

Model Identification and Robust Nonlinear Model Predictive Control of a Twin Rotor MIMO System

Rahideh, Akbar

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author

For additional information about this publication click this link.

<http://qmro.qmul.ac.uk/jspui/handle/123456789/1885>

Information about this research object was correct at the time of download; we occasionally make corrections to records, please therefore check the published record when citing. For more information contact scholarlycommunications@qmul.ac.uk

**Model Identification and Robust Nonlinear
Model Predictive Control of a Twin Rotor
MIMO System**

A thesis submitted by

Akbar Rahideh

in partial fulfilment of
the requirements of the degree of

Doctor of Philosophy

in the

School of Engineering and Materials Science

Queen Mary, University of London

Mile End Road

London, E1 4NS

UK

2009

School of Engineering and Materials Science
Queen Mary, University of London

PhD THESIS

DECLARATION

This thesis entitled

**Model Identification and Robust Nonlinear
Model Predictive Control of a Twin Rotor
MIMO System**

was composed by me and is based on my own work. Where the work of the others has been used, it is fully acknowledged in the text, captions, tables and illustrations. This thesis has not been submitted for any other qualification.

Name AKBAR RAHIDEH

Signed 

Date 17/04/2009

Abstract

This thesis presents an investigation into a number of model predictive control (MPC) paradigms for a nonlinear aerodynamics test rig, a twin rotor multi-input multi-output system (TRMS). To this end, the nonlinear dynamic model of the system is developed using various modelling techniques. A comprehensive study is made to compare these models and to select the best one to be used for control design purpose. On the basis of the selected model, a state-feedback multistep Newton-type MPC is developed and its stability is addressed using a terminal equality constraint approach. Moreover, the state-feedback control approach is combined with a nonlinear state observer to form an output-feedback MPC. Finally, a robust MPC technique is employed to address the uncertainties of the system.

In the modelling stage, analytical models are developed by extracting the physical equations of the system using the Newtonian and Lagrangian approaches. In the case of the black-box modelling, artificial neural networks (ANNs) are utilised to model the TRMS. Finally, the grey-box model is used to enhance the performance of the white-box model developed earlier through the optimisation of parameters using a genetic algorithm (GA) based approach. Stability analysis of the autonomous TRMS is carried out before designing any control paradigms for the system.

In the control design stage, an MPC method is proposed for constrained nonlinear systems, which is the improvement of the multistep Newton-type control strategy. The stability of the proposed state-feedback MPC is guaranteed using terminal equality constraints. Moreover, the formerly proposed MPC algorithm is combined with an unscented Kalman filter (UKF) to formulate an output-feedback MPC. An extended Kalman filter (EKF) based on a state-dependent model is also introduced, whose performance is found to be better compared to that of the UKF. Finally, a robust MPC is introduced and implemented on the TRMS based on a polytopic uncertainty that is cast into linear matrix inequalities (LMI).

Table of Contents

Title Page	1
Abstract	3
Table of Contents	4
List of Figures	8
List of Tables	17
Acronyms	18
Acknowledgments	20
Chapter 1 Introduction	21
1.1 Background	21
1.2 Recent Developments	23
1.3 Motivation, Aims and Objectives	31
1.4 Contributions.....	33
1.5 Outline of the Thesis	35
1.6 Publications.....	36
1.6.1 Journal Articles	36
1.6.2 Conference Articles.....	37
Chapter 2 Nonlinear White-box Modelling	38
2.1 Introduction.....	38
2.2 The TRMS.....	39
2.3 The Newtonian-based Modelling of the TRMS.....	42
2.3.1 Modelling of the Vertical Channel Interface Circuit	42
2.3.2 Modelling of the Horizontal Channel Interface Circuit	43
2.3.3 Modelling of the Main Motor	44
2.3.4 Modelling of the Tail Motor	47
2.3.5 Modelling of the Force due to Propellers Rotation.....	48
2.3.6 Modelling of the 1DOF Vertical TRMS	50
2.3.7 Modelling of the 1DOF Horizontal TRMS.....	55
2.3.8 Modelling of the 2DOF TRMS.....	58
2.4 Results of the Newtonian-based Modelling.....	62
2.4.1 Parameter Identification.....	64

2.4.2	Results of the Vertical 1DOF TRMS Model	66
2.4.3	Results of the Horizontal 1DOF TRMS Model	70
2.4.4	Results of the 2DOF TRMS Model (Newtonian-based).....	74
2.5	The Euler-Lagrangian-based Modelling of the TRMS	81
2.5.1	The TRMS Particles Definition	81
2.5.2	Energy Expression	85
2.5.3	Lagrange's Equations.....	88
2.5.4	Implementation of the Lagrange-based Model	89
2.6	Results of the 2DOF TRMS Model (Lagrangian-based).....	92
2.7	Comparative Study.....	98
2.8	Conclusions.....	98
Chapter 3	Nonlinear Black-box Modelling.....	100
3.1	Introduction.....	100
3.2	The Fundamentals of MLP Networks.....	101
3.2.1	A Neuron.....	101
3.2.2	The Multilayer Perceptron	103
3.2.3	Designing the Excitation Signal.....	103
3.2.4	Model Structure Selection.....	104
3.2.5	Training algorithm	105
3.3	Model Validation	106
3.4	Results.....	108
3.4.1	The 1DOF Horizontal Model.....	108
3.4.2	The 1DOF Vertical Model	121
3.4.3	The 2DOF Model	132
3.5	Conclusions.....	140
Chapter 4	Nonlinear Grey-box Modelling.....	142
4.1	Introduction.....	142
4.2	Analytical Equations of the TRMS.....	143
4.3	GA-based Parameters Identification	145
4.4	Results.....	147
4.5	Conclusions.....	152
Chapter 5	Stability Analysis of the TRMS.....	153
5.1	Introduction.....	153
5.2	Equilibrium Points	154
5.3	Local Stability	155
5.4	Phase Plane Analysis	155

5.5	Limit Cycles.....	162
5.6	Stability in the Sense of Lyapunov	162
5.6.1	The 1DOF Horizontal	163
5.6.2	The 1DOF Vertical.....	164
5.6.3	The 2DOF TRMS.....	166
5.7	Conclusions.....	168
Chapter 6	Nonlinear Model Predictive Control and Stability	170
6.1	Introduction.....	170
6.2	Newton-type MPC Method.....	172
6.2.1	Methodology	172
6.2.2	Objective Function	177
6.2.3	Constraints	179
6.3	Stability	180
6.4	The TRMS.....	181
6.5	Results.....	185
6.6	Conclusions.....	194
Chapter 7	Output-Feedback Model Predictive Control.....	195
7.1	Introduction.....	195
7.2	Observer-based MPC Strategy	196
7.2.1	Methodology	196
7.2.2	The UKF	197
7.2.3	The State-Dependent EKF	197
7.3	Simulation Results	199
7.4	Experimental Procedure	207
7.4.1	State Estimator	208
7.4.2	Output-Feedback Nonlinear MPC.....	213
7.5	Conclusions.....	219
Chapter 8	Robust Model Predictive Control.....	220
8.1	Introduction.....	220
8.2	Polytopic Model.....	221
8.3	Problem Formulation	223
8.4	Robust MPC.....	224
8.5	Results.....	226
8.6	Conclusions.....	230
Chapter 9	Conclusions and Future Work.....	231

9.1	Conclusions.....	231
9.2	Suggestions for Future Work.....	235
Bibliography	239
Appendix 1	The TRMS Parameters	253
Appendix 2	Equation Derive	259
Appendix 3	Statistical Characteristics of Signals	262
Appendix 4	Kalman Filter	267
A4.1	EKF.....	267
A4.2	UKF.....	268
Appendix 5	Source Code of the Programmes	272
A5.1	Simulation Programmes.....	272
A5.2	Experimental Programmes.....	291

List of Figures

Figure 2.1	The TRMS	40
Figure 2.2	The TRMS	40
Figure 2.3	Coupled MIMO model of the TRMS	41
Figure 2.4	The graphical relationship between U_v and V_v	43
Figure 2.5	The graphical relationship between U_h and V_h	44
Figure 2.6	The circuit diagram of the main DC motor with its propeller	45
Figure 2.7	Simulation of the main DC motor	46
Figure 2.8	The circuit diagram of the tail DC motor with its propeller	47
Figure 2.9	Simulation of the tail DC motor	48
Figure 2.10	Simulation of the vertical force	49
Figure 2.11	Simulation of the horizontal force	49
Figure 2.12	Gravity forces and propulsive force in the vertical plane.....	50
Figure 2.13	Friction torque profile of the vertical channel	53
Figure 2.14	The complete block diagram of the vertical 1DOF TRMS model	54
Figure 2.15	The inside of the 'nonlinear equations' block.....	54
Figure 2.16	Propulsive force in the horizontal plane	55
Figure 2.17	Friction torque profile of the horizontal channel.....	56
Figure 2.18	The complete block diagram of the horizontal 1DOF TRMS model...	58
Figure 2.19	The inside of the 'nonlinear equations' block.....	58
Figure 2.20	The complete block diagram of dynamic simulation of the TRMS	61
Figure 2.21	Inside the 'four-degree nonlinear equations' block	62
Figure 2.22	The front view of the TRMS and the pitch angle reference point.....	63
Figure 2.23	The Top view of the TRMS and the yaw angle reference point.....	63
Figure 2.24	Pitch angle with sine input (frequency 0.02 Hz and amplitude 1.5 V).	67
Figure 2.25	Pitch angle with sine input (frequency 0.05 Hz and amplitude 0.8 V).	68
Figure 2.26	Pitch angle with sine input (frequency 0.1 Hz and amplitude 1 V).....	68
Figure 2.27	Pitch angle with sine input (frequency 0.5 Hz and amplitude 2 V).....	68
Figure 2.28	Pitch angle with sine input (frequency 1 Hz and amplitude 2.5 V).....	69
Figure 2.29	Pitch angle with square input (frequency 0.02 Hz and amplitude 0.5 V)	69
Figure 2.30	Pitch angle with square input (frequency 0.05 Hz and amplitude 0.8 V)	69
Figure 2.31	Pitch angle with square input (frequency 0.1 Hz and amplitude 1 V)..	70
Figure 2.32	Pitch angle with square input (frequency 0.5 Hz and amplitude 2.5 V)	70
Figure 2.33	Yaw angle with sine input (frequency 0.05 Hz and amplitude 0.3 V).	71

Figure 2.34	Yaw angle with sine input (frequency 0.1 Hz and amplitude 0.4 V) ...	71
Figure 2.35	Yaw angle with sine input (frequency 0.2 Hz and amplitude 0.5 V) ...	71
Figure 2.36	Yaw angle with sine input (frequency 0.5 Hz and amplitude 2 V)	72
Figure 2.37	Yaw angle with square input (frequency 0.05 Hz and amplitude 0.2 V)	72
Figure 2.38	Yaw angle with square input (frequency 0.1 Hz and amplitude 0.3 V)	72
Figure 2.39	Yaw angle with square input (frequency 0.2 Hz and amplitude 0.4 V)	73
Figure 2.40	Yaw angle with square input (frequency 0.5 Hz and amplitude 1 V) ..	73
Figure 2.41	Yaw angle with square input (frequency 1 Hz and amplitude 2.5 V) ..	73
Figure 2.42	Pitch angle and PSD (case 1).....	75
Figure 2.43	Yaw angle and PSD (case 1)	75
Figure 2.44	Pitch angle and PSD (case 2).....	75
Figure 2.45	Yaw angle and PSD (case 2)	76
Figure 2.46	Pitch angle and PSD (case 3).....	76
Figure 2.47	Yaw angle and PSD (case 3)	76
Figure 2.48	Pitch angle and PSD (case 4).....	77
Figure 2.49	Yaw angle and PSD (case 4)	77
Figure 2.50	Pitch angle and PSD (case 5).....	77
Figure 2.51	Yaw angle and PSD (case 5)	78
Figure 2.52	Pitch angle and PSD (case 6).....	78
Figure 2.53	Yaw angle and PSD (case 6)	78
Figure 2.54	Pitch angle and PSD (case 7).....	79
Figure 2.55	Yaw angle and PSD (case 7)	79
Figure 2.56	Pitch angle and PSD (case 8).....	79
Figure 2.57	Yaw angle and PSD (case 8)	80
Figure 2.58	Pitch angle and PSD (case 9).....	80
Figure 2.59	Yaw angle and PSD (case 9)	80
Figure 2.60	The TRMS	83
Figure 2.61	Front view of the TRMS with $\alpha_h = 0$	83
Figure 2.62	Top view of the TRMS	84
Figure 2.63	The complete block diagram of the Lagrangian-based TRMS model..	91
Figure 2.64	The inside of the nonlinear Lagrangian equations block.....	91
Figure 2.65	Pitch angle and PSD (case 1).....	92
Figure 2.66	Yaw angle and PSD (case 1)	92
Figure 2.67	Pitch angle and PSD (case 2).....	93
Figure 2.68	Yaw angle and PSD (case 2)	93
Figure 2.69	Pitch angle and PSD (case 3).....	93
Figure 2.70	Yaw angle and PSD (case 3)	94

Figure 2.71	Pitch angle and PSD (case 4).....	94
Figure 2.72	Yaw angle and PSD (case 4)	94
Figure 2.73	Pitch angle and PSD (case 5).....	95
Figure 2.74	Yaw angle and PSD (case 5)	95
Figure 2.75	Pitch angle and PSD (case 6).....	95
Figure 2.76	Yaw angle and PSD (case 6)	96
Figure 2.77	Pitch angle and PSD (case 7).....	96
Figure 2.78	Yaw angle and PSD (case 7)	96
Figure 2.79	Pitch angle and PSD (case 8).....	97
Figure 2.80	Yaw angle and PSD (case 8)	97
Figure 2.81	Pitch angle and PSD (case 9).....	97
Figure 2.82	Yaw angle and PSD (case 9)	98
Figure 3.1	A neuron with a bias and R inputs	101
Figure 3.2	Some commonly used activation functions.....	102
Figure 3.3	A typical MLP network with R inputs and three layers.....	103
Figure 3.4	The NNARX model structure of a SISO system	105
Figure 3.5	The structure of the NN-based model in terms of the 1DOF horizontal dynamics	109
Figure 3.6	Model response with the gradient descent method.....	112
Figure 3.7	Correlation validation of the neural model with the GD training method	112
Figure 3.8	Model response with the GDM training method	112
Figure 3.9	Correlation validation of the model with the GDM training method ...	113
Figure 3.10	Model response with the GDA learning	113
Figure 3.11	Correlation validation of the model with the GDA training method..	113
Figure 3.12	Model response with the GDX training algorithm	114
Figure 3.13	Correlation validation of the model with the GDX learning method .	114
Figure 3.14	Model response with the resilient method.....	114
Figure 3.15	Correlation validation of the model with the resilient training method	115
Figure 3.16	Model response with the CGF method	115
Figure 3.17	Correlation validation of the model with the CGF training method...	115
Figure 3.18	Model response with the CGP method	116
Figure 3.19	Correlation validation of the model with the CGP training method...	116
Figure 3.20	Model response with the CGB method.....	116
Figure 3.21	Correlation validation of the model with the CGB training method ..	117
Figure 3.22	Model response with the SCG training method.....	117
Figure 3.23	Correlation validation of the model with the SCG training method...	117

Figure 3.24	Model response with the BFGS quasi-Newton method	118
Figure 3.25	Correlation validation of the model with the BFGS learning method	118
Figure 3.26	Model response with the one step secant method.....	118
Figure 3.27	Correlation validation of the model with the OSS training method ...	119
Figure 3.28	Model response with the LM method.....	119
Figure 3.29	Correlation validation of the model with the LM training algorithm.	119
Figure 3.30	Model response with the Bayesian regularisation method	120
Figure 3.31	Correlation validation of the model with the BR training method	120
Figure 3.32	The structure of the NN-based model in terms of its 1DOF vertical dynamics	123
Figure 3.33	Model response with the gradient descent method.....	123
Figure 3.34	Correlation validation of the model with the GD training method.....	123
Figure 3.35	Model response with the GDM method	124
Figure 3.36	Correlation validation of the model with the GDM training method .	124
Figure 3.37	Model response with the GDA learning algorithm	124
Figure 3.38	Correlation validation of the model with the GDA training method..	125
Figure 3.39	Model response with the GDX learning method	125
Figure 3.40	Correlation validation of the model with the GDX learning method .	125
Figure 3.41	Model response with the resilient method.....	126
Figure 3.42	Correlation validation of the model with the resilient training method	126
Figure 3.43	Model response with the CGF method.....	126
Figure 3.44	Correlation validation of the model with the CGF training method...	127
Figure 3.45	Model response with the CGP training method.....	127
Figure 3.46	Correlation validation of the model with the CGP training method...	127
Figure 3.47	Model response with the CGB method.....	128
Figure 3.48	Correlation validation of the model with the CGB training method ..	128
Figure 3.49	Model response with the SCG method.....	128
Figure 3.50	Correlation validation of the model with the SCG training method...	129
Figure 3.51	Model response with the BFGS quasi-Newton method	129
Figure 3.52	Correlation validation of the model with the BFGS learning method	129
Figure 3.53	Model response with the one step secant method.....	130
Figure 3.54	Correlation validation of the model with the OSS training method ...	130
Figure 3.55	Model response with the LM method.....	130
Figure 3.56	Correlation validation of the model with the LM training algorithm.	131
Figure 3.57	Model response with the Bayesian regularisation method	131
Figure 3.58	Correlation validation of the model with the BR training method	131
Figure 3.59	The structure of the NN-based model of the TRMS.....	133

Figure 3.60	Yaw angle response with the LM learning method.....	134
Figure 3.61	Pitch angle response with the LM learning method	135
Figure 3.62	Correlation validation between V_v and α_v of the LM method	135
Figure 3.63	Correlation validation between V_h and α_h of the LM method.....	135
Figure 3.64	Correlation validation between V_h and α_v of the LM method.....	135
Figure 3.65	Correlation validation between V_v and α_h of the LM method	136
Figure 3.66	Yaw angle response with the BFGS learning method.....	136
Figure 3.67	Pitch angle response with the BFGS learning method	136
Figure 3.68	Correlation validation between V_v and α_v of the BFGS method	136
Figure 3.69	Yaw angle response with the BR learning method.....	137
Figure 3.70	Pitch angle response with the BR learning method.....	137
Figure 3.71	Correlation validation between V_v and α_v of the BR method.....	137
Figure 3.72	Yaw angle response with the CGB learning method.....	137
Figure 3.73	Pitch angle response with the CGB learning method	138
Figure 3.74	Correlation validation between V_v and α_v of the CGB method.....	138
Figure 3.75	Yaw angle response with the RP learning method	138
Figure 3.76	Pitch angle response with the RP learning method	138
Figure 3.77	Correlation validation between V_v and α_v of the RP method	139
Figure 3.78	Yaw angle response with the SCG learning method	139
Figure 3.79	Pitch angle response with the SCG learning method.....	139
Figure 3.80	Correlation validation between V_v and α_v of the SCG method	139
Figure 4.1	The block diagram of the iterative identification procedure	146
Figure 4.2	The control and data flow of a GA	146
Figure 4.3	The yaw angle of the beam (case 1)	148
Figure 4.4	The pitch angle of the beam (case 1).....	148
Figure 4.5	The yaw angle of the beam (case 2)	149
Figure 4.6	The pitch angle of the beam (case 2).....	149
Figure 4.7	The yaw angle of the beam (case 3)	149
Figure 4.8	The pitch angle of the beam (case 3).....	149
Figure 4.9	The yaw angle of the beam (case 4)	149
Figure 4.10	The pitch angle of the beam (case 4).....	149
Figure 4.11	The yaw angle of the beam (case 5)	150
Figure 4.12	The pitch angle of the beam (case 5).....	150
Figure 4.13	The yaw angle of the beam (case 6)	150
Figure 4.14	The pitch angle of the beam (case 6).....	150
Figure 4.15	The yaw angle of the beam (case 7)	150
Figure 4.16	The pitch angle of the beam (case 7).....	150
Figure 4.17	The yaw angle of the beam (case 8)	151

Figure 4.18	The pitch angle of the beam (case 8).....	151
Figure 4.19	The yaw angle of the beam (case 9).....	151
Figure 4.20	The pitch angle of the beam (case 9).....	151
Figure 4.21	The yaw angle of the beam (case 10).....	151
Figure 4.22	The pitch angle of the beam (case 10).....	151
Figure 4.23	The yaw angle of the beam (case 11).....	152
Figure 4.24	The pitch angle of the beam (case 11).....	152
Figure 5.1	The block diagram of the 2DOF TRMS.....	156
Figure 5.2	The block diagram of the decoupled TRMS.....	156
Figure 5.3	The phase portrait of the main motor.....	159
Figure 5.4	The phase portrait of the main motor with a higher inductance.....	159
Figure 5.5	The phase portrait of the tail motor.....	160
Figure 5.6	The phase portrait of the tail motor with a higher inductance.....	160
Figure 5.7	The phase portrait of the vertical dynamics.....	160
Figure 5.8	The phase portrait of the horizontal dynamics.....	161
Figure 5.9	The 3D phase portrait of the horizontal dynamics.....	161
Figure 5.10	The 3D phase portrait of the vertical dynamics.....	161
Figure 5.11	The Lyapunov function and the sets N and N^*	165
Figure 6.1	The block diagram of a generic MPC.....	171
Figure 6.2	The MPC approach of a SISO plant at time k	174
Figure 6.3	The TRMS.....	182
Figure 6.4	Block diagram of the proposed Newton-type MPC approach.....	186
Figure 6.5	Flowchart of the proposed Newton-type MPC approach.....	187
Figure 6.6	Square wave response of the horizontal angle (case 1).....	189
Figure 6.7	Square wave response of the vertical angle (case 1).....	189
Figure 6.8	Tail motor manipulated signal, U_h , (case 1).....	190
Figure 6.9	Main motor manipulated signal, U_v , (case 1).....	190
Figure 6.10	Sine wave response of the horizontal angle (case 2).....	190
Figure 6.11	Sine wave response of the vertical angle (case 2).....	190
Figure 6.12	Tail motor manipulated signal, U_h , (case 2).....	190
Figure 6.13	Main motor manipulated signal, U_v , (case 2).....	190
Figure 6.14	Square wave response of the horizontal angle (case 3).....	191
Figure 6.15	Square wave response of the vertical angle (case 3).....	191
Figure 6.16	Tail motor manipulated signal, U_h , (case 3).....	191
Figure 6.17	Main motor manipulated signal, U_v , (case 3).....	191
Figure 6.18	Square wave response of the horizontal angle (case 4).....	191
Figure 6.19	Square wave response of the vertical angle (case 4).....	191
Figure 6.20	Tail motor manipulated signal, U_h , (case 4).....	192

Figure 6.21	Main motor manipulated signal, U_v , (case 4)	192
Figure 6.22	Sine wave response of the horizontal angle (case 5)	192
Figure 6.23	Sine wave response of the vertical angle (case 5)	192
Figure 6.24	Tail motor manipulated signal, U_h , (case 5)	192
Figure 6.25	Main motor manipulated signal, U_v , (case 5)	192
Figure 6.26	Sawtooth wave response of the horizontal angle (case 6)	193
Figure 6.27	Sawtooth wave response of the vertical angle (case 6)	193
Figure 6.28	Tail motor manipulated signal, U_h , (case 6)	193
Figure 6.29	Main motor manipulated signal, U_v , (case 6)	193
Figure 6.30	Two successive results of internal model output.....	193
Figure 6.31	The objective function values related to the case of Figures 6.6 and 6.7	194
Figure 7.1	Flowchart of the unscented Kalman observer procedure	198
Figure 7.2	Block diagram of the output-feedback Newton-type MPC approach ..	200
Figure 7.3	Flowchart of the output-feedback Newton-type MPC approach.....	201
Figure 7.4	Square wave response of the horizontal angle (case 1)	202
Figure 7.5	Square wave response of the vertical angle (case 1)	202
Figure 7.6	Tail motor manipulated signal, U_h , (case 1)	203
Figure 7.7	Main motor manipulated signal, U_v , (case 1)	203
Figure 7.8	Square wave response of the horizontal angle (case 2)	203
Figure 7.9	Square wave response of the vertical angle (case 2)	203
Figure 7.10	Tail motor manipulated signal, U_h , (case 2)	203
Figure 7.11	Main motor manipulated signal, U_v , (case 2)	203
Figure 7.12	Square wave response of the horizontal angle (case 3)	204
Figure 7.13	Square wave response of the vertical angle (case 3)	204
Figure 7.14	Tail motor manipulated signal, U_h , (case 3)	204
Figure 7.15	Main motor manipulated signal, U_v , (case 3)	204
Figure 7.16	Sine wave response of the horizontal angle (case 4)	204
Figure 7.17	Sine wave response of the vertical angle (case 4)	204
Figure 7.18	Tail motor manipulated signal, U_h , (case 4)	205
Figure 7.19	Main motor manipulated signal, U_v , (case 4)	205
Figure 7.20	Sine wave response of the horizontal angle (case 5)	205
Figure 7.21	Sine wave response of the vertical angle (case 5)	205
Figure 7.22	Tail motor manipulated signal, U_h , (case 5)	205
Figure 7.23	Main motor manipulated signal, U_v , (case 5)	205
Figure 7.24	Sawtooth wave response of the horizontal angle (case 6)	206
Figure 7.25	Sawtooth wave response of the vertical angle (case 6)	206
Figure 7.26	Tail motor manipulated signal, U_h , (case 6)	206

Figure 7.27	Main motor manipulated signal, U_v , (case 6)	206
Figure 7.28	The performance of the UKF (S_v)	206
Figure 7.29	The performance of the UKF (S_h)	206
Figure 7.30	The overall block diagram of the UKF and the EKF test	208
Figure 7.31	Inside the 'EKF' block	209
Figure 7.32	Measured and EKF-based estimated yaw angle (case 1)	210
Figure 7.33	Measured and UKF-based estimated yaw angle (case 1)	210
Figure 7.34	Measured and EKF-based estimated pitch angle (case 1)	210
Figure 7.35	Measured and UKF-based estimated pitch angle (case 1)	210
Figure 7.36	Measured and EKF-based estimated ω_h (case 1)	210
Figure 7.37	Measured and UKF-based estimated ω_h (case 1)	210
Figure 7.38	Measured and EKF-based estimated ω_v (case 1)	211
Figure 7.39	Measured and UKF-based estimated ω_v (case 1)	211
Figure 7.40	EKF and UKF-based estimated S_h (case 1)	211
Figure 7.41	EKF and UKF-based estimated S_v (case 1)	211
Figure 7.42	Measured and EKF-based estimated yaw angle (case 2)	211
Figure 7.43	Measured and UKF-based estimated yaw angle (case 2)	211
Figure 7.44	Measured and EKF-based estimated pitch angle (case 2)	212
Figure 7.45	Measured and UKF-based estimated pitch angle (case 2)	212
Figure 7.46	Measured and EKF-based estimated ω_h (case 2)	212
Figure 7.47	Measured and UKF-based estimated ω_h (case 2)	212
Figure 7.48	Measured and EKF-based estimated ω_v (case 2)	212
Figure 7.49	Measured and UKF-based estimated ω_v (case 2)	212
Figure 7.50	EKF and UKF-based estimated S_h (case 2)	213
Figure 7.51	EKF and UKF-based estimated S_v (case 2)	213
Figure 7.52	Sawtooth wave response of the horizontal angle (case 1)	214
Figure 7.53	Sawtooth wave response of the vertical angle (case 1)	214
Figure 7.54	Tail motor manipulated signal, U_h , (case 1)	214
Figure 7.55	Main motor manipulated signal, U_v , (case 1)	214
Figure 7.56	Sine wave response of the horizontal angle (case 2)	215
Figure 7.57	Sine wave response of the vertical angle (case 2)	215
Figure 7.58	Tail motor manipulated signal, U_h , (case 2)	215
Figure 7.59	Main motor manipulated signal, U_v , (case 2)	215
Figure 7.60	Sine wave response of the horizontal angle (case 3)	215
Figure 7.61	Sine wave response of the vertical angle (case 3)	215
Figure 7.62	Tail motor manipulated signal, U_h , (case 3)	216
Figure 7.63	Main motor manipulated signal, U_v , (case 3)	216
Figure 7.64	Square wave response of the horizontal angle (case 4)	216

Figure 7.65	Square wave response of the vertical angle (case 4)	216
Figure 7.66	Tail motor manipulated signal, U_h , (case 4)	216
Figure 7.67	Main motor manipulated signal, U_v , (case 4)	216
Figure 7.68	Square wave response of the horizontal angle (case 5)	217
Figure 7.69	Square wave response of the vertical angle (case 5)	217
Figure 7.70	Tail motor manipulated signal, U_h , (case 5)	217
Figure 7.71	Main motor manipulated signal, U_v , (case 5)	217
Figure 7.72	Step response of the horizontal angle (case 6)	217
Figure 7.73	Step response of the vertical angle (case 6).....	217
Figure 7.74	Tail motor manipulated signal, U_h , (case 6)	218
Figure 7.75	Main motor manipulated signal, U_v , (case 6)	218
Figure 7.76	Step response of the horizontal angle with external force (case 7)	218
Figure 7.77	Step response of the vertical angle (case 7).....	218
Figure 7.78	Tail motor manipulated signal, U_h , (case 7)	218
Figure 7.79	Main motor manipulated signal, U_v , (case 7)	218
Figure 8.1	The graphical representation of a polytopic uncertainty [85].....	222
Figure 8.2	The sub-polytopic uncertainty	226
Figure 8.3	Yaw angle response with initial value of 1.2 rad (case 1).....	228
Figure 8.4	Pitch angle with initial value of 0.65 rad (case 1)	228
Figure 8.5	Tail rotor applied voltage(case 1)	228
Figure 8.6	Main rotor applied voltage (case 1).....	228
Figure 8.7	Yaw angle response with initial value of -0.65 rad (case 2).....	228
Figure 8.8	Pitch angle with initial value of -0.35 rad (case 2).....	228
Figure 8.9	Tail rotor applied voltage (case 2).....	229
Figure 8.10	Main rotor applied voltage (case 2).....	229
Figure 8.11	Yaw angle response with initial value of -1.27 rad (case 3).....	229
Figure 8.12	Pitch angle with initial value of -0.47 rad (case 3)	229
Figure 8.13	Tail rotor applied voltage (case 3).....	229
Figure 8.14	Main rotor applied voltage (case 3).....	229
Figure 8.15	Yaw angle response with initial value of 0.64 rad (case 4).....	230
Figure 8.16	Pitch angle with initial value of 0.43 rad (case 4)	230
Figure 8.17	Tail rotor applied voltage (case 4).....	230
Figure 8.18	Main rotor applied voltage (case 4).....	230
Figure 9.1	A wavelet-based network with p inputs and a single output	236

List of Tables

Table 2.1	The main differences between a helicopter and a TRMS	42
Table 2.2	The constraints of the yaw and pitch angles	64
Table 2.3	The characteristics of the input signals	74
Table 2.4	Comparative study of the Lagrangian and Newtonian models	98
Table 3.1	Comparison of the training algorithms	121
Table 3.2	Performance comparison of the training algorithms.....	133
Table 3.3	Performance comparison of the training algorithms.....	140
Table 4.1	Comparative study of the white- and grey-box models	152
Table 6.1	The parameters of the proposed MPC	189

Acronyms

AI	Artificial Intelligence
ANN	Artificial Neural Network
ARIMA	AutoRegressive Integrated Moving Average
BFGS	Broyden Fletcher Goldfarb Shanno
BR	Bayesian Regularisation
CGB	Conjugate Gradient with Beale-Powell restart
CGF	Conjugate Gradient with Fletcher-Reeves
CGP	Conjugate Gradient with Polak-Ribiere updates
DC	Direct Current
DMC	Dynamics Matrix Control
DOF	Degree Of Freedom
EKF	Extended Kalman Filter
GA	Genetic Algorithm
GD	Gradient Descent
GDA	Gradient Descent with Adaptive learning rate
GDM	Gradient Descent with Momentum
GDX	Gradient Descent with Adaptive learning rate and Momentum
LM	Levenberg-Marquardt
LMI	Linear Matrix Inequalities
LQG	Linear Quadratic Gaussian
MIMO	Multi-Input Multi-Output
MLP	MultiLayer Perceptron
MPC	Model Predictive Control
MSE	Mean Squared Error
NMPC	Nonlinear Model Predictive Control
NN	Neural Network

NNARMAX	Neural Network AutoRegressive Moving Average with eXternal input
NNARX	Neural Network AutoRegressive eXternal input
NNFIR	Neural Network Finite Impulse Response
NNOE	Neural Network Output Error
NNSSIF	Neural Network State-Space Innovation Form
ODE	Ordinary Differential Equation
OSS	One Step Secant
PDP	Parallel Distributed Processing
PRBS	Pseudo Random Binary Sequence
PSD	Power Spectral Density
RB	Resilient Backpropagation
RBF	Radial Basis Function
RHC	Receding Horizon Control
SCG	Scaled Conjugate Gradient
SISO	Single-Input Single-Output
TRMS	Twin Rotor MIMO System
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter

Acknowledgments

At Queen Mary, University of London, I have had the opportunity to meet some wonderful people.

First of all, I would like to especially thank my first supervisor Dr M Hasan Shaheed for his valuable guidance, patience, enthusiasm and his critical comments. He provided an exciting, stimulating and open research environment at the department, which encouraged me to pursue my research eagerly. I would also like to thank my second supervisor Dr H J C Huijberts for his support, encouragement, and his pertinent and constructive comments. It has been a great honour for me to work with them and learn from them.

I would like to take this opportunity to thank Dr Ali Akbar Safavi for all his valuable assistance and constructive feedback during my academic life.

I must also acknowledge the unconditional love and support provided by my wife. Without her support, it would not be possible to start, continue and accomplish my study abroad. Her presence in both pleasant and unpleasant conditions of my life made it possible to improve my self confidence.

CHAPTER 1

Introduction

1.1 Background

The twin rotor MIMO system (TRMS) is a laboratory platform designed for control experiments by Feedback Instruments Ltd [1]. There are some resemblances between a TRMS and a helicopter. For instance, like a helicopter it has a strong cross-coupling between the main and tail rotors. It is a highly nonlinear and complex system. Some of its states and outputs are inaccessible for measurements. Thus, from the control perspective, it can be perceived as a challenging problem. The control objective is to move the beam of the TRMS as quickly and accurately as possible to the desired attitudes in terms of both the pitch and azimuth angles.

To develop a control paradigm on the basis of a model-based control approach, it is essential to have an accurate model of the plant. Depending on the background knowledge of a system, the identification methods can be categorised into three different classes. If the identification is based only on the input-output measured data, the identification process is called black-box or empirical modelling. On the other hand, white-box or analytical modelling refers to the model development techniques based on the physical characteristics of a process. A hybrid method comprising white- and black-box modelling is called a grey-box identification in which a system is roughly modelled according to its governing physical equations and subsequently improved using a black-box identification approach based on a set of input-output data [2]. Each of the aforementioned classes of modelling can be categorised as either linear or nonlinear.

The analytical modelling of an aerodynamic system can be achieved using the Newtonian, Lagrangian or any other related approaches. The straightforward application of Newton's laws to model simple systems has been supplemented by a useful and powerful general approach for finding the equations of motion for most types of dynamical systems, introduced by the French mathematician Joseph Louis

Lagrange [3]. In this investigation, the TRMS is analytically modelled using both the Newtonian and Lagrangian-based approaches.

Among the black-box modelling approaches, ANNs look very promising. An ANN simulates a human brain in a really simple form and therefore, it can obtain some of the specific features of a brain such as learning, adaptation, association, generalisation, and in some cases, even forgetfulness. A neuron is the fundamental element of ANNs, and can be considered a processor. One of the most popular types of neuron receives weighted inputs, sums them up and then passes the result into a linear or nonlinear activation function to formulate the output. In some cases, the activation functions possess some specific characteristics such as monotonicity and differentiability. Model identification of nonlinear systems is one of the most efficient applications of ANNs.

Receding horizon control (RHC) or MPC was introduced in the late seventies and early eighties [4]-[8] and has been developed considerably since then. It is a class of control approach that uses an explicit model of the plant and tries to evaluate the manipulated variables through an optimisation method. The optimisation procedure minimises an objective function, subject to a given set of constraints, to obtain the optimum values of manipulated variables that makes the controlled variables follow the reference trajectories as closely as possible with the minimum possible control effort. The first element of the calculated input signals is sent to the plant as input and others are rejected; this process is well-known as the receding horizon. Therefore, MPC is not a single control approach and it can be considered to be a class of control strategies with the following common features [9]:

- All MPC methods need a plant-model to predict the process outputs during a specific time, called the prediction horizon.
- They have an objective function that needs to be minimised in order to find a set of optimal manipulated variables during the control horizon.
- At each step, only the first set of manipulated variables is sent to the plant and others are discarded.

Despite most of the practical processes being nonlinear, the majority of the MPC techniques implemented on the industrial processes are based on linear models. One of the rationales for this is that a linear model is easy and fast to develop, compared to a nonlinear one. Another reason refers to the stability and, more generally,

robustness problems that are more complicated in nonlinear cases. Some of the nonlinear models and/or constraints lead to non-convex nonlinear optimisation problems that are relatively more complex to solve. Last but not least, in some cases, a linear model provides satisfactory performance.

Stability of any control system is a minimum requirement. The Lyapunov theory of stability is a fundamental tool for stability of linear and nonlinear systems. Although the stability analysis of a linear system using the Lyapunov criteria is straightforward, in the case of nonlinear systems such as MPC approaches, even linear MPC, stability proof is a challenging issue. Therefore, further action should be taken into consideration to guarantee the stability of such systems. For instance, since the closed-loop stability of a general MPC with a finite horizon is not guaranteed, several schemes have so far been proposed to address the issue. Terminal state equality constraints, terminal cost, terminal inequality constraints (dual mode), terminal contractive constraints, infinite horizon control, quasi-infinite horizon control, and control Lyapunov function approaches are some of the most popular methods used to guarantee the closed-loop stability of model predictive controllers.

Robust stability, i.e. stability in the face of model mismatch, is a more general form of stability. Stability analysis, in general, assumes the model is absolutely accurate and there is no uncertainty in the model. However, in practical situations it is almost impossible to model a plant without any approximation. Therefore, robust stability is vital when dealing with practical systems. Generally, providing the robust stability of a nonlinear system is much more challenging than providing its nominal stability.

Due to the aforementioned issues related to nonlinear model predictive control (NMPC), the application of this method in practical situations is still very limited. But its potential is really great since nonlinear MPC is essential in those areas where a process is highly nonlinear and frequently requires change in operation regions [9]. These challenging issues should be seriously addressed for future applications of control engineering. In this investigation, a number of nonlinear MPC paradigms are designed and then implemented on the TRMS.

1.2 Recent Developments

Some investigations have been reported to address the modelling and control of a TRMS using various analytical and artificial intelligence (AI)-based approaches. For

instance, nonlinear modelling of a 1 degree of freedom (1DOF) TRMS using radial basis function (RBF) networks has been addressed in [10], which presents nonlinear system identification method for modelling air vehicles of complex configuration. Dynamic modelling and optimal control of a TRMS in terms of its 1DOF dynamics has been presented in [11]. In that approach, the proposed model is employed in the design of a feedback linear quadratic Gaussian (LQG) compensator. Performance analysis of 4 types of conjugate gradient algorithms in the nonlinear dynamic modelling of a 1DOF TRMS using feedforward NNs has been reported in [12]. Dynamic modelling of a TRMS in hovering position using NN-based and parametric linear approaches has been presented in [13]. Mat Darus *et al.* [14] have proposed a parametric modelling of a TRMS using genetic algorithms (GAs); in that, the global search technique of GAs is used to identify the parameters of the TRMS based on one-step-ahead prediction.

Modelling of unmanned aerial vehicles (UAVs) and helicopters has widely been reported in the literature. For instance, dynamic modelling and feedback control of a side-by-side rotor tandem helicopter has been proposed in [15]. The investigation deals with the development of a dynamic model and design of a feedback controller for reducing torque oscillations of a side-by-side rotor tandem helicopter, which is a MIMO system. Wu *et al.* [16] have presented a nonlinear flight controller design for a type of helicopter by a trajectory linearisation method. In this paper, the authors have presented the development of a nonlinear dynamic model of a 3DOF flight control experiment apparatus, which is a helicopter with three propellers driven by direct current (DC) motors. Exact linearisation and non-interacting control of a 4 rotors helicopter via dynamic feedback has been reported in [17]. The authors have designed a feedback control in order to transform the nonlinear system to a linear and controllable one that is known as the exact linearisation problem. They have reduced the system from an input-output point of view to an aggregate of independent single-input single-output (SISO) channels that is known as non-interacting control problem or input-output decoupling problem. Hover control via a Lyapunov-based approach for an autonomous model helicopter has been proposed by Mahony *et al.* [18]. In this article, a dynamic model based on physical insight has been presented for a reduced scale autonomous helicopter using Newton's equations. Real-time stabilisation and tracking of a four-rotor mini rotorcraft have been presented in [19]. The authors have developed a control approach and implemented it on a mini rotorcraft; they have obtained the dynamic model of the four-rotor rotorcraft via a Lagrange approach. Nonlinear dynamics and control of a wind-milling gyroplane rotor have been proposed by Somov *et al.* [20]. In this investigation, the mathematical models of

auto-rotation and flapping have been extracted and used for nonlinear dynamic modelling. The development of a nonlinear model and a nonlinear control strategy for a scale model helicopter has been presented in [21]. The nonlinear 3DOF model of the helicopter is based on the Lagrangian equations. Cerro *et al.* [22] have proposed the identification of a small unmanned helicopter model using GAs. In this article, a hybrid (analytical and empirical) model of the helicopter, its stability and parameter identification using GAs have been presented. Mathematical modelling of a UAV during take off with nose-wheel off the ground has been discussed in [23]. The mathematical model is formulated according to the Newton-Euler method. Jun *et al.* [24] have presented the state estimation of an autonomous helicopter using the Kalman filtering.

