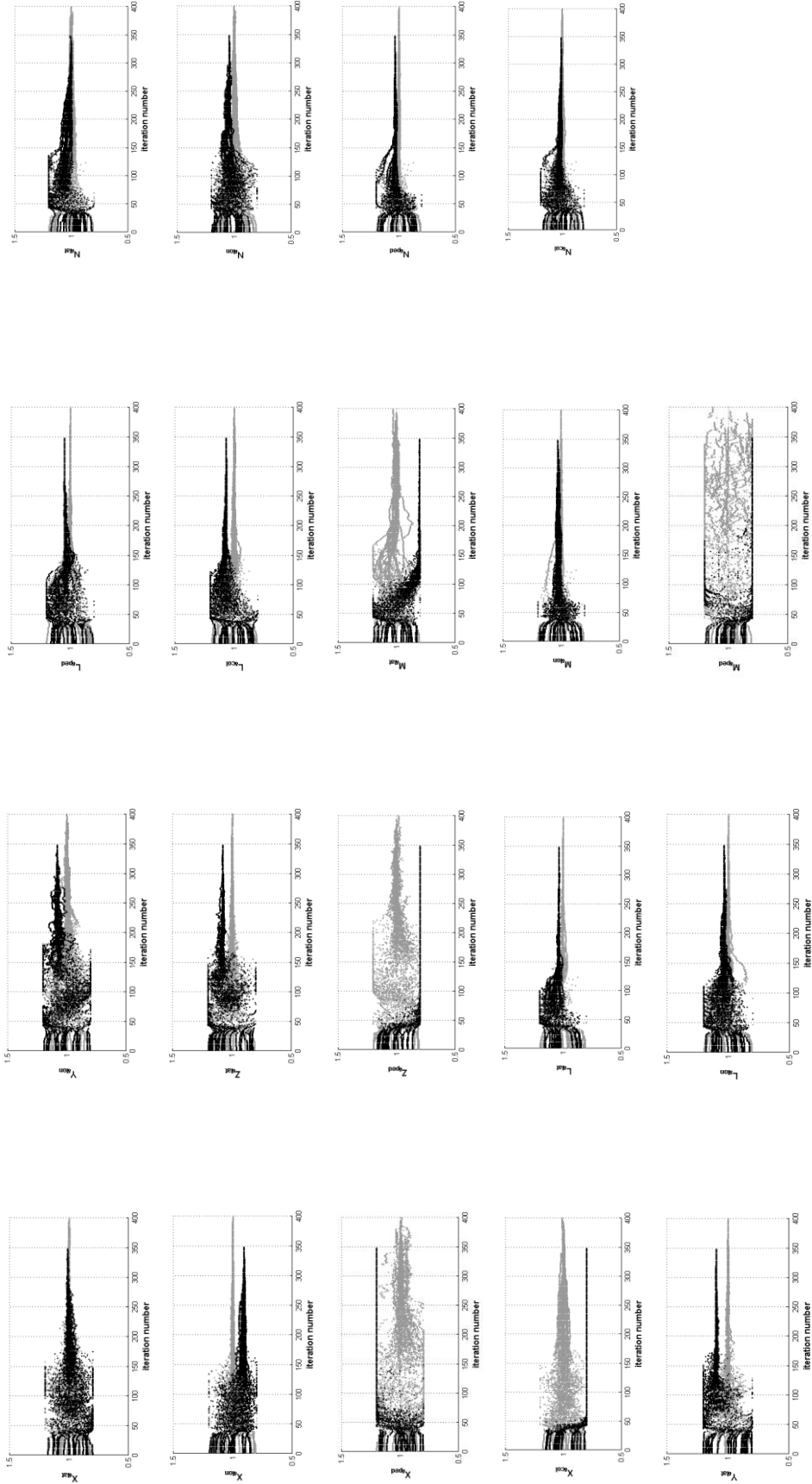


Figure 6.49. Convergence of Control Derivatives (Case 5.2 vs. Case 5.5)

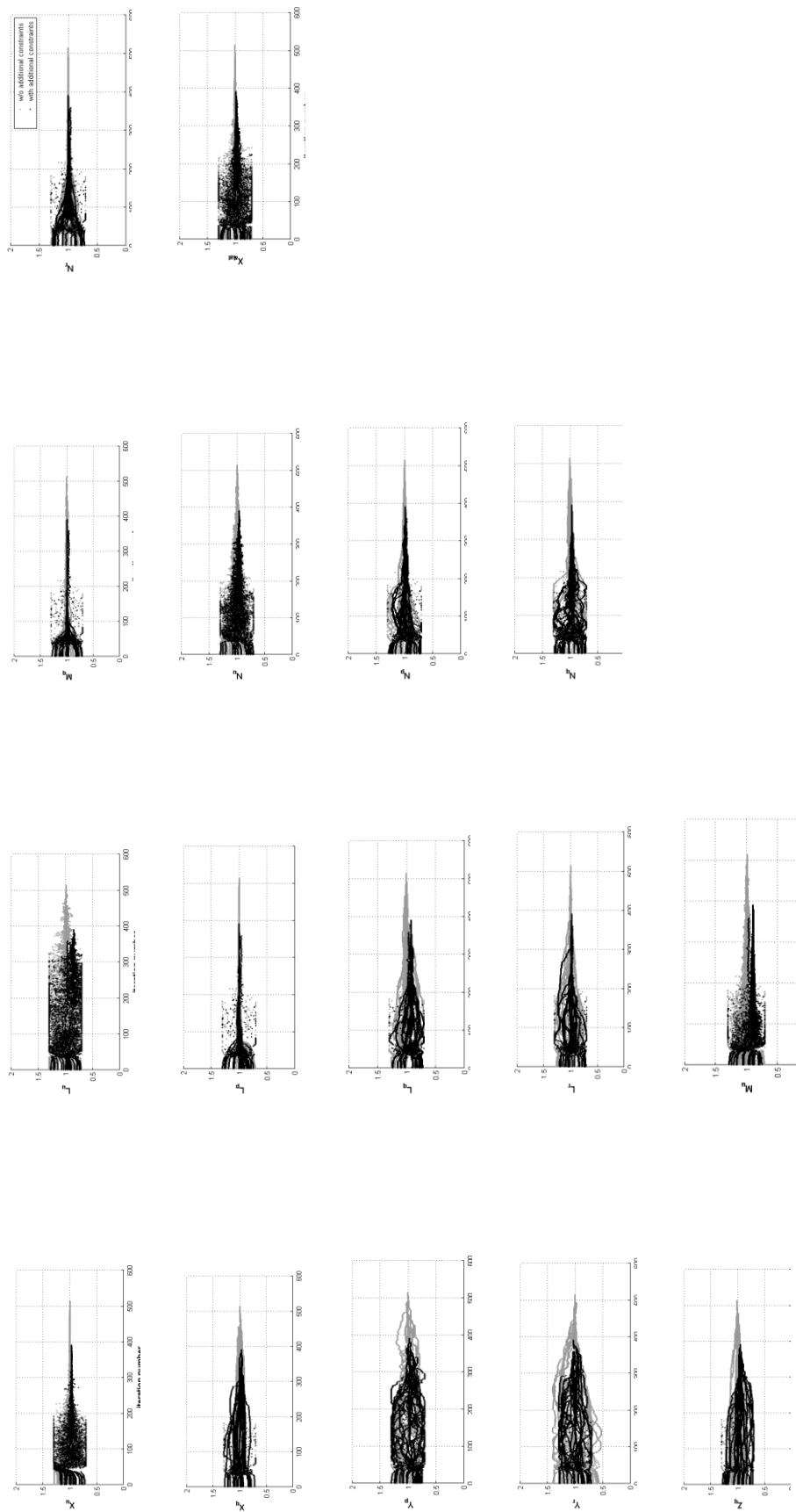


Figure 6.50. Convergence of Stability Derivatives (Case 5.3 vs. Case 5.6)

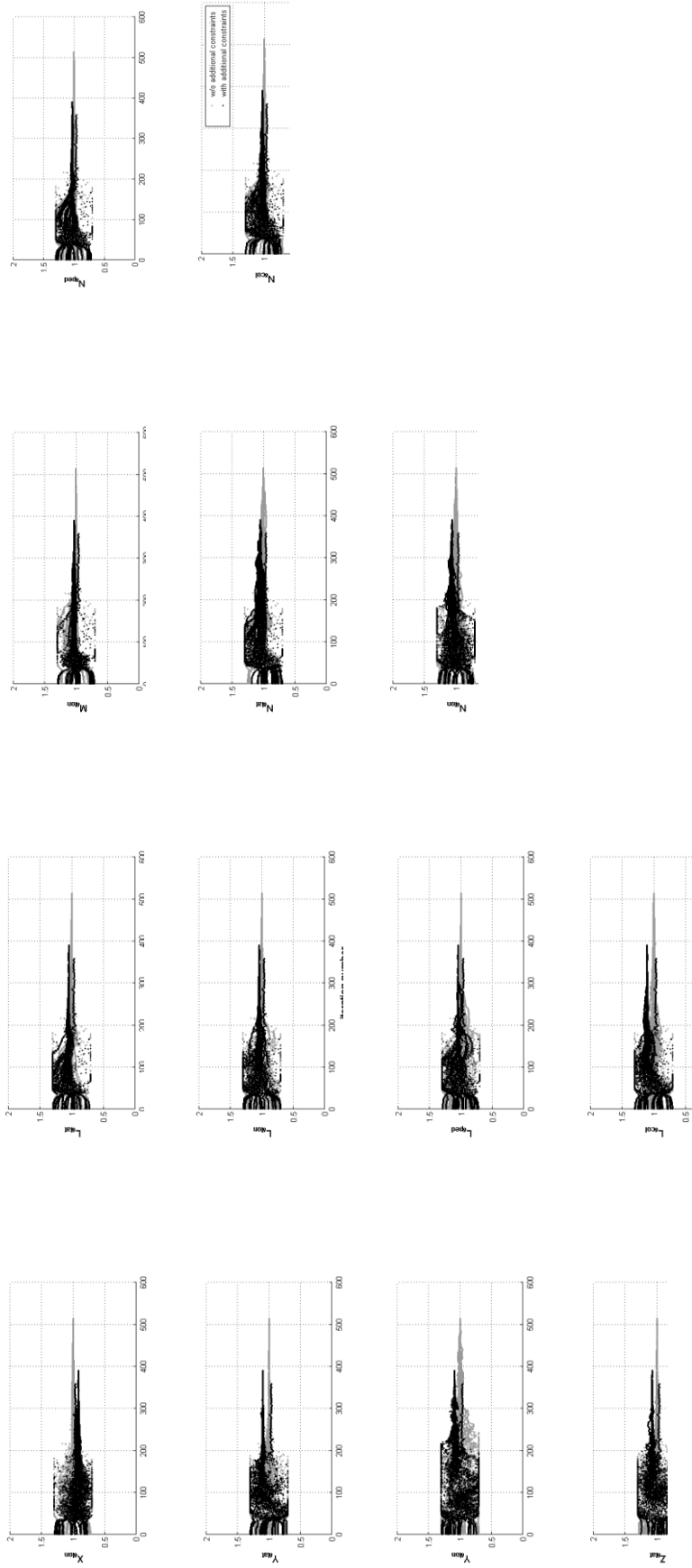


Figure 6.51. Convergence of Control Derivatives (Case 5.3 vs. Case 5.6)