In this research, the TRMS has analytically been modelled using both the Newtonian and Lagrangian approaches. Although both methods are supposed to produce exactly the same results, due to different assumptions in the case of each method, the results are slightly different. Moreover, the system has empirically been modelled using NN-based approaches. It is clear that empirical modelling is valid only in the range of input data used for identification. Also, black-box modelling techniques suffer from the loss of the process structure. One of the advantages of black-box modelling is easiness of finding an acceptable model in comparison with white-box modelling. It has resulted in less error compared to the analytical model adopted in the case of the TRMS. It is noted that in all previous works on the NN-based modelling of a TRMS [10], [12], [13] the authors have modelled the system in terms of 1DOF dynamics of the system. However, in the present research both 1DOF and 2DOF dynamics models have been developed.

Applications of grey-box modelling have widely been reported in the literature. For example, a systematic improvement method of stochastic grey-box modelling has been proposed in [25]. The features of this approach allow to identify the model deficiencies and to reveal their structural origin based on interplay between stochastic differential equation modelling, statistical tests and non-parametric modelling. Non-measurable variable estimation using grey-box modelling has been presented in [26] which its novelty is to estimate unknown variables using experiments, *a-priori* information and physical insights of a given process. Tan *et al.* [27] have proposed an evolutionary grey-box modelling technique based on a hybrid approach of genetic evolution and Boltzmann learning. The technique uses *a-priori* knowledge of the white-box global structure of a physical system, whilst incorporating accurate black-box for non-measurable local nonlinearities. Grey-box

identification of three classes of block-oriented models, i.e. Hammerstein, Wiener models, and the feedback block-oriented models, has been introduced by Pearson *et al.* [28]. In this investigation, the approach is restricted to those processes with nonlinear steady state characteristics that are known *a-priori* and with no steady state multiplicity. A grey-box modelling mechanism based on NNs for chemical processes has been presented in [29]. Here, the process dynamics has initially been modelled using an analytically derived model and an NN has then been used to compensate for the model mismatch. Oussar *et al.* [30] have shown that dynamic semi-physical neural modelling can be a powerful strategy for the design of models in accordance with the legibility of knowledge-based models and the flexibility of training from experimental data. Hybrid grey-box modelling of a pickling process has been proposed in [31]. The modelling procedure follows a structured approach, which includes basic modelling, data acquisition, model calibration, hybrid expanded modelling, stochastic modelling and model assessment. An approach to model identification of the autoregressive integrated moving average (ARIMA) family using GAs can be found in [32], where a GA-based model identification has been proposed to overcome the problem of local optima; the method is suitable for any ARIMA model. Chang [33] has proposed system identification and control based on GAs for a class of nonlinear systems to identify the unknown parameters of a system with previously known structure. A flexible scheme for parameter estimation in stochastic grey-box models has been presented in [34]. The estimation scheme is based on the EKF and features maximum likelihood as well as maximum *a-posteriori* estimation on multiple independent data sets. Sohlberg [35] has proposed a rational modelling procedure of a pilot heating process using a grey-box modelling approach. A simplified nonlinear continuous model has been formed and its unknown parameters have been estimated using measured data extracted from the process. Finally, the model has been utilised for designing a model predictive controller.

MPC can be defined as a class of control algorithms that utilises an explicit process model to predict the future response of the plant [36]. In recent years MPC, as one of the modern computer optimisation control techniques, has achieved a significant level of acceptability and great development in control theory and applications. Linear MPC has become popular with the publication of some papers on model predictive heuristic control [4], [5] and dynamics matrix control (DMC) [6]-[8]. As evident from the literature, MPC has never been applied to a TRMS; however, some researchers have exploited MPC to solve some aerospace and robot control problems. Poignet and Gautier [37] have proposed nonlinear MPC for a robotic

manipulator and implemented it with a linearised and optimised dynamic model. Wroblewski [38] has applied MPC to control a 6DOF manipulator in an attempt to reduce the error at the end of the prediction horizon rather than to find the optimal solution in order to decrease the computational burden. A nonlinear MPC approach has also been reported to control multiple autonomous helicopters in a complex environment; the approach combines stabilisation of vehicle dynamics and decentralised trajectory generation by incorporating a potential function [39]. Modelling, control and decision making for flying robots have been discussed in [40], where the authors have designed a tracking control layer using nonlinear MPC and integrated it with a trajectory generation for logistical action planning. The application of an MPC paradigm for a parallel mechanism has been presented in [41]. In that investigation, a predictive function control strategy based on a simplified dynamic model has been implemented. Kawati *et al.* [42] have addressed a vision-based robot motion control via a nonlinear RHC approach. There, the authors have proposed a stabilising RHC scheme which is based on a control Lyapunov function and a corresponding feedback control law. Gu *et al.* [43] have used a receding horizon controller for regulating a non-holonomic mobile robot. In that study, the stability of RHC has been guaranteed by adding a terminal-state penalty to the cost function and a terminal-state region to the optimisation constraints. Localisation and navigation of a mobile robot using an NMPC technique has been reported in [44]. Vanek *et al.* [45] have proposed a practical approach for real-time trajectory tracking of UAV formation. In that case, an MPC outer loop controller specifies the command signals for the H_∞ locally controlled dynamics.

Note that in the case of a severe nonlinear system, a single linear model cannot provide acceptable results in all operating regions. In other words, a highly nonlinear system cannot be modelled linearly to be adequate in all operating regions, unless the process always works in the neighbourhood of the point of interest. In the case of the TRMS, a linear MPC is insufficient to obtain satisfactory performance in all operating regions. As mentioned before, depending on the degrees of nonlinearity of a system, it is sometimes possible to find a linear model to be valid in some specific operating points, but generally, for a highly nonlinear system such as the TRMS, it is difficult to find a linear model to be adequate in all operating regions. Note that a nonlinear plant can be modelled using multiple linear modelling [46]-[51] or adaptive linear modelling [52]-[55] or linear time-varying modelling [56] approaches. In the case of multiple-modelling, the operating region of a nonlinear system is divided into several sub-regions and for each of the sub-regions, a linear model is developed. Therefore, according to the current operating point of the

system, the appropriate linear model is used to predict the output of the process. For instance, Aufderheide *et al.* [46], [47] have proposed an MPC based on a multiple model strategy. In that research, a bank of first-order plus deadtime models has been designed according to the range of gains, dominant time constants and time delays. A multiple model adaptive control strategy for multivariable dynamic matrix control has been proposed in [48]. The proposed method combines the outputs of multiple linear DMC controllers, each with its own step response model. Wan *et al.* [49] have presented a scheduled stabilising output-feedback MPC for constrained nonlinear systems. They have designed a set of local output-feedback predictive controllers with their estimated regions of stability covering the desired operating region, and implemented them as a single scheduled MPC which switches online between the set of local controllers. A multiple model adaptive approach has been proposed in an MPC framework to account for variability and handle the constraints [50], [51]. Note that in relatively high order systems with multiple inputs, it is a cumbersome task to find these linear models to cover all operating regions. Also, multiple-model MPC uses a linear model during the prediction horizon that cannot be adequate in highly nonlinear systems with a large prediction horizon.

An adaptive linear modelling method updates the linear model according to measurement data or linearisation of a nonlinear plant-model. For example, adaptive linear MPC has been presented to update linear model online based on measurement data to handle model uncertainties [52], [53]. Zhang *et al.* [54] have proposed a method that uses pseudo-partial derivative to dynamically linearise a nonlinear system at each step of predictive functional control in order to have the benefits of linear quadratic optimisation methods. In [55], a combination of parameter estimation and MPC algorithm has been used for max-plus linear discrete-event systems. Note that the proposed method covers only a specific small class of systems. Li and Biegler [57] have proposed multistep Newton-type control strategies for constrained nonlinear processes in which the nonlinear model is linearised around a nominal trajectory and a quadratic problem is solved over the horizon. The extension of the mentioned approach can be found in [58] that put the performance index into an augmented form and performed some modifications such as extending the output prediction horizon to infinity. O'Brien *et al.* [56] have used a state-dependent model to design a model predictive controller for a wastewater system. Based on the state-dependent state-space model, a linear time-varying model has been constructed using the future trajectory determined by the predictive controller. A Newton-type scheme for the approximate on-line solution of optimisation problems for optimal feedback control has been proposed in [59]. Fast reaction to

disturbances can be achieved by delivering approximations of the exact optimal feedback control. Diehl *et al.* [60] have investigated a Newton-type method for online optimisation in NMPC, the so-called real-time iteration scheme. In their investigation, only one Newton-type iteration has been performed per sampling time to have a fast online optimisation algorithm.

Note that the closed-loop stability of a generic MPC with finite-horizon cost function cannot be guaranteed [61] and therefore, further investigation needs to be carried out, e.g. modifying the cost function and/or constraints. Terminal state equality constraints technique has widely been proposed to stabilise linear/nonlinear and discrete/continuous MPC systems [62]-[65]. By adding a terminal cost to the objective function, Bitmead *et al.* [61] have proposed a stable predictive control for unconstrained linear systems. Rawlings *et al.* [66] have proposed an infinite horizon controller to guarantee the stability of both stable and unstable linear plants. A quasi-infinite horizon scheme has been proposed to stabilise stable and unstable nonlinear model predictive controllers with input constraints [67]. Scokaert *et al.* [68] have investigated conditions under which suboptimal model predictive controllers are stabilised. Robustness properties of nonlinear receding horizon controllers with terminal equality constraints have been investigated with respect to gain and additive perturbations [69]. Michalska *et al.* [70] have proposed a robust dual mode receding horizon controller for a wide class of nonlinear systems with state and control constraints and model error. A complete survey on both linear and nonlinear MPC focusing on sufficient conditions to guarantee stability and robustness can be found in [71]. Comprehensive review on stability and robustness of nonlinear MPC has been addressed in [72]. Jadbabaie *et al.* [73] have proposed a stable nonlinear MPC using unconstrained finite horizon optimal control. The method approximates the tail of the infinite horizon cost-to-go using an appropriate control Lyapunov function.

In this thesis, it is initially assumed that all state variables of the system are accessible, and therefore state-feedback MPC is taken into consideration. Moreover, an output-feedback nonlinear MPC is obtained using a combination of state-feedback nonlinear MPC and a state observer [74]-[77]. This will be discussed in Chapter 6. Stability, robustness and performance of output-feedback nonlinear MPC have been investigated in [74]; however, here the stability of state-feedback nonlinear MPC is taken into account. Kothare and Morari [77] have used an additional state constraint, called a contractive constraint, to prove strong nominal stability properties of MPC for constrained nonlinear systems.

So far, several paradigms have been developed in the field of robust control. Comprehensive surveys on robust MPC can be found in [71], [78] and [79]. An investigation on inherent robustness of MPC [71], in which the uncertainty is ignored, has been carried out by De Nicolao *et al.* [69] and Magni *et al.* [80] in the case of discrete and continuous-time frameworks respectively. Zero terminal state constraints have been used to study the robustness of nonlinear receding-horizon controllers with respect to gain and additive perturbations [69]. An inverse optimality result has been derived using nonlinear analogue of the fake Riccati equation for receding-horizon control scheme [80]. Open-loop min-max MPC was first introduced by Campo *et al.* [81] where the min-max strategy was originally proposed by Witsenhausen [82]. In this approach, the worst-case scenario over a set of uncertain plants is minimised to find the optimum sequence of manipulated variables. However, it has been demonstrated using a counterexample that robust stability cannot be guaranteed by the min-max paradigm alone [78], [83]. As iterated in the literature, the open-loop min-max is really conservative [78], [84], [85] and therefore a feedback version of MPC has been introduced in various forms [85]-[90]. Kothare *et al.* [85] have proposed an interesting technique in which, at each time step, a state-feedback control law minimises a worst-case infinite horizon objective function, subject to constraints. This approach has been improved by Cuzzola *et al.* [86] using several Lyapunov functions based on [91] each one corresponds to different vertex of uncertainty's polytope instead of a single Lyapunov function used in [85]. Mao [87] has pointed out and amended the mistake in [86]. A further improvement has been obtained based on two previous papers [85], [86] by adding N free control moves before the linear feedback law [88]. The concept of an asymptotically stable invariant ellipsoid has been used to develop a robust constrained MPC paradigm. The approach gives a sequence of explicit control laws corresponding to a sequence of asymptotically stable invariant ellipsoids constructed off-line, one within another in state-space [89]. Scokaert and Mayne [90] have outlined the details of min-max MPC formulation that introduce the notion of presenting feedback in the control implementation. Wang and Rawlings [92] have addressed a robust MPC method that guarantees stability and offset-free set point tracking in the presence of model uncertainty. The robust regulator uses a tree trajectory to forecast the time-varying model uncertainty.

Robust output-feedback MPC has recently been reported in the literature [84], [93], [94]. Wan and Kothare [93] have developed an off-line robust output-feedback MPC; first, an off-line robust constrained state-feedback MPC and a state estimator are independently designed, and then the robust stabilisability of the combined controller

and estimator is analysed. If the robust stability criterion is not satisfied, the design of the controller and estimator are iteratively repeated by specifying new design parameters to achieve a satisfactory result. A robust output-feedback MPC of constrained, linear, discrete-time systems in the face of bounded state and output disturbances has been proposed by Mayne *et al.* [84]. The approach consists of a stable Luenberger state estimator, where its error is bounded by an invariant set, and a robustly stabilising, tube-based, MPC. Ding *et al.* [94] have presented a synthesis approach for output-feedback robust MPC for systems with polytopic description, bounded state disturbance and measurement noise. A sequence of output-feedback laws is calculated off-line based on state estimators by solving LMI optimisation problems. At each time step, a suitable output-feedback law is chosen on-line from this sequence. Note that most of these robust output-feedback MPC approaches have difficulties in real time situation.

In this investigation, a nonlinear MPC is designed and implemented on the TRMS. The main challenging feature of this study is to find a nonlinear MPC method that can be implemented on the TRMS and also to provide the stability and more generally robustness of the closed-loop system.

1.3 Motivation, Aims and Objectives

Control of aerodynamic systems has been an area of interest ever since I was introduced to this subject. An additional motivation to carry out this investigation stems from the prospective use of AI-based techniques like GAs and ANNs for system modelling and control purposes. The control challenges of a highly nonlinear system like the TRMS, its resemblance to a helicopter and its cross-coupled dynamics have also inspired me. The practical nature of this research including prospective implementation of the MPC approaches for nonlinear systems in real time and addressing the issues of stability and robustness have also driven my enthusiasm for this research.

Although the TRMS does not have the ability to fly, it possesses a striking similarity to a helicopter in terms of system nonlinearities and cross-coupled modes. The TRMS, therefore, can be perceived as an unconventional and complex air vehicle with a flexible main body. These system characteristics present formidable challenges in modelling, control design, control analysis and implementation. The main aim of this research is to design a model-based predictive control for the TRMS

which can be extended to other nonlinear aerodynamic systems. The specific aims and objectives of the investigation are depicted below:

- **Nonlinear dynamic modelling of the TRMS**

Accurate modelling is a prerequisite for the success of any model-based control technique. Therefore, various modelling approaches need to be investigated to figure out a suitable model for the TRMS. In this case, white-box, black-box and grey-box modelling approaches are employed. In the case of white-box modelling, the governing equations of the system are extracted using both the Newtonian and Lagrangian techniques and the parameters are identified according to physical knowledge of the system. NNs are then employed to develop the nonlinear black-box model of the system. Using the advantages of GAs, the parameters of the white-box model are retuned to achieve a grey-box model of the TRMS. The models are verified and compared using convincing tests such as overlapping, PSD and correlation techniques to select the most suitable one.

- **Stability of the plant**

Before going through the control design, the stability of the model needs to be investigated. In this case, the Lyapunov technique and the invariant set theorem are employed to investigate and analyse the global asymptotically stability of the system.

- **Control specifications**

Here a set of specifications is outlined for the control system design purpose:

- Test the stability of the system in the face of the coupling effect between the two channels, disturbance and step change.
- The ability to continuously follow a reference trajectory.
- Zero steady state error response.
- The ability to follow a step change in a short time (say, rise time less than 10 seconds).

- **State-feedback MPC for nonlinear systems**

Assuming the accessibility of the state variables, a state-feedback MPC approach for nonlinear systems is developed. The idea originates from the multistep Newton-type MPC approach and the prospective use of a state-dependent model.

- **Stability of the state-feedback MPC**

As has been emphasised previously, stability is the minimum requirement and must be guaranteed for any control system. Terminal equality constraints will be used to guarantee the stability of the control approach.

- **State observers**

It is a well-known fact that full states are not accessible in the case of most practical systems and therefore, a state estimator such as the Luenberger observer or the Kalman filter needs to be used to estimate the state variables. Note that these observers can be utilised for linear systems and, in the case of nonlinear plants, EKF is one of the popularly used state estimators. However, when dealing with highly nonlinear systems, an EKF may exhibit significant performance degradation and therefore, other types of nonlinear estimators such as a UKF are required. In this study, a UKF is employed to observe the state variables of the TRMS. Moreover, an EKF based on a state-dependent state-space model is developed.

- **Output-feedback MPC for nonlinear systems**

The state-feedback model-based predictive control paradigm is extended to an output-feedback MPC using the state observers.

- **Robust MPC**

It is noted that robustness analysis in nonlinear MPC is much more difficult than stability analysis due to the fact that robustness means stability in the presence of model error and uncertainty [9]. One of the aims of this research is to utilise robust MPC techniques for a highly nonlinear MIMO system, the TRMS. It is envisaged that the proposed techniques can be extended to other mechanical and aerospace engineering problems.

1.4 Contributions

In this research, three different modelling techniques are employed to dynamically model a nonlinear system. On the basis of the most adequate model, a nonlinear MPC is developed, addressing issues of stability and robustness. The main contributions of this investigation are as follows:

- Analytical dynamic models of the TRMS in terms of 1DOF and 2DOF are developed. It is noted that the crude models supplied by the manufacturer [1] have not taken all the necessary forces into consideration, and therefore do not represent the TRMS accurately. In order to improve the models, the motors of the system are modelled according to their governing equations, and the necessary parameters such as the motors moment of inertia and load torque coefficients are precisely identified. Moreover, the TRMS has a set of flat cables which connect the power supply to the motors and also send the feedback signals to the computer. That the force originates from the flat cables has been considered, and all the related effective parameters are optimally obtained.
- Both 1DOF and 2DOF dynamic empirical models based on NNs are investigated with various learning approaches. Among the learning algorithms, the so-called Levenberg-Marquardt technique is found to be the most promising one. The obtained models are validated and compared with several convincing tests such as overlapping, PSD and correlation validation tests.
- Grey-box modelling of the system that merges the advantages of both white- and black-box modelling is investigated and applied for further improvement of the plant-model. The parameters of the developed analytical model are retuned by an evolutionary optimisation method, GAs, to find the optimum parameters. The performances of the white-box and grey-box approaches are compared to illustrate the model improvement achieved.
- A stable MPC approach based on a discrete state-space model is applied to the nonlinear system. A state-dependent model is used to find the same number of linear models as the prediction horizon at each instant. The stability of the control system is guaranteed using the terminal equality constraints approach. It is noted that the proposed method has been able to achieve a very fast and precise response to various reference signals.
- The proposed state-feedback RHC is extended to an output-feedback MPC. To this end, a UKF is utilised to observe the state variables. Moreover, an EKF based on a state-dependent state-space model is applied and the results achieved in this case are found to be as good as that of the UKF.

- On the basis of the analytic nonlinear model, a polytopic model is developed and used in designing a robust MPC. The robustness is achieved by the Lyapunov criterion. The overall optimisation problem is cast as a quadratic optimisation problem involving LMI.

1.5 Outline of the Thesis

A brief outline of the thesis is presented below. Chapters 2 to 4 deal with the modelling of the system, Chapter 5 presents the stability of the unforced TRMS and Chapters 6 to 8 address the MPC paradigms.

- **Chapter 2** presents the analytical model of the TRMS in terms of its vertical 1DOF, horizontal 1DOF and 2DOF using the Newtonian as well as Lagrangian methods. The identification of the TRMS parameters has been explained in detail. The model response and the real TRMS response are compared with each other to validate the accuracy of the model. For further assessment of the model, the PSD of the real system and the developed model are compared.
- **Chapter 3** describes the NN-based empirical model of the TRMS in terms of its vertical 1DOF, horizontal 1DOF and 2DOF. Multilayer perceptron (MLP) networks have been used to model the plant. The NN-based model response and the real TRMS response are compared with each other to validate the accuracy of the model. A comparative study between analytical and empirical models is also presented in this chapter.
- **Chapter 4** discusses the grey-box nonlinear dynamic modelling approach of the TRMS. GAs have been used to optimally retune the parameters of the plant-model. In order to show the efficacy and validity of the approach, the response of the grey-box model has been compared with that of the white-box paradigm.
- **Chapter 5** presents the stability analysis of the TRMS. The equilibrium points of the nonlinear systems are calculated and the local stability of the 2DOF system is proven using Lyapunov's linearisation method. The global

stability of the 1DOF TRMS is considered using the Lyapunov and invariant set theorem.

- **Chapter 6** discusses a stable MPC for nonlinear systems. The theoretical background of the proposed nonlinear MPC has been described. An approach based on multistep Newton-type MPC has been used to control the TRMS in terms of 2DOF dynamics of the system. The stability of the closed-loop system is provided using terminal equality constraints.
- **Chapter 7** is the extension of the proposed state-feedback MPC presented in Chapter 6 to an output-feedback MPC using a UKF. The UKF is believed to show better and more reliable results compared to its counterpart, EKF. In this chapter, an EKF is also employed which is based on a state-dependent state-space model rather than the Jacobian-based linearisation.
- **Chapter 8** presents the development of a polytopic model and the design of a robust model-based predictive control approach. In this chapter, the procedure of polytopic modelling is described and the robustness of the controller is provided on the basis of polytopic uncertainty.
- **Chapter 9** consists of conclusions, discussion and outline of future work.

1.6 Publications

Publications extracted from this research are listed below.

1.6.1 Journal Articles

- **A. Rahideh, H. M. Shaheed**, "NN-based Modelling of a 2DOF Twin Rotor MIMO System using Conjugate Gradient Learning Algorithms", *IMechE Journal of Aerospace Engineering*, Vol. 222, No. 6, 2008, pp. 757-771.
- **A. Rahideh, H. M. Shaheed, H. J. C. Huijberts**, "Dynamic Modelling of a TRMS using Analytical and Empirical Approaches", *Control Engineering Practice*, Vol. 16, 2008, pp. 241-259.

- **A. Rahideh, H. M. Shaheed**, “Mathematical Dynamic Modelling of a Twin-rotor Multiple Input-Multiple Output System”, *IMechE Journal of Systems and Control Engineering*, Vol. 221, No. 1, January 2007, pp. 89-101.

1.6.2 Conference Articles

- **A. Rahideh, H. M. Shaheed**, “Output Feedback Nonlinear Model Predictive Control of a Twin Rotor MIMO System”, *11th International Conference on Climbing and Walking Robots*, Coimbra, Portugal, September 2008, pp. 463-470.
- **A. Rahideh, H. M. Shaheed, H. J. C. Huijberts**, “Stable Adaptive Model Predictive Control for Nonlinear Systems”, *American Control Conference*, Washington, Seattle, June 2008, pp. 1673-1678.
- **A. Rahideh, H. M. Shaheed**, “Dynamic Modelling of a Twin Rotor System using Grey Box Approach”, *International Symposium on Mechatronics and its Applications*, Jordan, May 2008.
- **A. Rahideh, A. A. Safavi, H. M. Shaheed**, "NN-Based Modelling of a 2DOF TRMS using RPROP Learning Algorithm", *European Control Conference*, Kos, Greece, July 2007, pp. 2648-2654.
- **A. Rahideh, A. A. Safavi, H. M. Shaheed**, "Applications of Neural Networks to Modelling of a 2DOF TRMS", *Iranian Conference on Electrical Engineering*, Tehran, Iran, May 2007, pp. 86-91.
- **A. Rahideh, H. M. Shaheed, A. A. Safavi, H. J. C. Huijberts**, “Model Predictive Control of a Twin Rotor MIMO System”, *12th IEEE International Conference on Methods and Models in Automation and Robotics*, Poland, August 2006, pp. 453-458.

CHAPTER 2

Nonlinear White-box Modelling

In this chapter, a TRMS is analytically modelled in terms of its 1DOF and 2DOF dynamics using both the Newtonian and Lagrangian methods. The adequacy of the model is discussed by comparing the response of the model and the real TRMS in the time and frequency domains. A detailed description of the TRMS is also provided in this chapter.

2.1 Introduction

Modelling is the fundamental stage of almost any modern control technique. Depending on the *a-priori* knowledge of a plant, it can be modelled using white-box or black-box modelling approaches. In some cases, the combination of white- and black-box approaches is employed to simultaneously achieve the advantages of both techniques; these techniques are well-known as grey-box modelling approaches. White-box modelling is a type of plant-model representation in which a detailed physical and/or chemical knowledge of the plant is needed. In other words, when full physical insight of a plant is available, one can analytically model the plant to estimate or predict its behaviour. In most cases, the final product of white-box models is a set of ordinary or partial differential equations. In the field of control engineering, the ordinary differential equations (ODEs) are usually expressed in a standard form of state-space equations regardless of their linear or nonlinear features. In the case of mechanical systems, the Newtonian or Lagrangian techniques are commonly utilised to find the system's governing equations [3]. Clearly, the models based on the Newtonian and Lagrangian approaches should reach the same set of equations if the assumptions behind them are similar. However, sometimes the assumptions are considered to be different due to the differing complexity of the approaches. Although the procedure to develop an analytical plant-model is almost systematic, the model parameters identification is a challenging and time-consuming process which is the main drawback of white-box modelling techniques. A plant-model is usually employed to design and pre-test a control paradigm where the

controller cannot be tested directly on the real plant. Also, a plant-model is needed in the case of model-based control approaches such as MPC. In this investigation, both of the aforementioned applications of modelling are utilised. Model validation tests are carried out to assess whether the model represents the plant adequately. In dealing with analytical models, both the time and frequency domains validation tests need to be employed to check the degree of adequacy of the plant-model. The so-called overlapping test is the most commonly used validation test in all types of modelling approaches; in this test, the model response is compared with that of the real plant in the time domain to visualise the discrepancy between them. The PSD test is a well-known test to check the adequacy of the model in the frequency domain. In this investigation, both the time and frequency domain tests are used to validate the analytical model of the TRMS.

2.2 The TRMS

The TRMS is a laboratory set-up designed by Feedback Instrument Ltd [1] and is a suitable test platform for assessment and implementation of advanced control techniques. The system is connected to a computer through a fast interface to transfer control signals to the actuators and to receive the corresponding feedback signals from the sensors. Moreover, the real-time workshop toolbox of MATLAB/Simulink provides an opportunity for the designer to facilitate the controller design procedure using advanced control toolboxes and other useful built-in functions. The system possesses two propellers perpendicular to each other, one for vertical movement and the other for horizontal motion. However, each one of them significantly affects the motion of the other. The propellers are driven by two high speed permanent magnet DC motor drives. The input voltage signals to the motors are restricted between -2.5 and +2.5 Volts. However, the minimum and maximum terminal voltages of the motors are specified to be -18 and +18 Volts respectively. The difference between the input voltage and the terminal voltage stems from the fact that an input voltage signal from the processor, here the computer, is mainly a low-current voltage signal that needs to be transferred to a corresponding motor terminal voltage using interface circuits and power electronics devices.

The two propellers are connected to each other using a free beam pivoted on the top of the tower of the TRMS. The beam can rotate in both the vertical and horizontal planes. In order to make the system stable in steady state positions, a counter weight beam, perpendicular to the free beam, is attached to the system and a weight is also

attached to the counter weight beam to model a load. It is noted that the position of the weight on the counter weight beam can be changed resulting a corresponding change in the dynamics of the system. The TRMS, shown in Figures 2.1 and 2.2, possesses two shaft encoders to accurately measure the angle of the beam in the horizontal and vertical planes.

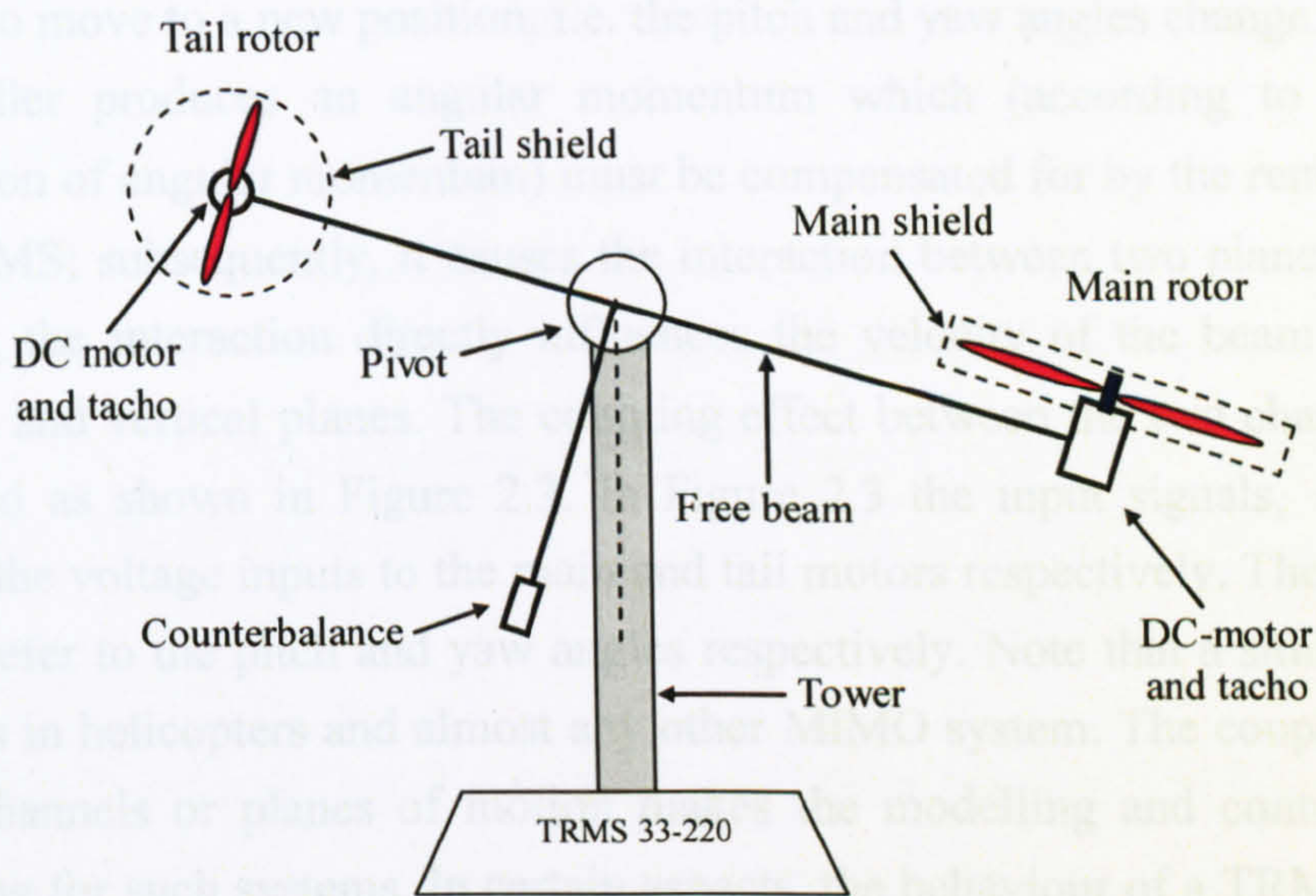


Figure 2.1 The TRMS

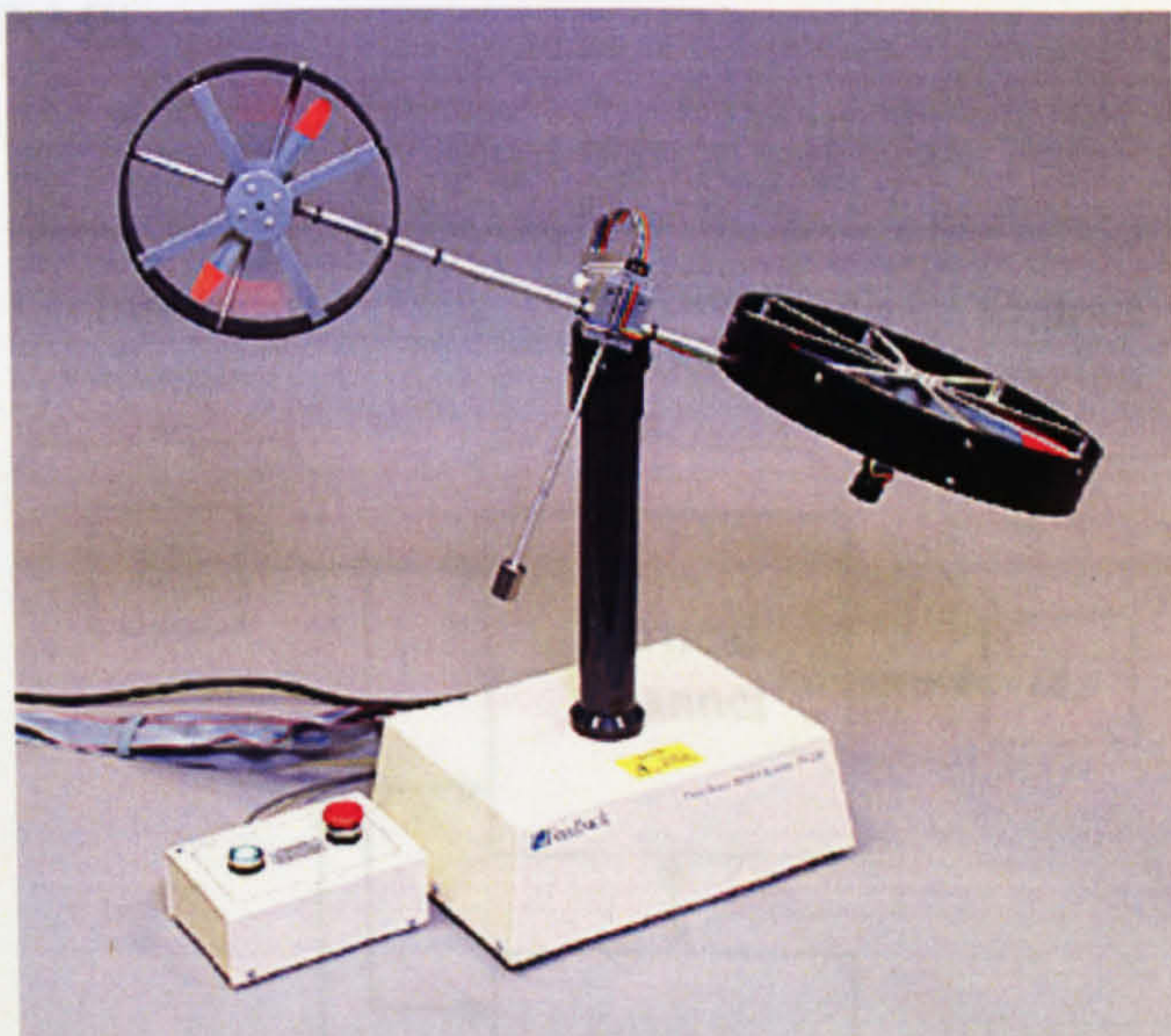


Figure 2.2 The TRMS

In addition, a tacho-generator is attached to the shaft of each motor for sensing the motor rotational velocity. Basically, the TRMS is a system with 2DOF that can also be used as two 1DOF systems if one of the attached screws is locked to physically restrict the motion in either the horizontal or the vertical plane.

The angle of attack of the blades in a typical helicopter is used to control the aerodynamic force; however, in the case of the TRMS there is no change in the angle of attack of the blades and the aerodynamic force is controlled by varying the rotational velocity of the motors. Therefore, the control input signals are the supply voltages of the DC motors. A change in the magnitude of the voltage results in a corresponding change of the rotational velocity of the propeller, which in turn forces the beam to move to a new position, i.e. the pitch and yaw angles change. Rotation of the propeller produces an angular momentum which (according to the law of conservation of angular momentum) must be compensated for by the remaining body of the TRMS; subsequently, it causes the interaction between two planes of motion. Moreover, the interaction directly influences the velocity of the beam in both the horizontal and vertical planes. The coupling effect between the two channels can be represented as shown in Figure 2.3. In Figure 2.3 the input signals, U_v and U_h , represent the voltage inputs to the main and tail motors respectively. The outputs, α_v and α_h , refer to the pitch and yaw angles respectively. Note that a similar coupling also exists in helicopters and almost any other MIMO system. The coupling between various channels or planes of motion makes the modelling and control problems challenging for such systems. In certain aspects, the behaviour of a TRMS resembles that of a helicopter. Nonetheless, a helicopter is different from a TRMS with respect to a number of fundamental aspects. Table 2.1 lists the main discrepancies between a helicopter and a TRMS.

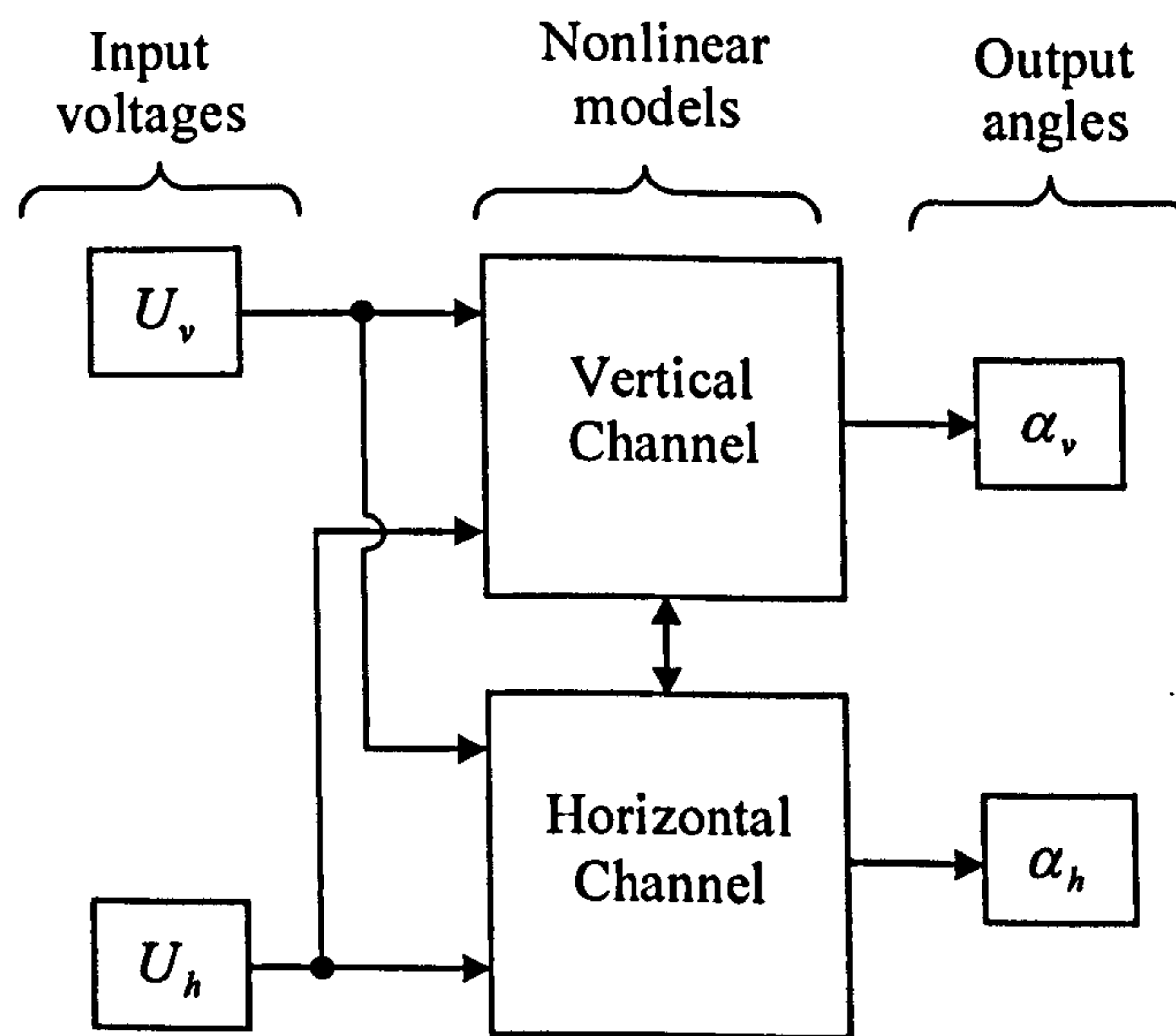


Figure 2.3 Coupled MIMO model of the TRMS

Table 2.1 The main differences between a helicopter and a TRMS

	TRMS	Helicopter
Pivot point is located at:	Midway between the two rotors	The main rotor head
Lift generation or vertical control via:	Speed control of the main rotor	Collective pitch control*
Yaw is controlled by changing the:	Tail rotor speed	Pitch angle of all the blades of the tail rotor
Cyclical control**:	No	Yes <i>For directional control</i>
Ability to fly:	No	Yes

* Pitch angles of all blades of the main rotor are changed but at constant rotor speed.

** Cyclical control changes the alignment of the main rotor blades in order to change the direction of a helicopter.

2.3 The Newtonian-based Modelling of the TRMS

Modern methods of design and adaptation of real time controllers require high quality mathematical plant-models. For high order nonlinear cross-coupled systems, classical modelling methods based on Lagrange's equations are often employed; however, it is also possible to use a simpler approach, based on Newton's equations of motion. In this study, the TRMS, as shown in Figure 2.2, is modelled using both the Newtonian and Lagrangian approaches. The Newtonian-based model of the system is presented in this section. In order to model the TRMS, the system is divided into several subsystems to form a modular system. At the first stage of modelling, the interface circuits between the computer and the TRMS have been modelled based on measured data. The permanent magnet DC motors used to propel the system have been modelled at the second stage. Then, the relationship between the force applied to the beam and the rotational velocity of each motor has been obtained. Thereafter, the dynamic equations that associate the aerodynamic force of the propellers to the position of the beam (pitch and yaw angles) have been extracted. Finally, the parameters of the 1DOF vertical, 1DOF horizontal, and 2DOF systems have been identified respectively.

2.3.1 Modelling of the Vertical Channel Interface Circuit

As mentioned before, the input voltage signal of the main motor, U_v , from the computer differs from the motor terminal voltage, V_v , and the relationship between

these two sets of values is slightly nonlinear. The low-current signal from the computer is unable to excite the electric motor and therefore, the signal has to be converted to a voltage to supply to the motor terminals. The model of the interface circuit is obtained by gradually changing the input signal in the MATLAB/Simulink environment from -2.5 to 2.5 with a step size of 0.1, and simultaneously measuring the relative motor terminal voltage values. Figure 2.4 shows the relationship between the input and the output of the vertical channel interface circuit. As it is evident from Figure 2.4, the relationship is not linear; however, it can linearly be approximated by $V_v = k_2 U_v$ for simplicity, whenever necessary. Where, k_2 is the approximate gain which models the vertical interface circuit and is obtained using the least squared error approach. The optimum value of k_2 is found to be 8.5.

It is noted that the interface circuit consists of several parts such as computer ports, a data acquisition board, electronic and power electronic devices/circuits, and filters which connect the computer to the electric motors.

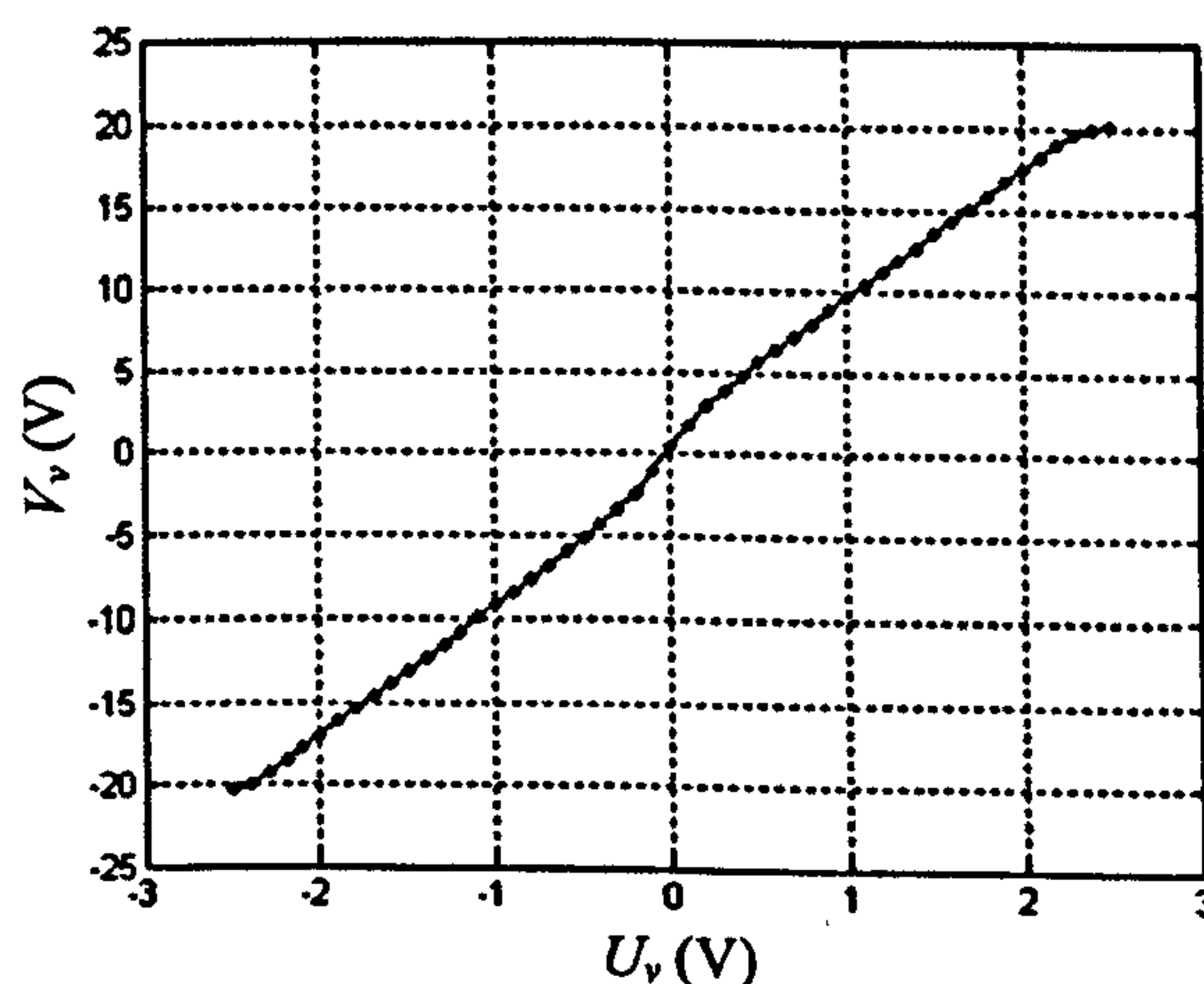


Figure 2.4 The graphical relationship between U_v and V_v

2.3.2 Modelling of the Horizontal Channel Interface Circuit

Like the main motor, the input voltage signals of the tail motor, U_h , in the computer are converted to the tail motor terminal voltages, V_h , using the interface circuit. Figure 2.5 shows their relationship. The corresponding data is presented in Table A1.2 in Appendix 1. A linear approximation can be obtained as $V_h = k_1 U_h$, where, the optimum value of k_1 is found to be 6.5.

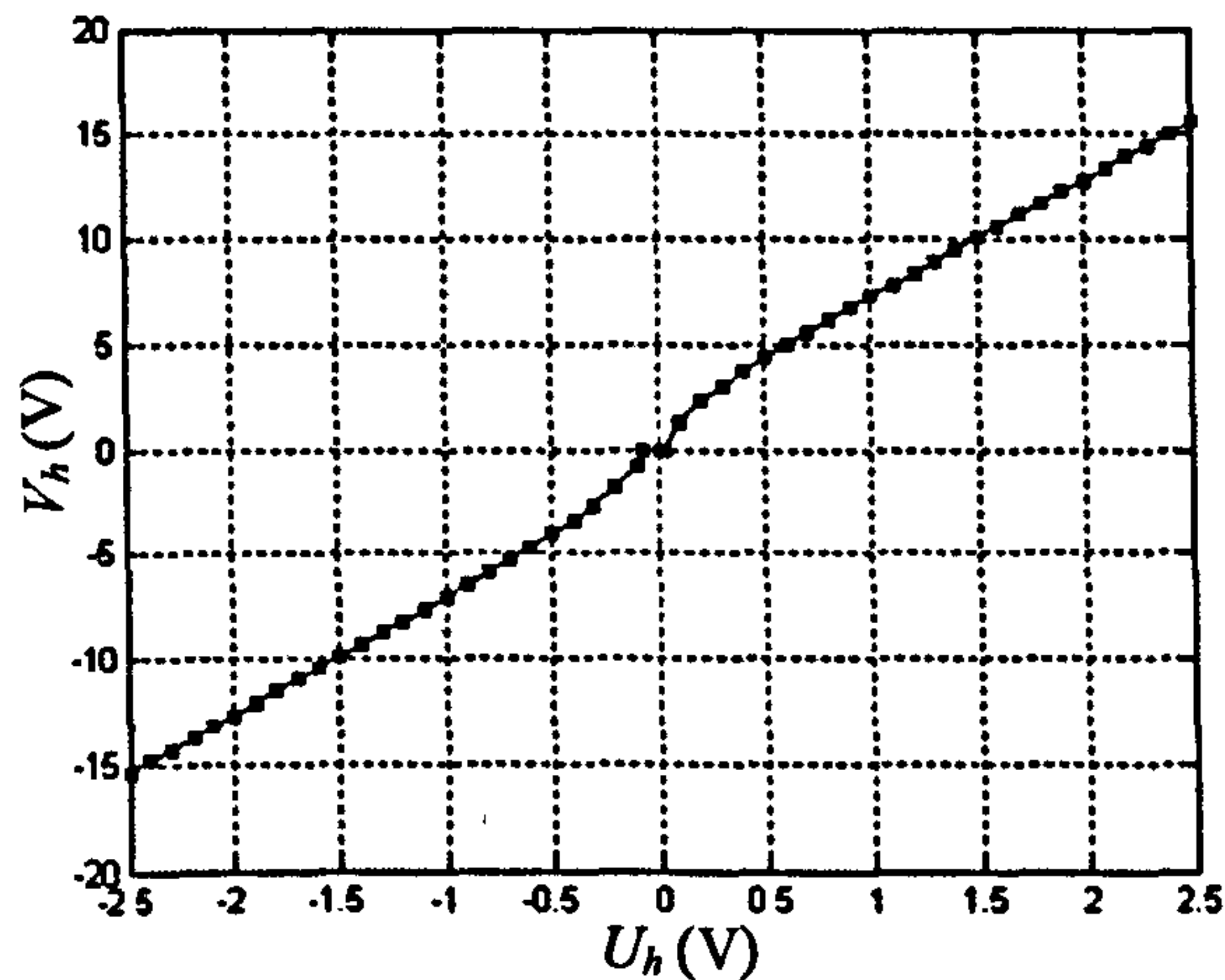


Figure 2.5 The graphical relationship between U_h and V_h

Note that the motor terminal voltages for the same input signal from the computer may differ depending on warm or cold conditions of the system.

2.3.3 Modelling of the Main Motor

As mentioned previously, two permanent magnet DC motors of the Maxon Company drive the main and tail propellers. The motors are identical with different mechanical loads; the discrepancy stems from the fact that the size and shape of the vertical and horizontal propellers are significantly different. Based on the principle theory of electromechanical systems, an electric motor needs two sets of windings to be able to work properly; one is so-called field winding, producing a magnetic field, and the other is entitled armature winding, which carries the armature current. Basically, the armature winding of a DC motor is located on the non-stationary part, i.e. the rotor, and its field winding is placed on the stationary part, the stator. In the case of a permanent magnet DC motor, a set of permanent magnets substitutes the field winding for generating a constant magnetic flux. The electric motors with especial applications to control drives are usually designed to have long length rotors with small cross sectional area, i.e. small radius, to be able to react very fast (accelerate rapidly) in response to reference signals. This allows the motors to have a higher torque to moment of inertia ratio.

According to the circuit diagram of the main motor shown in Figure 2.6, equations (2.1) to (2.5) can be written. According to Kirchhoff's voltage law, the terminal voltage of the motor is equal to the algebraic sum of the voltage drop due to the armature resistance, voltage over the armature inductance and the electromotive

force. In general, electromotive force of an electric motor is proportional to the rotational velocity of the rotor multiplied by the magnetic field. Since in a permanent magnet motor the magnetic field is constant, therefore, its electromotive force can be changed only by changing the rotational velocity of its rotor.

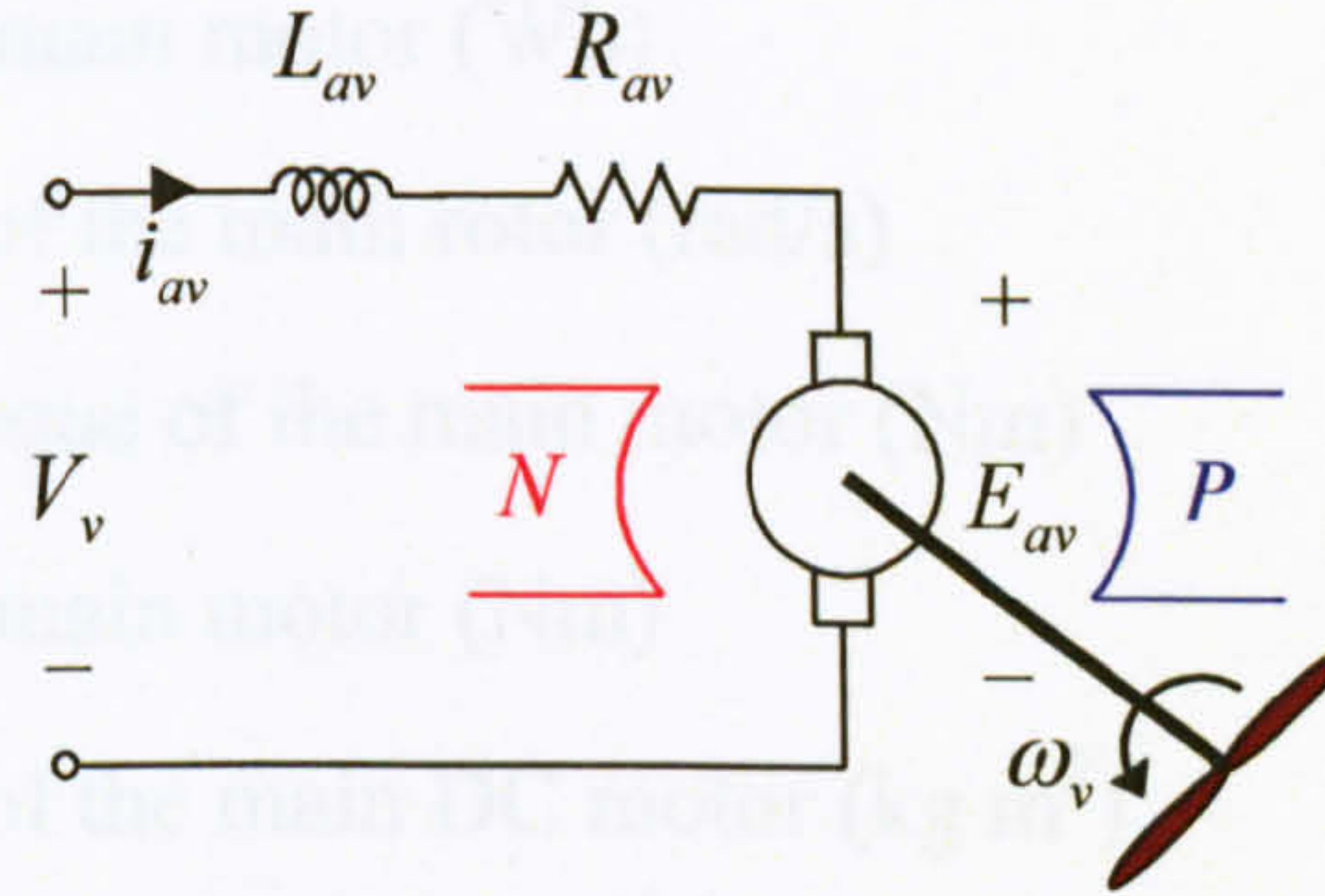


Figure 2.6 The circuit diagram of the main DC motor with its propeller

Newton's second law is used to derive equation (2.3), i.e. the sum of exerted torques to a rotational body is proportionate to the product of its moment of inertia and its rotational acceleration. In this study, three applied torques are considered: electromagnetic torque, load torque, and viscous friction torque. Electromagnetic torque of a permanent magnet electric motor depends directly on its armature current, and the equation of a load torque can vary depending on the type of the load. In the case of propeller loads, the load torque expression can be approximated by a parabolic function and considering the direction of rotation. All the parameters are listed in Table A1.3 of Appendix 1 and the file containing the parameters can be found in Appendix 5.

$$V_v = E_{av} + R_{av}i_{av} + L_{av} \frac{di_{av}}{dt} \quad (2.1)$$

$$E_{av} = k_{av}\phi_v\omega_v \quad (2.2)$$

$$M_{ev} = M_{Lv} + J_{mr} \frac{d\omega_v}{dt} + B_{mr}\omega_v \quad (2.3)$$

$$M_{ev} = k_{av}\phi_v i_{av} \quad (2.4)$$

$$M_{Lv} = \begin{cases} k_{tvp}\omega_v |\omega_v| & \text{if } \omega_v \geq 0 \\ k_{tvn}\omega_v |\omega_v| & \text{if } \omega_v < 0 \end{cases} \quad (2.5)$$

where,

V_v : Main DC motor terminal voltage (V)

E_{av} : Electromotive force of the main motor (V)

R_{av} : Armature resistance of the main motor (Ω)

L_{av} : Armature inductance of the main motor (H)

i_{av} : Armature current of the main motor (A)

k_{av} : Constant (Nm/AWb)

ϕ_v : Magnetic flux of the main motor (Wb)

ω_v : Rotational velocity of the main rotor (rad/s)

M_{ev} : Electromagnetic torque of the main motor (Nm)

M_{Lv} : Load torque of the main motor (Nm)

J_{mr} : Moment of inertia of the main DC motor (kg m^2)

B_{mr} : Viscous friction coefficient of the main DC motor ($\text{kg m}^2/\text{s}$)

k_{tvp}, k_{tvn} : Constants ($\text{Nms}^2/\text{rad}^2$).

The simulation of the main DC motor is illustrated in Figure 2.7. It is clear that the model is a second-order nonlinear system. The motor model is nonlinear due to the nonlinearities of the load torque. In the case of a linear load torque, the model has two time constants, one is electrical and the other is mechanical. Normally, the mechanical time constant is significantly greater than the electrical one and therefore, the mechanical one is typically considered to be the dominant time constant of the system.

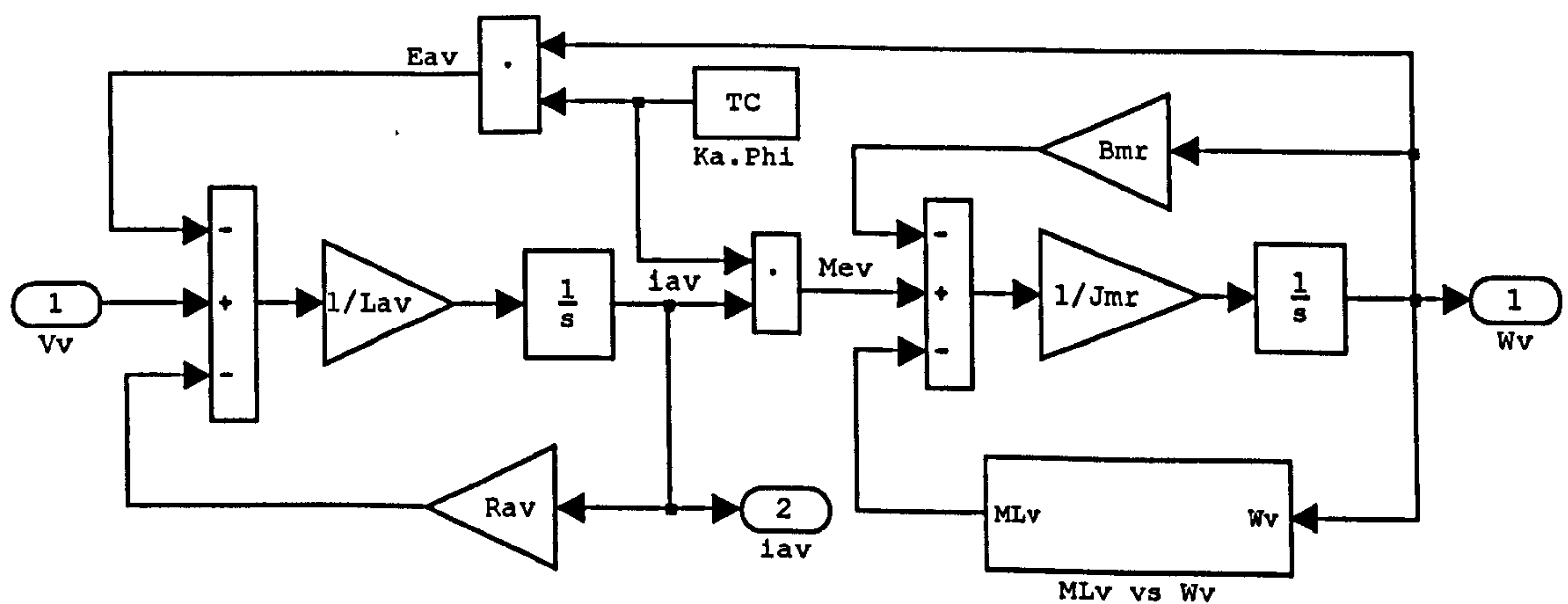


Figure 2.7 Simulation of the main DC motor

2.3.4 Modelling of the Tail Motor

From the circuit diagram of the tail motor, as shown in Figure 2.8, equations (2.6) to (2.10) can be written.

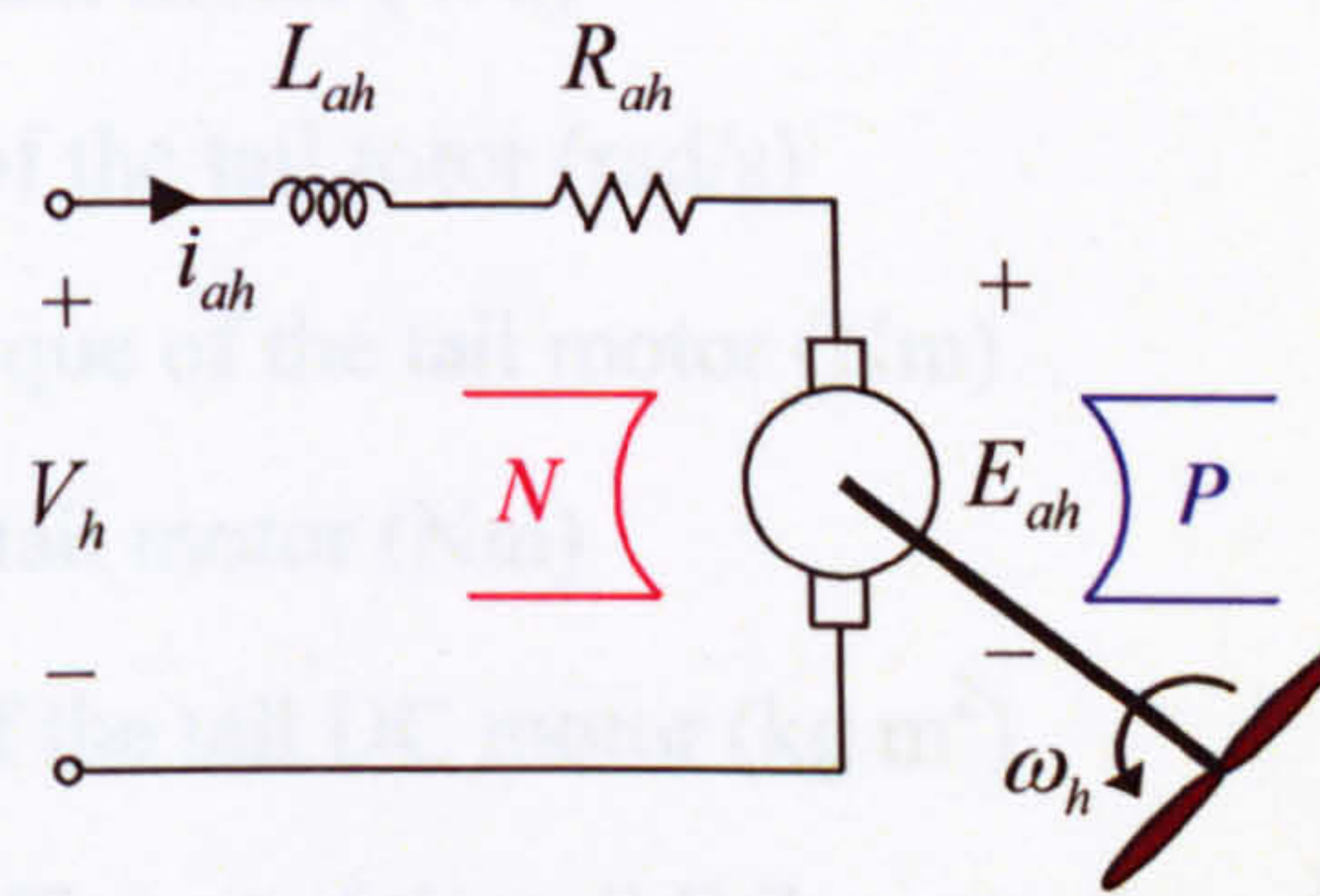


Figure 2.8 The circuit diagram of the tail DC motor with its propeller

From a physical point of view, once the permanent magnet DC motor is supplied with a voltage source, the armature winding initially draws high amount of current because electromotive force is primarily zero. Subsequently, in the presence of a magnetic field, the armature winding on the rotor is forced to rotate due to the generated electromagnetic torque and therefore, electromotive force increases gradually. Depending on the mechanical load torque and friction torque, the motor may accelerate and after a while rest at a steady state level in which the acceleration becomes zero. As a consequence, the electromagnetic torque just equals the sum of the load torque and the friction torque.

$$V_h = E_{ah} + R_{ah}i_{ah} + L_{ah} \frac{di_{ah}}{dt} \quad (2.6)$$

$$E_{ah} = k_{ah}\phi_h\omega_h \quad (2.7)$$

$$M_{eh} = M_{Lh} + J_{tr} \frac{d\omega_h}{dt} + B_{tr}\omega_h \quad (2.8)$$

$$M_{eh} = k_{ah}\phi_h i_{ah} \quad (2.9)$$

$$M_{Lh} = \begin{cases} k_{thp}\omega_h |\omega_h| & \text{if } \omega_h \geq 0 \\ k_{thn}\omega_h |\omega_h| & \text{if } \omega_h < 0 \end{cases} \quad (2.10)$$

where,

V_h : Tail DC motor terminal voltage (V)

E_{ah} : Electromotive force of the tail motor (V)

R_{ah} : Armature resistance of the tail motor (Ω)

L_{ah} : Armature inductance of the tail motor (H)

i_{ah} : Armature current of the tail motor (A)

k_{ah} : Constant (Nm/AWb)

ϕ_h : Magnetic flux of the tail motor (Wb)

ω_h : Rotational velocity of the tail rotor (rad/s)

M_{eh} : Electromagnetic torque of the tail motor (Nm)

M_{Lh} : Load torque of the tail motor (Nm)

J_{tr} : Moment of inertia of the tail DC motor (kg m^2)

B_{tr} : Viscous friction coefficient of the tail DC motor ($\text{kg m}^2/\text{s}$)

k_{thp}, k_{thn} : Constants ($\text{Nms}^2/\text{rad}^2$).

The simulation of the tail DC motor is illustrated in Figure 2.9. Analogous to the main motor, the model of the tail motor is a nonlinear, second-order system.

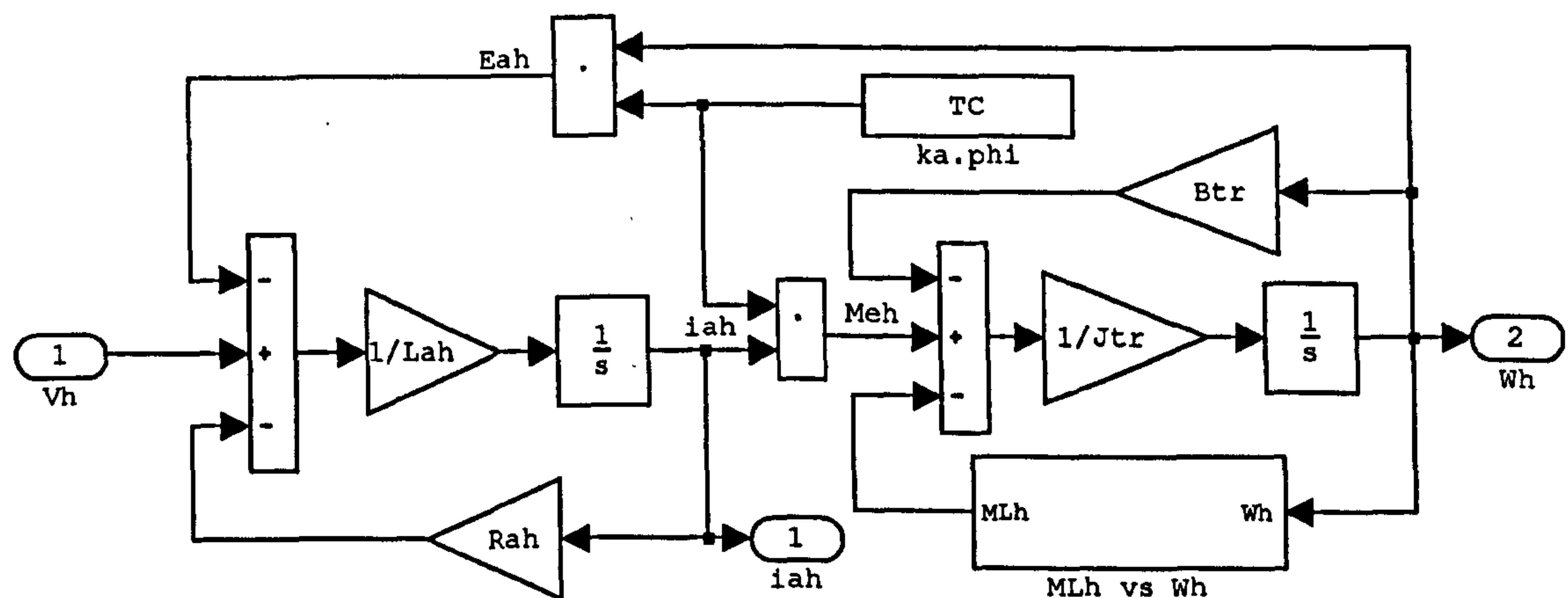


Figure 2.9 Simulation of the tail DC motor

2.3.5 Modelling of the Force due to Propellers Rotation

After modelling the DC motors, the rotational velocities of the two propellers are converted to the applied horizontal and vertical forces to the beam. To this end, the rotational speed and the corresponding applied force to the beam have been measured at several operating points and a suitable curve has been fitted on the gathered data. The data in the positive and negative rotational velocity regions are asymmetric, and therefore, two separate curves are fitted on two regions for more accuracy. Figures 2.10 and 2.11 depict the simulation of the vertical and horizontal

forces respectively. A visual comparison between the measured data and the fitted curve can be found in Figure A1.1 of Appendix 1.

$$F_h(\omega_h) = \begin{cases} k_{fhp} \omega_h |\omega_h| & \text{if } \omega_h \geq 0 \\ k_{fhn} \omega_h |\omega_h| & \text{if } \omega_h < 0 \end{cases} \quad (2.11)$$

$$F_v(\omega_v) = \begin{cases} k_{fvp} \omega_v |\omega_v| & \text{if } \omega_v \geq 0 \\ k_{fvn} \omega_v |\omega_v| & \text{if } \omega_v < 0 \end{cases} \quad (2.12)$$

where,

F_h : Nonlinear function of aerodynamic force due to the tail rotor (N)

F_v : Nonlinear function of aerodynamic force due to the main rotor (N)

k_{fhp} , k_{fhn} , k_{fvp} , k_{fvn} : Constants (Ns²/rad²).

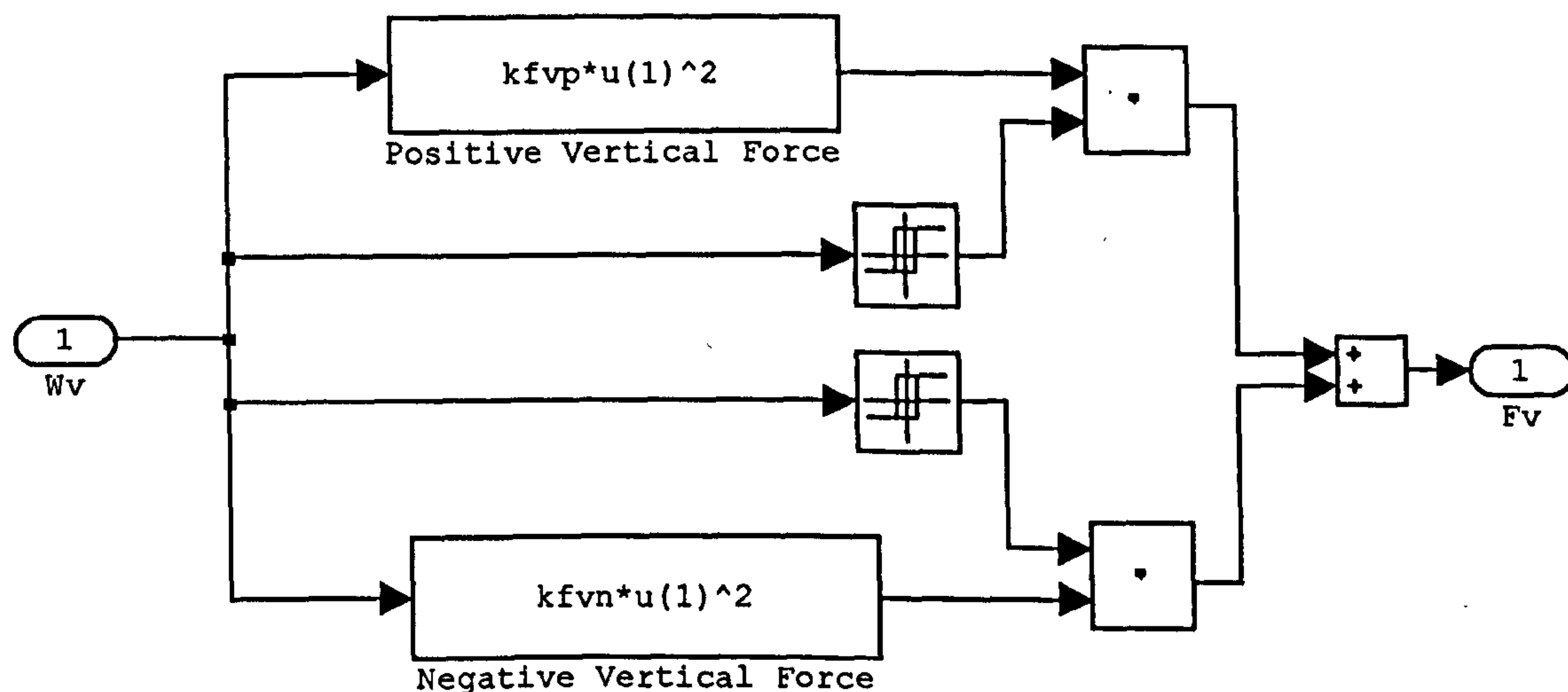


Figure 2.10 Simulation of the vertical force

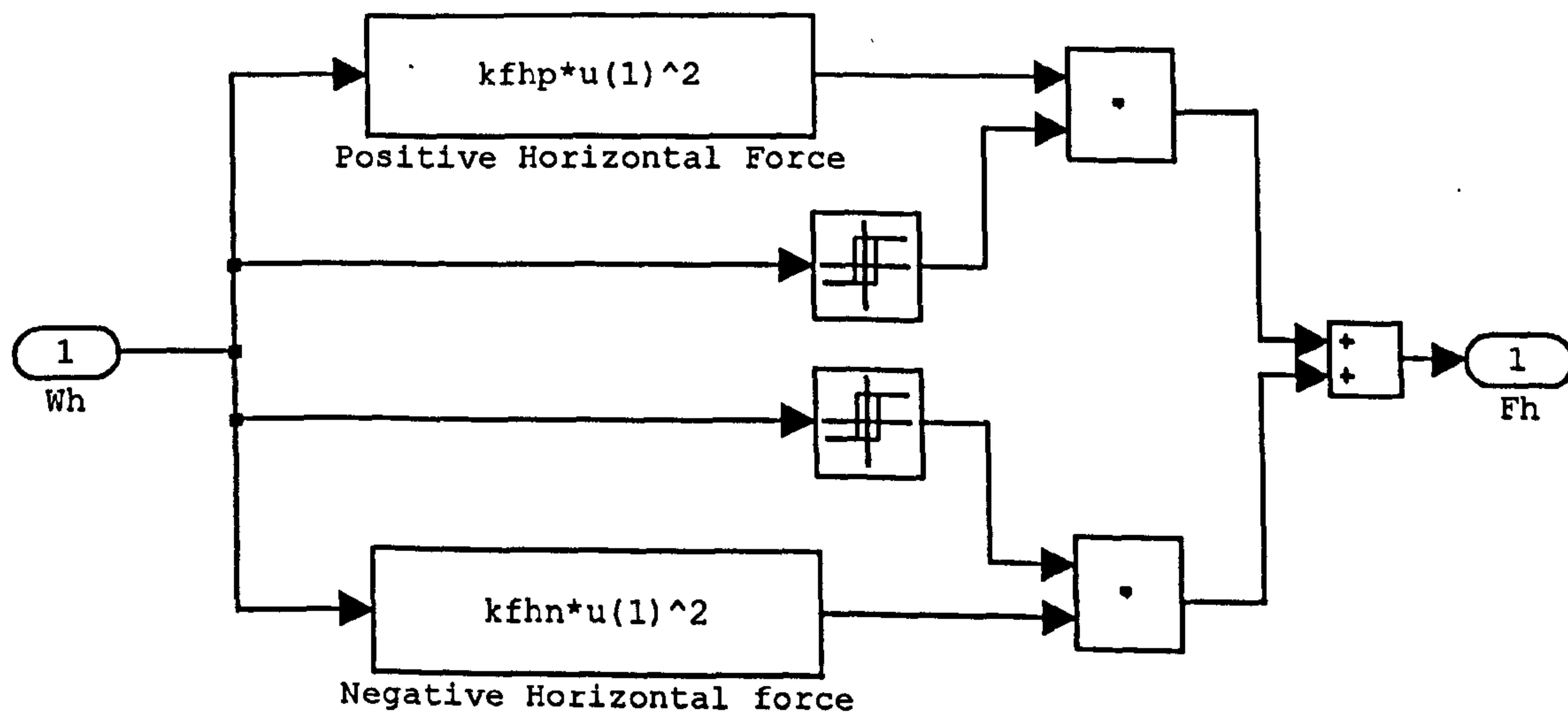


Figure 2.11 Simulation of the horizontal force

2.3.6 Modelling of the 1DOF Vertical TRMS

To model the dynamical equations of the system, the TRMS user manual [1] has been used to some extent to improve the crude model provided by the manufacturer.

2.3.6.1 Torques due to the gravity force

Assume the TRMS is observed from its front view as shown in Figure 2.12, therefore, the applied gravity forces to the TRMS can be divided into six parts as in Figure 2.12. The first term, $g(m_{tr} + m_{ts})$, is related to the mass of the tail rotor and its shield. The gravity force caused by the mass of the beam in the tail and main sides forms the second, $m_t g$, and third, $m_m g$, components. The mass of the main shield and rotor exerts a gravitational force which is represented by the fourth part, $g(m_{mr} + m_{ms})$. The fifth part, $m_{cb} g$, represents the gravity force due to the counter weight mass and finally the last part, $m_b g$, is due to the mass of the counter weight beam.