Table 6.15. *Percentage Estimation Errors of Stability Derivatives – Case 5<sup>11</sup>*

parameter	symbol	estimation error %					
		Case 5.1	Case 5.2	Case 5.3	Case 5.4	Case 5.5	Case 5.6
		[-10% 10%]	[-20% 20%]	[-30% 30%]	[-10% 10%] + add. const.	[-20% 20%] + add. const.	[-30% 30%] + add. const.
x <sub>1</sub>	X <sub>u</sub>	0.075	0.170	0.135	7.045	2.928	3.684
x <sub>2</sub>	X <sub>v</sub>	3.364	4.233	2.799	NA	NA	NA
x <sub>3</sub>	X <sub>w</sub>	0.007	0.406	0.474	NA	NA	NA
x <sub>4</sub>	X <sub>p</sub>	0.148	0.258	0.642	9.032	20.000	30.000
x <sub>5</sub>	X <sub>q</sub>	0.019	0.043	1.478	9.449	5.257	3.522
x <sub>6</sub>	X <sub>r</sub>	1.158	1.775	0.534	10.000	20.000	28.560
x <sub>7</sub>	Y <sub>u</sub>	0.058	2.107	2.937	NA	NA	NA
x <sub>8</sub>	Y <sub>v</sub>	0.054	0.167	0.183	5.938	13.994	25.241
x <sub>9</sub>	Y <sub>w</sub>	0.156	0.927	1.077	NA	NA	NA
x <sub>10</sub>	Y <sub>p</sub>	1.624	0.263	1.677	NA	NA	NA
x <sub>11</sub>	Y <sub>q</sub>	0.716	1.202	7.008	9.953	20.000	25.111
x <sub>12</sub>	Y <sub>r</sub>	0.031	0.015	0.020	3.362	3.207	1.984
x <sub>13</sub>	Z <sub>u</sub>	0.528	0.826	0.625	NA	NA	NA
x <sub>14</sub>	Z <sub>v</sub>	1.004	5.680	0.421	NA	NA	NA
x <sub>15</sub>	Z <sub>w</sub>	0.082	0.111	0.433	NA	NA	NA
x <sub>16</sub>	Z <sub>p</sub>	0.370	0.365	0.168	0.491	0.981	1.472
x <sub>17</sub>	Z <sub>q</sub>	0.014	0.172	0.059	0.995	0.457	1.009
x <sub>18</sub>	Z <sub>r</sub>	1.008	1.838	0.053	10.000	20.000	30.000
x <sub>19</sub>	L <sub>u</sub>	1.074	0.955	1.112	9.991	5.005	13.089
x <sub>20</sub>	L <sub>v</sub>	0.230	0.191	0.076	7.712	20.000	30.000
x <sub>21</sub>	L <sub>w</sub>	0.027	0.042	0.212	10.000	20.000	30.000
x <sub>22</sub>	L <sub>p</sub>	0.041	0.024	0.008	0.278	0.345	0.018
x <sub>23</sub>	L <sub>q</sub>	0.084	0.208	0.162	3.936	5.862	6.998
x <sub>24</sub>	L <sub>r</sub>	0.064	0.070	0.048	1.736	2.334	1.449
x <sub>25</sub>	M <sub>u</sub>	0.875	0.648	0.931	10.000	12.234	10.826
x <sub>26</sub>	M <sub>v</sub>	6.859	5.478	6.949	NA	NA	NA
x <sub>27</sub>	M <sub>w</sub>	0.196	0.041	1.309	10.000	7.266	3.189
x <sub>28</sub>	M <sub>p</sub>	0.345	0.105	0.009	10.000	20.000	30.000
x <sub>29</sub>	M <sub>q</sub>	0.007	0.011	0.089	0.928	0.662	0.360
x <sub>30</sub>	M <sub>r</sub>	3.514	2.413	1.330	NA	NA	NA
x <sub>31</sub>	N <sub>u</sub>	0.800	0.655	0.629	5.163	3.701	3.747
x <sub>32</sub>	N <sub>v</sub>	0.319	0.226	0.035	4.671	17.154	24.020
x <sub>33</sub>	N <sub>w</sub>	0.114	0.270	0.371	NA	NA	NA
x <sub>34</sub>	N <sub>p</sub>	0.185	0.057	0.095	1.282	3.207	2.121
x <sub>35</sub>	N <sub>q</sub>	0.196	0.139	0.090	10.000	7.967	3.123
x <sub>36</sub>	N <sub>r</sub>	0.030	0.012	0.031	0.189	0.640	0.028

<sup>11</sup> “add. const.” denotes “additional constraints”

NA : Not Applicable

Table 6.16. *Percentage Estimation Errors of Control Derivatives – Case 5*<sup>12</sup>

parameter	symbol	estimation error %					
		Case 5.1	Case 5.2	Case 5.3	Case 5.4	Case 5.5	Case 5.6
		[-10% 10%]	[-20% 20%]	[-30% 30%]	[-10% 10%] + add. const.	[-20% 20%] + add. const.	[-30% 30%] + add. const.
x <sub>37</sub>	X <sub>δlat</sub>	0.115	0.076	0.241	0.015	0.974	1.581
x <sub>38</sub>	X <sub>δlon</sub>	0.016	0.014	0.079	9.995	9.717	8.732
x <sub>39</sub>	X <sub>δped</sub>	1.637	1.339	1.209	10.000	20.000	30.000
x <sub>40</sub>	X <sub>δcol</sub>	0.038	0.037	0.469	10.000	20.000	30.000
x <sub>41</sub>	Y <sub>δlat</sub>	0.282	0.269	0.279	9.414	9.523	9.592
x <sub>42</sub>	Y <sub>δlon</sub>	0.148	0.260	0.027	8.257	7.638	10.266
x <sub>43</sub>	Y <sub>δped</sub>	0.078	0.086	0.090	NA	NA	NA
x <sub>44</sub>	Y <sub>δcol</sub>	0.599	0.324	0.892	NA	NA	NA
x <sub>45</sub>	Z <sub>δlat</sub>	0.347	0.443	0.385	9.161	7.799	7.460
x <sub>46</sub>	Z <sub>δlon</sub>	0.015	0.120	0.055	NA	NA	NA
x <sub>47</sub>	Z <sub>δped</sub>	0.589	0.176	0.138	10.000	20.000	30.000
x <sub>48</sub>	Z <sub>δcol</sub>	0.012	0.093	0.028	NA	NA	NA
x <sub>49</sub>	L <sub>δlat</sub>	0.077	0.081	0.072	2.554	3.441	4.574
x <sub>50</sub>	L <sub>δlon</sub>	0.250	0.262	0.246	2.503	3.180	3.542
x <sub>51</sub>	L <sub>δped</sub>	0.069	0.063	0.062	4.825	4.751	4.259
x <sub>52</sub>	L <sub>δcol</sub>	0.191	0.180	0.171	3.334	6.593	10.435
x <sub>53</sub>	M <sub>δlat</sub>	0.093	0.471	0.363	7.977	19.450	30.000
x <sub>54</sub>	M <sub>δlon</sub>	0.193	0.234	0.164	2.631	2.386	2.971
x <sub>55</sub>	M <sub>δped</sub>	1.763	2.787	3.004	10.000	20.000	30.000
x <sub>56</sub>	M <sub>δcol</sub>	0.634	0.422	0.672	NA	NA	NA
x <sub>57</sub>	N <sub>δlat</sub>	0.198	0.006	0.300	1.978	0.152	4.374
x <sub>58</sub>	N <sub>δlon</sub>	0.272	0.214	0.091	5.999	4.137	6.528
x <sub>59</sub>	N <sub>δped</sub>	0.058	0.023	0.015	3.132	3.434	2.811
x <sub>60</sub>	N <sub>δcol</sub>	0.021	0.086	0.177	1.349	1.022	2.771

<sup>12</sup> “add. const.” denotes “additional constraints”