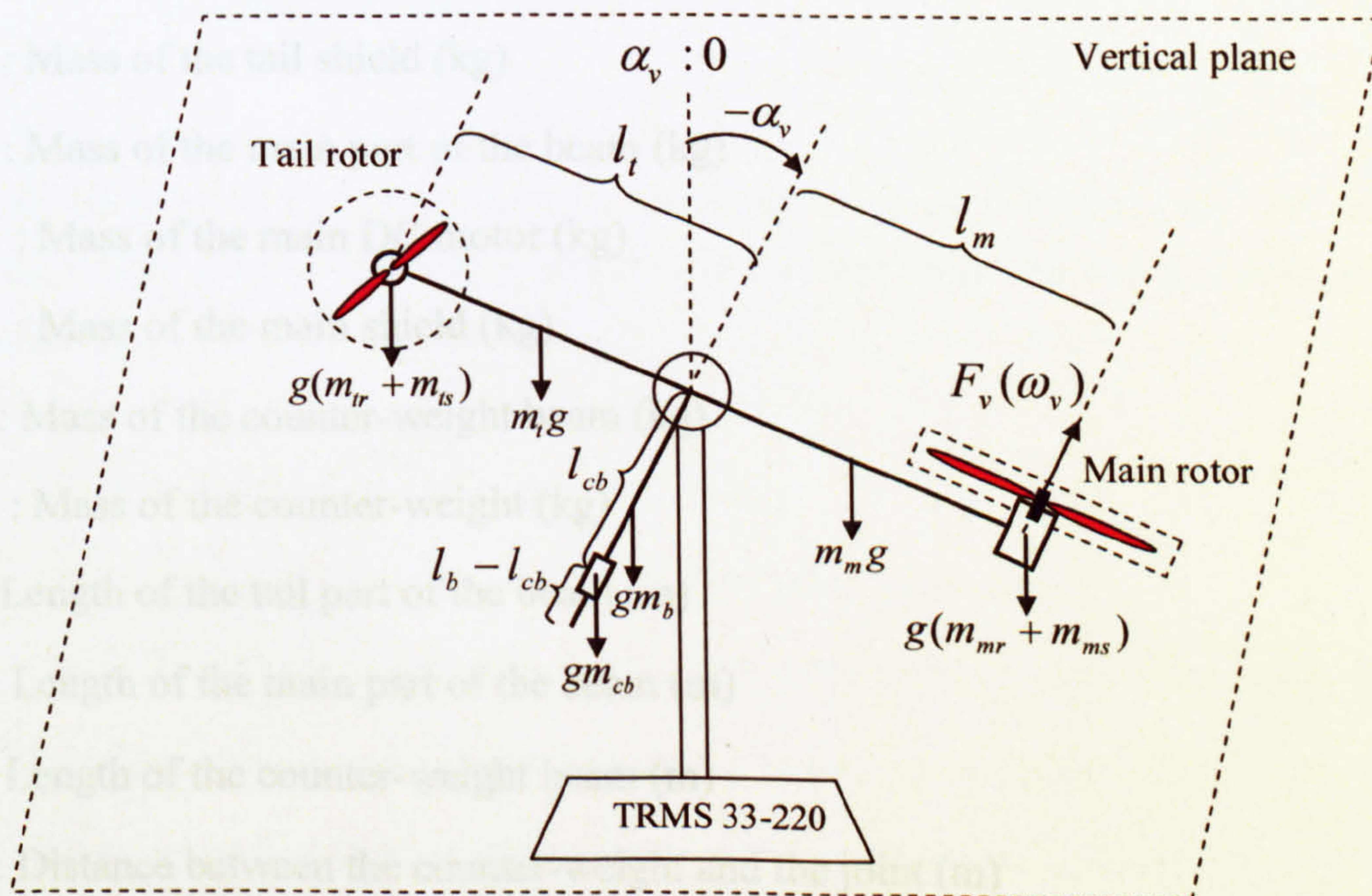


Figure 2.12 Gravity forces and propulsive force in the vertical plane

Therefore, the torque corresponding to the gravity forces is as follows,

$$M_{v1} = g \left\{ \left[\left(\frac{m_t}{2} + m_{tr} + m_{ts} \right) l_t - \left(\frac{m_m}{2} + m_{mr} + m_{ms} \right) l_m \right] \cos \alpha_v - \left(\frac{m_b}{2} l_b + m_{cb} l_{cb} \right) \sin \alpha_v \right\} \quad (2.13)$$

and can be expressed as,

$$M_{v1} = g \{ (A - B) \cos \alpha_v - C \sin \alpha_v \} \quad (2.14)$$

where,

$$A = \left(\frac{m_t}{2} + m_{tr} + m_{ts} \right) l_t$$

$$B = \left(\frac{m_m}{2} + m_{mr} + m_{ms} \right) l_m$$

$$C = \left(\frac{m_b}{2} l_b + m_{cb} l_{cb} \right)$$

and,

M_{v1} : Torque corresponding to the gravity forces in the vertical plane (Nm)

g : Gravitational acceleration (m/s^2)

m_t : Mass of the tail part of the beam (kg)

m_{tr} : Mass of the tail DC motor (kg)

m_{ts} : Mass of the tail shield (kg)

m_m : Mass of the main part of the beam (kg)

m_{mr} : Mass of the main DC motor (kg)

m_{ms} : Mass of the main shield (kg)

m_b : Mass of the counter-weight beam (kg)

m_{cb} : Mass of the counter-weight (kg)

l_t : Length of the tail part of the beam (m)

l_m : Length of the main part of the beam (m)

l_b : Length of the counter-weight beam (m)

l_{cb} : Distance between the counter-weight and the joint (m)

α_v : Vertical position (pitch angle) of the TRMS beam (rad).

2.3.6.2 Torque due to the propulsive force of the main rotor

In order to consider the propulsive forces applied to the beam in the vertical plane, it is needed to refer to Figure 2.12. The moment of the propulsive force produced by the main rotor is calculated by the product of the propeller force of the main motor

and the distance between the main motor and the pivot point. Therefore, it can be written as,

$$M_{v2} = l_m F_v(\omega_v) \quad (2.15)$$

where M_{v2} is the torque corresponding to the propulsive forces in the vertical plane (Nm).

2.3.6.3 Friction torque

While the free beam and the main rotor are moving in the vertical plane, the ambient air pressure causes an opposite force which prevents the body from further possible movement. This preventing force is deemed as a friction force. It is assumed that the friction force consists of viscous, Coulomb and static types, so the friction torque in the vertical plane can be expressed as,

$$M_{v3} = k_{vfv} \Omega_v + k_{cfv} \text{sign}(\Omega_v) + \begin{cases} k_{sfv} & \text{if } \Omega_v = 0^+ \\ -k_{sfv} & \text{if } \Omega_v = 0^- \\ 0 & \text{otherwise} \end{cases} \quad (2.16)$$

where the first term corresponds to the viscous friction torque, the second part represents the Coulomb friction torque and the last term denotes the static friction torque, and

M_{v3} : Friction torque in the vertical plane (Nm)

Ω_v : Angular velocity (pitch velocity) of the TRMS beam (rad/s)

k_{vfv} : Coefficient of the viscous friction torque in the vertical plane (Nms/rad)

k_{cfv} : Constant of the Coulomb friction torque in the vertical plane (Nm)

k_{sfv} : Constant of the static friction torque in the vertical plane (Nm).

This form of friction torque representation, equation (2.16), suffers from a discontinuity at $\Omega_v = 0$ (see Figure 2.13a) and may render some problems during the controller design. Therefore, the friction shown in Figure 2.13b is used for modelling; since, this profile covers viscous, Coulomb and static frictions and the discontinuity has been removed using an acceptable approximation.

$$M_{v3} = M_{fric,v} \quad (2.17)$$

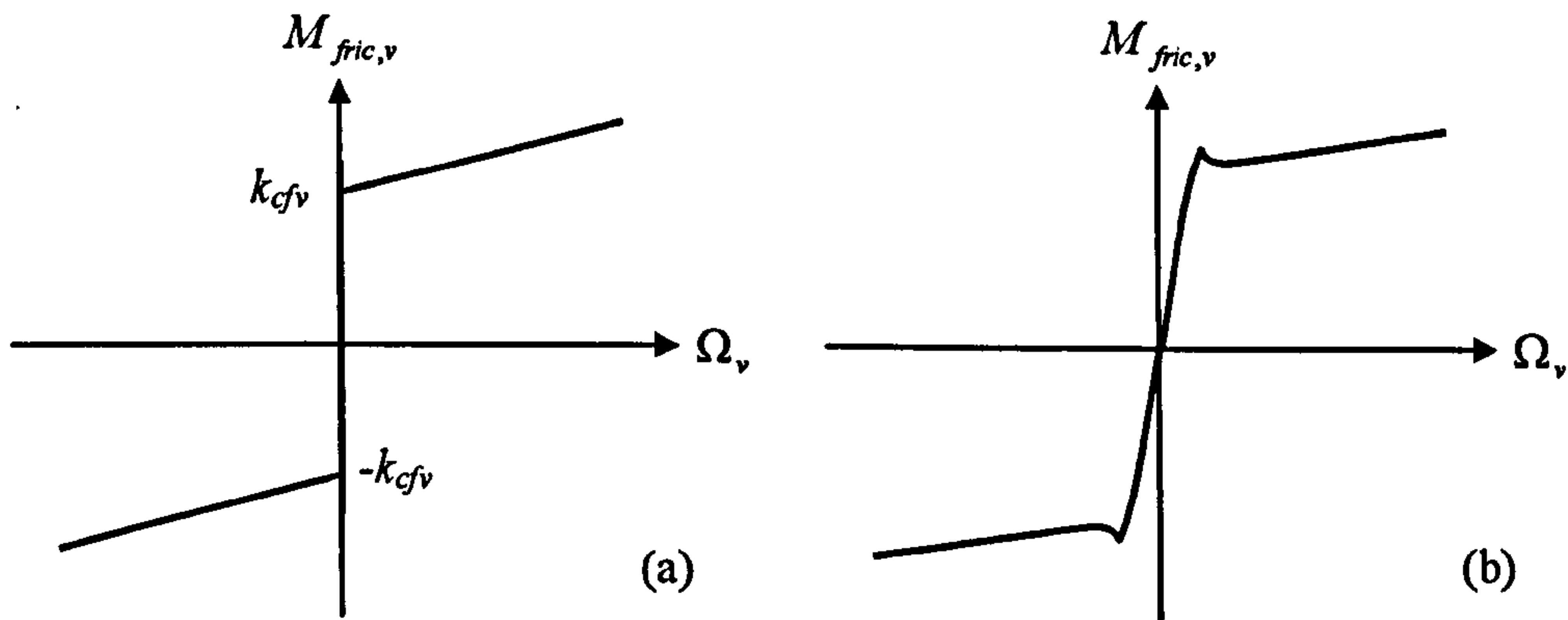


Figure 2.13 Friction torque profile of the vertical channel

2.3.6.4 Total torques in the vertical plane

According to Figure 2.12, one can determine and augment the components of the moment of inertia related to the vertical plane as,

$$J_v = m_{mr}l_m^2 + m_m \frac{l_m^2}{3} + m_{cb}l_{cb}^2 + m_b \frac{l_b^2}{3} + m_{tr}l_t^2 + m_t \frac{l_t^2}{3} + \left(\frac{m_{ms}}{2}r_{ms}^2 + m_{ms}l_m^2\right) + (m_{ts}r_{ts}^2 + m_{ts}l_t^2) \quad (2.18)$$

In the above equation, the tail shield is considered like a hoop perpendicular to the horizontal plane. The same process can be applied to the main shield, but parallel with the horizontal plane. The tail motor and its propellers are deemed as a particle of mass m_{tr} , located at l_t far from the pivot point. The main motor accompanying by its propellers can be considered in the same manner. The counter-weight is assumed as a particle of mass m_{cb} , located at the far end of the counter weight beam.

From Newton's second law of motion, $\sum_{i=1}^3 M_{vi} = J_v \frac{d^2\alpha_v}{dt^2}$, the following equations can be written.

$$\frac{d\Omega_v}{dt} = \frac{l_m F_v(\omega_v) - M_{fric,v} + g[(A-B)\cos\alpha_v - C\sin\alpha_v]}{J_v} \quad (2.19)$$

$$\frac{d\alpha_v}{dt} = \Omega_v \quad (2.20)$$

where,

r_{ms} : Radius of the main shield (m)

r_{ts} : Radius of the tail shield (m)

J_v : Moment of inertia about the horizontal axis (kg m^2).

The complete block diagram of the vertical 1DOF TRMS model is shown in Figure 2.14. In this figure, the first block from left is a look-up table which models the interface circuit. The main motor model is located inside the corresponding subsystem, the second block from left. The third block is the subsystem that receives the rotational velocity of the main motor and returns its propulsive force. The inside of these two mentioned subsystems have previously been depicted in Figures 2.7 and 2.10. Figure 2.15 shows the inside of the 'nonlinear equations' block according to equations (2.19) and (2.20). It is clear that the 1DOF vertical model of the TRMS is a fourth-order system. The control voltage U_v is the input signal to the main channel which is converted to the terminal voltage of the main motor, V_v . The output of the model is the pitch angle and four state variables are the armature current of the main motor, the rotational velocity of the main propellers, the angular velocity of the beam, and the pitch angle. Note that by changing the initial values of the integrators, the response of the system with non-zero initial state variables can be modelled.

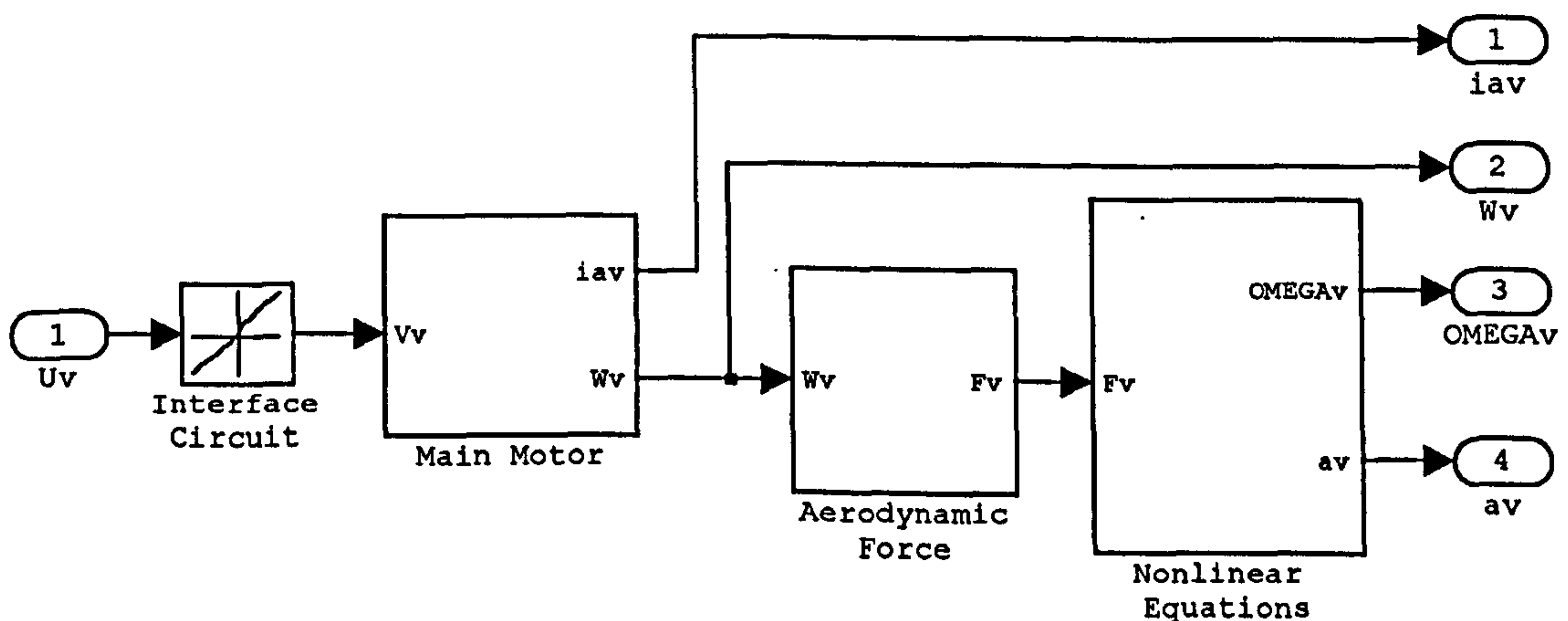


Figure 2.14 The complete block diagram of the vertical 1DOF TRMS model

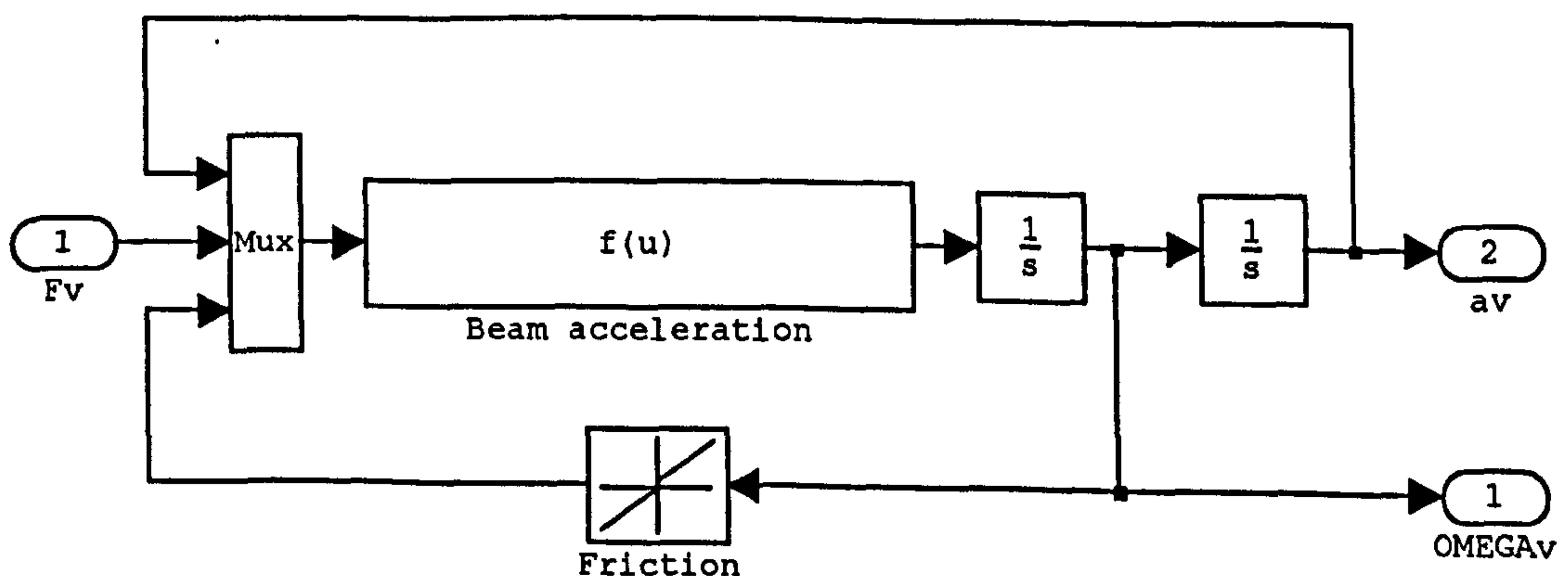


Figure 2.15 The inside of the 'nonlinear equations' block

2.3.7 Modelling of the 1DOF Horizontal TRMS

Assume that the main rotor is kept unexcited and just affects the system like a simple load. The only input to the system is the tail voltage signal and the output is the yaw angle. In this case, the TRMS is prevented from any vertical movement by locking the screw provided.

2.3.7.1 Torque due to the propulsive force of the tail rotor

Figure 2.16 shows the propulsive forces applied to the beam in the horizontal plane. According to this figure, the moment of the propulsive force produced by the tail rotor is written as,

$$M_{h1} = l_t F_h(\omega_h) \cos \alpha_v \quad (2.21)$$

Although in the case of the 1DOF horizontal model the pitch angle is constant, the propulsive torque has been defined as a function of both the pitch angle and the tail rotor rotational velocity because this equation is intended to be used in the case of the 2DOF dynamics.

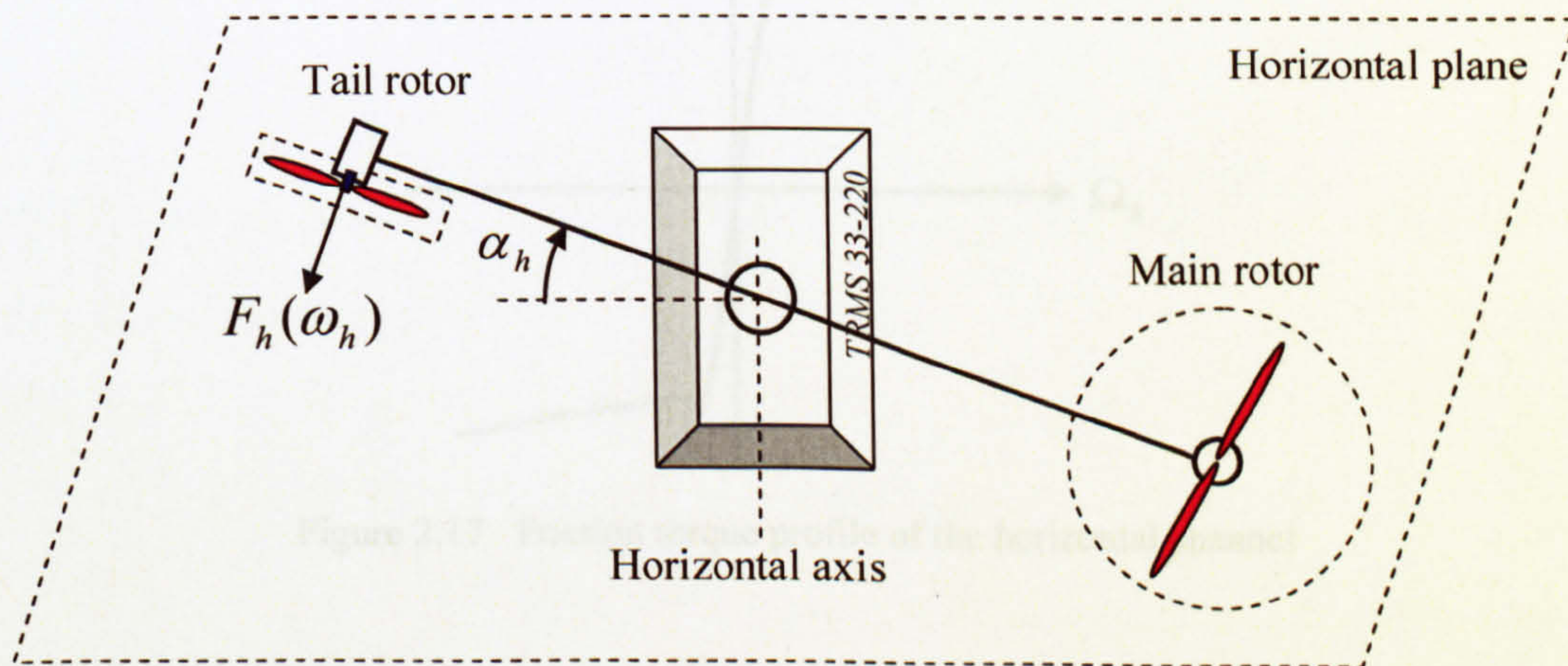


Figure 2.16 Propulsive force in the horizontal plane

2.3.7.2 Friction torque

Similar to the vertical channel, it is assumed that the friction force of the system in the horizontal plane consists of the viscous, Coulomb and static types. The moment of friction due to the angular velocity of the beam in the horizontal plane is represented as,

$$M_{h2} = k_{vfh}\Omega_h + k_{cfh}\text{sign}(\Omega_h) + \begin{cases} k_{sfh} & \text{if } \Omega_h = 0^+ \\ -k_{sfh} & \text{if } \Omega_h = 0^- \\ 0 & \text{otherwise} \end{cases} \quad (2.22)$$

where,

M_{h2} : Friction torque in the horizontal plane (Nm)

Ω_h : Angular velocity (azimuth velocity) of the TRMS beam (rad/s)

k_{vfh} : Coefficient of the viscous friction torque in the horizontal plane (Nms/rad)

k_{cfh} : Constant of the Coulomb friction torque in the horizontal plane (Nm)

k_{sfh} : Constant of the static friction torque in the horizontal plane (Nm).

The improved version of the friction torque profile can be seen in Figure 2.17 where

$$M_{h2} = M_{fric,h}.$$

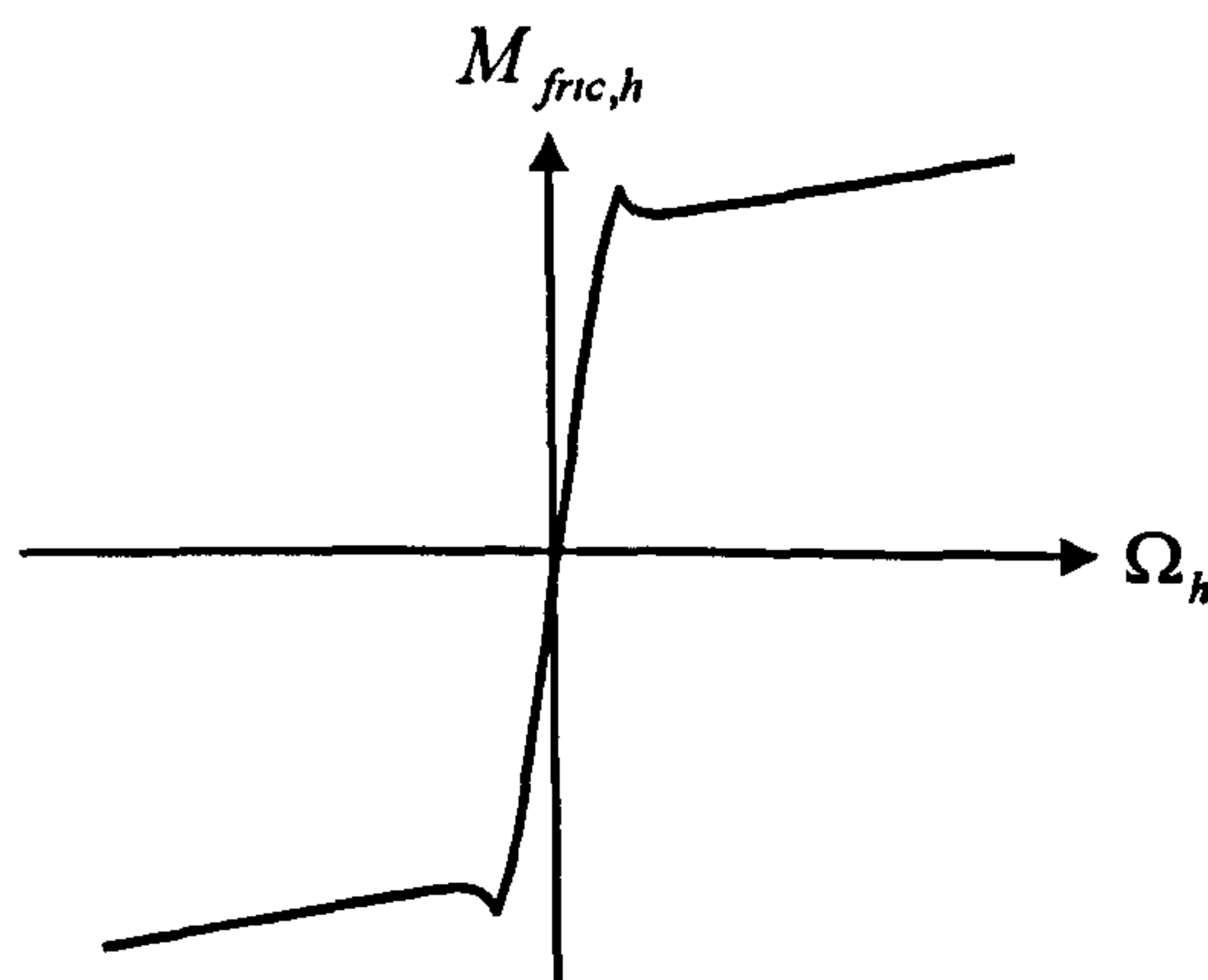


Figure 2.17 Friction torque profile of the horizontal channel

2.3.7.3 Flat cable torque

The TRMS possesses a set of flat cables that connect the power supply to the motors and also transmit the signals from measurement devices into the computer. For more accuracy of the dynamic system, the torque generated by the flat cables is considered like the torque of a spring as,

$$M_{h3} = \begin{cases} k_{chp} \alpha_h & \text{if } \alpha_h \geq 0 \\ k_{chn} \alpha_h & \text{if } \alpha_h < 0 \end{cases} \quad (2.23)$$

where α_h is the yaw angle of the beam (rad), k_{chp} and k_{chn} are constants (Nm/rad) and M_{h3} is the flat cable torque (Nm). It is of great importance to consider this torque despite of its small amplitude because it is the only resistive force in the horizontal channel which is independent of the beam speed.

2.3.7.4 Total torques in the horizontal plane

Considering Figure 2.16, the components of the moment of inertia about the vertical axis are determined as in equation (2.24). More details on moment of inertia can be found in Appendix 2.

$$J_h = D \cos^2 \alpha_v + E \sin^2 \alpha_v + F \quad (2.24)$$

where,

$$D = \left(\frac{m_m}{3} + m_{mr} + m_{ms}\right) l_m^2 + \left(\frac{m_t}{3} + m_{tr} + m_{ts}\right) l_t^2$$

$$E = \frac{m_b}{3} l_b^2 + m_{cb} l_{cb}^2$$

$$F = m_{ms} r_{ms}^2 + \frac{m_{ts}}{2} r_{ts}^2$$

Again, based on Newton's second law of motion, $\sum_{i=1}^3 M_{hi} = J_h \frac{d^2 \alpha_h}{dt^2}$, the following equations can be written.

$$\frac{d\Omega_h}{dt} = \frac{l_t F_h(\omega_h) \cos \alpha_v - M_{fric,h} - M_{h3}}{D \cos^2 \alpha_v + E \sin^2 \alpha_v + F} \quad (2.25)$$

$$\frac{d\alpha_h}{dt} = \Omega_h \quad (2.26)$$

where,

J_h : Moment of inertia about the vertical axis (kg m^2).

The complete block diagram of the horizontal 1DOF TRMS model is shown in Figure 2.18. Figure 2.19 shows the inside of the nonlinear equations block based on equations (2.24) and (2.25). The control voltage U_h is converted to the input of the tail motor, V_h , which drives the rotor.

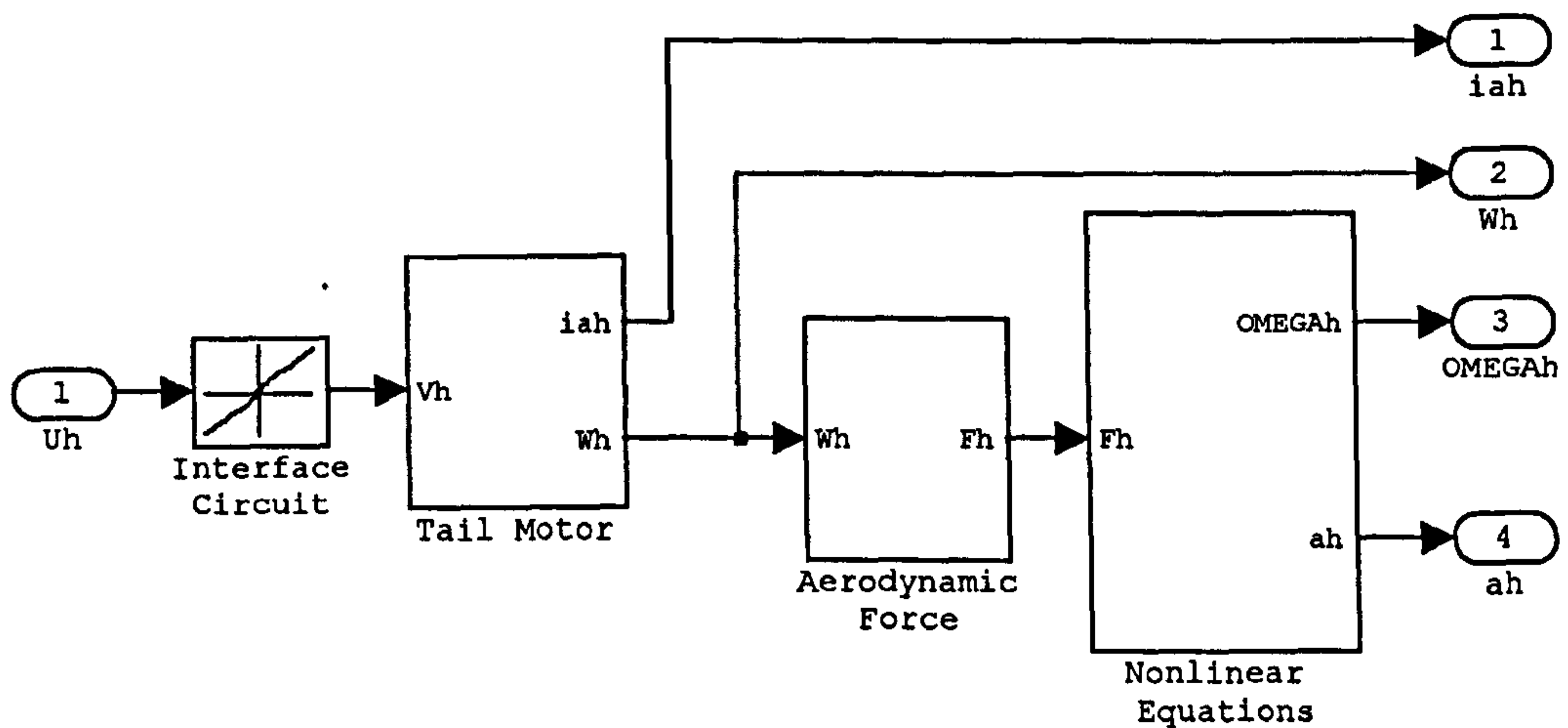


Figure 2.18 The complete block diagram of the horizontal 1DOF TRMS model

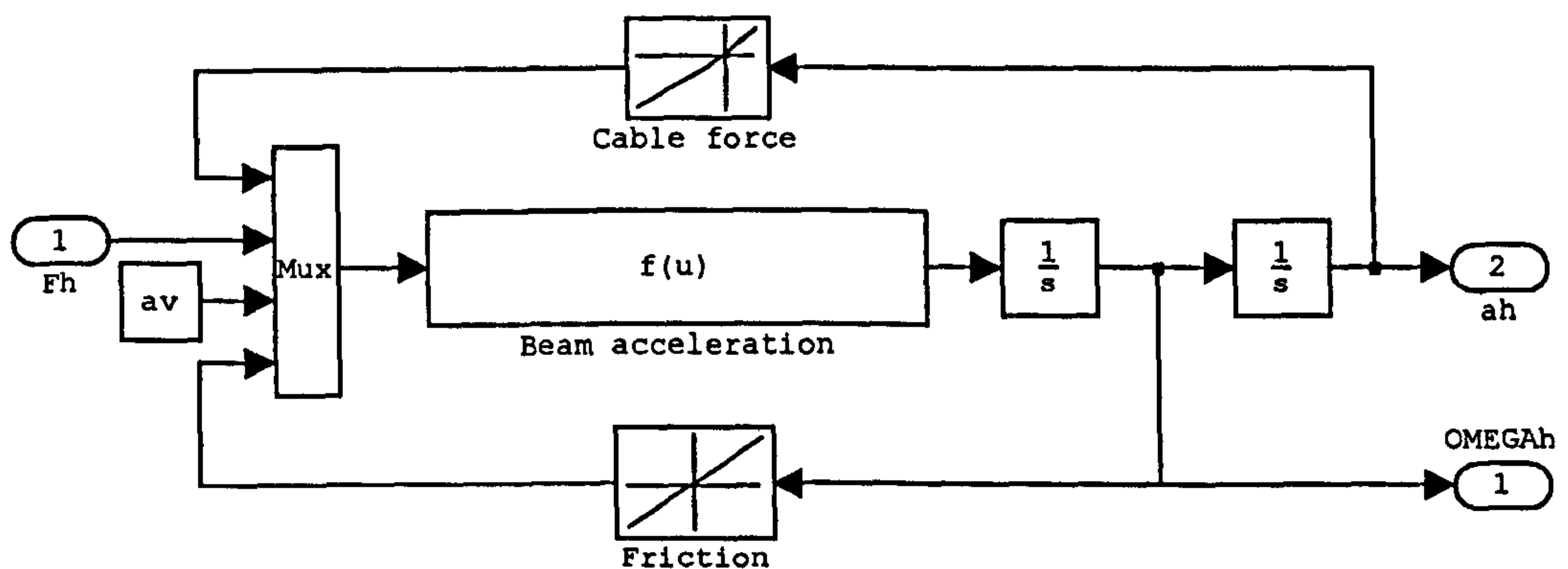


Figure 2.19 The inside of the 'nonlinear equations' block

2.3.8 Modelling of the 2DOF TRMS

All the equations related to gravity, propulsive and friction forces in terms of the 2DOF dynamics of the system are the same as that of the 1DOF dynamics. In this section, the torques produced by the cross-coupling of both channels are presented.

2.3.8.1 Centrifugal torque

Due to the rotation of the beam in the horizontal plane, a centrifugal force affects the beam angle in the vertical plane. The moment of the centrifugal force can be expressed as equation (2.27). More explanation about this equation can be found in Appendix 2.

$$M_{v4} = -\Omega_h^2 \left\{ \left(\frac{m_t}{2} + m_{tr} + m_{ts} \right) l_t^2 + \left(\frac{m_m}{2} + m_{mr} + m_{ms} \right) l_m^2 - \left(\frac{m_b}{2} l_b^2 + m_{cb} l_{cb}^2 \right) \right\} \sin \alpha_v \cos \alpha_v \quad (2.27)$$

This can be rewritten as,

$$M_{v4} = -\Omega_h^2 H \sin \alpha_v \cos \alpha_v \quad (2.28)$$

where, M_{v4} is the centrifugal torque in the vertical plane (Nm) and,

$$H = A l_t + B l_m - \frac{m_b}{2} l_b^2 - m_{cb} l_{cb}^2$$

2.3.8.2 Gyroscopic torque

Let us consider the rotating propeller of the main rotor as a spinning disc. Whilst the mentioned disc is rotating in the horizontal plane, and since the angular velocity of the beam in the horizontal plane is considerably lower than the main rotor rotational velocity, the gyroscopic effect can be expressed as the following simplified equation [96],

$$M_{v5} = k_g F_v(\omega_v) \Omega_h \cos \alpha_v \quad (2.29)$$

where,

k_g : Gyroscopic constant (ms/rad)

M_{v5} : Gyroscopic torque (Nm).

2.3.8.3 The effect of the tail motor rotation on the vertical angle

This term represents the effect of the tail propeller speed on the vertical plane movement of the beam, and can be modelled as equation (2.30). It implies that a constant tail motor speed produces no force on the vertical plane, and only if the tail motor accelerates, the acceleration produces a force on the vertical plane. This effect has been observed experimentally.

$$\Omega_{vh} = k_t \omega_h \quad (2.30)$$

where,

Ω_{vh} : The effect of the tail propeller speed on the vertical plane movement of the beam (rad/s)

k_t : Constant.

2.3.8.4 Total torques in the vertical plane

From Figure 2.12, the components of the moment of inertia about the horizontal axis have been obtained as in equation (2.18).

According to Newton's second law of motion, it is clear that, $\sum_{i=1}^5 M_{vi} = J_v \frac{d^2 \alpha_v}{dt^2}$, and the following equations can be written,

$$\frac{dS_v}{dt} = \frac{l_m F_v(\omega_v) - M_{fric,v} + g[(A - B)\cos\alpha_v - C\sin\alpha_v] - 0.5\Omega_h^2 H \sin 2\alpha_v}{J_v} + \frac{k_g F_v(\omega_v)\Omega_h \cos\alpha_v}{J_v} \quad (2.31)$$

$$\Omega_v = S_v + \frac{k_t \omega_h}{J_v} \quad (2.32)$$

$$\frac{d\alpha_v}{dt} = \Omega_v \quad (2.33)$$

where,

S_v : Angular velocity of the TRMS beam in the vertical plane without the effect of the horizontal channel (rad/s).

2.3.8.5 The effect of the main motor rotation on the horizontal angle

This term denotes the effect of the main propeller speed on the horizontal plane movement of the beam, and can approximately be modelled as,

$$\Omega_{hv} = k_m \omega_v \cos\alpha_v \quad (2.34)$$

where,

Ω_{hv} : The effect of the main propeller speed on the horizontal plane movement of the beam (rad/s)

k_m : Constant.

2.3.8.6 Total torques in the horizontal plane

Considering Figure 2.16, the components of the moment of inertia about the vertical axis can be expressed as in equation (2.24).

Again, from Newton's second law of motion, $\sum_{i=1}^3 M_{hi} = J_h \frac{d^2 \alpha_h}{dt^2}$, the following equations can be written,

$$\frac{dS_h}{dt} = \frac{l_t F_h(\omega_h) \cos \alpha_v - M_{fric,h} - M_{cable}(\alpha_h)}{D \cos^2 \alpha_v + E \sin^2 \alpha_v + F} \quad (2.35)$$

$$\Omega_h = S_h + \frac{k_m \omega_v \cos \alpha_v}{D \cos^2 \alpha_v + E \sin^2 \alpha_v + F} \quad (2.36)$$

$$\frac{d\alpha_h}{dt} = \Omega_h \quad (2.37)$$

where,

S_h : Angular velocity of the TRMS beam in the horizontal plane without the effect of the vertical channel (rad/s).

The complete block diagram of the 2DOF TRMS model is shown in Figure 2.20.

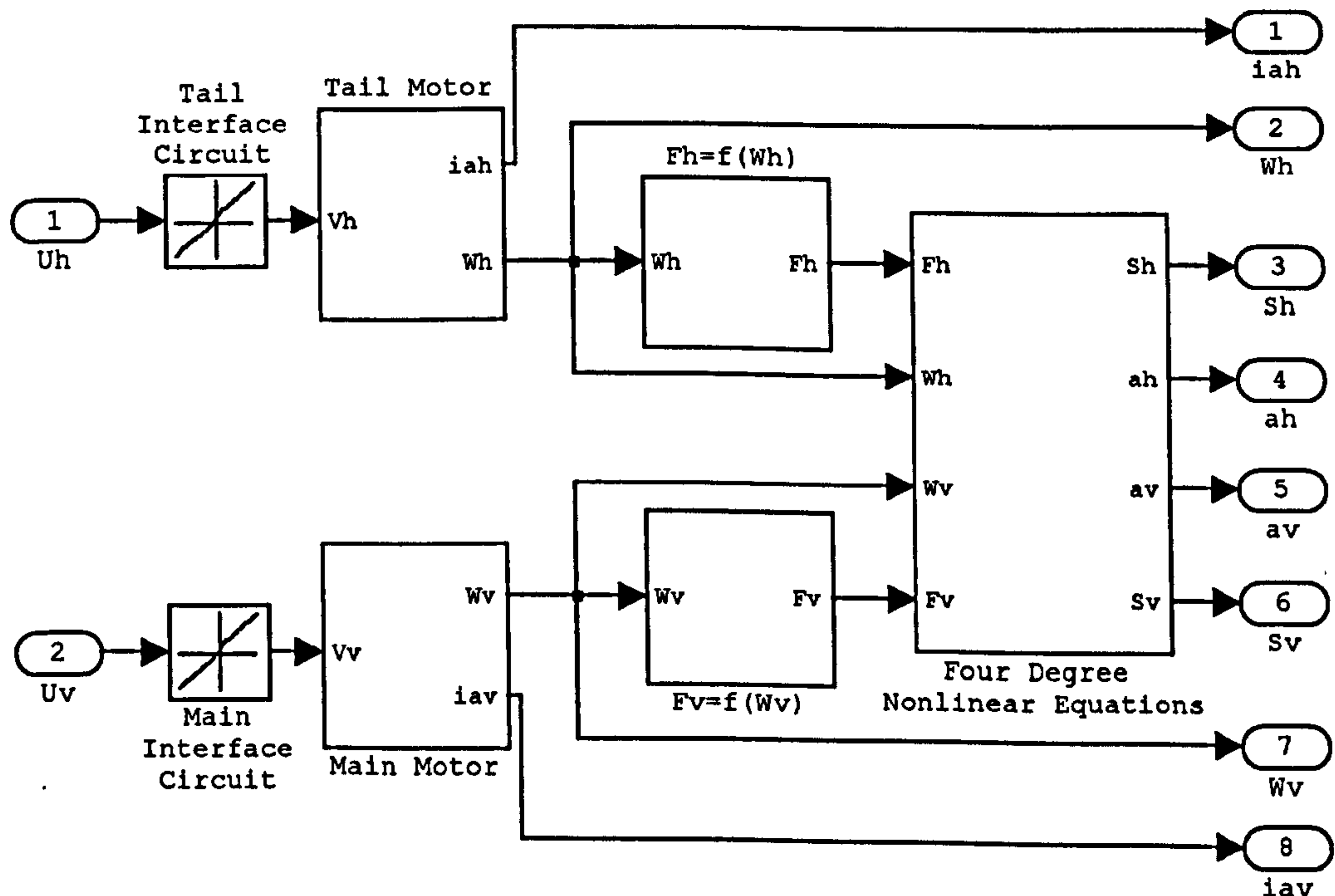


Figure 2.20 The complete block diagram of dynamic simulation of the TRMS

Figure 2.21 shows the inside of the 'four-degree nonlinear equations' block according to equations (2.31) to (2.33) and also (2.35) to (2.37). The control voltages U_h and U_v are input signals that are proportional to the terminal voltages of the DC motors. Clearly, the overall model is a two-input, two-output system with 8 state variables.

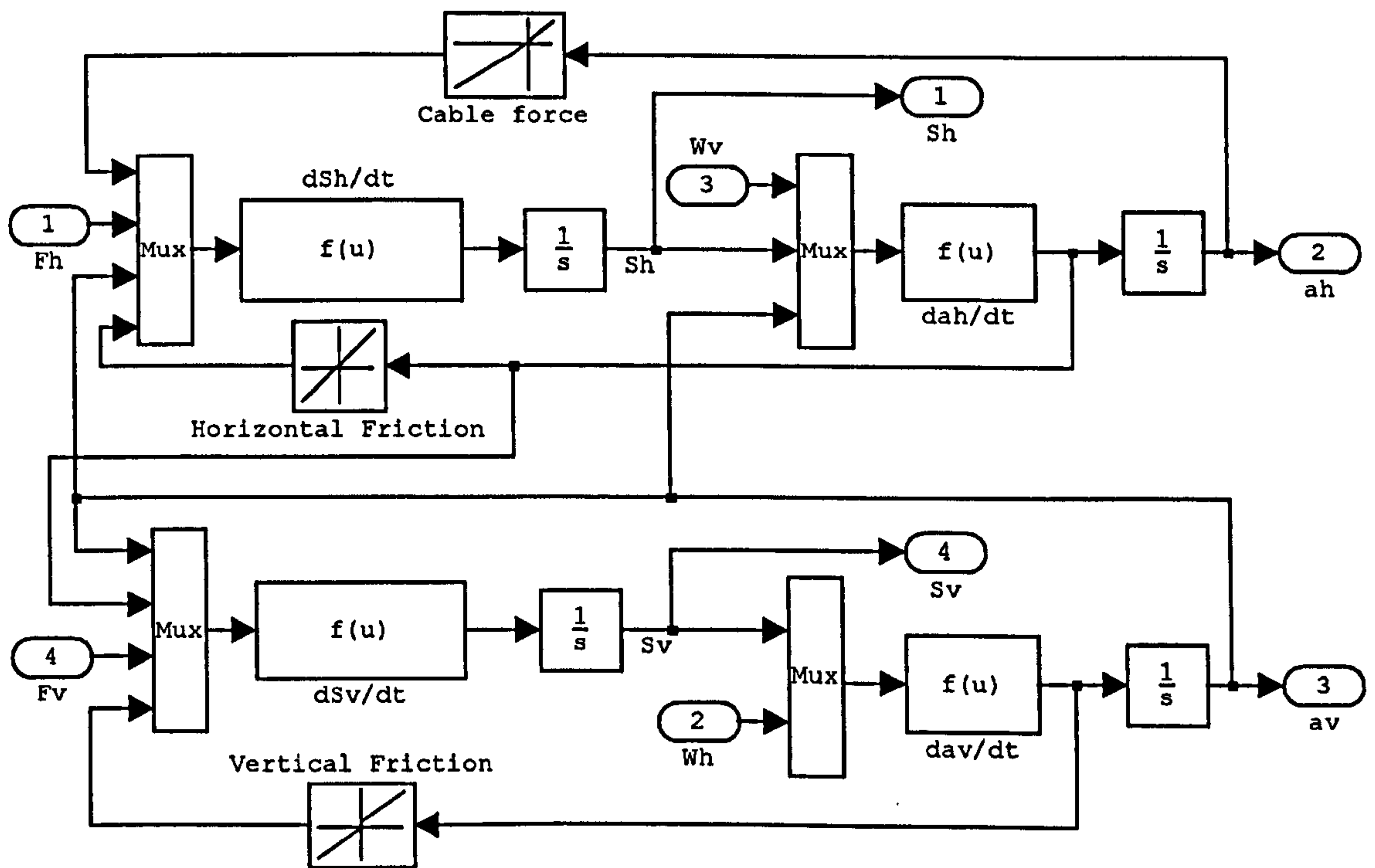


Figure 2.21 Inside the 'four-degree nonlinear equations' block

2.4 Results of the Newtonian-based Modelling

The position of the minimum energy of the TRMS (equilibrium point) is assumed to be the reference point of both the horizontal (yaw) and vertical (pitch) angles. At the equilibrium point, the angle between the tower and the counterweight beam in the vertical plane is -0.6048 radians (-34.67 degrees) therefore, this discrepancy should be considered. According to equation (2.31) the angle between the tower and the counterweight beam at the equilibrium point can be calculated as,

$$(A - B) \cos(\alpha_{v_0}) - C \sin(\alpha_{v_0}) = 0$$

$$\alpha_{v_0} = \tan^{-1}((A - B)/C) \quad (2.38)$$

Note that this angle may be varied by changing the position of the counter-weight along the counter-weight beam. Figures 2.22 and 2.23 show the sign of directions and the zero position of the pitch and yaw angles respectively. Figure 2.22 is the front view of the TRMS and Figure 2.23 shows its top view.

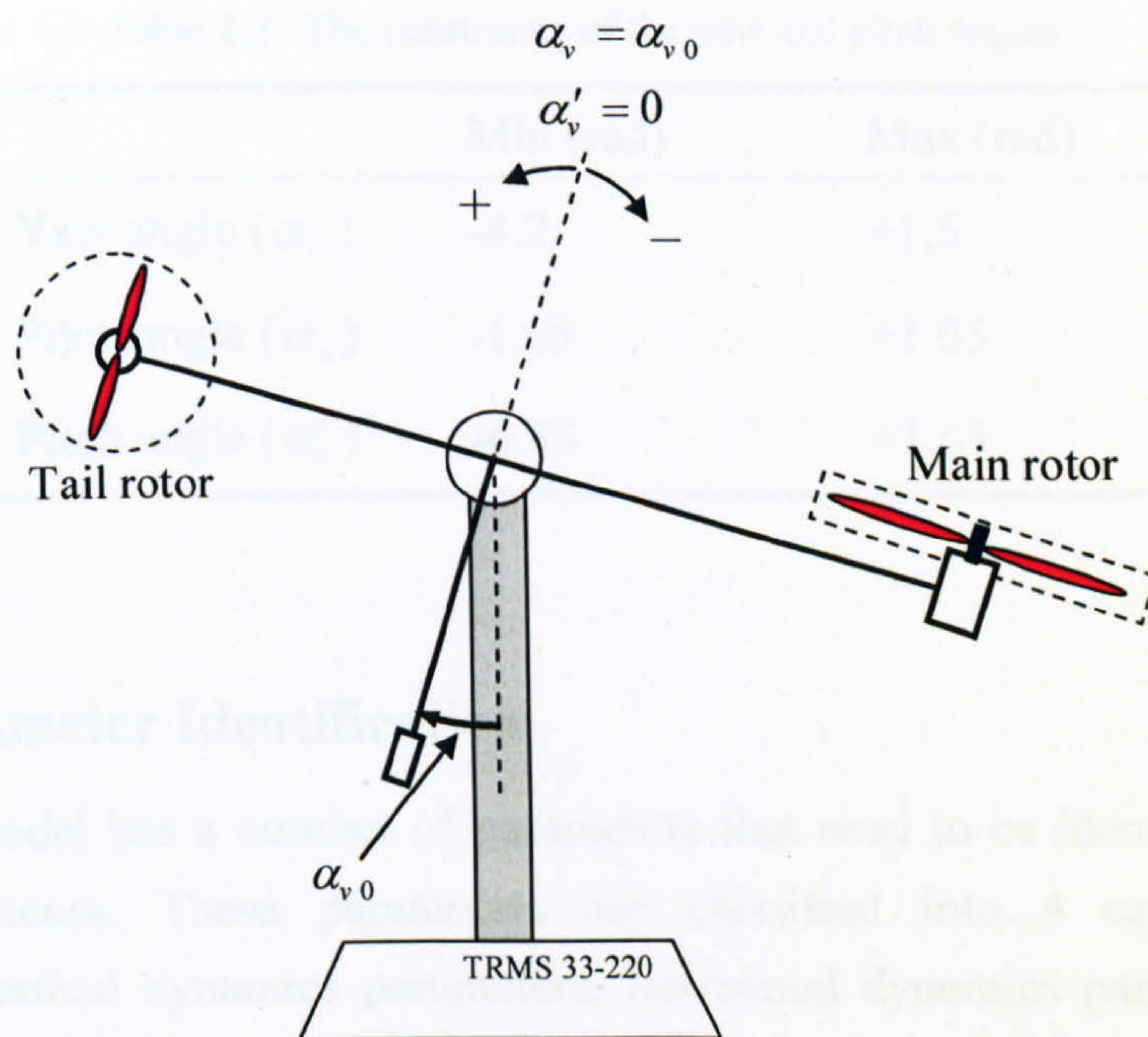


Figure 2.22 The front view of the TRMS and the pitch angle reference point

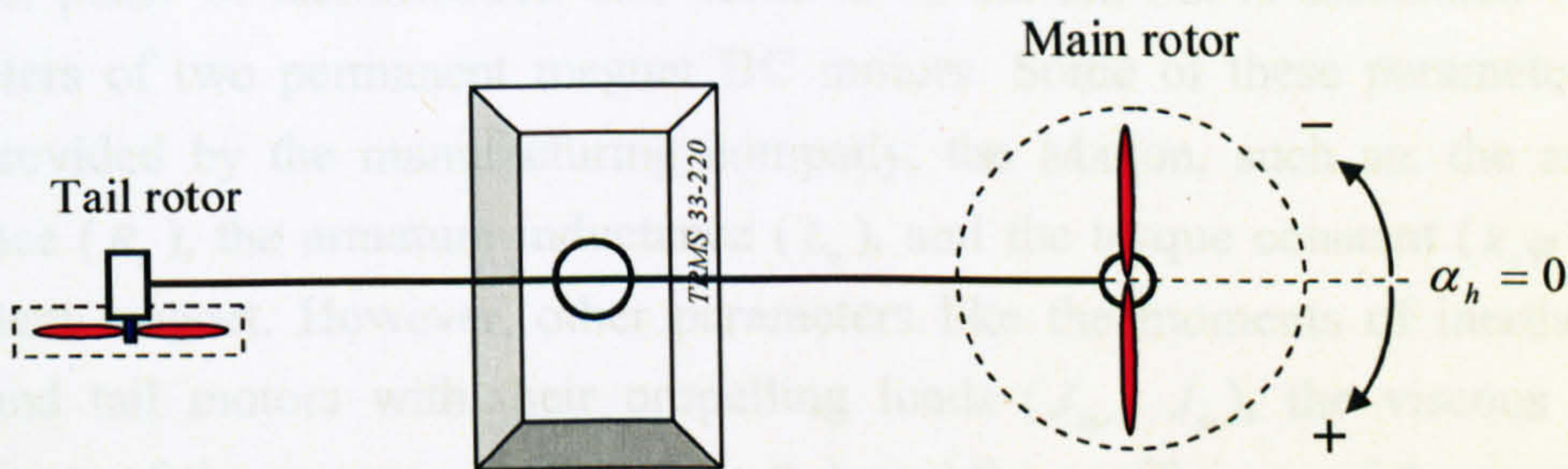


Figure 2.23 The Top view of the TRMS and the yaw angle reference point

The initial values of the two shaft encoders are set to the initial positions of the beam. For instance, if the initial position of the vertical angle is 30° , the zero of the vertical shaft encoder is adapted on this angle. It is also clear from the results that the response of the real TRMS always starts from zero for both the vertical and horizontal angles. Consequently, to obtain a consistent model of the real TRMS, one needs to substitute α_v in all equations by $\alpha_v = \alpha'_v + \alpha_{v0}$.

Therefore, the outputs of the model will be α'_v and α_h instead of α_v and α_h . Additionally, the yaw and pitch angles are mechanically restricted between their corresponding upper and lower limits. The minimum and maximum angles that the beam can reach are listed in Table 2.2 for both the channels.

Table 2.2 The constraints of the yaw and pitch angles

	Min (rad)	Max (rad)
Yaw angle (α_h)	-4.2	+1.5
Pitch angle (α_v)	-1.05	+1.05
Pitch angle (α'_v)	-0.45	+1.65

2.4.1 Parameter Identification

The TRMS model has a number of parameters that need to be identified during the modelling process. These parameters are classified into 4 categories: motor parameters, vertical dynamics parameters, horizontal dynamics parameters and the parameters related to the interaction between the two channels.

The first phase of identification that needs to be carried out is associated with the parameters of two permanent magnet DC motors. Some of these parameters have been provided by the manufacturing company, the Maxon, such as: the armature resistance (R_a), the armature inductance (L_a), and the torque constant ($k_a\phi$) of the permanent magnet. However, other parameters like the moments of inertia of the main and tail motors with their propelling loads (J_{mr} , J_{tr}), the viscous friction coefficients of the motors rotation (B_{mr} , B_{tr}), and the coefficients of the motors load torque (k_{tvp} , k_{tvn} , k_{thp} , k_{thn}), that are different in negative and positive rotational velocity ranges due to the asymmetrical propellers need to be identified. It is noted that the rotational velocity of the motors can be measured using two tachogenerators attached to the shafts of the motors. The existence of the tachogenerators facilitates the motors parameters identification process. To this end, a step input voltage was applied to the main motor terminals and subsequently the rotational velocity of the motor started to increase and became constant after a while. It is clear from equations (2.1) to (2.5) that the moment of inertia (J_{mr}) affects only the transient response and other parameters (B_{mr} , k_{tvp} and k_{tvn}) have effect on both transient and steady state responses. First, a positive step voltage has been applied to the main motor and according to the steady state rotational velocity of the motor, two parameters, B_{mr} and k_{tvp} , have been identified. Note that the coefficient k_{tvp} has a dominant effect at high speed range and B_{mr} shows more influence at low speed region. It stems from the fact that B_{mr} is the coefficient of the motor speed and k_{tvp} is the coefficient of the motor speed squared. The same procedure was followed for the negative region to obtain k_{tvn} . As pointed out earlier, the moment of inertia (J_{mr}) influences the transient

response and it should definitely be greater than the moment of inertia of the motor alone, supplied by the manufacture. Therefore, J_{mr} can be defined by comparing the rise time of the real and model rotational velocities because, it is proportional to the rise time. The parameters of the tail motor have been achieved in the same manner. Note that it is not straightforward to theoretically calculate the moment of inertia related to the propeller and therefore it should be experimentally identified.

The next step is to find the parameters that relate the rotational velocity of the motors and the aerodynamic forces, according to equations (2.11) and (2.12), i.e. k_{fvp} , k_{fvn} , k_{fhp} , k_{fhn} . For instance, in order to identify the coefficient of the aerodynamic force due to the main motor positive rotation, k_{fvp} , one can apply a positive voltage to the motor, and measure the resultant rotational velocity and also the resultant force applied to the beam (exactly at the point that the motor is located) in the vertical plane using a force sensor. However, in the case of the TRMS this force is low and an accurate and sensitive low range force sensor is required. It is recommended that the measurement is carried out with several rotational velocities for more accuracy as it has been done in this investigation. The same procedure has been adopted to measure other similar parameters, i.e. k_{fvn} , k_{fhp} , k_{fhn} .

The remaining parameters which required to be identified to complete the 1DOF horizontal model are the coefficients of the flat cable model (k_{chp} , k_{chn}), and the friction parameters. As mentioned previously, the flat cable is modelled like a spring. To identify these parameters, one can manually change the position of the beam in the horizontal plane while the TRMS is unexcited, and measure the applied torque due to the flat cable (applied force multiply by the distance between the measurement point and the pivot point) and the corresponding angle of the beam. The coefficient can be calculated as the measured torque divided by the measured angle of the beam. This method is used for the negative angle region; however, again it is suggested to repeat the measurement for several positions of the beam and fit a linear or even nonlinear curve on the measured data. The measured values are listed in Table A1.6 of Appendix 1. To validate the flat cable coefficients, one can manually displace the TRMS beam to an arbitrary initial angle and release it without any external force, the beam will oscillate and eventually stop along the equilibrium point, i.e. the beam comes to rest at a position of the minimum energy. By comparing the results of the real TRMS and the plant-model, the accuracy of the coefficients can be verified and even improved.

In order to find the friction parameters, it is primitively assumed that the only effective friction on the system is viscous friction, which is proportional to the beam velocity. Increasing the viscous friction coefficient causes the overshoot of the beam angle to decrease. Comparing the model and the real responses, the parameters can be estimated. Then, the constant of Coulomb friction is obtained by further consideration of the model response.

Note that the flat cables have no considerable effect on the 1DOF vertical model. The friction parameters of the 1DOF vertical model are identified using the same method as used in the case of the horizontal model.

The parameters related to the interaction between the two channels (k_t , k_m) and the gyroscopic constant, (k_g) are the last parameters need to be obtained. For identification the parameter associated with the effect of the tail motor rotation on the vertical angle (k_t), only the tail motor is excited and because the main motor is left unexcited, the movement of the beam in the vertical plane is solely due to the tail motor rotation. Comparing the simulation and the real TRMS responses can help to find the parameter. It is clear that the gyroscopic force has no effect on the system in this situation as the propulsive force of the main motor is zero. The same technique is used for the identification of k_m . The gyroscopic constant is obtained by exciting both motors and changing the parameter to make the model response as close as possible to the real response.

2.4.2 Results of the Vertical 1DOF TRMS Model

The response of the developed vertical 1DOF model is presented in Figures 2.24 to 2.32. In order to validate the models, they have been assessed with various input signals. The results here have been selected in order to illustrate the performance of the models in various operating regions without repeating the results with similar characteristics. The PSD test is used for further validation. The PSD can be considered to be an estimation of the Fourier transform technique. The overlapping test is carried out to validate the model performance in the time domain and the PSD is utilised to show the performance of the model in the frequency domain. The Welch's method has been employed to find the PSD of the signals. The corresponding function is available in MATLAB.

Figure 2.24 shows the overlapping and PSD tests of the 1DOF vertical model where

the input signal is a sine wave with a frequency of 0.02 Hz and an amplitude of 1.5 V. It is clear from the results that the model responses closely followed the real TRMS responses in the case of both the overlapping and PSD tests. Note that due to the system nonlinearity, the model has been evaluated using several input signals with various frequencies and amplitudes. Changing the frequency, amplitude and even the shape of the input signals shows that the model response intimately matches the real system response and therefore, the model adequately represents the TRMS in terms of its 1DOF vertical dynamics. It is clear from the results that the system is nonlinear because with the variation of the frequency and/or the amplitude of the input signals, the shape of the output response completely changes (compare Figures 2.29 and 2.31). Also, the nonlinearity of the system is evident from Figure 2.29 in which the positive and negative parts of the pitch angle response are different.

The model has been tested in the range of frequencies less than 1 Hz because for input signals with frequencies greater than 1 Hz, the beam has no considerable movement. It is because the dominant time constant of the motor is considerably greater than 1 second and if the polarity of the input signal changes in less than 1 second (e.g. 0.5 seconds or less) the motor cannot rotate in any direction and only vibrates in its initial position. The PSD tests with respect to various input signals reveals that there are at least two dominant peaks in each figure, one is related to the frequency of the excitation signal and the other is the dominant natural frequency of the system. Therefore, the dominant natural frequency of the vertical channel occurs at about 0.4 Hz.

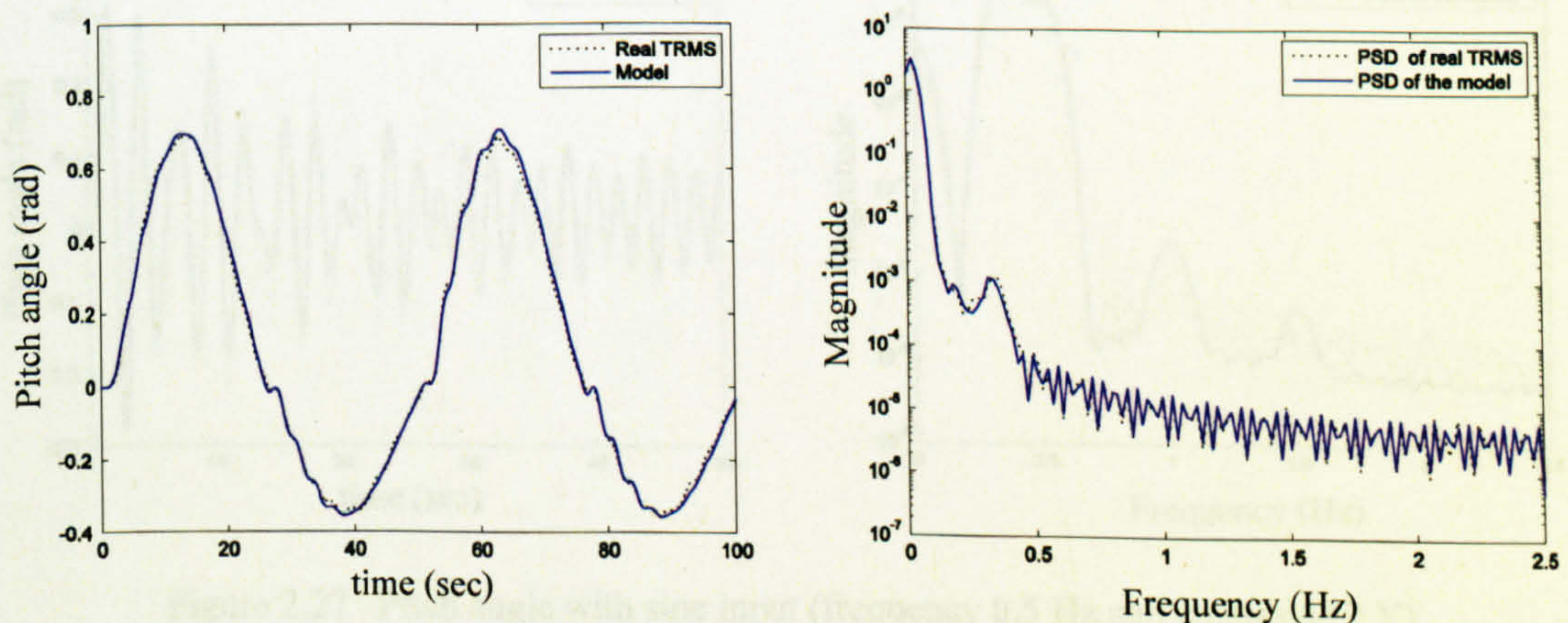


Figure 2.24 Pitch angle with sine input (frequency 0.02 Hz and amplitude 1.5 V)

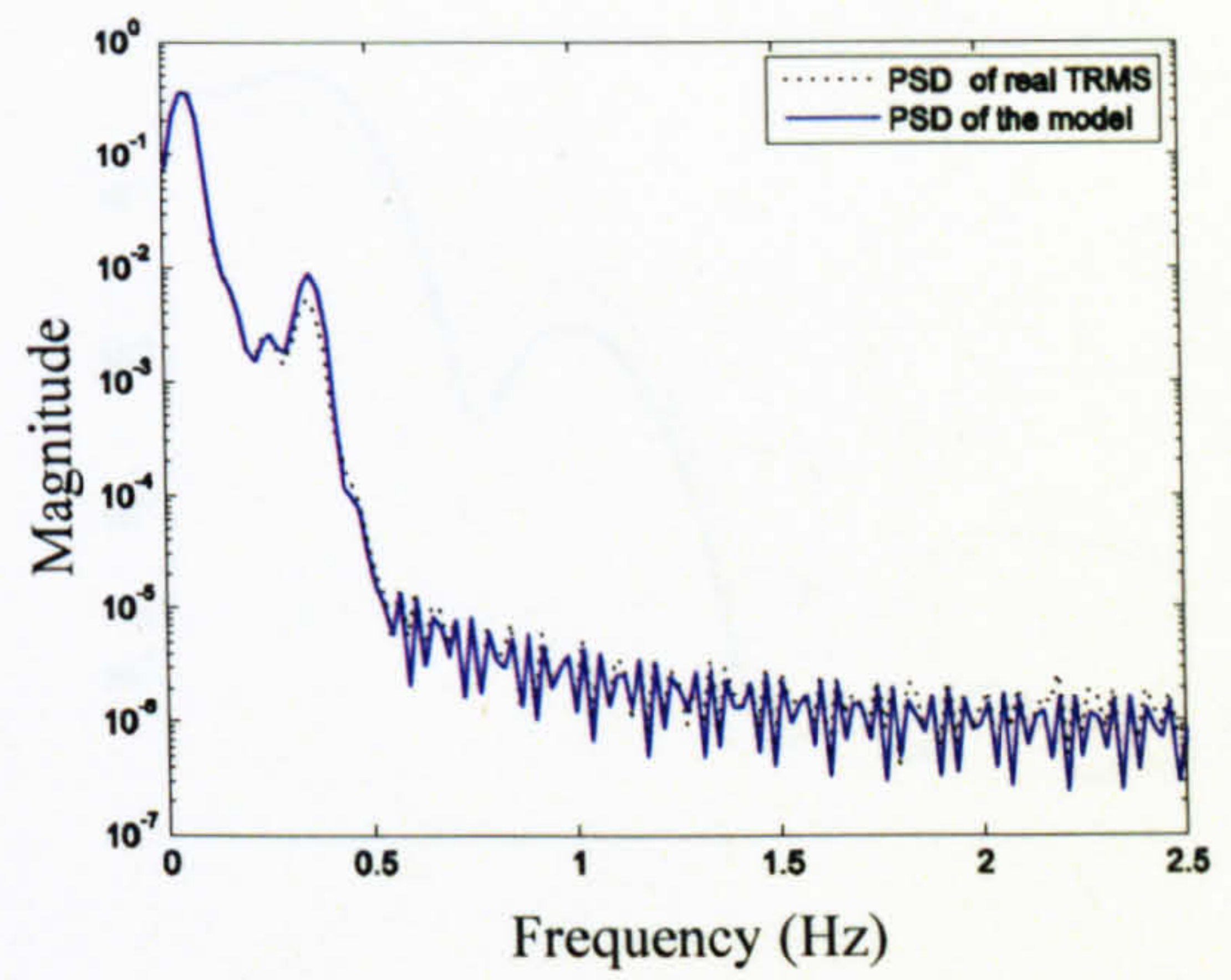
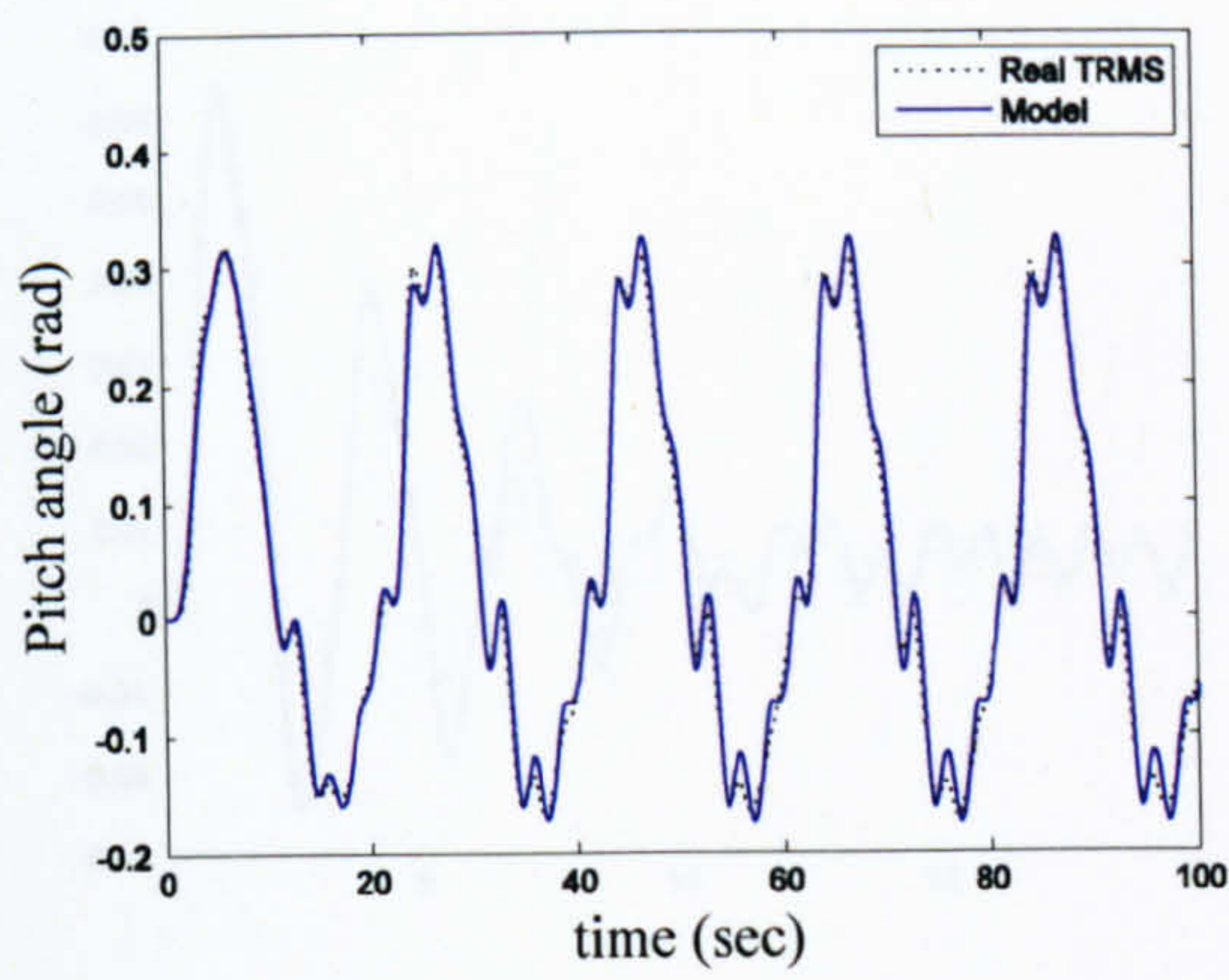


Figure 2.25 Pitch angle with sine input (frequency 0.05 Hz and amplitude 0.8 V)

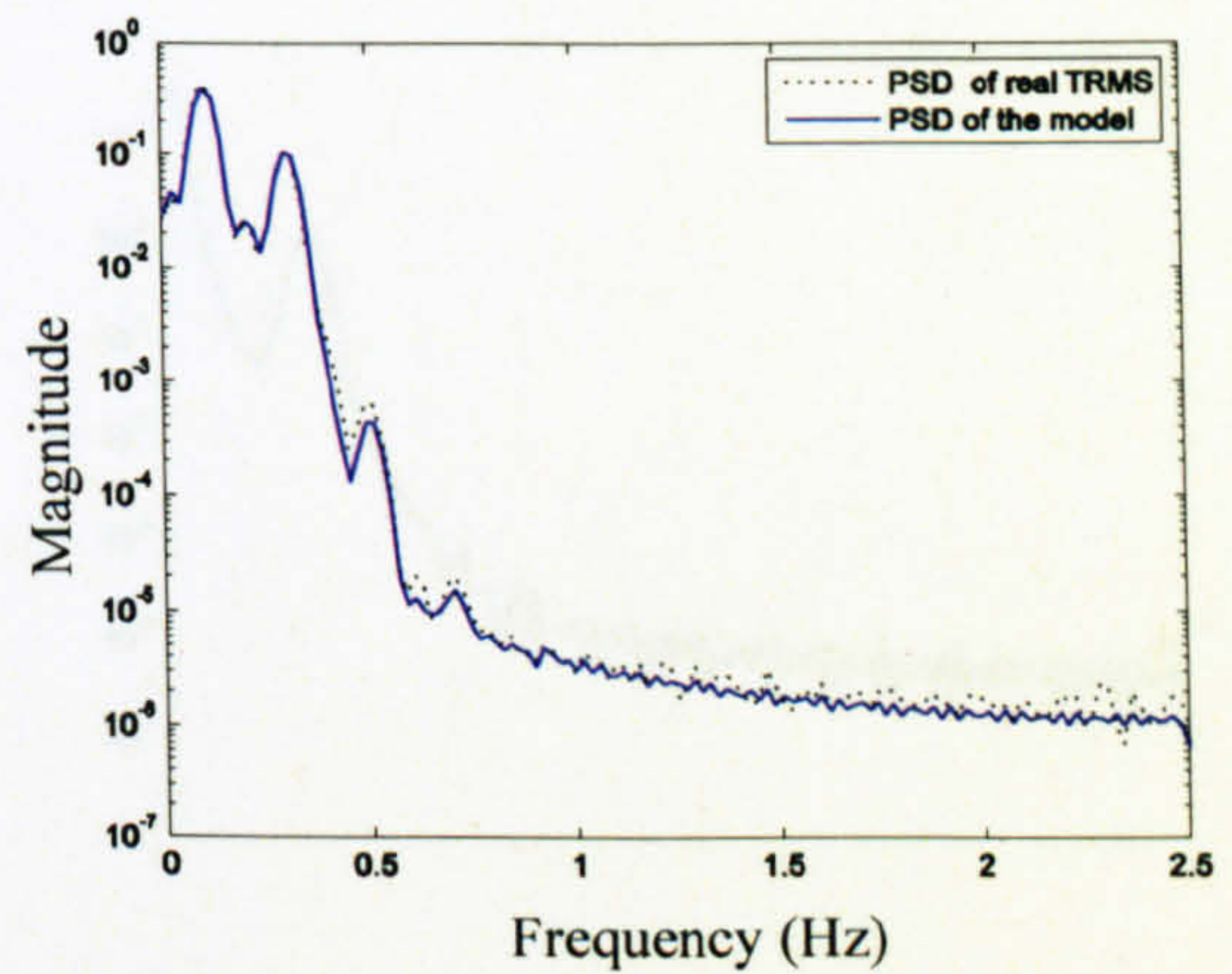
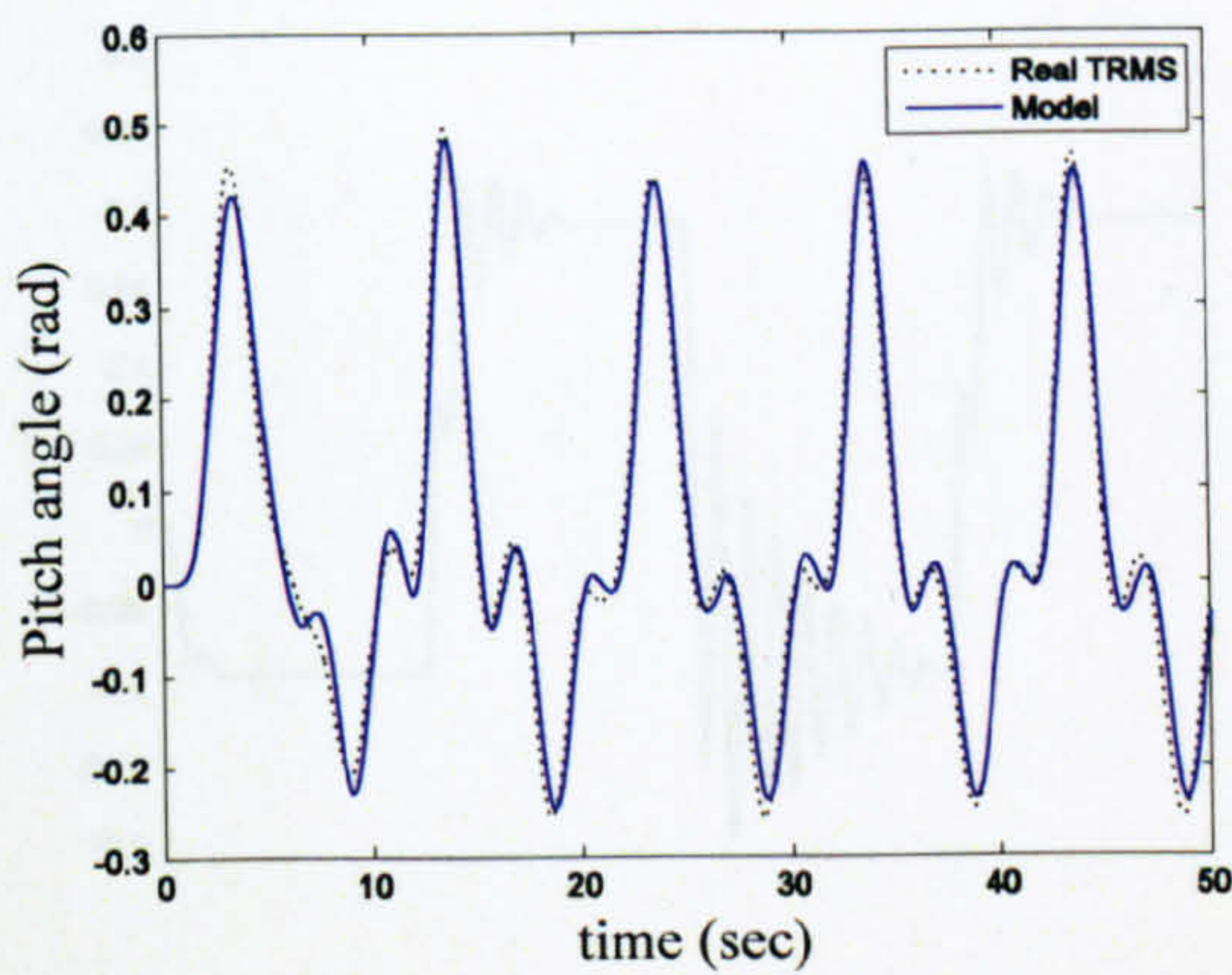


Figure 2.26 Pitch angle with sine input (frequency 0.1 Hz and amplitude 1 V)