NA : Not Applicable

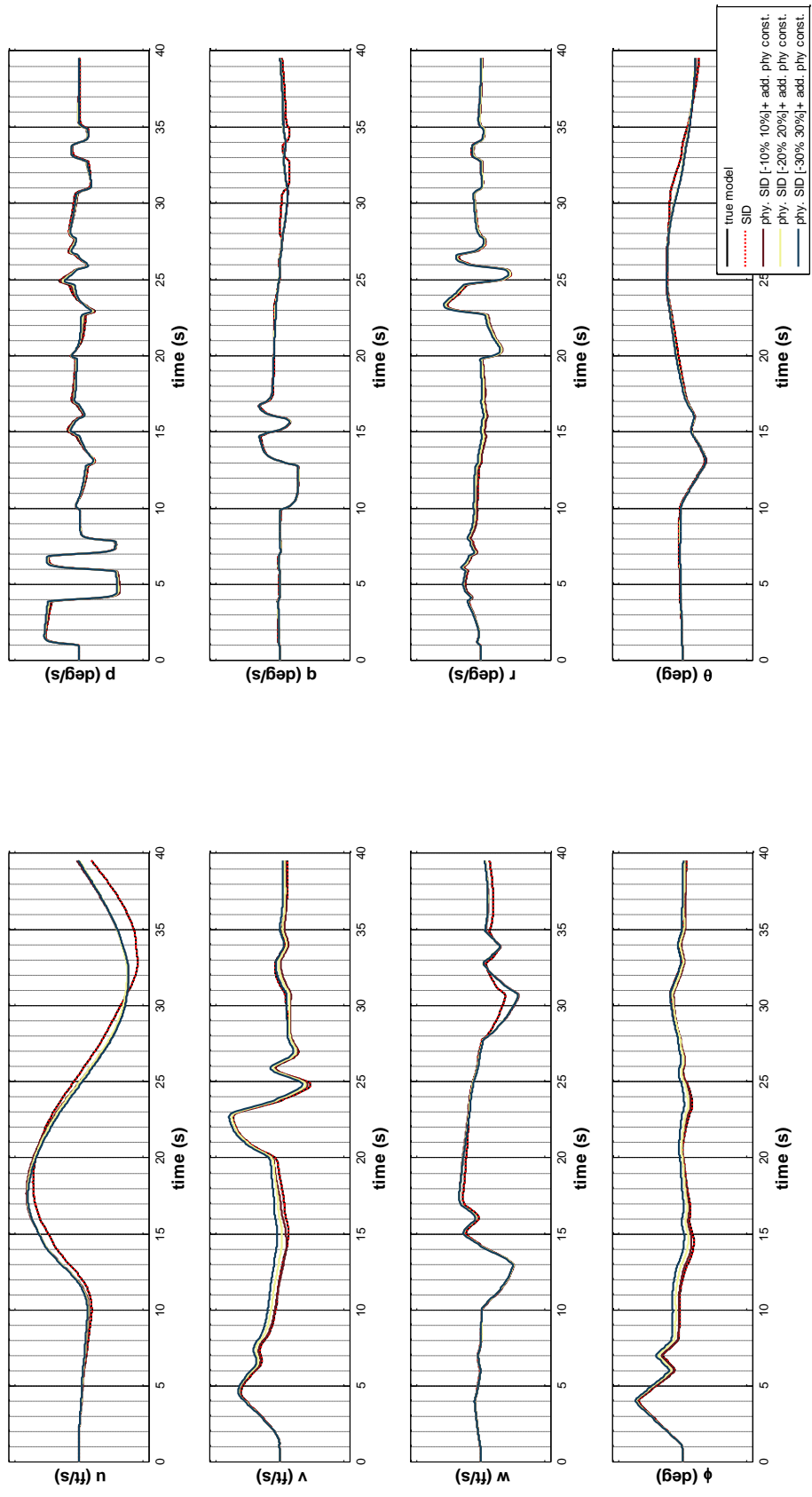


Figure 6.52. Comparison of Outputs for 3-2-1-1 Excitation Signals (Identification Using Linear Model Simulation)

The doublet input is applied in four channels sequentially for verification. The inputs and the outputs of this test case are given in Figure 6.11 and Figure 6.53 respectively. TIC values are calculated according to Eq.(117). Results are listed in Table 6.17.

Table 6.17. *TIC Values for Case 5 (for Verification Signal)*<sup>13</sup>

optimization batch for constraint set		TIC value (Case 2.1 to Case.2.3)	TIC value (Case 5.1 to Case 5.3)
Case 2.1 vs. Case 5.1	[-10% 10%]	0.0021	0.0103
Case 2.2 vs. Case 5.2	[-20% 20%]	0.0022	0.0036
Case 2.3 vs. Case 5.3	[-30% 30%]	0.0023	0.0038

---

<sup>13</sup>Remember that the scenarios Case 2.1 to Case 2.3 are the same with Case 5.1 to Case 5.3.

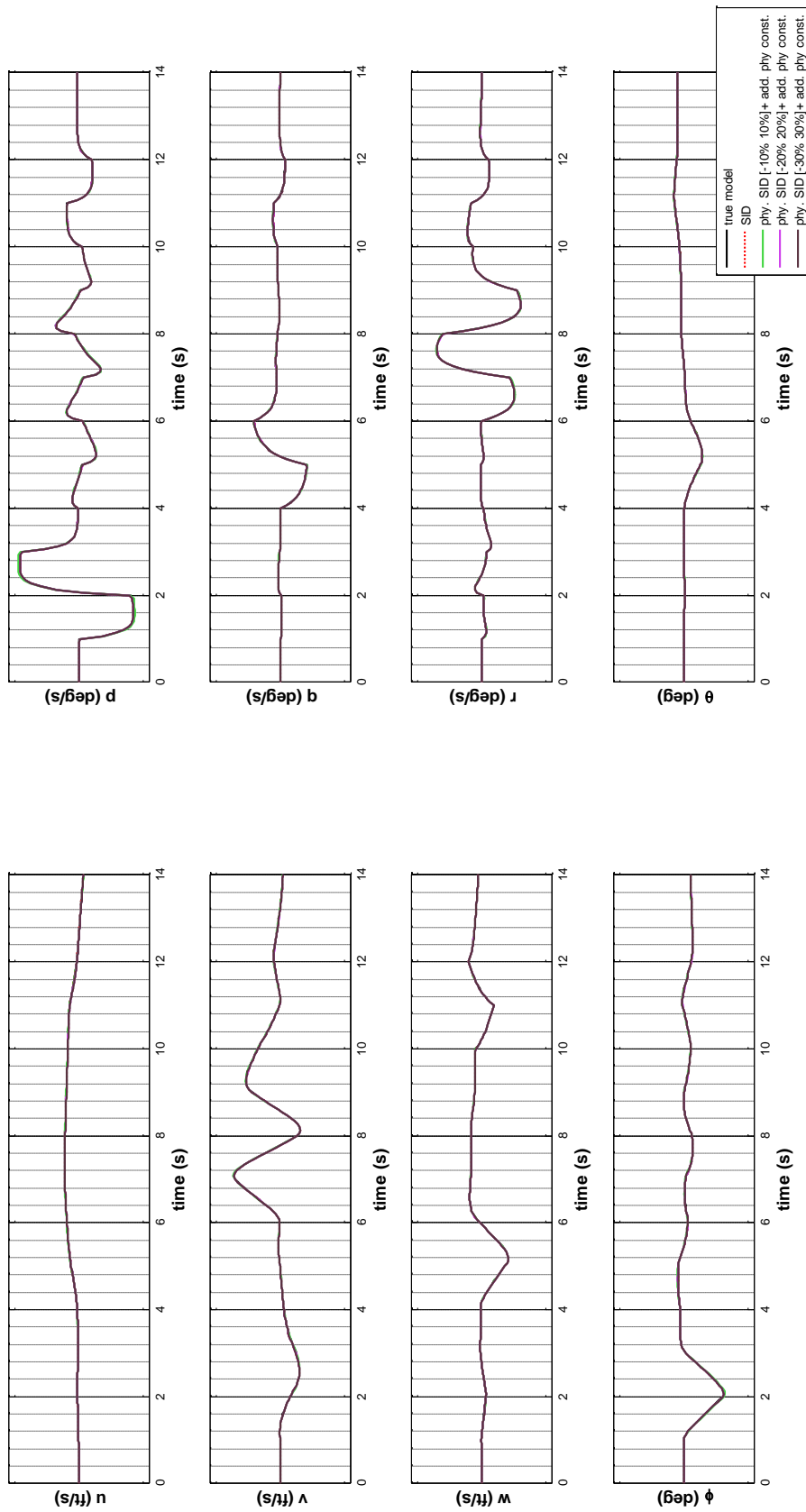


Figure 6.53. Comparison of Outputs for Doublet Excitation Signals (Identification Using Linear Model Simulation)



## 6.6. Discussion on Results

### 6.6.1. Comparison of Algorithm Type: IP or SQP

We compare these two algorithms: IP and SQP. If we only consider the results of Case 1 and Case 2 we see that the interior-point” algorithm is superior to SQP in many aspects. The quality of convergence seems better for the Interior-Point” algorithm compared to SQP. All of the derivatives converge easily in Case 1 where the IP algorithm is utilized whereas there are some deficiencies in Case 2 where the SQP algorithm is used. Convergence behavior of the IP algorithm is quite obvious in the presented. Moreover, the convergence rate is another advantage of the IP algorithm. As we mentioned above, with considering “estimation error” values and the TIC values, again the IP results are superior to those of SQP. In summary, by inspecting the results of Case 1 and Case 2 we can conclude that IP is better than SQP. However, results of Case 3 will modify our opinion.

Case 3 show that IP results are quite sensitive to the midpoint of the constraints. This was an expected phenomenon since IP algorithm sets the initial iteration point to the midpoint of the finite bounds. In the light of this, the optimizations are repeated for the different asymmetric constraint levels (i.e. [-10% 20%], [-10% 30%]) where the midpoint is different than the true value. In Case3, it is observed that some certain derivatives do not converge to the same value for different constraint levels. Moreover, for those parameters, the asymmetric constraint conditions ([-10% 20%], [-10% 30%]) do not converge to the true value. The true values can only be achieved with symmetric constraint levels ([-10% 10%]). However, for SQP algorithm, the results do not change too much with asymmetric constraint levels (Figure 6.23 and Figure 6.24). It is observed that parameters which have low level of significance do not converge for SQP algorithm whereas they converge to different values for different unrepeatable constraint levels with IP algorithm. This finding is summarized in Figure 6.25. Since symmetric constraint settings are not always feasible due to the real-life uncertainties, the IP algorithm results can be misleading for our problem. SQP algorithm seems to

be more trustworthy since the results are not affected too much from asymmetric constraints.

### **6.6.2. Comparison of Constraint Levels ([-10% 10%] ... [-90% 90%]))**

Numerical results indicate that there is not a distinct change in the convergence of the results due to expansion of the constraint limits. It is observed that some deterioration occurs in convergence with increasing constraint levels. This happens especially for the derivatives that are not much significant. These deteriorations do not affect the results too much. It is important to point out that as long as the sign information of the parameters is correct or consistent with helicopter flight dynamics the level of constraint levels do not blow up the optimization. However, if we go beyond 100% percent error in the constraint levels we may end up with false results.

### **6.6.3. Discussion on the Results Regarding the Initial Values of T Matrix**

In Case 4, we examined the performance of our solution method for arbitrary initialization of T matrix. The analysis showed that it is still possible find good results. This is an important outcome that it is not necessary for matrix  $C$  to be equal to identity for the methodology to work well.

### **6.6.4. Discussion on the Case with Additional Physical Constraints**

Analysis showed that when we impose additional physical constraints on the problem by using priori information on some well-known parameters, the degree of freedom of the optimization process decreases. The optimization algorithm forces the rest of the parameters to compensate for the absence of these well-known parameters. This causes the insignificant parameters to converge to constraint limits instead of true values. However, it was showed that the methodology still works for most of the significant parameters in such scenarios.

### 6.6.5. Discussion on Parameter Accuracy

One important metric for parameter accuracy is “sensitivity”. The inverse of the sensitivity, named as insensitivity, is a metric for how much a parameter can be changed from the estimated value without causing error in minimization of the objective function more than a given amount [70]. The formulation of the insensitivity is presented in Eq. (118) where  $\mathcal{H}$  is hessian of the objective function given in Eq. (105).

$$I_{x_i} = (\mathcal{H}_{x_i x_i})^{-1/2} \quad (118)$$

The parameter insensitivities are generally presented as normalized percentages of the converged parameter values [17]: The formulation is given in Eq. (119).

$$\bar{I}_{x_i} = \left| \frac{I_{x_i}}{x_i} \right| \times 100\% \quad (119)$$

As a rule of thumb, the parameters which have  $\bar{I}_{x_i} < 10\%$  is assumed as reliable. For our problem, the (normalized) insensitivity values are calculated for each parameter by using the equations (118) and (119). According to the results, the derivatives  $X_q, Y_p, Y_r, Z_p, Z_q, Z_r, L_p, L_q, L_r, M_p, M_q, N_p, N_q, N_r$ . satisfy the reliability criteria defined above.