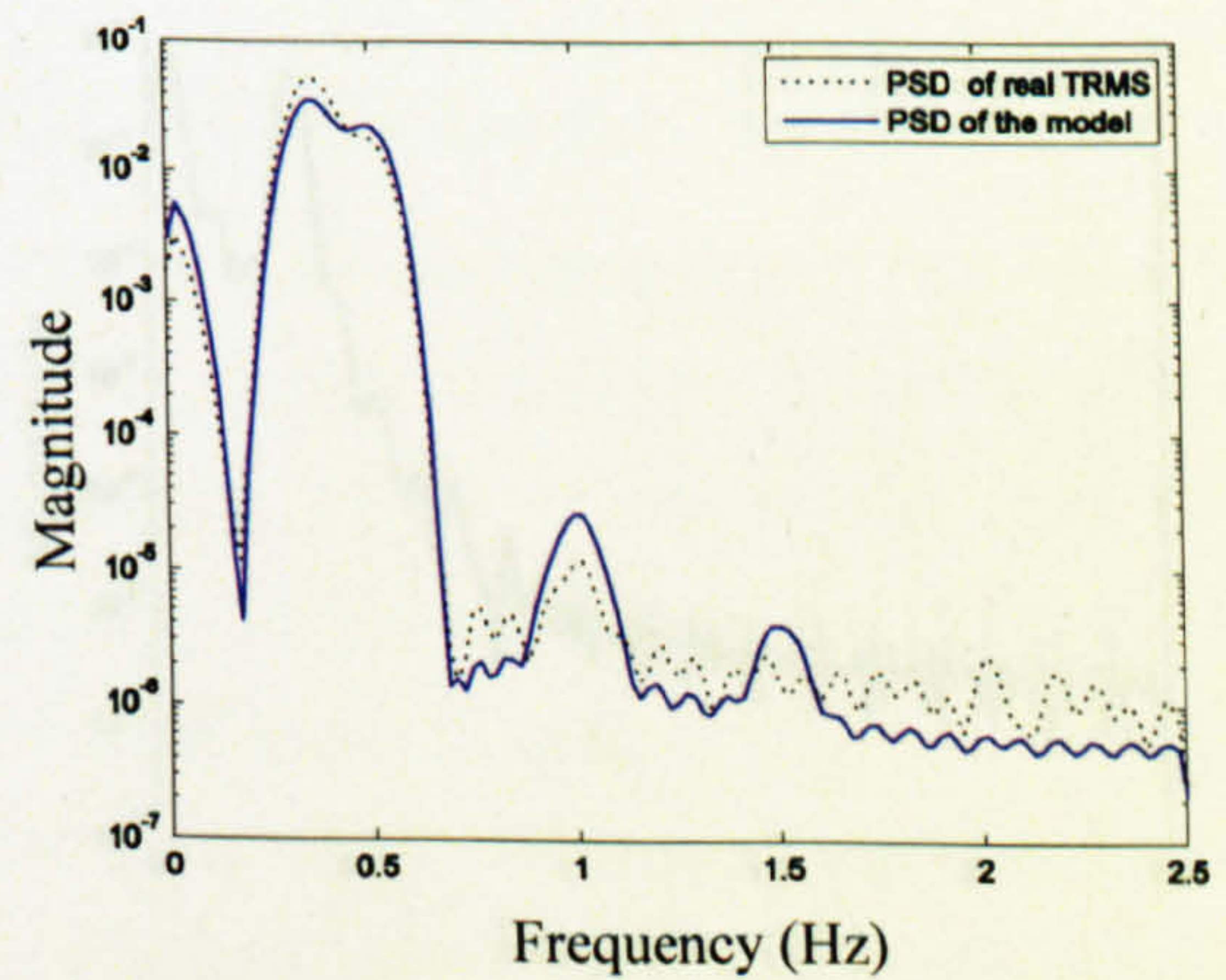
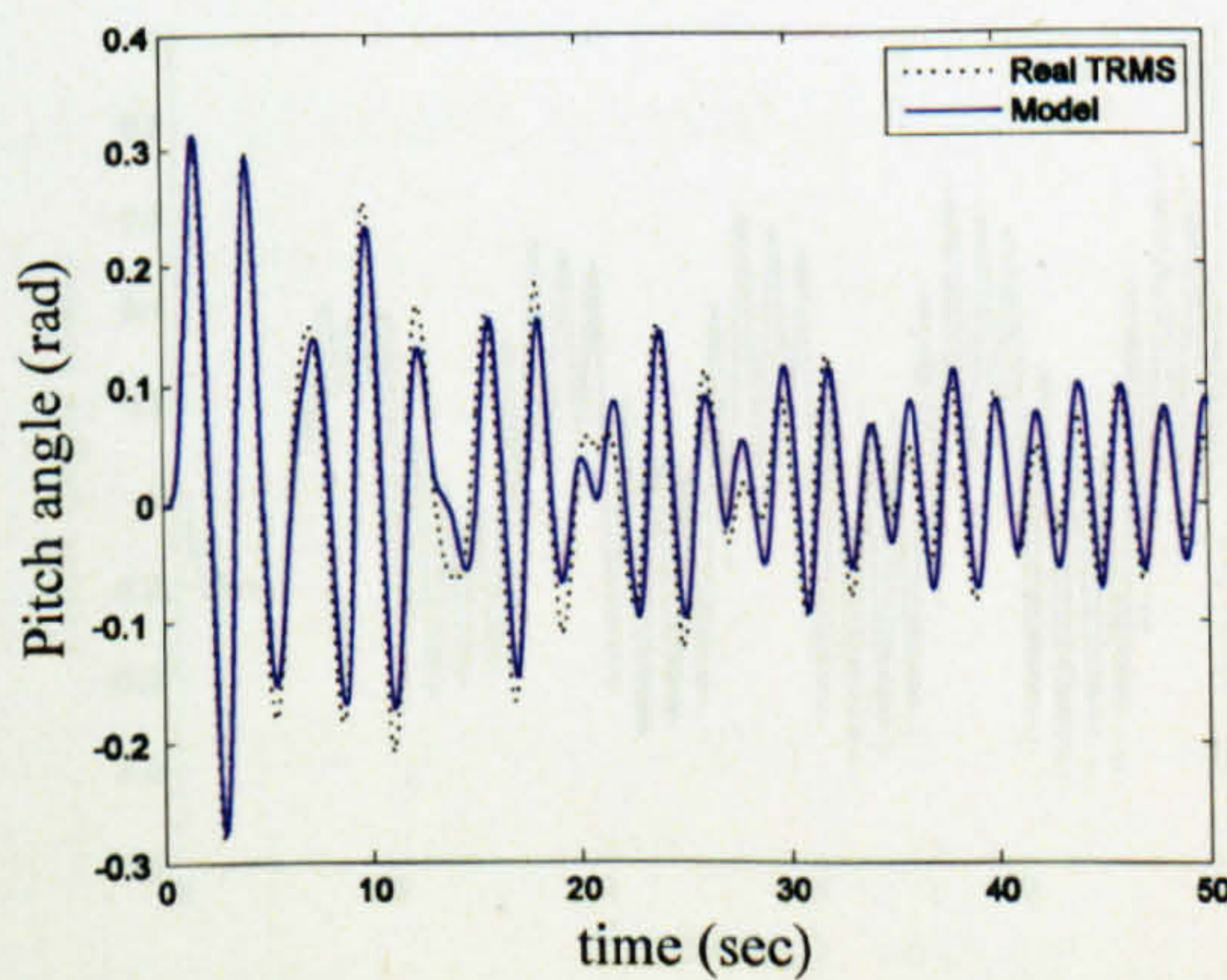


Figure 2.27 Pitch angle with sine input (frequency 0.5 Hz and amplitude 2 V)

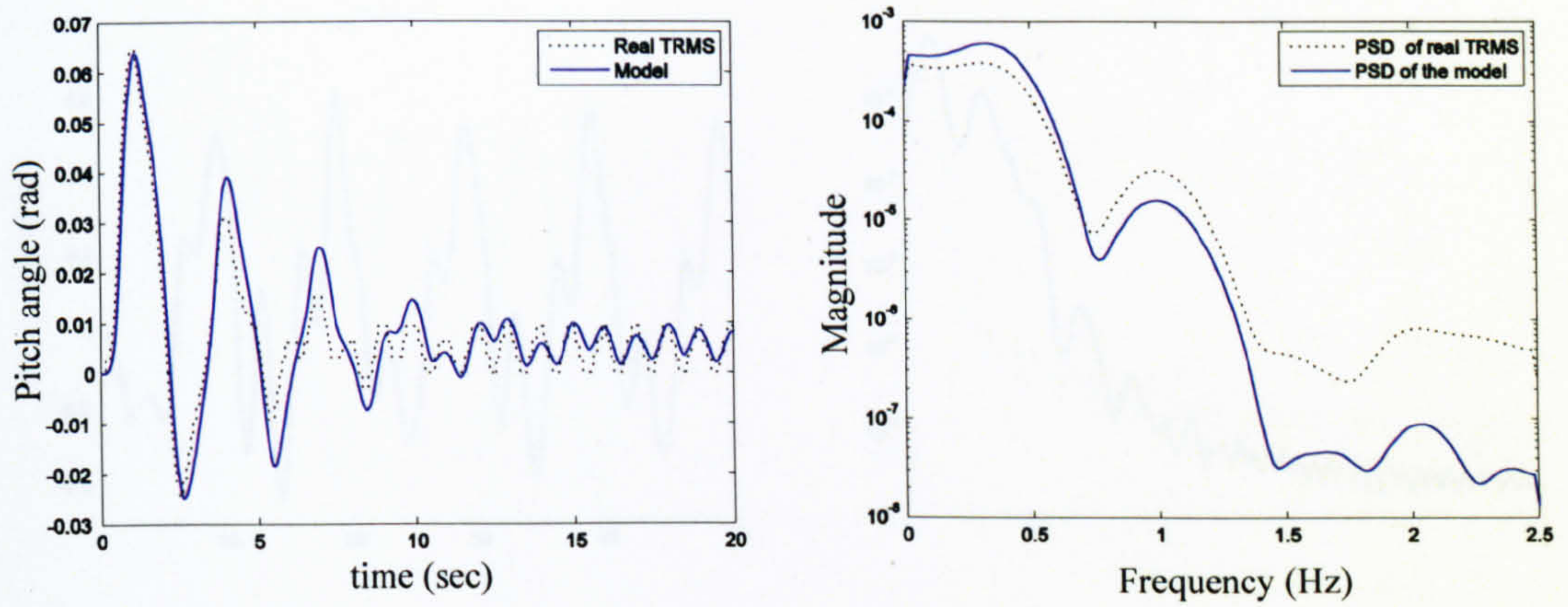


Figure 2.28 Pitch angle with sine input (frequency 1 Hz and amplitude 2.5 V)

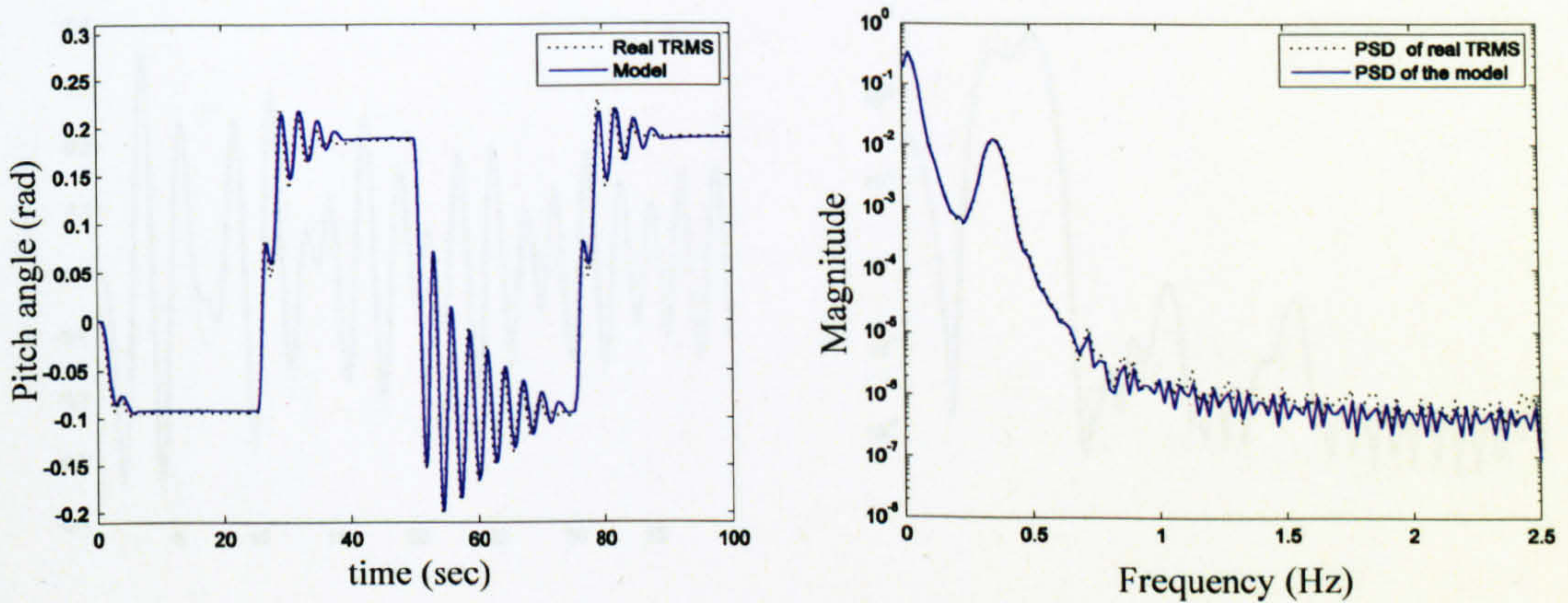


Figure 2.29 Pitch angle with square input (frequency 0.02 Hz and amplitude 0.5 V)

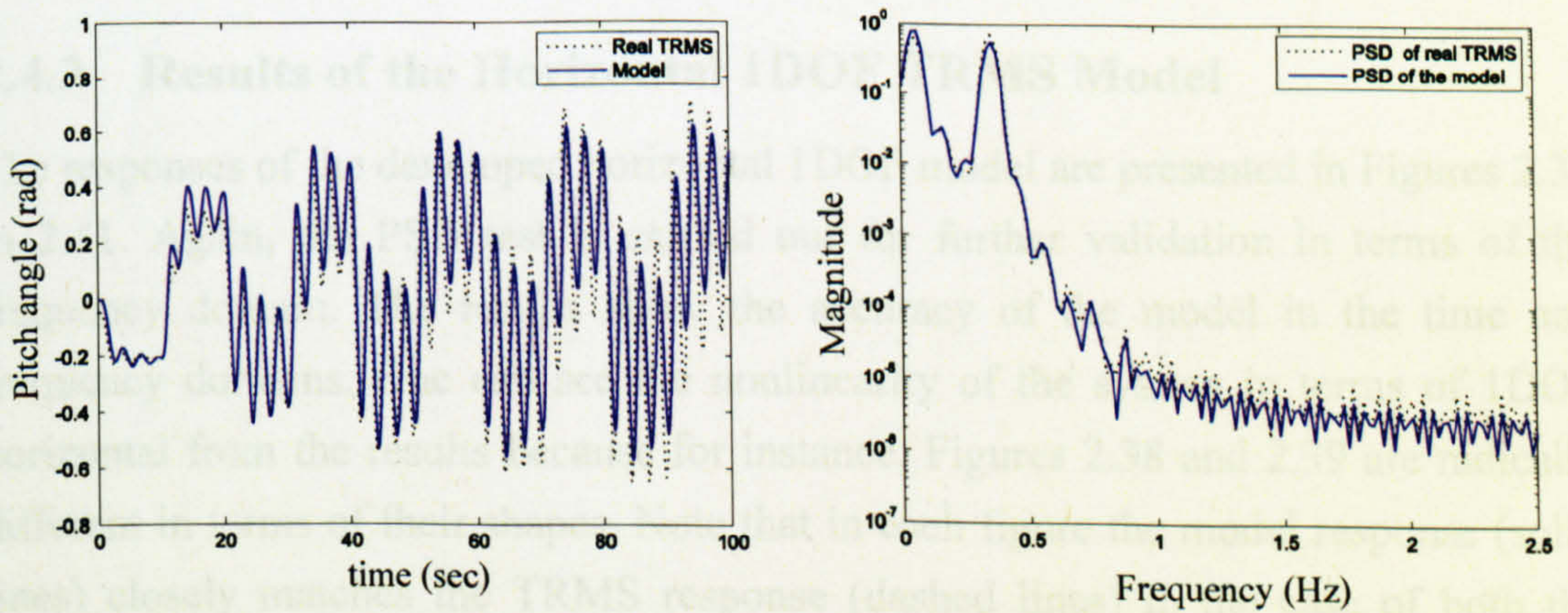


Figure 2.30 Pitch angle with square input (frequency 0.05 Hz and amplitude 0.8 V)

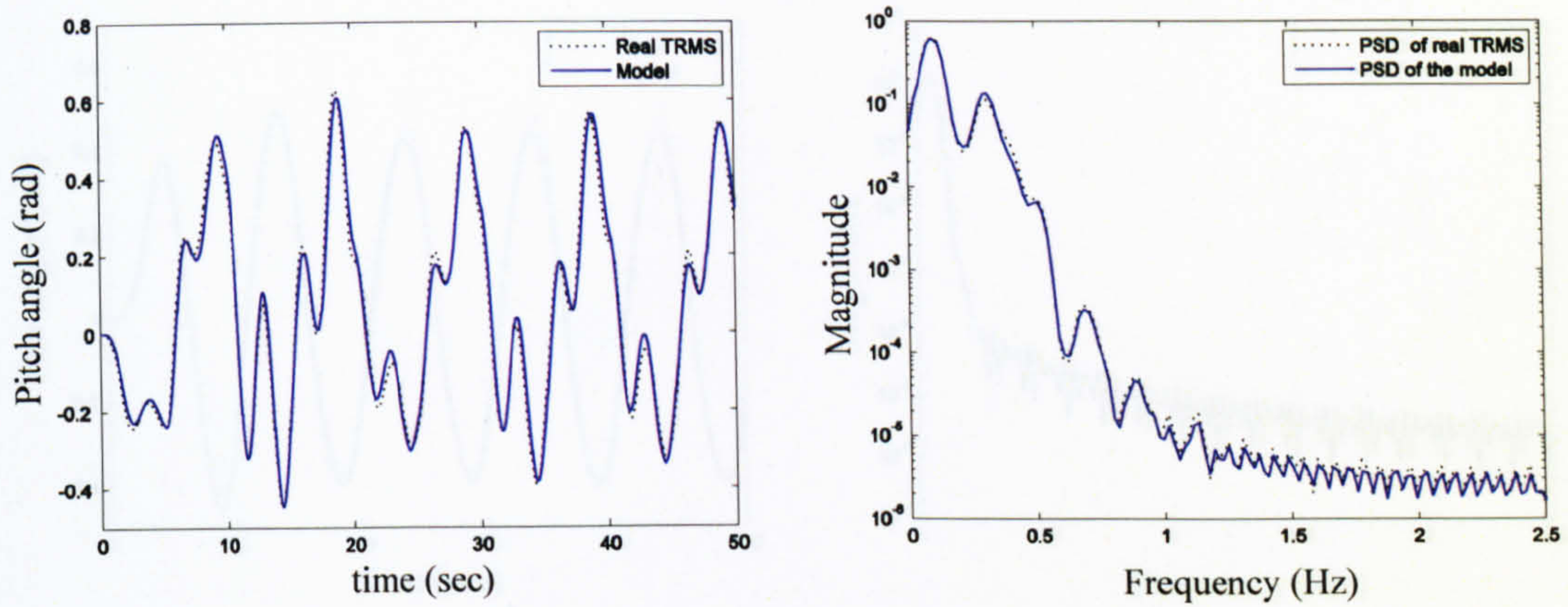


Figure 2.31 Pitch angle with square input (frequency 0.1 Hz and amplitude 1 V)

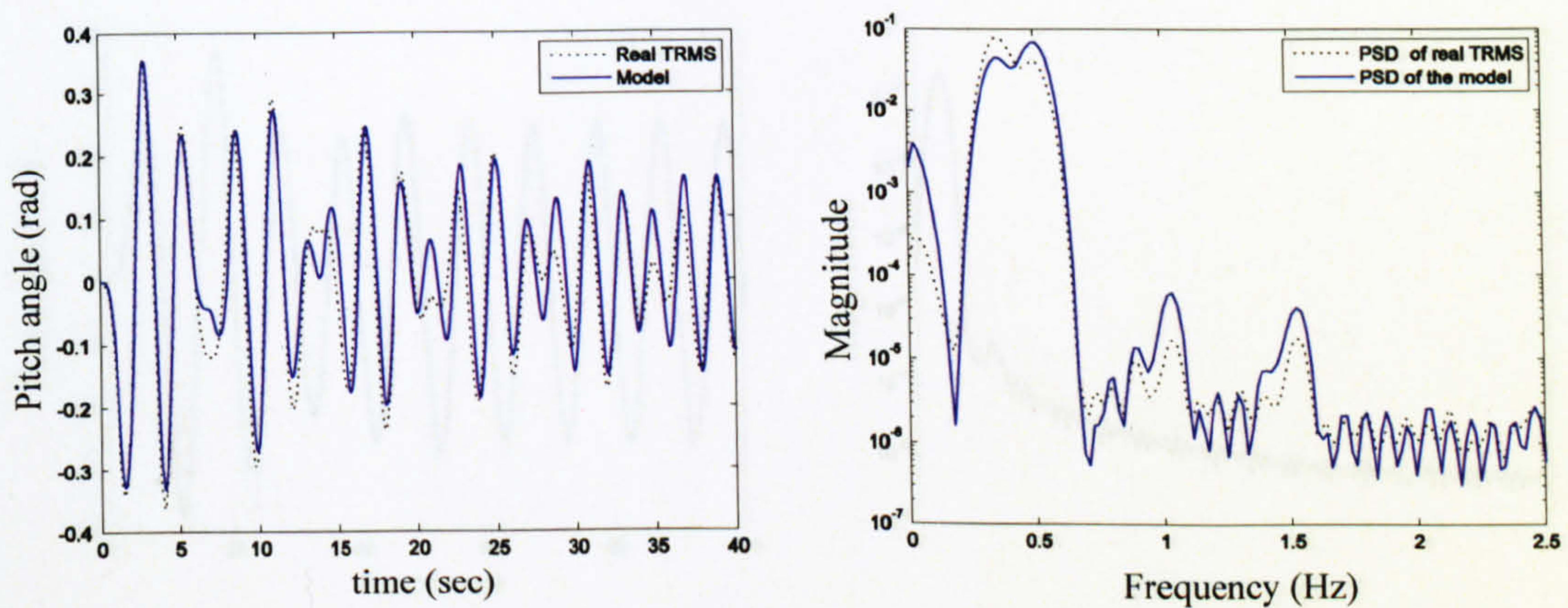


Figure 2.32 Pitch angle with square input (frequency 0.5 Hz and amplitude 2.5 V)

2.4.3 Results of the Horizontal 1DOF TRMS Model

The responses of the developed horizontal 1DOF model are presented in Figures 2.33 to 2.41. Again, the PSD test is carried out for further validation in terms of the frequency domain. The results show the accuracy of the model in the time and frequency domains. One can see the nonlinearity of the system in terms of 1DOF horizontal from the results because for instance, Figures 2.38 and 2.39 are radically different in terms of their shapes. Note that in each figure the model response (solid lines) closely matches the TRMS response (dashed lines) in the case of both the overlapping and PSD tests. Therefore, the model is adequate to accurately predict the output of the real system. Looking carefully at the PSD test results discloses that the dominant natural frequency of the horizontal channel seems to approximately occur at frequency of 0.05 Hz.

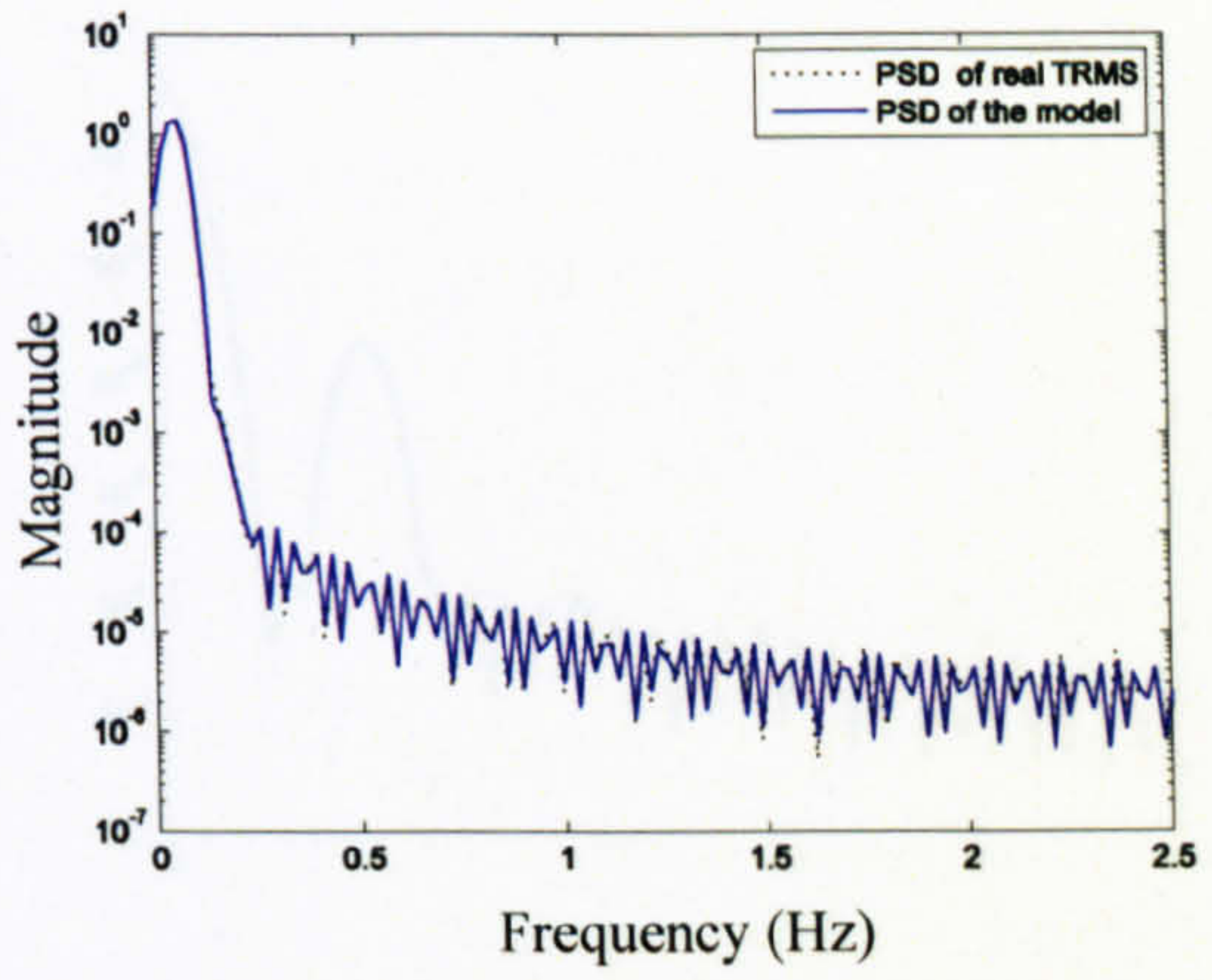
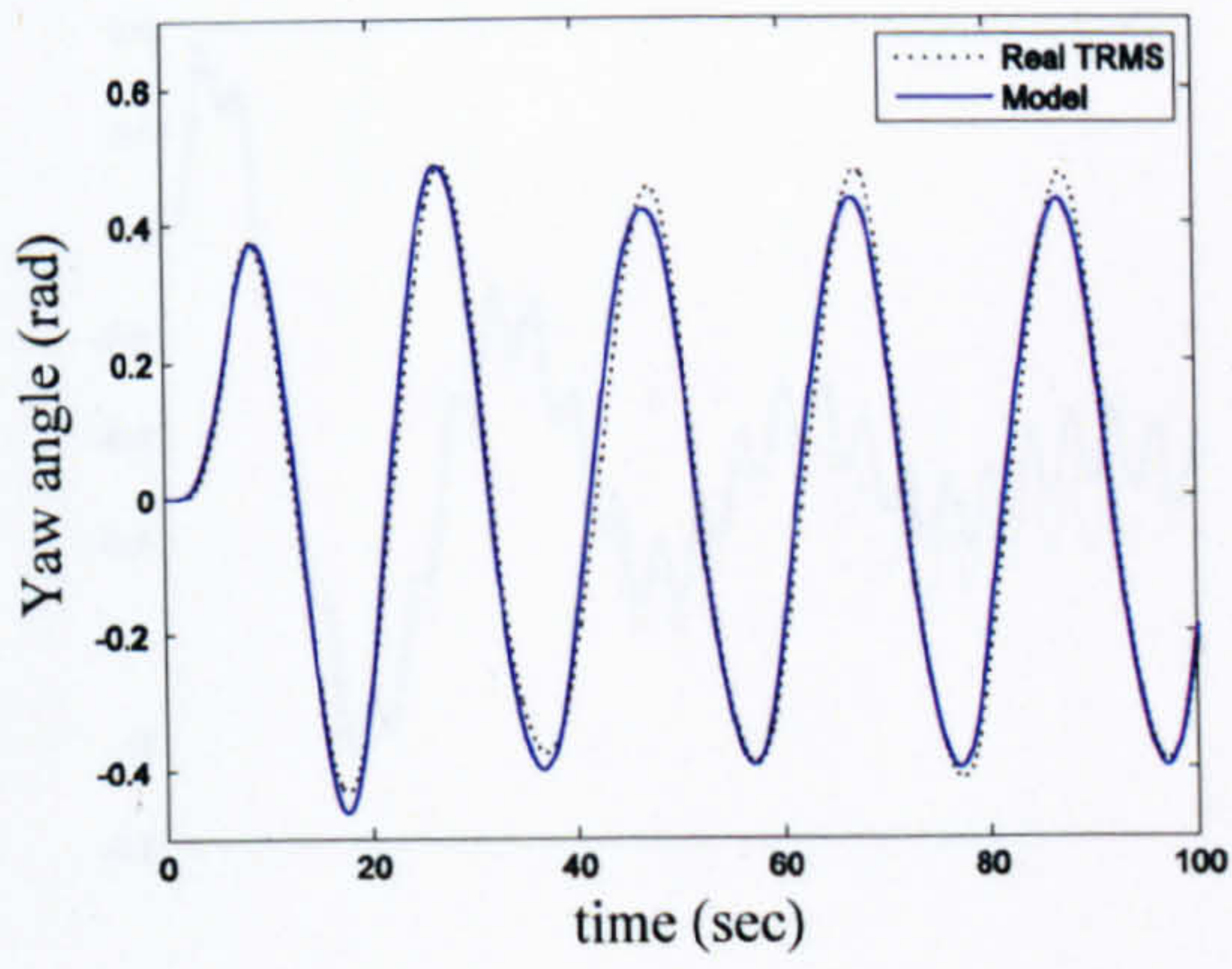


Figure 2.33 Yaw angle with sine input (frequency 0.05 Hz and amplitude 0.3 V)

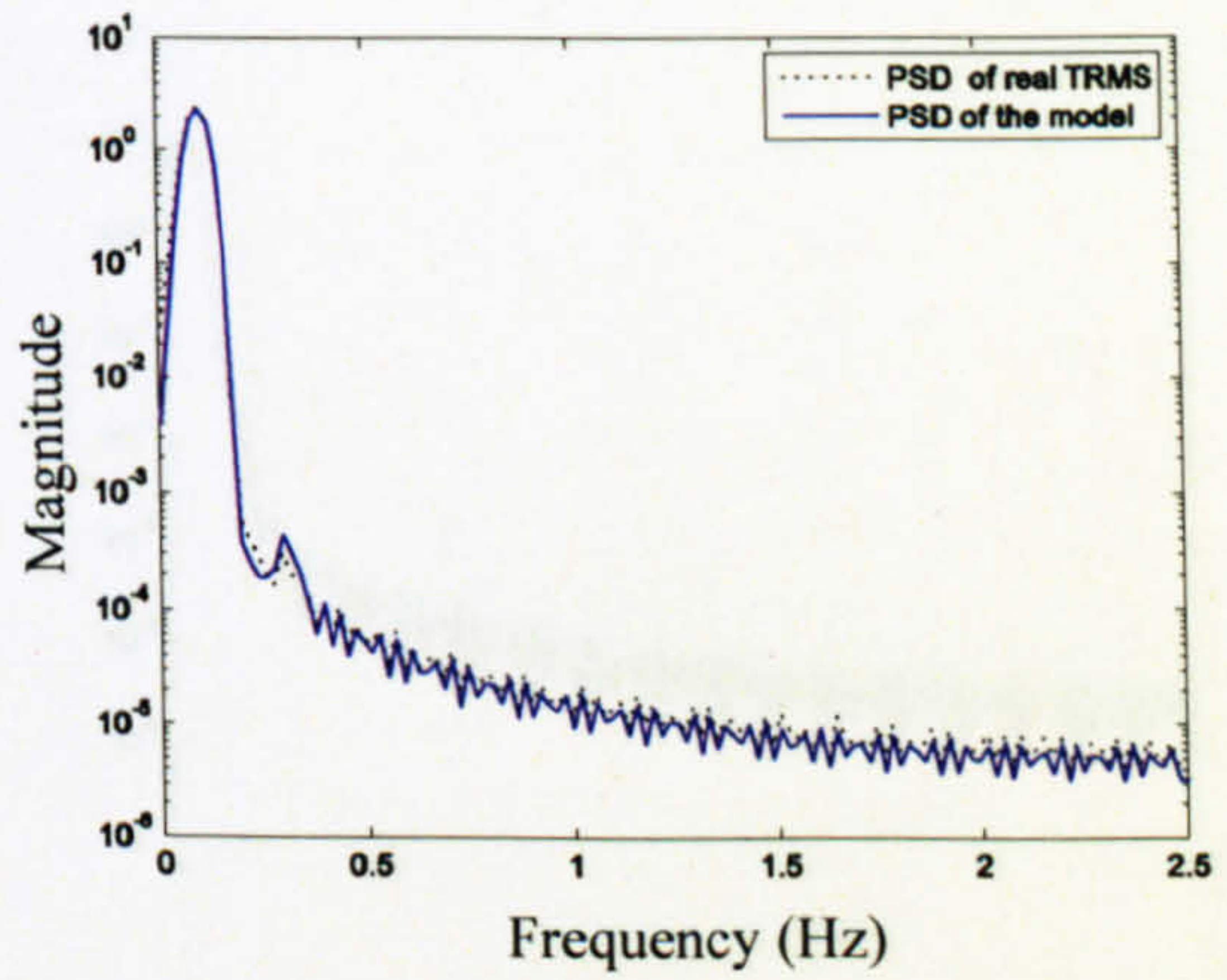
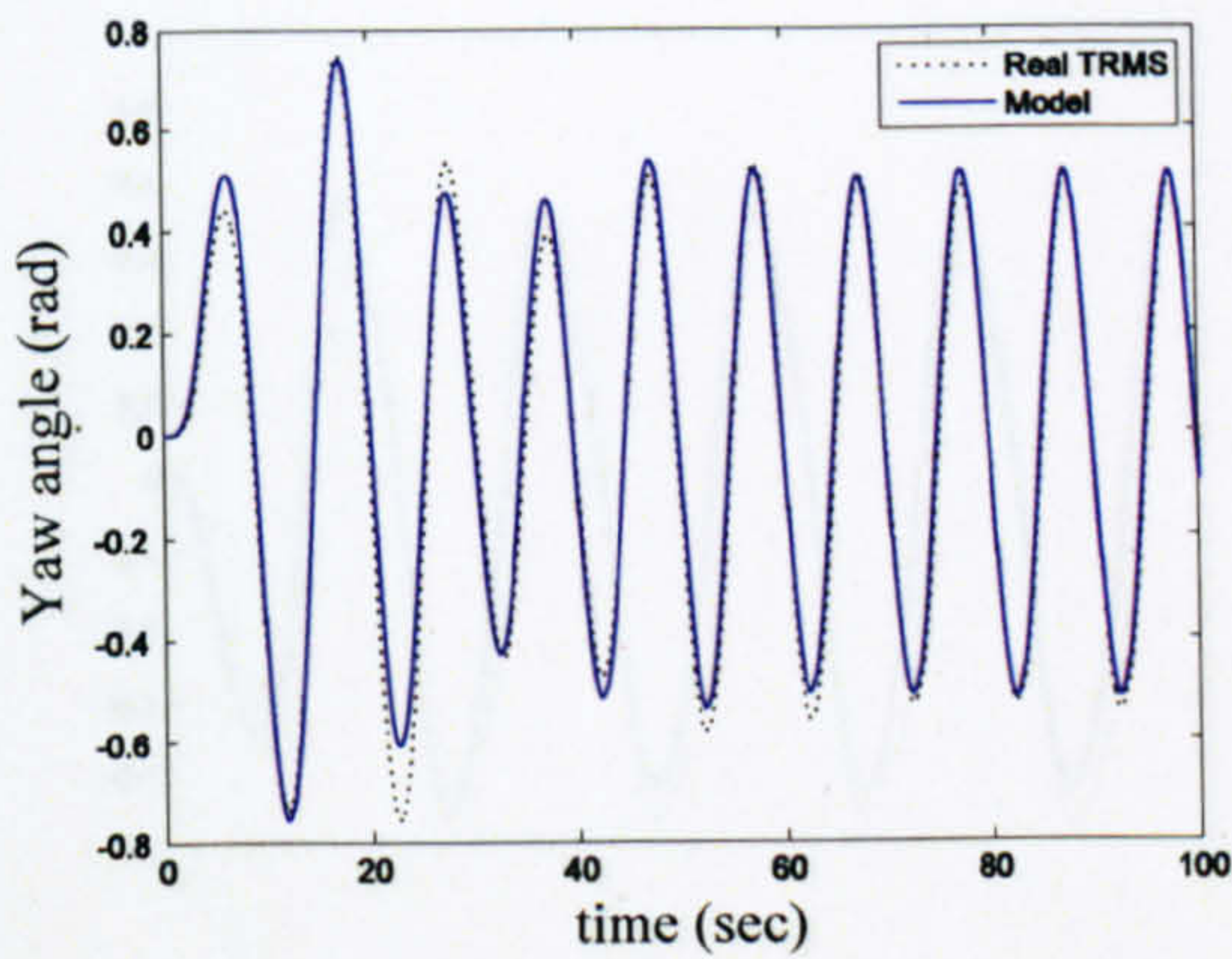


Figure 2.34 Yaw angle with sine input (frequency 0.1 Hz and amplitude 0.4 V)

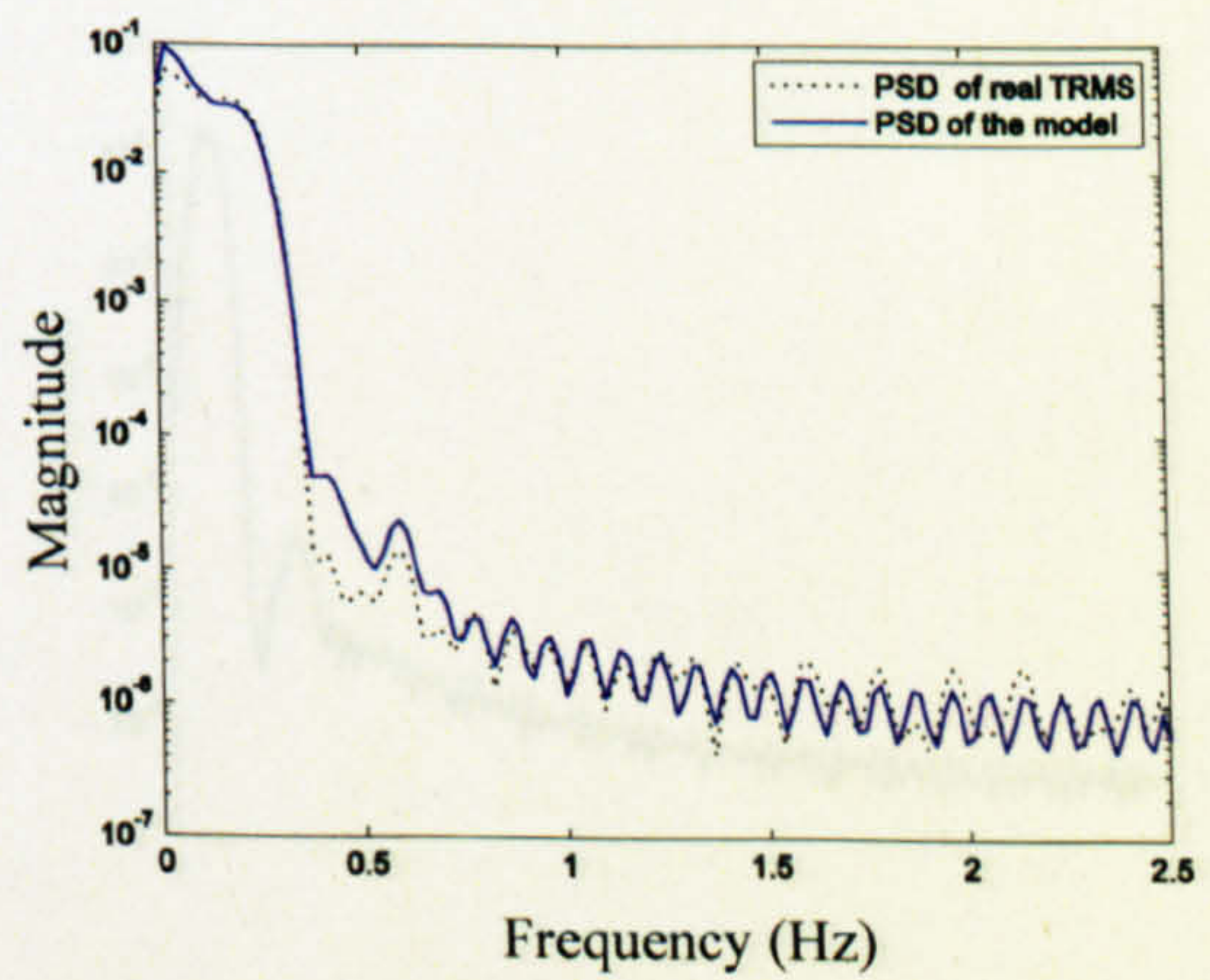
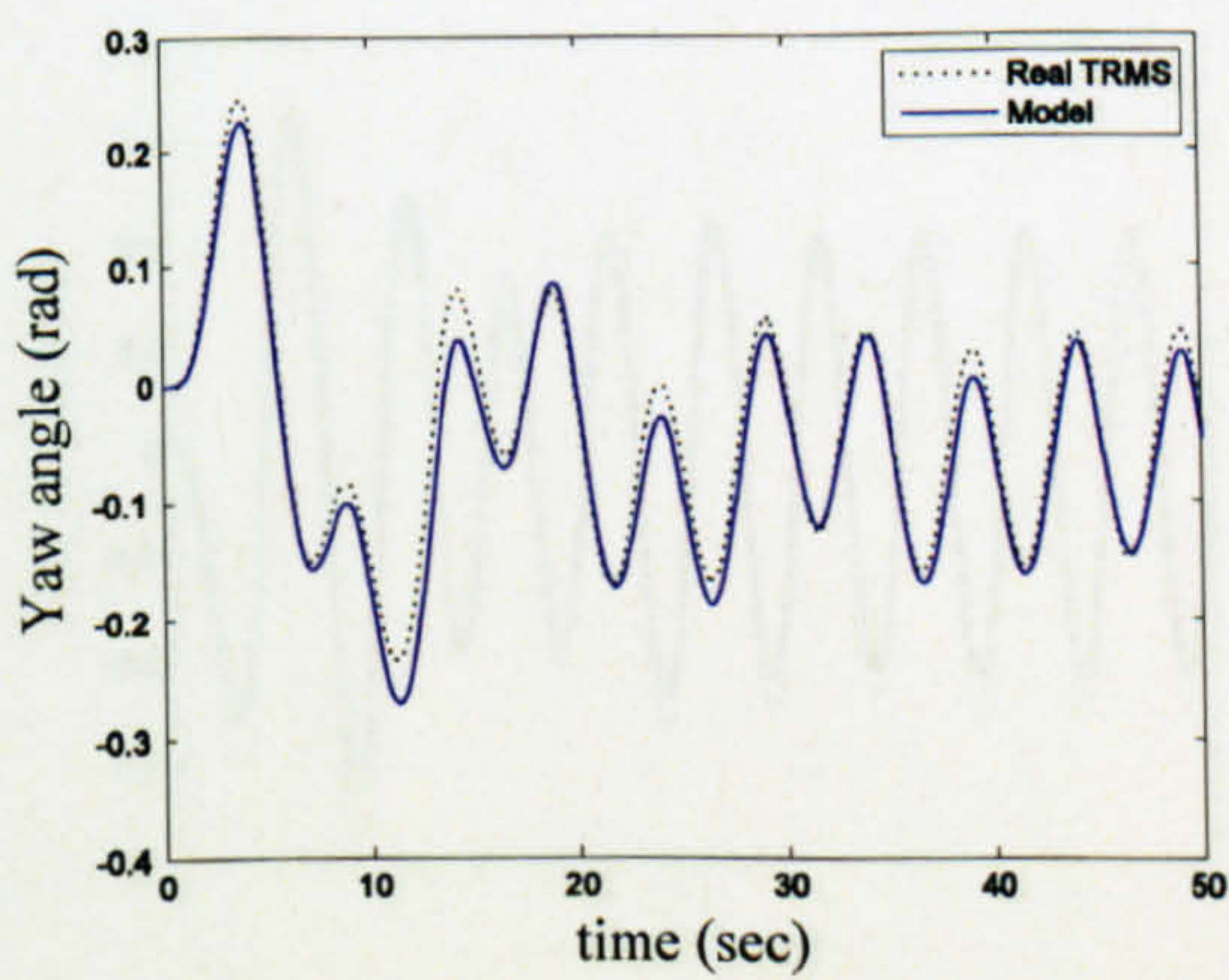