This finding is complying with the convergence results of the parameters defined in this chapter. We observe that the significant parameters which are mentioned in Chapter 3 converge well in the case studies defined in this chapter. Most of these parameters also satisfy the insensitivity criteria.

## 6.7. Implementation for Nonlinear Simulation

Up to now we have presented many test cases dealing with alternative optimization algorithms, different constraints and initial conditions. These are presented for proving the concept behind the methodology. For this purpose, we have only utilized the outputs of the linear model until now. From here on we will present the implementation of the methodology to nonlinear simulation data and real flight test data. SQP optimization algorithm and only one set of constraint level [-30% 30%]) is used for these.

We can now present the case of nonlinear simulation. In this case, the whole identification process is repeated using FLIGHTLAB nonlinear simulation data. “Nonlinear Response” utility is used for this purpose. Before executing a time response run, desired inputs and outputs are set from “Xanalysis” interface. Scripts are generated to define specific input-output data whose configuration is defined in the beginning of this chapter.

The corresponding time domain output comparisons are shown in Figure 6.54. The obtained TIC value for this case is 0.324 which is also acceptable.

As we mentioned above, we selected the model order as 8 in order to be compatible with the model defined in the linear case. However, it may be possible to obtain better output matching with a higher order model selection. For example, if we selected the order by including the main rotor, inflow and engine we may get more accurate fits. Time domain output comparisons of the model with an order of 11 are given in Figure 6.55. Parameter estimation for higher order models can be an another research study.

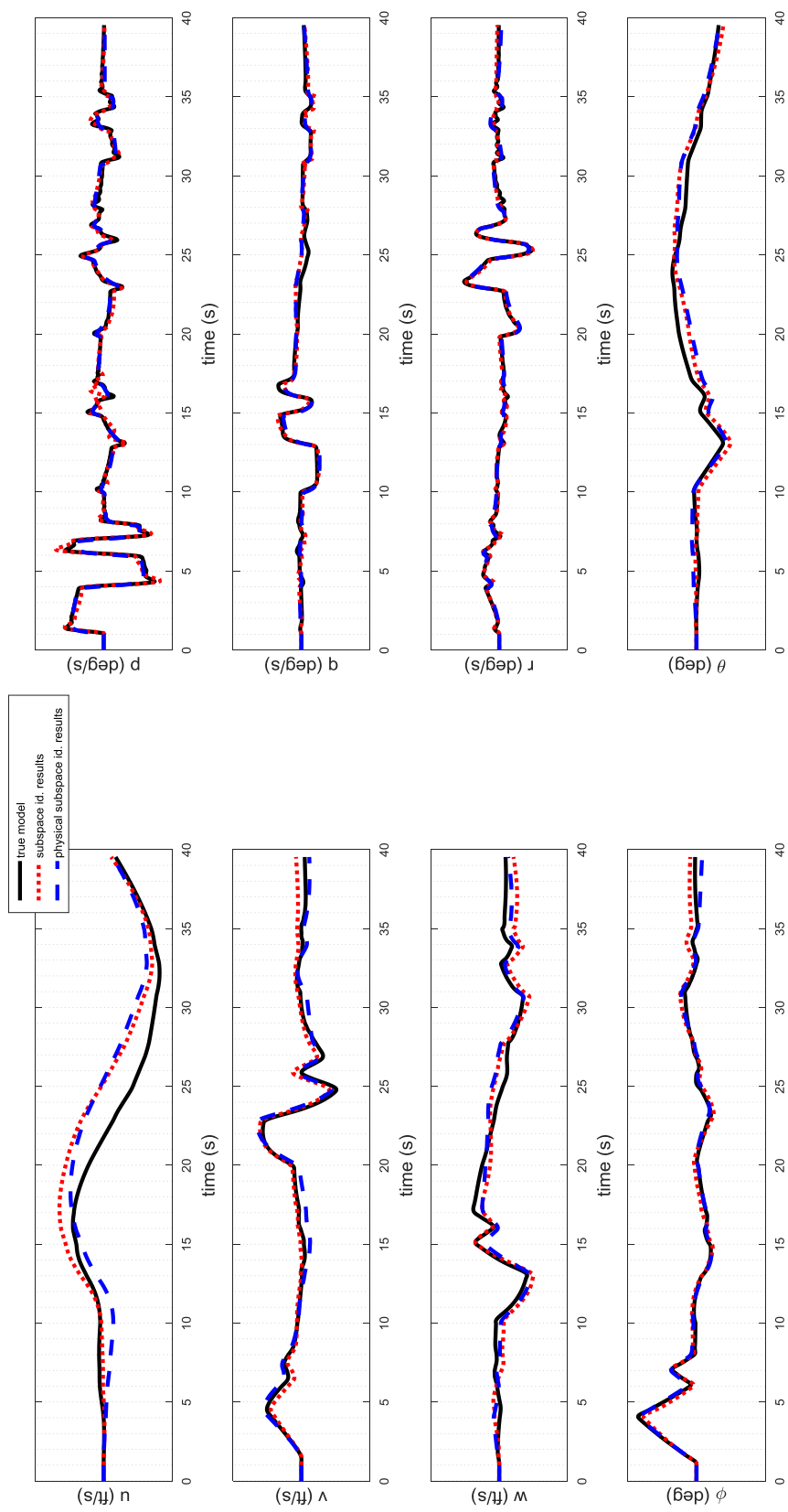


Figure 6.54. Comparison of Outputs for 3-2-1-1 Excitation Signals (Identification Using Nonlinear Model Simulation)

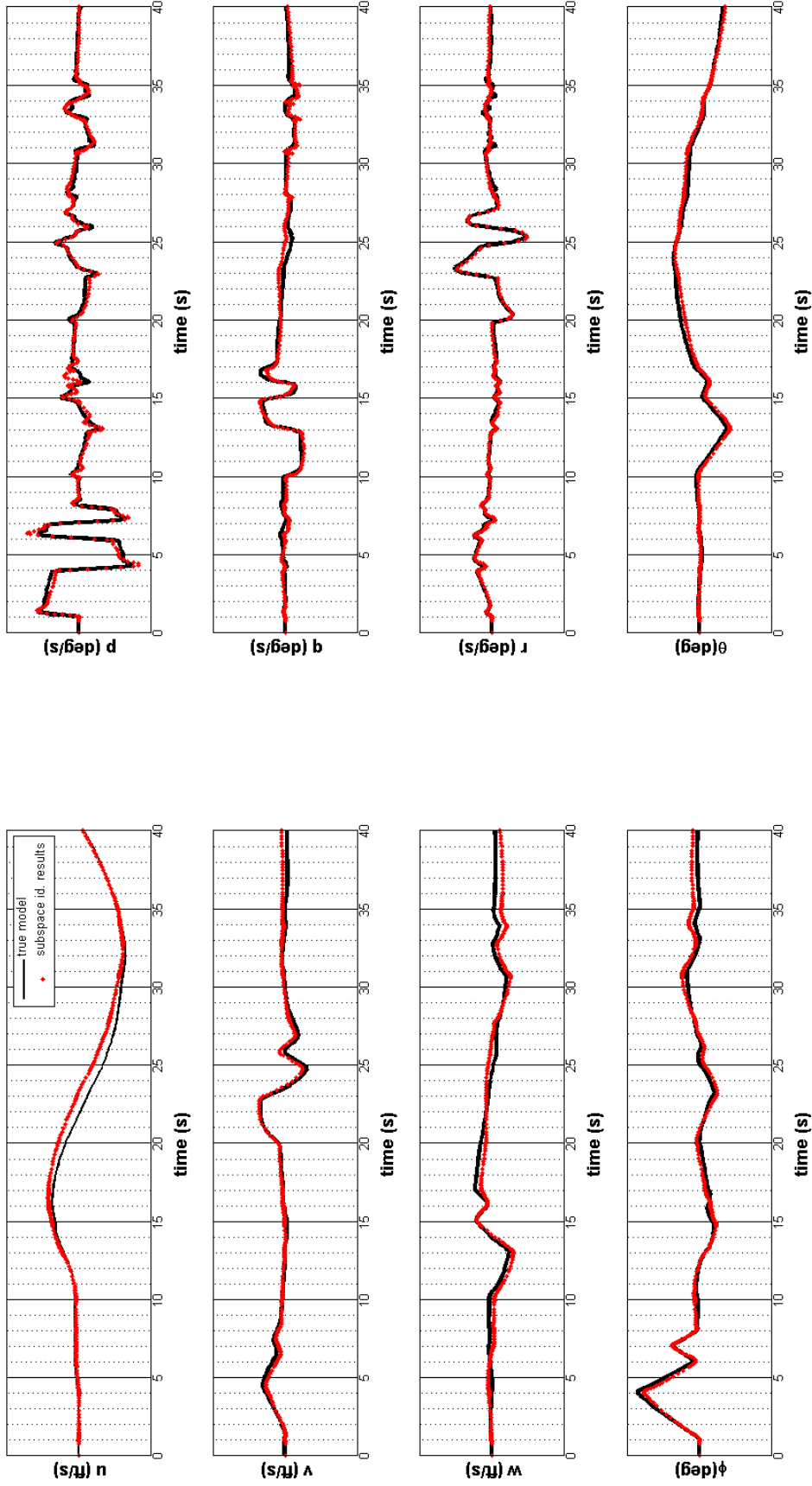


Figure 6.55. Comparison of Outputs for 3-2-1-1 Excitation Signals (Identification Using Nonlinear Model Simulation, for the order  $n$  is selected as 11)

## 6.8. Implementation to Real Flight Test Data

The proposed methodology is implemented to a previously recorded flight test data of a multi-role helicopter. The helicopter was instrumented with GPS aided INS, air data system and other various flight test equipment. GPS aided INS was the major data source for this study for obtaining the translational velocity components (longitudinal, lateral and vertical), angular velocity components (roll, pitch and yaw) and Euler angles (roll and pitch). A data consistency check was conducted between the GPS aided INS and air data systems for assuring the reliability of the test data. The pilot control inputs (lateral cyclic, longitudinal cyclic, pedal and collective) were measured by the relevant actuation systems during the flight test. The mentioned flight test data was recorded by both onboard and ground (telemetry) data acquisition systems.

INS data sampling frequency (150 Hz) and pilot input sampling frequency (62.5 Hz) were not equal for this case. In order to prepare the data for the analysis, pilot input recordings were up sampled to the INS frequency by linear interpolation. It can be mentioned that in this flight test the frequency range of pilot inputs is around 1 Hz which assures that there is no possibility of aliasing due to this up sampling process.

In this flight test, the helicopter was initially trimmed at 110 knots level flight condition at 7300 ft sea level altitude. 3 sets of longitudinal cyclic doublet inputs were applied sequentially by the pilot for excitation (Figure 6.56). The other control channels were only used for maintaining the trim condition. Since excitation to the helicopter is only given in the longitudinal channel, the flight test data being studied is only convenient for the identification of the longitudinal derivatives. Consequently  $L_p, L_q, L_r, M_p, M_q, M_r, N_p, N_q, N_r, L_{\delta_{lon}}, M_{\delta_{lon}}, N_{\delta_{lon}}$  are the derivatives being estimated and the rest of the derivatives are set to zero during the process. The initial values for the physical parameters are set to be equal to the linearization outputs of FLIGHTLAB model. The constraints are selected by trial and error here.

In the scope of real flight test data implementation, it is observed that a time delay correction is required in the pilot inputs. Time delays unique to each control channel are determined by matching the model behavior with the flight test data. They are found to be between 0.1-0.3 seconds. The main reason behind these time delays is evaluated to be associated with the quasi steady assumption of 6-DOF helicopter modeling [17]. The issue can be attributed to transient rotor dynamics.

The TIC value for flight test data implementation is obtained as 0.31. This indicates that the results are again acceptable. The comparison of the real flight test measurements with the outputs of the model obtained through subspace identification and the outputs of the model obtained through physical subspace identification are shown in Figure 6.57.

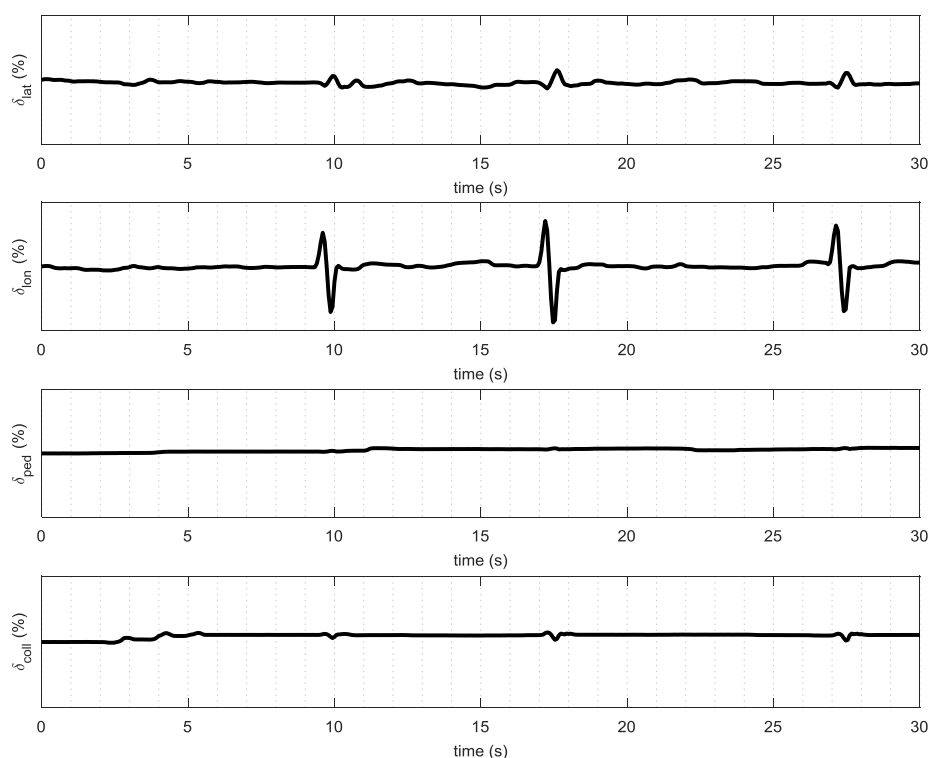


Figure 6.56. Input Signals of Real Flight Test



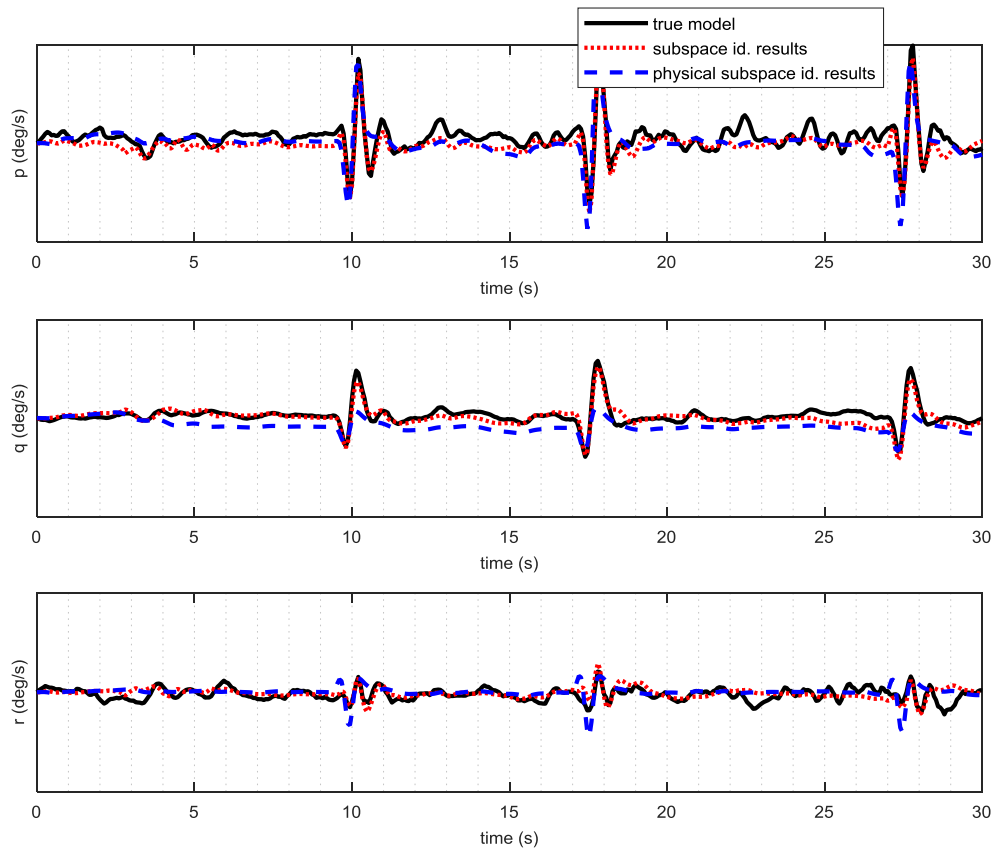


Figure 6.57. Comparison of Outputs for Longitudinal Cyclic Doublet Excitations (Identification Using Flight Test Data)



## CHAPTER 7

### CONCLUSIONS AND FUTURE WORK

#### 7.1. Conclusions

In this thesis, the subspace identification methodology is implemented on a multi-role twin engine helicopter. The studies are focused on the problem of physical parameter estimation from subspace identification results. Nonlinear optimization methodologies are searched for solving the physical parameter estimation.

In Chapter 2, the theory of subspace identification is presented. “Robust Subspace Identification” algorithm which proved to work well on practical data is explained in detail. The problem of physical parameter estimation is defined and a solution methodology is proposed.

In Chapter 3, the physical parameters that are aimed to be estimated are defined. The equations of motion for 6-DOF helicopter model are introduced and the model structure is presented with its inputs, outputs and the states. The main assumptions about the physical parameters are mentioned and the importance of these parameters is explained.

The optimization problem is introduced in Chapter 4. The objective function is defined and the reasoning behind the algorithm selection is explained. Accordingly, both SQP and IP algorithms are briefly mentioned. The basic assumptions about constraints, initial values and constant terms are also given in this chapter.

Implementation of subspace identification on the multi-role helicopter is introduced in Chapter 5. First, the model generation, trim analysis and linearization of

FLIGHTLAB model are described. Input signal design and linear response outputs are illustrated. After that, the implementation of subspace identification method by using generated input output data set. The eigenvalues of the estimated model are compared with the ones of the true model. Time domain simulation outputs of the estimated model are compared with the true one. Then the required objective function is constructed for physical parameter estimation. The methodology for constraint selection and initial value assessment are described for our optimization problem.

The numerical results obtained for different optimization problems are presented in Chapter 6. Five case studies are examined. Two algorithms are utilized: IP and SQP. The performances of these algorithms are analyzed for several constraint and initial value set. The convergence results are compared in many aspects like convergence behavior, convergence rate, etc. Percentage error values are calculated for the physical parameters for each test case. TIC values are computed between the estimated models and the true model. The optimization algorithms are repeated for several times starting from randomly selected initial conditions to test the confidence level. Consistent results are achieved for different initial conditions. The method that is proved with linear model data until this point is tested with nonlinear simulation and real flight data.

Numerical results of Case 1 and Case 2 indicate that there is not a distinct change in the convergence results due to expansion in constraint limits. Numerical results of Case 1, Case 2 and Case 3 present that both algorithms are capable of finding the physical parameters. However, SQP algorithm seems more trustworthy due to the consistency of the convergence results which are not affected by the symmetry of the constraints. Case 4 deals with the performance of optimization for arbitrary initialization of  $T$  matrix. The analysis results indicate that it is still possible to find a solution when the initial condition of  $T$  matrix is not equal to the inverse of  $C$  matrix.

The study showed that using the methodology it is possible to estimate all of the significant physical parameters with one identification test. The methodology uses the advantages of MIMO identification to estimate many parameters at a time. This is quite important for the helicopter systems which have coupled dynamics. Implementation of the methodology on real flight data showed that the physical subspace identification methodology is well applicable to real world practices.

## **7.2. Future Work**

In real flight test applications, the helicopters and even other conventional flight vehicles are generally excited by using one control input. Excitation of the system simultaneously from all of the input channels may cause the system to enter into a nonlinear flight regime. Therefore, these are avoided due to safety reasons and the limitations of the system identification theory that is based on linear assumptions. However, as a future study, blending all of the subspace identification results which are obtained from different flight maneuvering conditions into one optimization problem may provide a solution for the estimation of all the stability and control derivatives all at a time. To obtain a better estimation, weight function assignment can be practiced according to the reliability of the subspace identification results of the relevant dynamics. Weighting of the parameters according to the coherence function between the inputs and the outputs for optimization may give better results.

Identifying higher order modes of the helicopter like flapping dynamics, engine etc. which enlarges the size of the unknown parameters and correspondingly complicates the optimization problem. This can be another challenging issue to be solved.

There are many options for initial value assignment and constraint determination of the parameters. Analysis and test data obtained in the scope of flight stability, handling qualities and wind tunnel testing activities can be utilized for this purpose.