Figure 2.35 Yaw angle with sine input (frequency 0.2 Hz and amplitude 0.5 V)

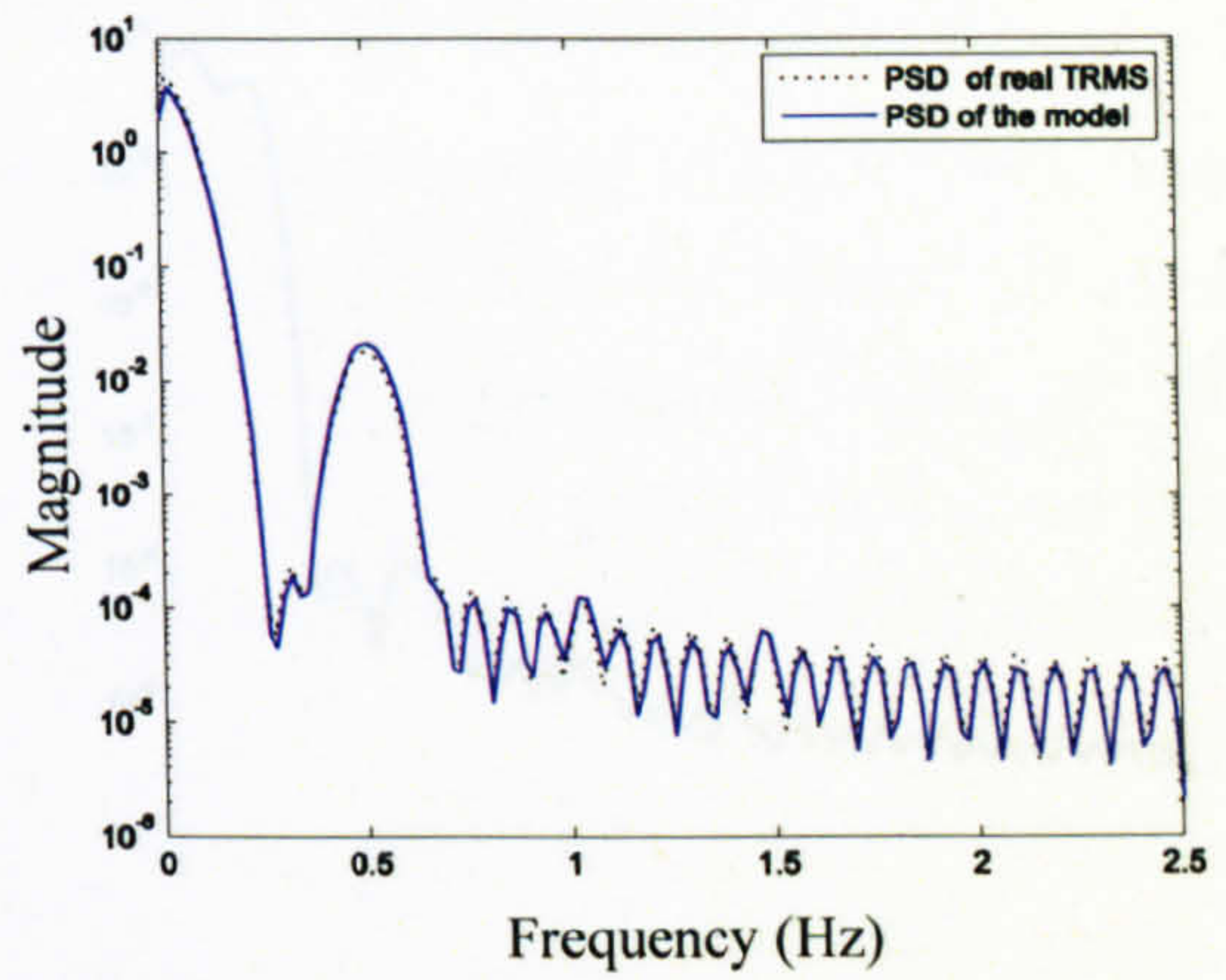
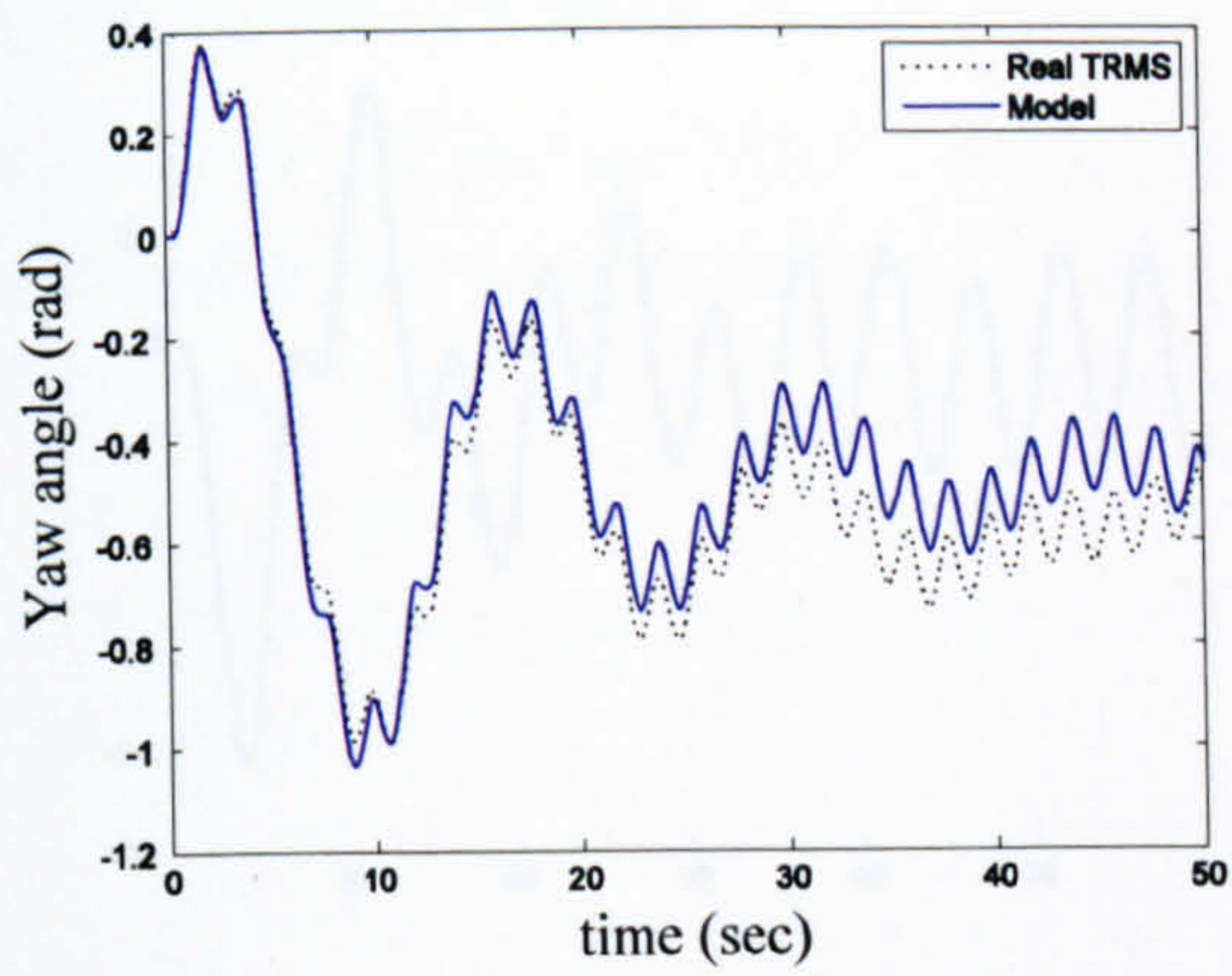


Figure 2.36 Yaw angle with sine input (frequency 0.5 Hz and amplitude 2 V)

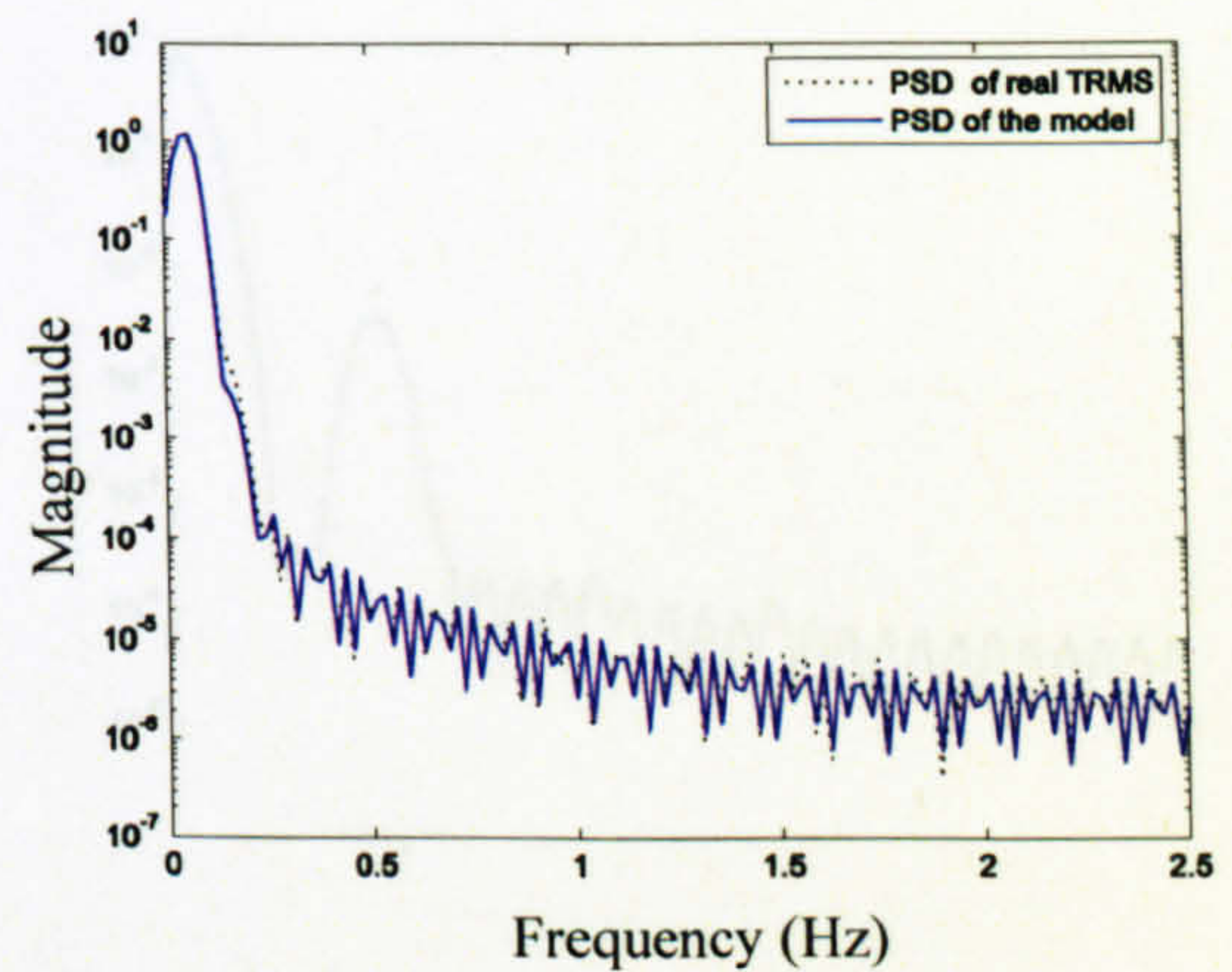
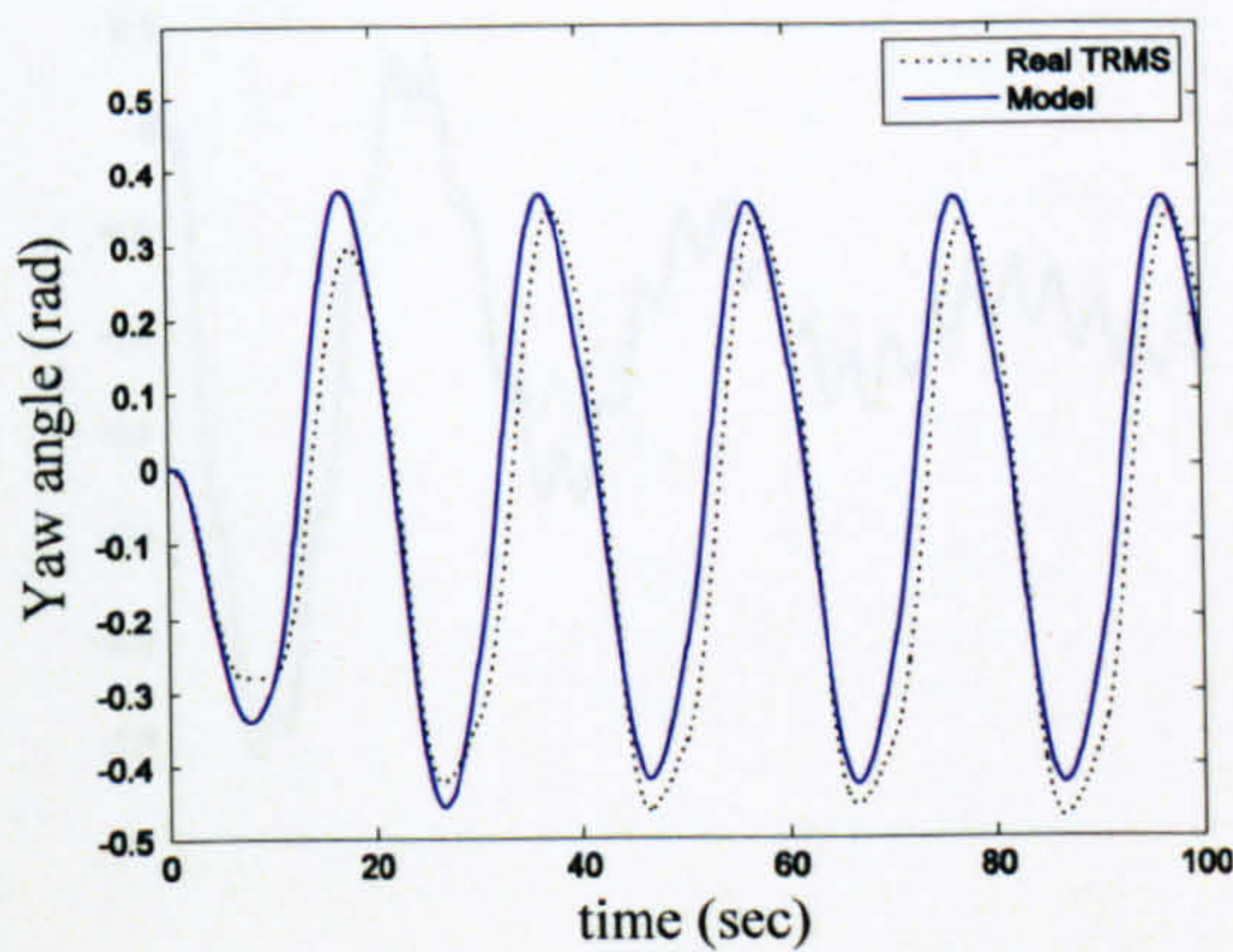


Figure 2.37 Yaw angle with square input (frequency 0.05 Hz and amplitude 0.2 V)

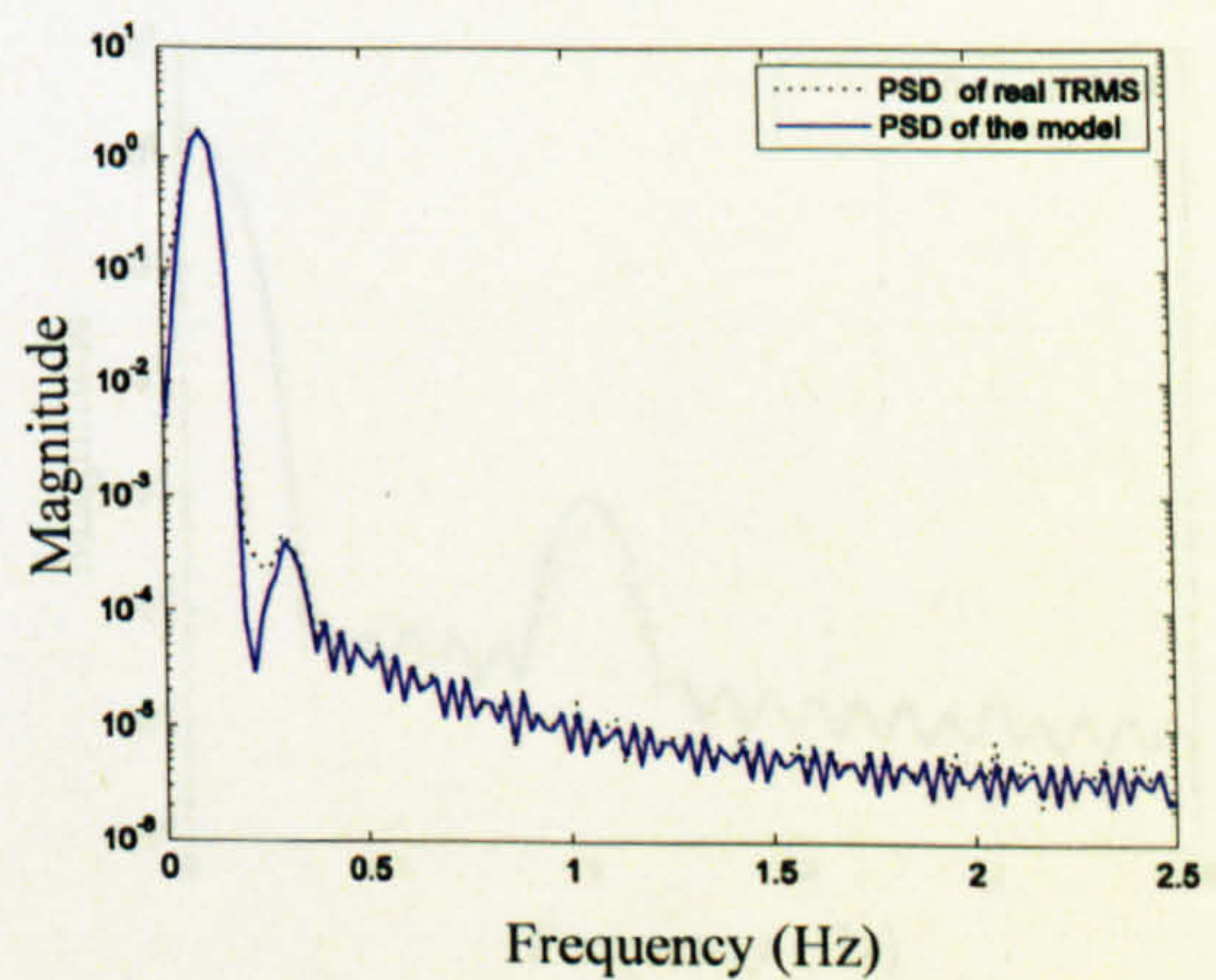
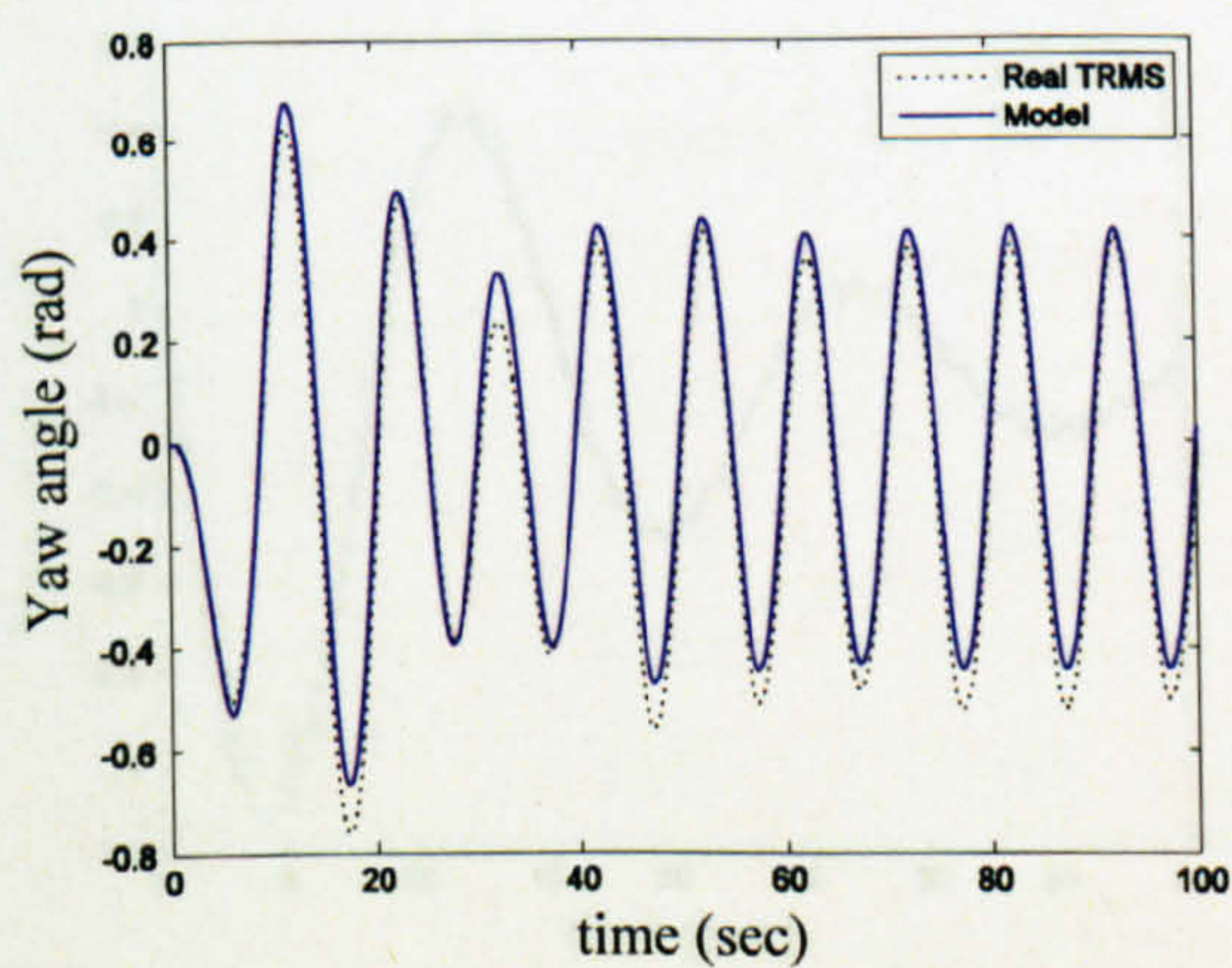


Figure 2.38 Yaw angle with square input (frequency 0.1 Hz and amplitude 0.3 V)

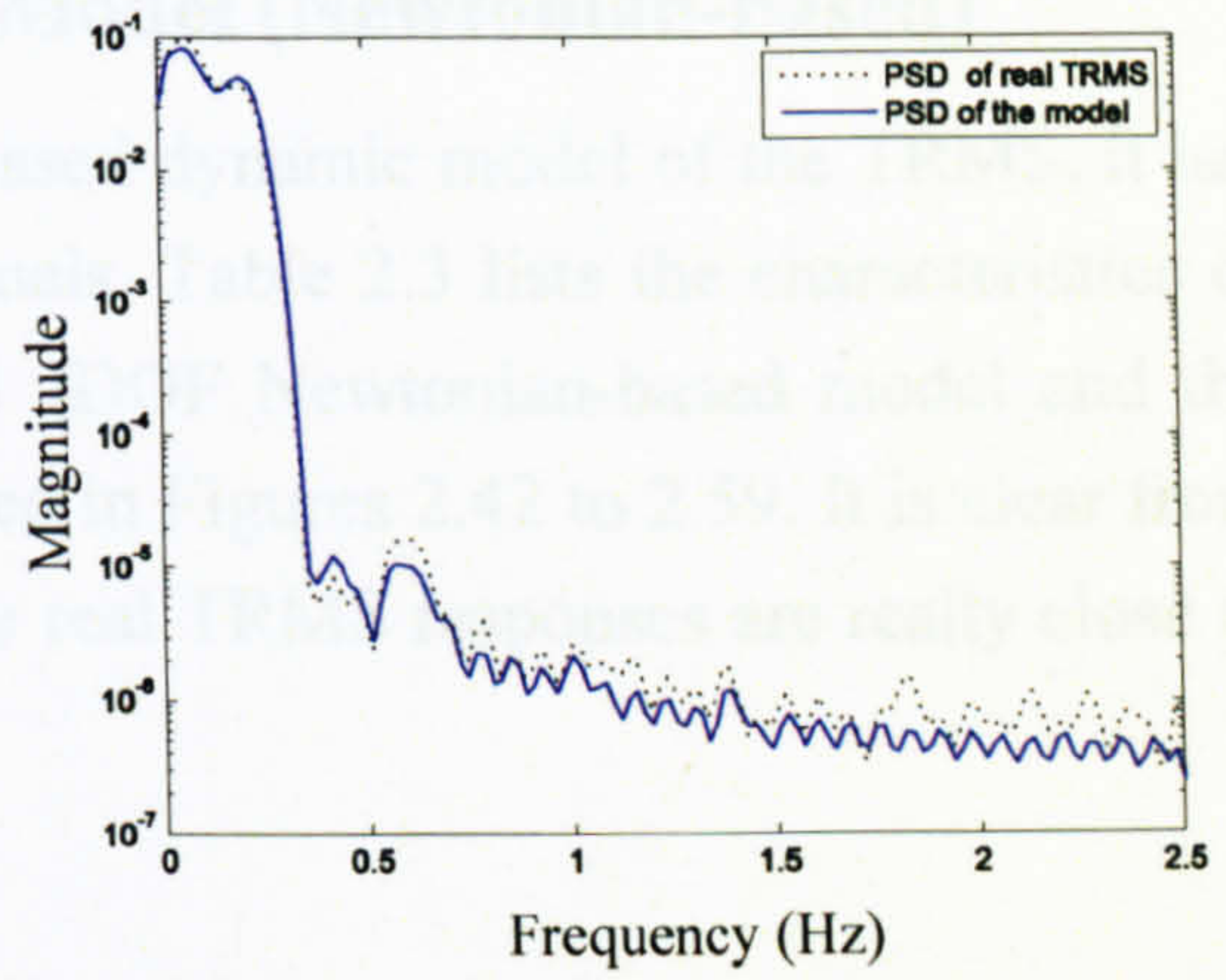
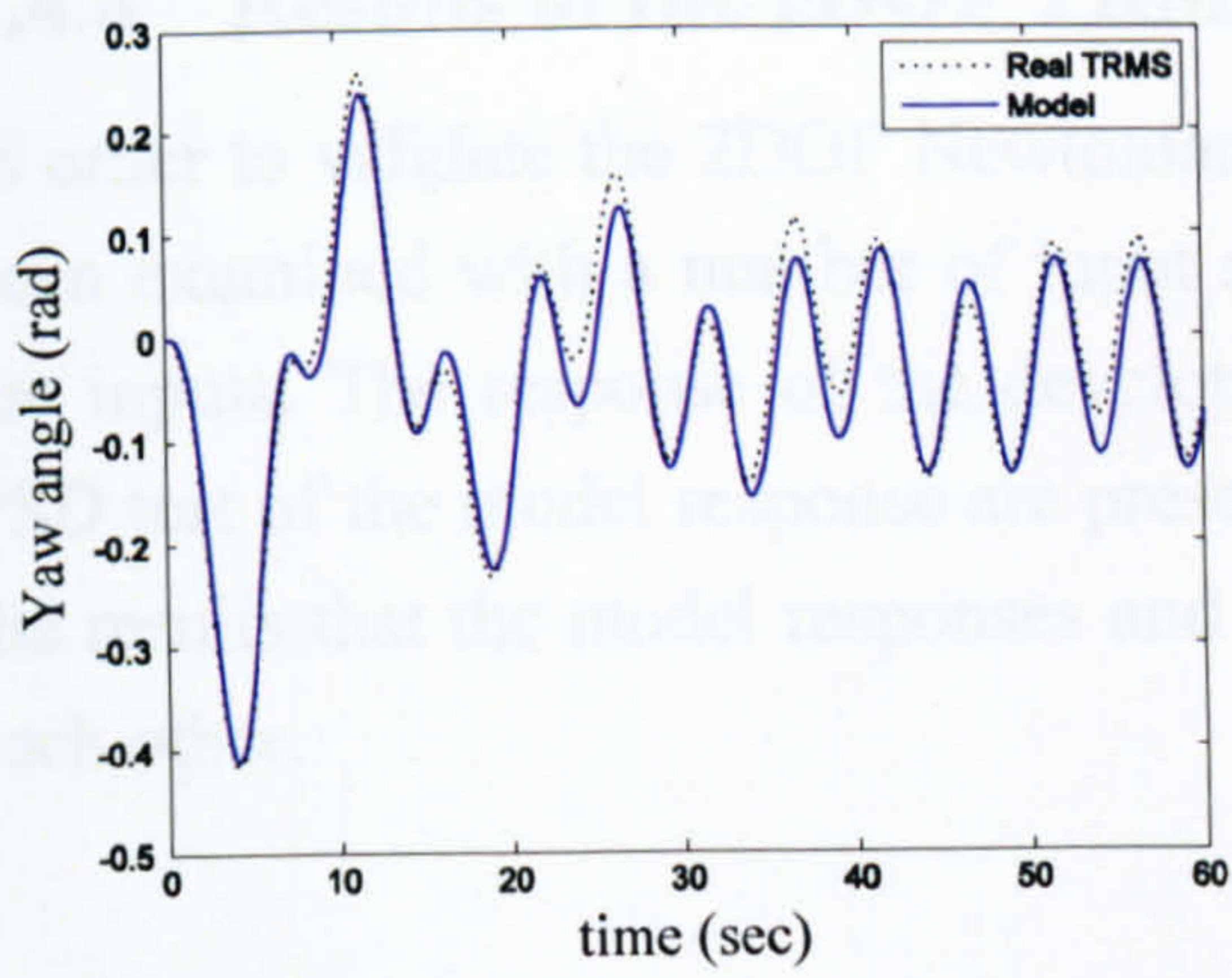


Figure 2.39 Yaw angle with square input (frequency 0.2 Hz and amplitude 0.4 V)

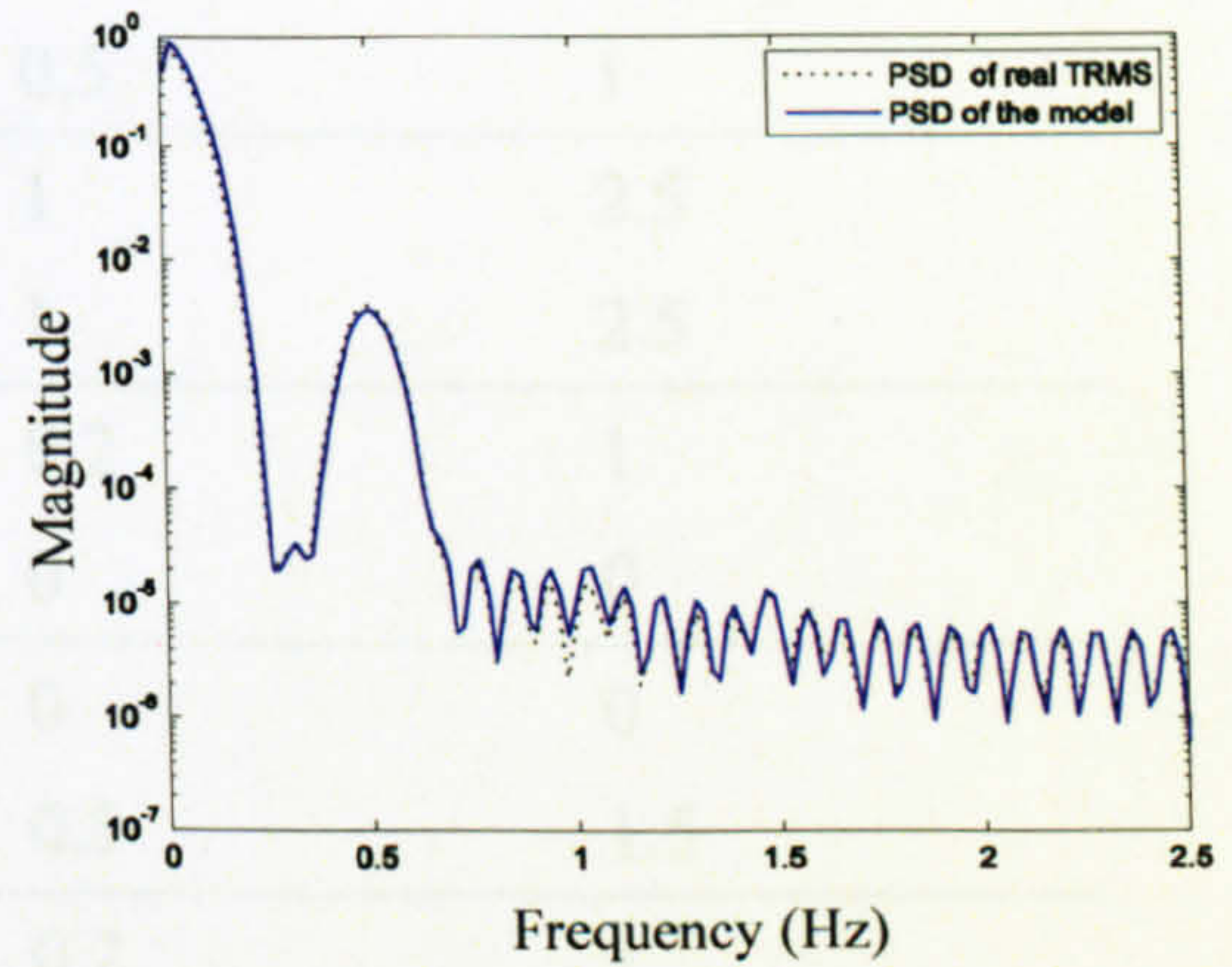
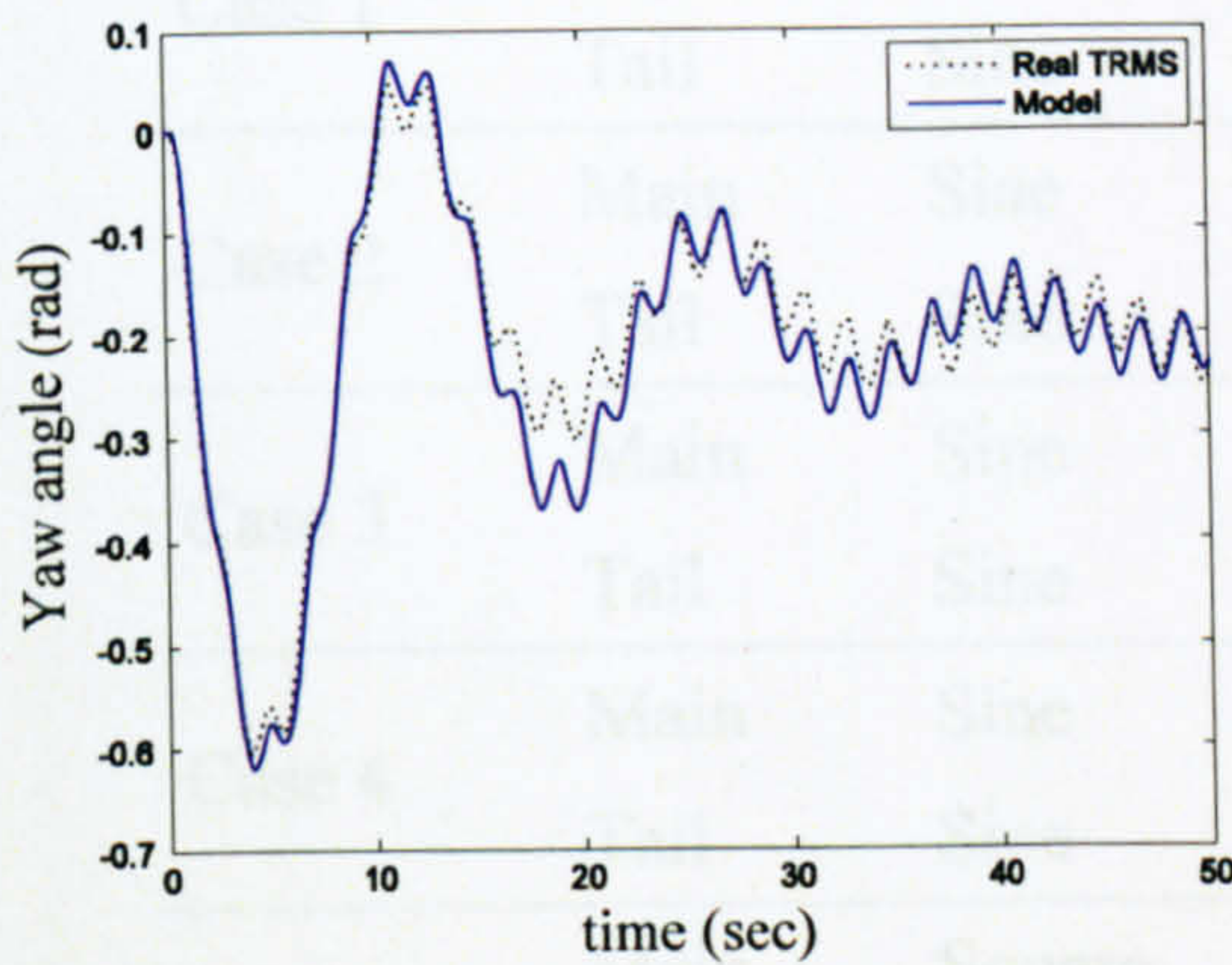


Figure 2.40 Yaw angle with square input (frequency 0.5 Hz and amplitude 1 V)