In this study all of the analyses are conducted for one single trim point. In future, this methodology can be repeated for several flight conditions and the obtained models can be stitched together to find a nonlinear model for the system from physical subspace identification.

## REFERENCES

- [1] D. Banerjee and J.W. Harding, 1991, AGARD, Rotorcraft System Identification, AGARD LS 178, “Assessment of Rotorcraft System Identification as Applied to the AH-64”, McDonnell Douglas Helicopter Company, Mesa, Arizona
- [2] J. Kaletka, 1991, AGARD, Rotorcraft System Identification, AGARD LS 178, “BO 105 Identification Results”, Deutsche Forschungsanstalt für Luft- und Raumfahrt e V (DLR) Institut für Flugmechanik, DW-3300 Braunschweig, Flughafen, Germany
- [3] G. D.Padfield, 1991, AGARD, Rotorcraft System Identification, AGARD LS 178, “SA 330 Puma Identification Results”, Defence Research Agency Aerospace Division RAE Bedford, UK
- [4] Klein, V., and Morelli, E. A., Aircraft System Identification: Theory and Practice, AIAA, 2006.
- [5] Hamel, P. G. (ed.), Rotorcraft System Identification, AGARD-AR-280, 1991.
- [6] R. V. Jategaonkar, 2006, “Flight Vehicle System Identification – A Time Domain Methodology”, American Institute of Aeronautics and Astronautics, Inc. 1801 Alexander Bell Drive, Reston, VA 20191
- [7] Tischler, M.B., and Cauffman, M. G., “Frequency-Response Method for Rotorcraft System Identification: Flight Applications to BO 105 Coupled Rotor/Fuselage Dynamics,” Journal of the American Helicopter Society, Vol. 37, (3), July 1992, pp. 3–17.

- [8] Fletcher, J.W., "Identification of UH-60 Stability Derivative Models in Hover from Flight Test Data: Journal of the American Helicopter Society, Vol. 40, (I), 1995.
- [9] Mettler, B., Tischler, M. B., and Kanade, T., "System Identification Modeling of a Small-Scale Unmanned Rotorcraft for Flight Control Design," Journal of the American Helicopter Society, Vol. 47, (1), January 2002, pp. 50–63.
- [10] Ham J. A., Gardner, G. K., and Tischler, M. B., "Flight-Testing and Frequency-Domain Analysis for Rotorcraft Handling Qualities," Journal of the American Helicopter Society, Vol. 40, (2), 1995, pp. 28-38.
- [11] Tomashofski, C. A., and Tischler, M. B., "Flight Test Identification of SH-2G Dynamics in Support of Digital Flight Control System Development," American Helicopter Society 55th Annual Forum Proceedings, Montreal, Canada, May 1999.
- [12] Harding, I. W., "Frequency-Domain Identification of Coupled Rotor/Body Models of an Advanced Attack Helicopter" American Helicopter Society 48th Annual Forum. Washington. June 1992.
- [13] Schroeder, I. A., Tischler, M. B., Watson, D. C. and Eshow, M. E., "Identification and Simulation Evaluation of Combat Helicopter in Hover, AIAA Journal of Guidance Control and Dynamics, Vol. 18, No. 1, pp. 31-38. Jan.-Feb. 1995.
- [14] Wei, W., Tischler, M. B., Cohen, K., System Identification and Controller Optimization of a Quadrotor Unmanned Aerial Vehicle in Hover, Journal of the American Helicopter Society, Vol. 62, 2017, DOI: 10.4050/JAHS.62.042007



- [15] Bhandari, S., Colgren R., “High-Order Dynamics Models of a Small UAV Helicopter Using Analytical and Parameter Identification Techniques”, Journal of the American Helicopter Society, Vol. 60, 2015, DOI: 10.4050/JAHS.60.022012.
- [16] Khaligh, S. P., Fahimi, F., Koch, R. C., “A System Identification Strategy for Nonlinear Model of Small-Scale Unmanned Helicopters” Journal of the American Helicopter Societ, Vol 61, 2016, DOI: 10.4050/JAHS.61.042002
- [17] Tischler, M. B. and Remple, R. K., “Aircraft and Rotorcraft System Identification: Engineering Methods with Flight-Test Examples”, American Institute of Aeronautics and Astronautics, Inc., Reston, VA, 2nd edition, 2012.
- [18] Geluardi, S., Nieuwenhuizen, F. M., Venrooij, J., Pollini, L, Bulthoff, H. H., “Frequency Domain System Identification of a Robinson R44 in Hover” Journal of the American Helicopter Society, Vol 63, 2018, DOI: 10.4050/JAHS.63.012009.
- [19] Van Overschee, P. and De Moor B., Subspace Identification for Linear Systems Theory - Implementation –Applications, Kluwer Academic Publishers, Boston/ London/ Dordrecht, 1996.
- [20] Budin M. Minimal realization of discrete linear systems from input-output observations. IEEE Transactions on Automatic Control, Vol. AC-16, no. 5, pp. 395-401, 1971.
- [21] Gopinath B. On the identification of linear time-invariant systems from input output data. The Bell System Technical Journal, Vol. 48, no. 5, pp. 1101-1113, 1969.

- [22] Liu R., Suen L.C. Minimal dimension realization and identifiability of input output sequences. IEEE Transactions on Automatic Control, Vol.AC-22, pp. 227- 232, 1977.
- [23] Akaike H. Stochastic theory of minimal realization. IEEE Transactions on Automatic Control, 19, pp. 667-674, 1974.
- [24] Akaike H. Markovian representation of stochastic processes by canonical variables. Siam J. Control, Vol. 13, no.1, pp. 162-173, 1975.
- [25] Aoki M. State space modeling of time series. Springer Verlag, Berlin, 1987.
- [26] Arun K.S., Kung S.Y. Balanced approximation of stochastic systems. SIAM J. Matrix Analysis and Applications, 11, pp. 42-68, 1990.
- [27] Larimore W.E. System identification, reduced order filtering and modeling via canonical variate analysis. Proc. of the American Control Conference, San Francisco, USA, 1983.
- [28] Larimore W.E. Canonical variate analysis in identification, filtering and adaptive control. Proc. 29th Conference on Decision and Control, Hawaii, USA, pp. 596-604, 1990.
- [29] Verhaegen M. A novel non-iterative MIMO state space model identification technique. Proc. 9th IFAC/IFORS Symp. on Identification and System Parameter Estimation, Budapest, Hungary, pp. 1453-1458, 1991.

- [30] Verhaegen, M., 1994, "Identification of the deterministic part of MIMO state space models given in innovations form from input–output data". *Automatica*, 30(1), 61–74.
- [31] Verhaegen, M., Verdult, V., *Filtering and System Identification: A Least Squares Approach*, Cambridge University Press, New York, NY, USA ©2007 ISBN:0521875129 9780521875127
- [32] Bittanti S., Lovera M., "Identification of Linear Models for a Hovering Helicopter Rotor", *IFAC System Identification*, Kitakyushu, Fukuoka, Japan, 1997.
- M. Verhaegen and A. Varga, "Some Experience with the MOESP Class of Subspace Model Identification Methods in identifying the BO105 Helicopter," tech. rep., German Aerospace Research Establishment, Institute for Robotics and System Dynamics, 1994.
- [33] Lovera M., "Identification of MIMO State Space Models for Helicopter Dynamics", *IFAC System Identification*, Rotterdam, Netherlands, 2003.
- [34] I. Ping Li, "Subspace-based System Identification for Helicopter Dynamic Modelling", *American Helicopter Society 63rd Annual Forum*, Virginia Beach, VA, May 1-3, 2007.
- [35] R.V., *Flight Vehicle System Identification: A Time Domain Methodology*, American Institute of Aeronautics and Astronautics Inc., Reston, VA, 2006.
- [36] Li, P., Postlethwaite, I., and Turner, M., "Subspace-Based System Identification for Helicopter Dynamic Modelling," *American Helicopter Society 63rd Annual Forum Proceedings*, Virginia Beach, VA, May 1–3, 2007.

- [37] Wartmann J., Seher-Weiss S. “Application of the Predictor-Based Subspace Identification Method to Rotorcraft System Identification”, 39th European Rotorcraft Forum, Moscow, Russia, 2013.
- [38] Seher-Weiss S., “ACT/FHS System Identification Including Rotor and Engine Dynamics”, AHS International 73rd Annual Forum & Technology Display, Fort Worth, Texas, USA, May 9–11, 2017.
- [39] Li, P., Postlethwaite, I., “Subspace and Bootstrap-Based Techniques for Helicopter Model Identification”, Journal of the American Helicopter Society, Vol. 56, 2011, DOI: 10.4050/JAHS.56.012002.
- [40] Bergamasco, M. and Lovera, M., “Continuous-time predictor-based subspace identification using Laguerre filters”. IET Control Theory Applications, 5(7):856–867, 5 2011a.
- [41] Bergamasco, M. and Lovera, M., Continuous-time predictor-based subspace identification for helicopter dynamics. In 37th European Rotorcraft Forum, Gallarate, Italia, 2011b.
- [42] Bergamasco, M. and Lovera, Rotorcraft system identification:an integrated time-frequency domain approach. In 2nd CEAS Specialist Conference on Guidance, Navigation & Control, Delft, The Netherlands, 2013.
- [43] Hespanha, J. P, Linear System Theory, Princeton University Press, 2009, Chap.13.

- [44] L. Liang Xie, L. Ljung, “Estimate Physical Parameters by Black-Box Modelling”, In: Proc. 21st Chinese Control Conference. Hangzhou, China. pp. 673–677, 2002.
- [45] Parrilo, P., Ljung, L., “Initialization of Physical Parameter Estimates”, 13th IFAC Symposium on System Identification, Rotterdam, Netherlands, 3 December 2003.
- [46] Lyzell, C., Enqvist, M., Ljung, L., “Handling Certain Structure Information in Subspace Identification”, Proceedings of the 15th IFAC Symposium on System Identification Saint-Malo, France, July 6-8, 2009
- [47] J. J. Ramos, Lopes dos Santos, P., “Mathematical Modeling, System Identification, and Controller Design of a Two Tank System”, Proceedings of the 46th IEEE Conference on Decision and Control 46th IEEE Conference on Decision and Control New Orleans, LA, USA, Dec. 12-14, 2007
- [48] J. Ramos, A., Mercère, G., Prot, O. “Parameter Estimation of Discrete and Continuous-Time Physical Models: A Similarity Transformation Approach”, 49th IEEE Conference on Decision and Control, Atlanta, GA, USA, December 15-17, 2010.
- [49] J. Ramos, Lopes dos Santos, P.,” Identifying Second-Order Models of Mechanical Structures in Physical Coordinates: An Orthogonal Complement Approach”, IEEE European Control Conference 1, Zurich, Switzerland, July 17-19, 2013.
- [50] Benson, H. Y., D. F. Shanno, and R. J. Vanderbei, “A Comparative Study of Large-Scale Nonlinear Optimization Algorithms” in High Performance

Algorithms and Software for Nonlinear Optimization, G. Di Pill0 and A. Murli (Eds.), Kluwer Academic, Nonvell, MA, pp. 95-128, 2003.

- [51] S.Avcioglu, A.T. Kutay, K. Leblebicioglu.,” Implementation of Physical Subspace Identification on a Realistic Helicopter, AIAA Science and Technology Forum and Exposition 2019, San Diego, CA,USA, January 7-12,2019.
  
- [52] Nocedal, J. and S. J. Wright. Numerical Optimization, Second Edition. Springer Series in Operations Research, Springer Verlag, 2006.
  
- [53] Byrd, R.H., Mary E. Hribar, and Jorge Nocedal, "An Interior Point Algorithm for Large-Scale Nonlinear Programming, SIAM Journal on Optimization," SIAM Journal on Optimization, Vol 9, No. 4, pp. 877–900, 1999
  
- [54] R.H. Byrd, M.E. Hribar, and J. Nocedal, “An interior point algorithm for large scale nonlinear programming”, SIAM J. Opt., 9(4):877–900, 1999.
  
- [55] Murray-Smith, D. J., 1998, “Methods for the External Validation of Continuous System Simulation Models: A Review,” Journal of Mathematical Modelling of Systems, Cilt. 4, No., s. 5–31.
  
- [56] Fischenberg, D., Jategaonkar, R., von Gruenhagen, W., 2004, “Aerodynamic Modeling and System Identification from Flight Data—Recent Applications at DLR,” Journal of Aircraft, Cilt. 41, No. 4, s. 681–691.
  
- [57] Ljung. L., System identification - Theory for the User, Englewood: Prentice Hall, 1987

- [58] Padfield, Gareth, “Helicopter Flight Dynamics: *The Theory and Application of Flying Qualities and Simulation Modeling*, American Institute of Aeronautics and Astronautics”, Washington D.C., 1996.
- [59] AGARD, Rotorcraft System Identification, AGARD AR 280, 1991.
- [60] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. “*Nonlinear Programming. Theory and Algorithms*”. John Wiley & Sons, second edition, 1993.
- [61] Jategaonkar, R.V., “*Flight Vehicle System Identification: A Time Domain Methodology*”, American Institute of Aeronautics and Astronautics Inc., Reston, VA, 2006.
- [62] Murray-Smith, D. J, “Methods for the External Validation of Continuous System Simulation Models: A Review,” *Journal of Mathematical Modelling of Systems*, Vol. 4, No., pp. 5–31, 1998.
- [63] Fischenberg, D., Jategaonkar, R., von Gruenhagen, W., “Aerodynamic Modeling and System Identification from Flight Data—Recent Applications at DLR,” *Journal of Aircraft*, Vol. 41, No. 4, pp. 681–691, 2004.
- [64] Willems J. From time series to linear systems. *Automatica*, Part I: Vol. 22, 1986
- [65] De Moor B. Mathematical concepts and techniques for modeling of static and dynamic systems. PhD thesis, Department of Electrical Engineering, Katholieke Universiteit Leuven, Belgium, 1988.

- [66] Kung S.Y. A new identification method and model reduction algorithm via singular value decomposition. 12th Asilomar Conf. on Circuits, Systems and Comp., pp. 705-714, Asilomar, CA, 1978.
- [67] FLIGHTLAB X-Analysis User Manual Advanced Rotorcraft Technology, Inc, June 2015
- [68] FLIGHTLAB Manual Vol I, Theory and Component References”, AdvanceRotorcraft Technology, April 2001.
- [69] FLIGHTLAB Manual Vol II, Scope, GSCOPE, PilotStation, CSGE, ModelEditor, Xanalysis”, Advance Rotorcraft Technology, April 2001.
- [70] Maine, R. E., and Iliff, K. W., “*Theory and Practice of Estimating the Accuracy of Dynamic Flight-Determined Coefficients*,” NASA RP 1077, 1981



## APPENDICES

### A. Proof of Theorem 2 [19]

From the formulas (38) and (40); we find that  $X_f$  can be written as linear combination of the past inputs  $U_p$  and the outputs  $Y_p$  as follows:

$$\begin{aligned}
 X_f &= A^i X_p + \Delta_i U_p \\
 &= A^i [\Gamma_i^\dagger Y_p - \Gamma_i^\dagger H_i U_p] + \Delta_i U_p \\
 &= A^i [\Delta_i - A^i \Gamma_i^\dagger H_i U_p] U_p + [A^i \Gamma_i^\dagger] Y_p \\
 &= L_p W_p
 \end{aligned} \tag{A.1}$$

with

$$L_p = A^i [\Delta_i - A^i \Gamma_i^\dagger H_i U_p | A^i \Gamma_i^\dagger]$$

with (A.2), the formula (39) can be rewritten as;

$$Y_f = \Gamma_i L_p W_p + H_i U_f \tag{A.2}$$

From this formula and using the definition of oblique projection ([19], equation 1.7) given in (A.3) the first claim of Theorem can be proven as;

$$A/B \mathbf{C} = [A/B^\perp][C/B^\perp]^\dagger C \quad (\text{A.1})$$

$$Y_f \Pi_{U_f^\perp} = \Gamma_i L_p W_p \Pi_{U_f^\perp} + H_i \underbrace{U_f \Pi_{U_f^\perp}}_{=0}$$

$$Y_f/U_f^\perp = \Gamma_i L_p W_p/U_f^\perp \quad (\text{A.2})$$

$$\underbrace{Y_f/U_f^\perp [W_p/U_f^\perp]^\dagger}_{O_i} W_p = \Gamma_i \underbrace{L_p W_p}_{=X_f}$$

$$O_i = \Gamma_i X_f$$

Where we have used the fact that  $[W_p/U_f^\perp][W_p/U_f^\perp]^\dagger W_p = W_p$ . This is not trivial, since  $W_p/U_f^\perp$  is rank deficient for purely deterministic systems which implies that  $[W_p/U_f^\perp][W_p/U_f^\perp]^\dagger$  is different from identity.

## B. State Space Model Representation In FLIGHTLAB

The similarity transformation approximation is utilized to convert the state space model structure from FLIGHTLAB form (Eq. B.1) to the model structure given in Eq. (112). The structure of the similarity transformation matrix  $T$  (Eq.B.2) is constructed according to the linear relation between the state space model structures given in Eq. B.1 and Eq. (112) where the stability derivative matrix  $A$  is given in Eq. (113) and the control derivative matrix is given in Eq.(114). The elements of FLIGHTLAB model matrices given in Eq. B.1 are matched with the elements of system model given in Eq. (113) and Eq.(114) with the equality written through Eq.(B.3) to Eq.(B.33).

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{\partial \dot{\phi}}{\partial \phi} & \frac{\partial \dot{\phi}}{\partial \theta} & \frac{\partial \dot{\phi}}{\partial u} & \frac{\partial \dot{\phi}}{\partial v} & \frac{\partial \dot{\phi}}{\partial w} & \frac{\partial \dot{\phi}}{\partial p} & \frac{\partial \dot{\phi}}{\partial q} & \frac{\partial \dot{\phi}}{\partial r} \\ \frac{\partial \dot{\theta}}{\partial \phi} & \frac{\partial \dot{\theta}}{\partial \theta} & \frac{\partial \dot{\theta}}{\partial u} & \frac{\partial \dot{\theta}}{\partial v} & \frac{\partial \dot{\theta}}{\partial w} & \frac{\partial \dot{\theta}}{\partial p} & \frac{\partial \dot{\theta}}{\partial q} & \frac{\partial \dot{\theta}}{\partial r} \\ \frac{\partial \dot{u}}{\partial \phi} & \frac{\partial \dot{u}}{\partial \theta} & \frac{\partial \dot{u}}{\partial u} & \frac{\partial \dot{u}}{\partial v} & \frac{\partial \dot{u}}{\partial w} & \frac{\partial \dot{u}}{\partial p} & \frac{\partial \dot{u}}{\partial q} & \frac{\partial \dot{u}}{\partial r} \\ \frac{\partial \dot{v}}{\partial \phi} & \frac{\partial \dot{v}}{\partial \theta} & \frac{\partial \dot{v}}{\partial u} & \frac{\partial \dot{v}}{\partial v} & \frac{\partial \dot{v}}{\partial w} & \frac{\partial \dot{v}}{\partial p} & \frac{\partial \dot{v}}{\partial q} & \frac{\partial \dot{v}}{\partial r} \\ \frac{\partial \dot{w}}{\partial \phi} & \frac{\partial \dot{w}}{\partial \theta} & \frac{\partial \dot{w}}{\partial u} & \frac{\partial \dot{w}}{\partial v} & \frac{\partial \dot{w}}{\partial w} & \frac{\partial \dot{w}}{\partial p} & \frac{\partial \dot{w}}{\partial q} & \frac{\partial \dot{w}}{\partial r} \\ \frac{\partial \dot{p}}{\partial \phi} & \frac{\partial \dot{p}}{\partial \theta} & \frac{\partial \dot{p}}{\partial u} & \frac{\partial \dot{p}}{\partial v} & \frac{\partial \dot{p}}{\partial w} & \frac{\partial \dot{p}}{\partial p} & \frac{\partial \dot{p}}{\partial q} & \frac{\partial \dot{p}}{\partial r} \\ \frac{\partial \dot{q}}{\partial \phi} & \frac{\partial \dot{q}}{\partial \theta} & \frac{\partial \dot{q}}{\partial u} & \frac{\partial \dot{q}}{\partial v} & \frac{\partial \dot{q}}{\partial w} & \frac{\partial \dot{q}}{\partial p} & \frac{\partial \dot{q}}{\partial q} & \frac{\partial \dot{q}}{\partial r} \\ \frac{\partial \dot{r}}{\partial \phi} & \frac{\partial \dot{r}}{\partial \theta} & \frac{\partial \dot{r}}{\partial u} & \frac{\partial \dot{r}}{\partial v} & \frac{\partial \dot{r}}{\partial w} & \frac{\partial \dot{r}}{\partial p} & \frac{\partial \dot{r}}{\partial q} & \frac{\partial \dot{r}}{\partial r} \end{bmatrix} \cdot \begin{bmatrix} \phi \\ \theta \\ u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} \quad (\text{B.1})$$

$$\begin{aligned}
& + \begin{bmatrix} \frac{\partial \dot{\phi}}{\partial \delta_{lon}} & \frac{\partial \dot{\phi}}{\partial \delta_{lat}} & \frac{\partial \dot{\phi}}{\partial \delta_{col}} & \frac{\partial \dot{\phi}}{\partial \delta_{ped}} \\ \frac{\partial \dot{\theta}}{\partial \delta_{lon}} & \frac{\partial \dot{\theta}}{\partial \delta_{lat}} & \frac{\partial \dot{\theta}}{\partial \delta_{col}} & \frac{\partial \dot{\theta}}{\partial \delta_{ped}} \\ \frac{\partial \dot{u}}{\partial \delta_{lon}} & \frac{\partial \dot{u}}{\partial \delta_{lat}} & \frac{\partial \dot{u}}{\partial \delta_{col}} & \frac{\partial \dot{u}}{\partial \delta_{ped}} \\ \frac{\partial \dot{v}}{\partial \delta_{lon}} & \frac{\partial \dot{v}}{\partial \delta_{lat}} & \frac{\partial \dot{v}}{\partial \delta_{col}} & \frac{\partial \dot{v}}{\partial \delta_{ped}} \\ \frac{\partial \dot{w}}{\partial \delta_{lon}} & \frac{\partial \dot{w}}{\partial \delta_{lat}} & \frac{\partial \dot{w}}{\partial \delta_{col}} & \frac{\partial \dot{w}}{\partial \delta_{ped}} \\ \frac{\partial \dot{p}}{\partial \delta_{lon}} & \frac{\partial \dot{p}}{\partial \delta_{lat}} & \frac{\partial \dot{p}}{\partial \delta_{col}} & \frac{\partial \dot{p}}{\partial \delta_{ped}} \\ \frac{\partial \dot{q}}{\partial \delta_{lon}} & \frac{\partial \dot{q}}{\partial \delta_{lat}} & \frac{\partial \dot{q}}{\partial \delta_{col}} & \frac{\partial \dot{q}}{\partial \delta_{ped}} \\ \frac{\partial \dot{r}}{\partial \delta_{lon}} & \frac{\partial \dot{r}}{\partial \delta_{lat}} & \frac{\partial \dot{r}}{\partial \delta_{col}} & \frac{\partial \dot{r}}{\partial \delta_{ped}} \end{bmatrix} \cdot \begin{bmatrix} \delta_{lon} \\ \delta_{lat} \\ \delta_{col} \\ \delta_{ped} \end{bmatrix}
\end{aligned}$$

$$T = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (B.2)$$

$$\frac{\partial \dot{\phi}}{\partial \phi} = \frac{\partial \dot{\phi}}{\partial \theta} = \frac{\partial \dot{\phi}}{\partial u} = \frac{\partial \dot{\phi}}{\partial v} = \frac{\partial \dot{\phi}}{\partial w} = \frac{\partial \dot{\phi}}{\partial \delta_{lat}} = \frac{\partial \dot{\phi}}{\partial \delta_{lon}} = \frac{\partial \dot{\phi}}{\partial \delta_{ped}} = \frac{\partial \dot{\phi}}{\partial \delta_{col}} = 0 \quad (B.3)$$

$$\frac{\partial \dot{\phi}}{\partial p} = 1 \quad (B.4)$$

$$\frac{\partial \dot{\phi}}{\partial q} = \sin \phi_0 \tan \theta_0 \quad (B.5)$$

$$\frac{\partial \dot{\phi}}{\partial r} = r \cos \phi_0 \tan \theta_0 \quad (\text{B.6})$$

$$\frac{\partial \dot{\theta}}{\partial q} = \cos \phi_0 \quad (\text{B.7})$$

$$\frac{\partial \dot{\theta}}{\partial r} = -\sin \theta_0 \quad (\text{B.8})$$

$$\frac{\partial \dot{\theta}}{\partial \varphi} = \frac{\partial \dot{\theta}}{\partial \theta} = \frac{\partial \dot{\theta}}{\partial u} = \frac{\partial \dot{\theta}}{\partial v} = \frac{\partial \dot{\theta}}{\partial w} = \frac{\partial \dot{\theta}}{\partial p} = \frac{\partial \dot{\theta}}{\partial \delta_{lat}} = \frac{\partial \dot{\theta}}{\partial \delta_{lon}} = \frac{\partial \dot{\theta}}{\partial \delta_{ped}} = \frac{\partial \dot{\theta}}{\partial \delta_{col}} = 0 \quad (\text{B.9})$$

$$\frac{\partial \dot{u}}{\partial \varphi} = 0 \quad (\text{B.10})$$

$$\frac{\partial \dot{u}}{\partial \theta} = -g \cos \theta_0 \quad (\text{B.11})$$

$$\frac{\partial \dot{u}}{\partial u} = X_u \quad (\text{B.12})$$

$$\frac{\partial \dot{u}}{\partial v} = X_v \quad (\text{B.13})$$

$$\frac{\partial \dot{u}}{\partial w} = X_w \quad (\text{B.14})$$

$$\frac{\partial \dot{u}}{\partial p} = X_p \quad (\text{B.15})$$

$$\frac{\partial \dot{u}}{\partial q} = X_q - w_0 \quad (\text{B.16})$$

$$\frac{\partial \dot{u}}{\partial r} = X_r + v_0 \quad (\text{B.17})$$

$$\frac{\partial \dot{v}}{\partial \varphi} = g \cos \phi_0 \cos \theta_0 \quad (\text{B.18})$$

$$\frac{\partial \dot{v}}{\partial \theta} = -g \sin \phi_0 \sin \theta_0 \quad (\text{B.19})$$

$$\frac{\partial \dot{u}}{\partial \delta_{lat}} = X_{\delta_{lat}} \quad (\text{B.20})$$

$$\frac{\partial \dot{u}}{\partial \delta_{lon}} = X_{\delta_{lon}} \quad (\text{B.21})$$

$$\frac{\partial \dot{u}}{\partial \delta_{ped}} = X_{\delta_{ped}} \quad (\text{B.22})$$

$$\frac{\partial \dot{u}}{\partial \delta_{col}} = X_{\delta_{col}} \quad (\text{B.23})$$

$$\frac{\partial \dot{v}}{\partial u} = Y_u \quad (\text{B.24})$$

$$\frac{\partial \dot{v}}{\partial v} = Y_v \quad (\text{B.25})$$

$$\frac{\partial \dot{v}}{\partial w} = Y_w \quad (\text{B.26})$$

$$\frac{\partial \dot{v}}{\partial p} = Y_p + w_0 \quad (\text{B.27})$$

$$\frac{\partial \dot{v}}{\partial q} = Y_q \quad (\text{B.28})$$

$$\frac{\partial \dot{v}}{\partial r} = Y_r - u_0 \quad (\text{B.29})$$

$$\frac{\partial \dot{v}}{\partial \delta_{lat}} = Y_{\delta_{lat}} \quad (\text{B.30})$$

$$\frac{\partial \dot{v}}{\partial \delta_{lon}} = Y_{\delta_{lon}} \quad (\text{B.31})$$

$$\frac{\partial \dot{v}}{\partial \delta_{ped}} = Y_{\delta_{ped}} \quad (\text{B.32})$$

$$\frac{\partial \dot{v}}{\partial \delta_{col}} = Y_{\delta_{col}} \quad (\text{B.33})$$





### C. True Model vs. Subspace Identification Results

A matrix (true model):

$$\begin{bmatrix} -0.045 & 0.002 & & & & & & & \\ 0.004 & -0.103 & & & & -116.225 & & & \\ -0.033 & & & & & 2.468 & & & \\ -0.004 & & & & 2.180 & & & & \\ 0.004 & & & & & & & & \\ -0.004 & & & & & & & & \\ 0 & 0 & 0 & 1 & -0.002 & 0.060 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0.032 & 0 & 0 \end{bmatrix}$$

A matrix (subspace identification result):

$$\begin{bmatrix} -0.012 & -0.100 & & & & & & & \\ -0.054 & -0.194 & & & & 0.682 & & & \\ 0.026 & & & & & -2.968 & & & \\ & & & & 0.146 & & & & \\ & & & & & & & & \\ 0.021 & & & & & & & & \\ 0.042 & 0.111 & 0.062 & -0.116 & -0.355 & -0.278 & -3.900 & 0.617 \\ 0 & 0.006 & -0.032 & 0.974 & -0.169 & 0.628 & -0.599 & -8.194 \end{bmatrix}$$

B matrix (true model):

$$\begin{bmatrix} -0.0142 & & & & 0.0461 \\ 0.0790 & & & & \\ & 0.2406 & 0.0031 & & \\ & & 0.0409 & & \\ & & & 0.0156 & \\ 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & \end{bmatrix}$$

B matrix (subspace identification result):

$$\begin{bmatrix} 0.005 & & & & 0.110 \\ -0.014 & & & & \\ & -0.029 & 0.080 & & \\ & & -0.874 & & \\ & & & -0.807 & 0.687 \\ -0.050 & -1.114 & 1.176 & 0.030 & \\ 2.654 & 0.585 & 0.854 & 0.548 & \end{bmatrix}$$

C matrix (true model):

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

C matrix (subspace identification result):

$$\begin{bmatrix} -1.254 & & & & & & & -0.134 \\ & -0.874 & & & & & & \\ & & & & & & & 0.051 \\ & & & & & & & 0.006 \\ & & -0.003 & & & & & 0.014 \\ & & & & & & & -0.001 \\ & & & & & & & \\ & -0.032 & & & & & & 0.0027 \\ & 0.010 & & & & & & -0.005 \\ -0.001 & & & & & & & -0.075 \end{bmatrix}$$

D matrix (true model):

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

D matrix (subspace identification result):

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$



## CURRICULUM VITAE

### PERSONAL INFORMATION

Surname, Name : Avcioglu, Sevil  
Nationality : Turkish (TC)  
Date and Place of Birth : 23 September 1978, Seydişehir  
Phone : +90 312 811 18 00  
E-mail : sevilavcioglu@metu.edu.tr

### EDUCATION

Degree	Institution	Year of Graduation
MS	METU Aerospace Engineering	2012
BS	METU Aeronautical Engineering	2002
High School	Konya Meram Science High School	1995

### WORK EXPERIENCE

Year	Place	Enrollment
2008-Present	Turkish Aerospace Industries (TAI)	Flight Mechanics Engineer
2002-2008	ROKETSAN A.Ş	Systems Engineer

2008-Present	Turkish Aerospace Industries (TAI)	Flight Mechanics Engineer
2002-2008	ROKETSAN A.Ş.	Systems Engineer

## **FOREIGN LANGUAGES**

English (Advanced), Italian (Intermediate)

## **PUBLICATIONS**

1. S.Avcioğlu, A.T. Kutay, K. Leblebicioğlu, "Implementation of Physical Subspace Identification on a Realistic Helicopter, AIAA Science and Technology Forum and Exposition 2019, San Diego, CA, USA, January 7-12, 2019.

## **HOBBIES**

Outdoor Sports, Painting, Cooking