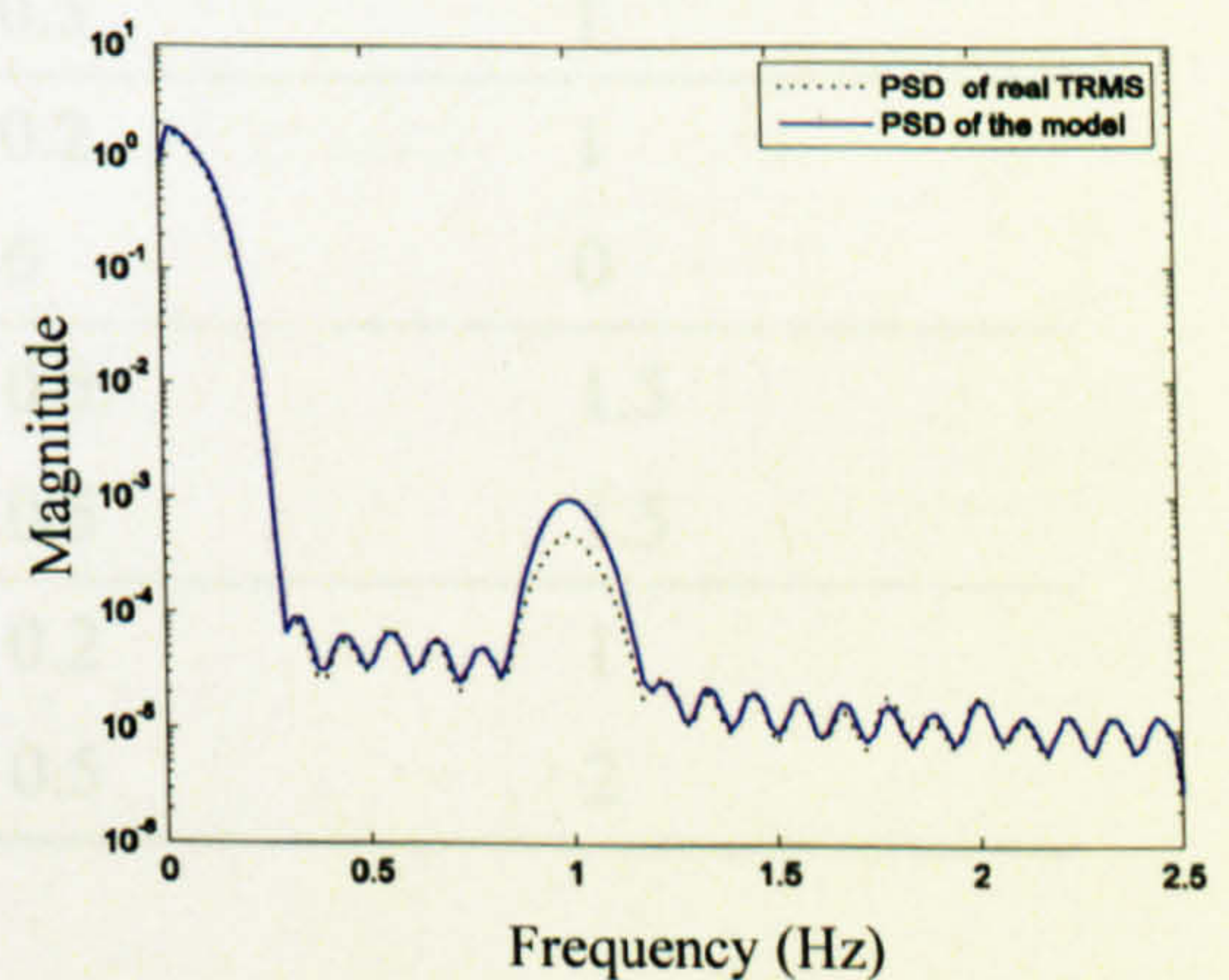
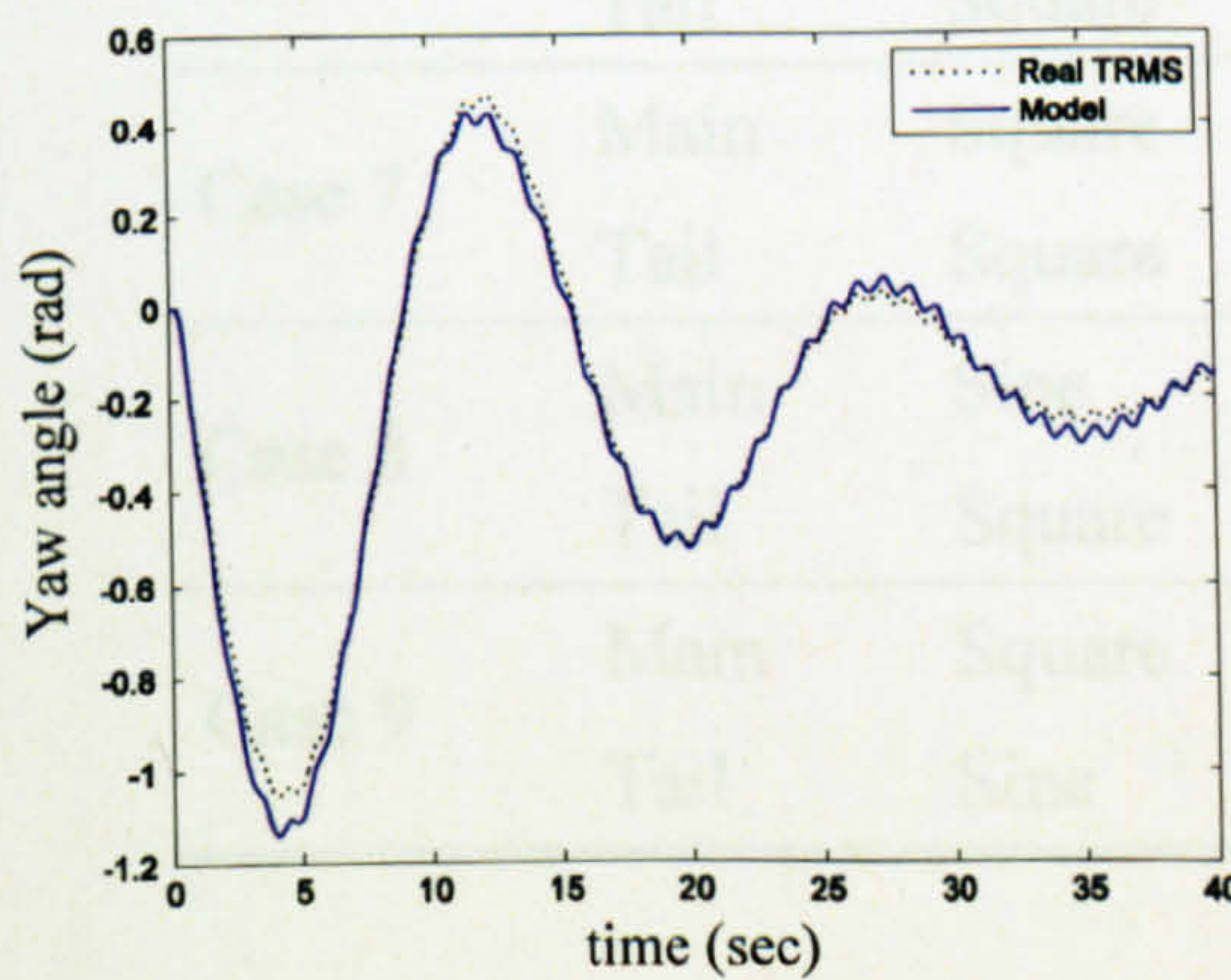


Figure 2.41 Yaw angle with square input (frequency 1 Hz and amplitude 2.5 V)

2.4.4 Results of the 2DOF TRMS Model (Newtonian-based)

In order to validate the 2DOF Newtonian-based dynamic model of the TRMS, it has been examined with a number of input signals. Table 2.3 lists the characteristics of the inputs. The response of the developed 2DOF Newtonian-based model and the PSD test of the model response are presented in Figures 2.42 to 2.59. It is clear from the results that the model responses and the real TRMS responses are really close to each other.

Table 2.3 The characteristics of the input signals

		Type	Frequency (Hz)	Amplitude (V)
Case 1	Main	Sine	0.5	1.5
	Tail	Sine	0.5	1
Case 2	Main	Sine	1	2.5
	Tail	Sine	1	2.5
Case 3	Main	Sine	0.2	1
	Tail	Sine	0	0
Case 4	Main	Sine	0	0
	Tail	Sine	0.5	1.5
Case 5	Main	Square	0.2	1
	Tail	Square	0.5	1.5
Case 6	Main	Square	0	0
	Tail	Square	0.5	1
Case 7	Main	Square	0.2	1
	Tail	Square	0	0
Case 8	Main	Sine	0.5	1.5
	Tail	Square	0.5	1.5
Case 9	Main	Square	0.2	1
	Tail	Sine	0.5	2

For example in case 1, the main and tail motors of the system have been excited with two sine waves with the same frequency of 0.5 Hz and different amplitudes of 1.5 V and 1 V respectively. The respective outputs (pitch and yaw angles) and their PSDs are shown in Figures 2.42 and 2.43. The strong interaction between the two channels is evident from the results. Figures 2.47 and 2.55 show the yaw angle of the system when the tail motor is left unexcited. In other words, these figures illustrate the effect

of the main rotor on the horizontal channel. In addition, the cross-coupling of the tail rotor on the pitch angle can be observed in Figures 2.48 and 2.52.

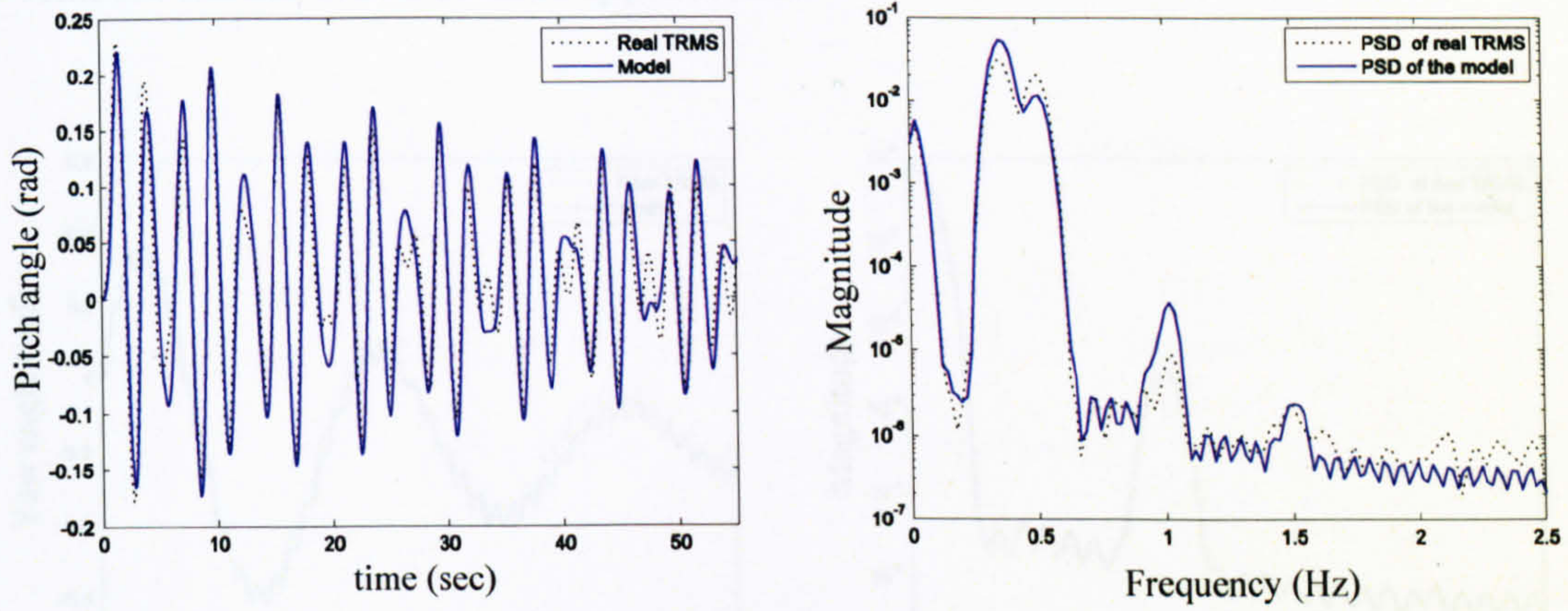


Figure 2.42 Pitch angle and PSD (case 1)

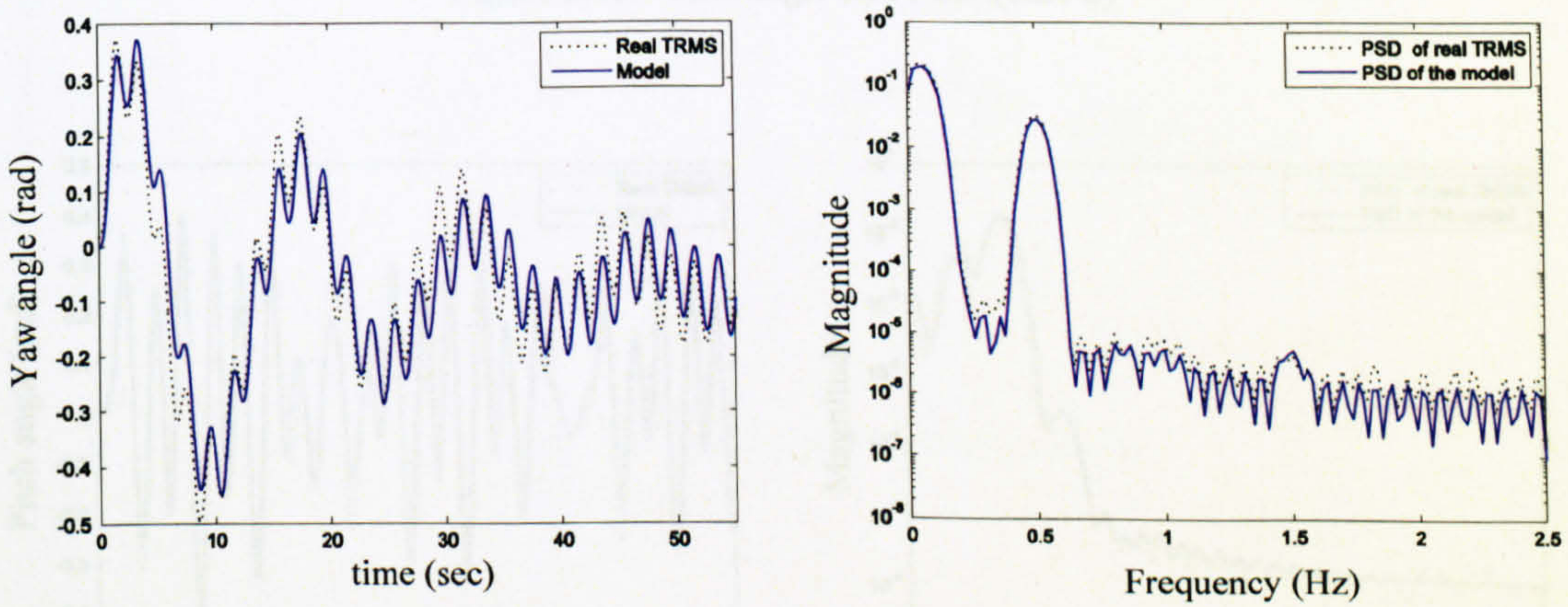


Figure 2.43 Yaw angle and PSD (case 1)

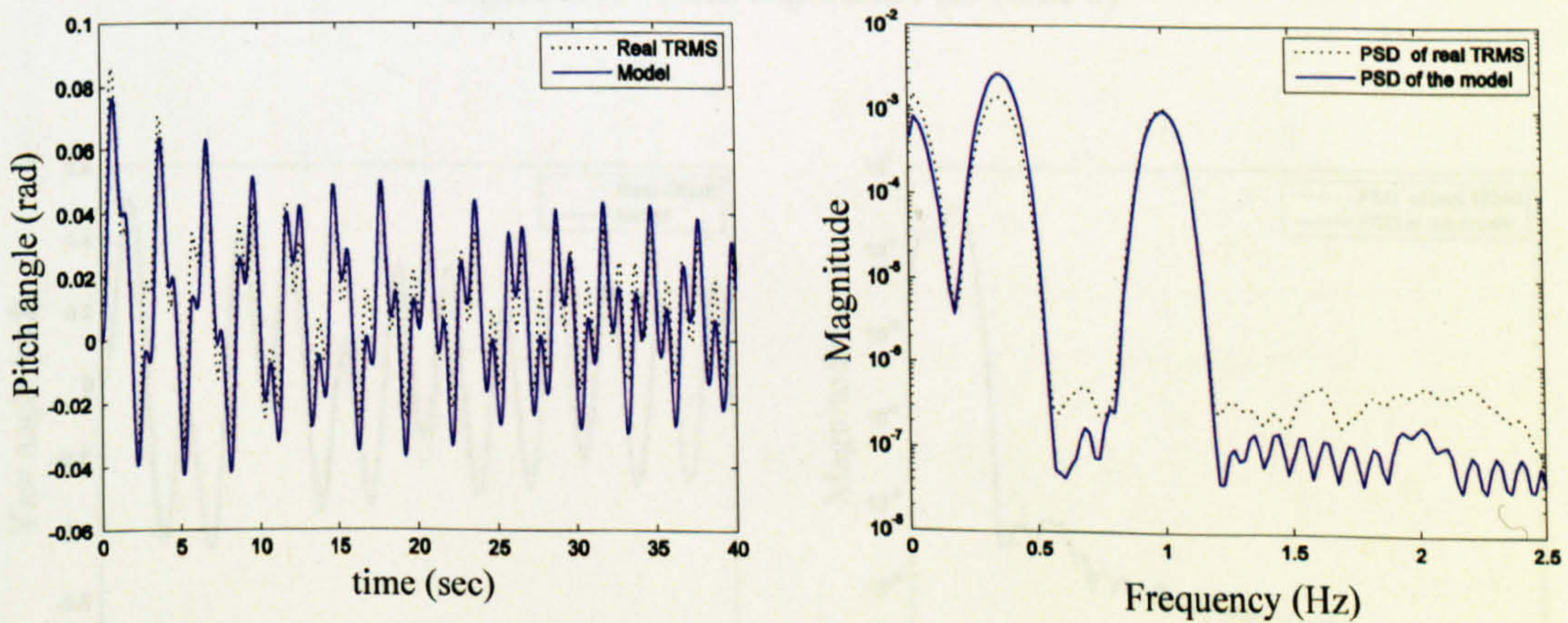


Figure 2.44 Pitch angle and PSD (case 2)

It is clear from the figures that the model is adequate to represent the behaviour of the real system at both high and low frequency ranges. As a matter of fact, the PSD results also prove the adequacy of the developed model at high frequency range that is vital for model-based control approaches.

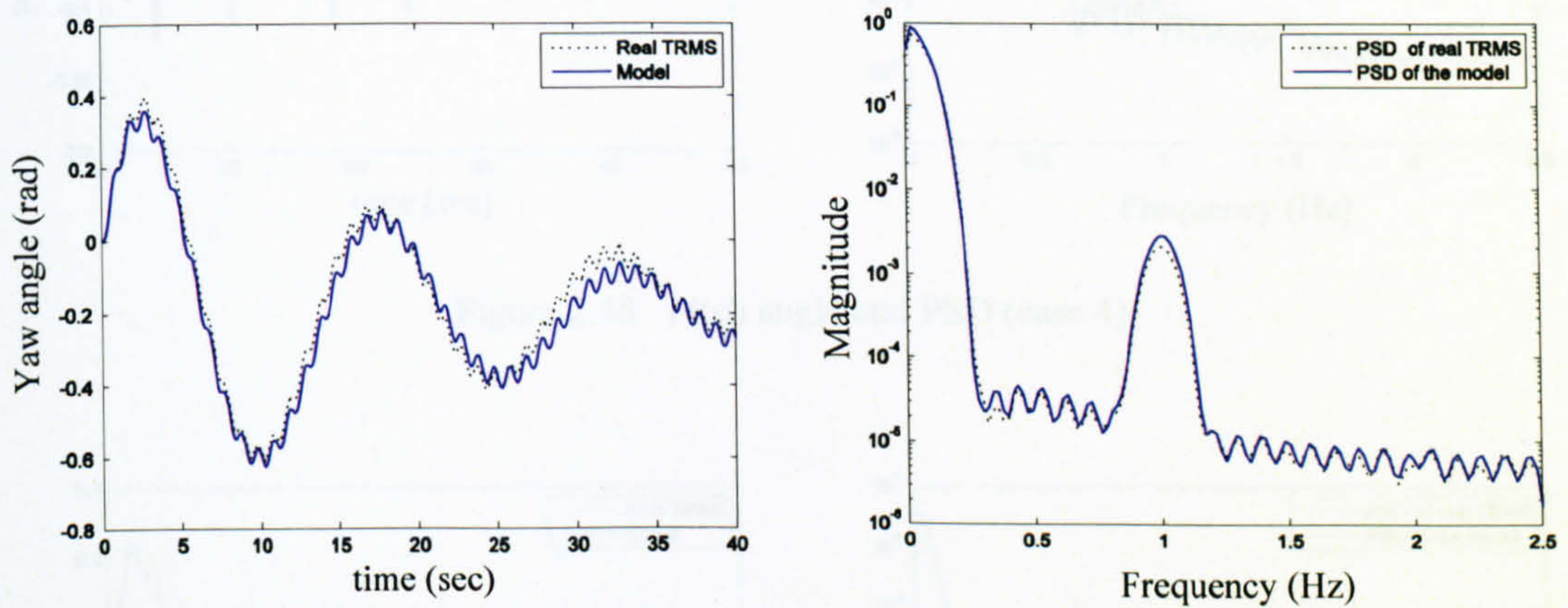


Figure 2.45 Yaw angle and PSD (case 2)

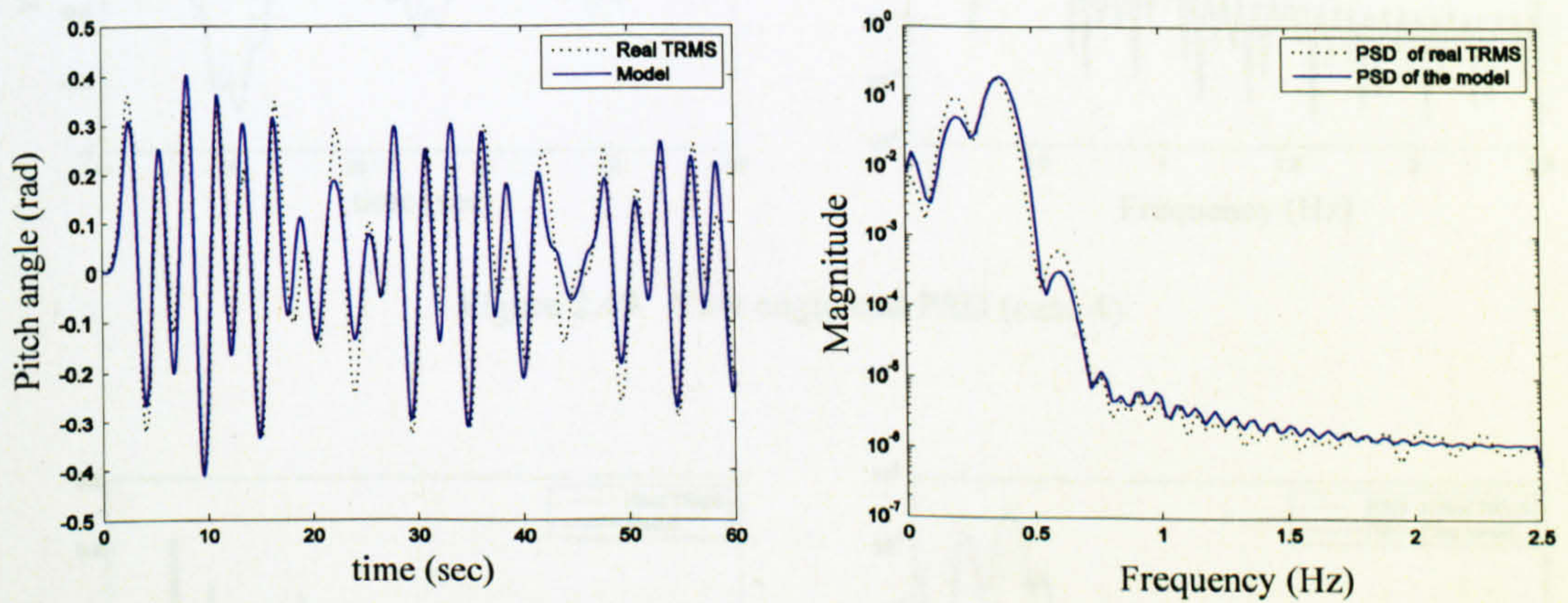


Figure 2.46 Pitch angle and PSD (case 3)

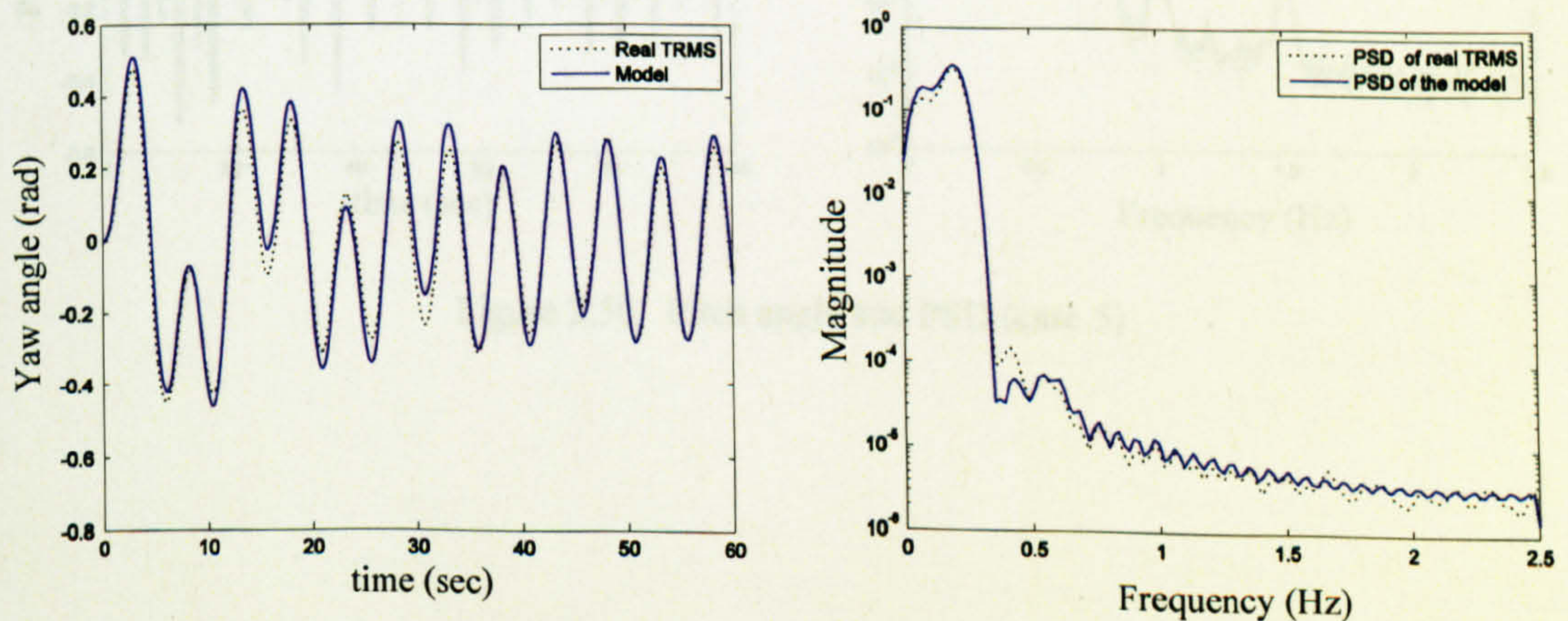


Figure 2.47 Yaw angle and PSD (case 3)

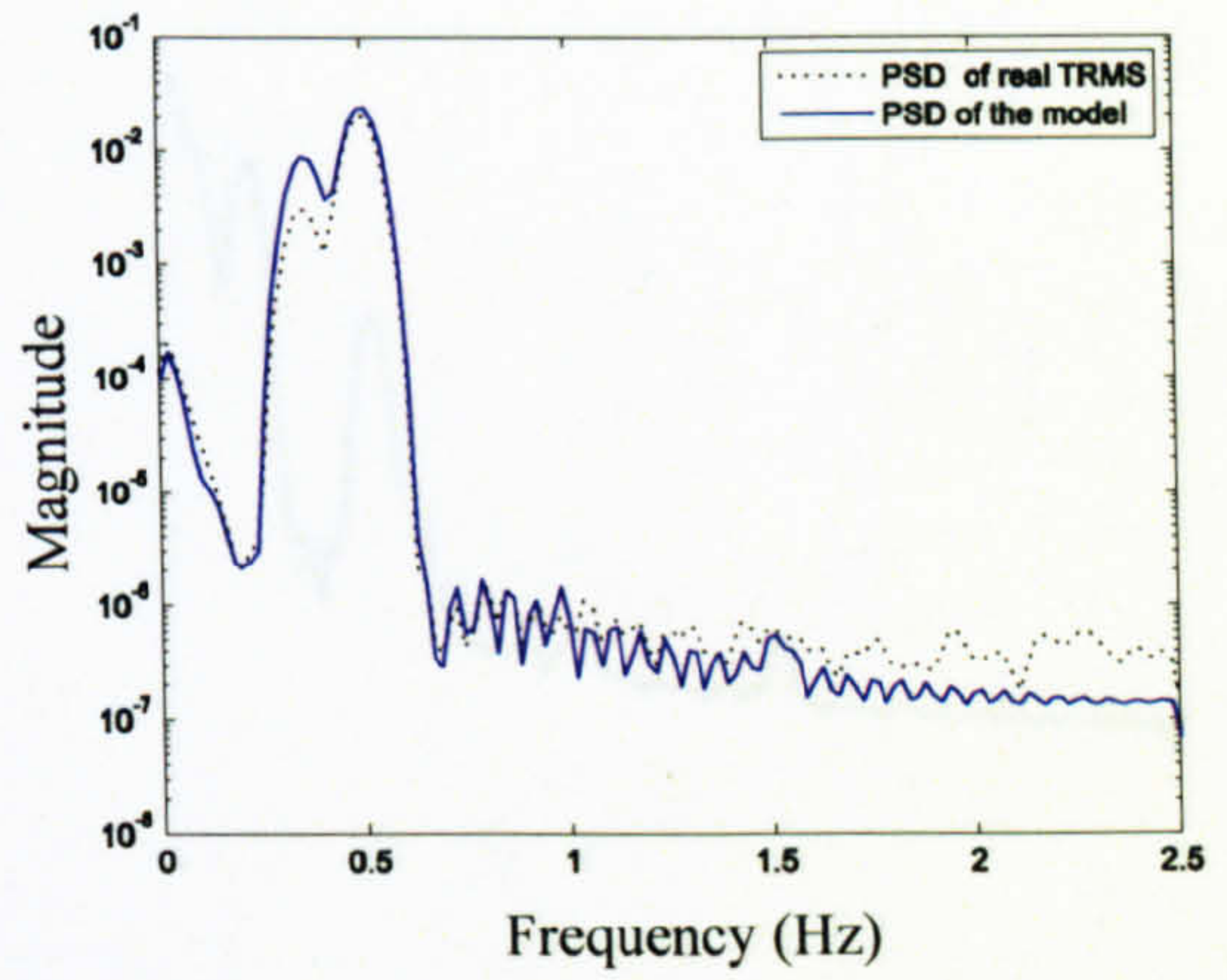
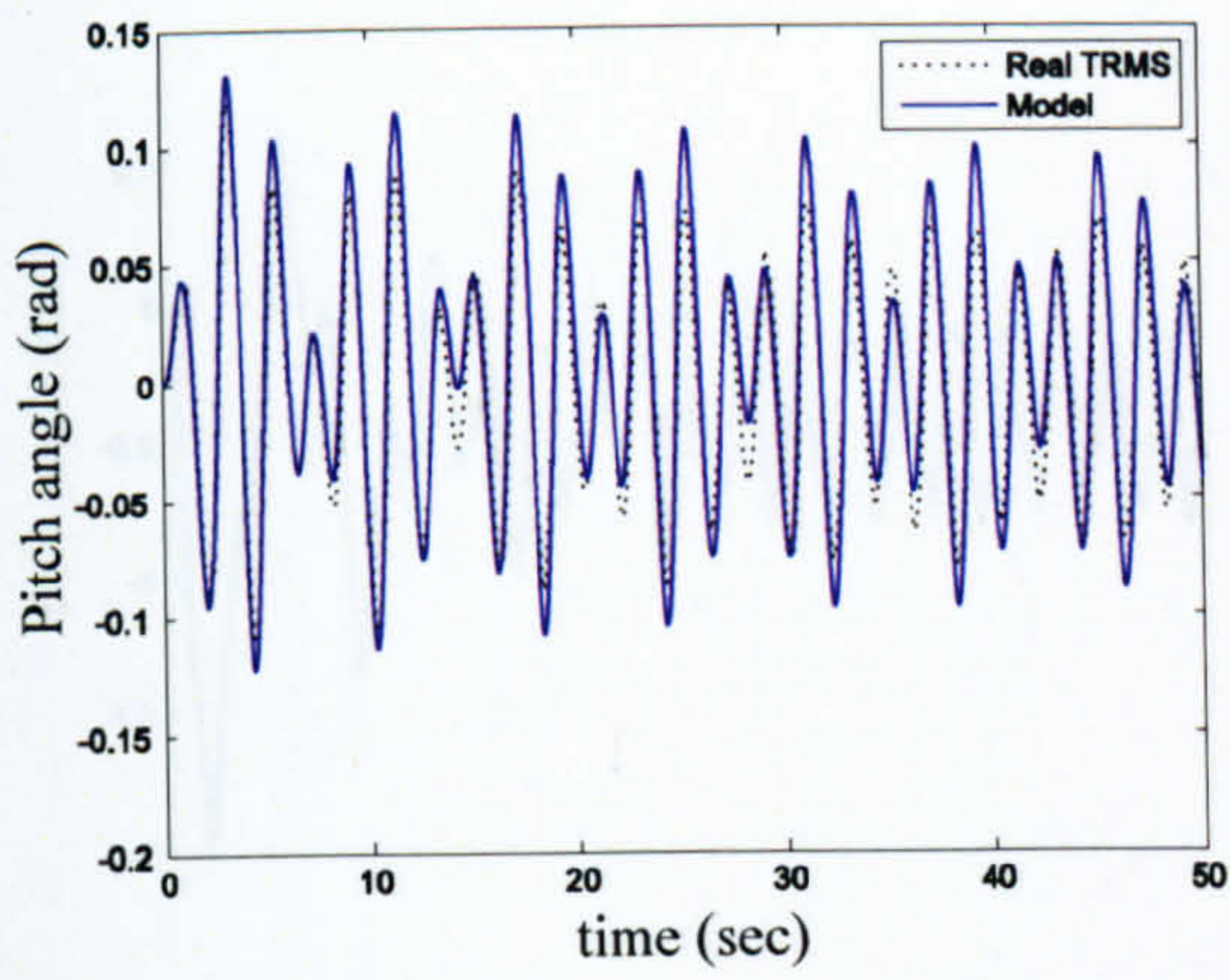


Figure 2.48 Pitch angle and PSD (case 4)

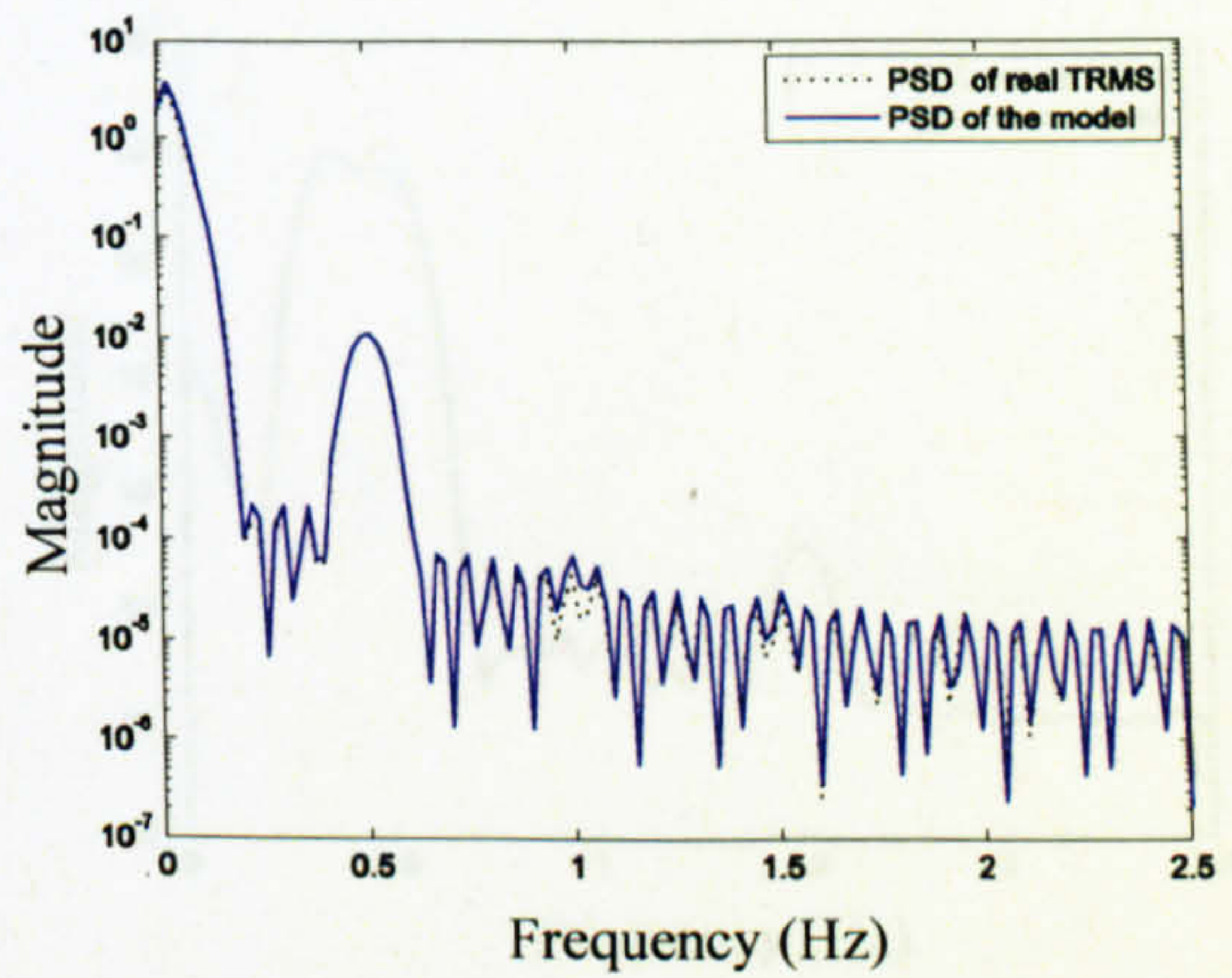
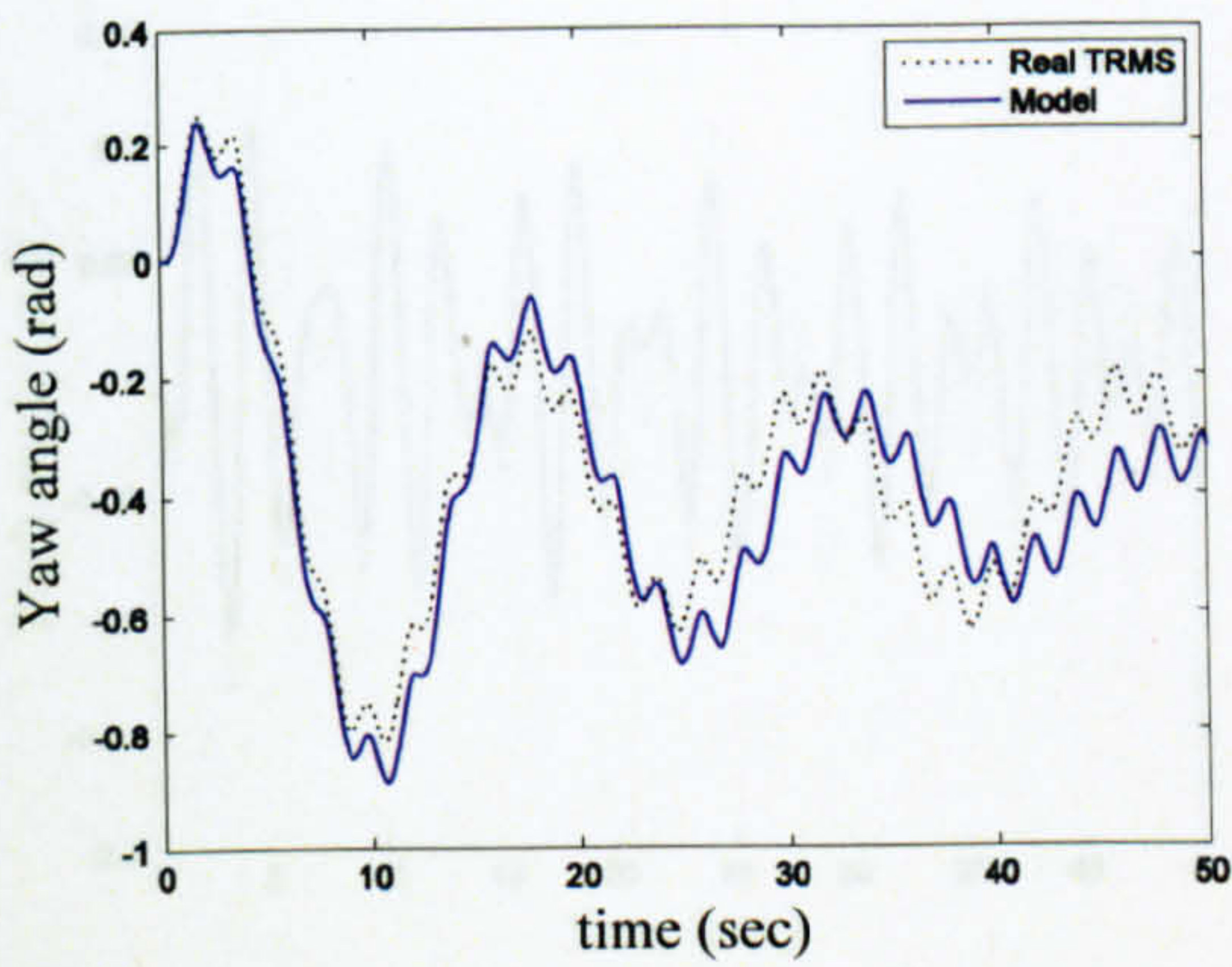


Figure 2.49 Yaw angle and PSD (case 4)

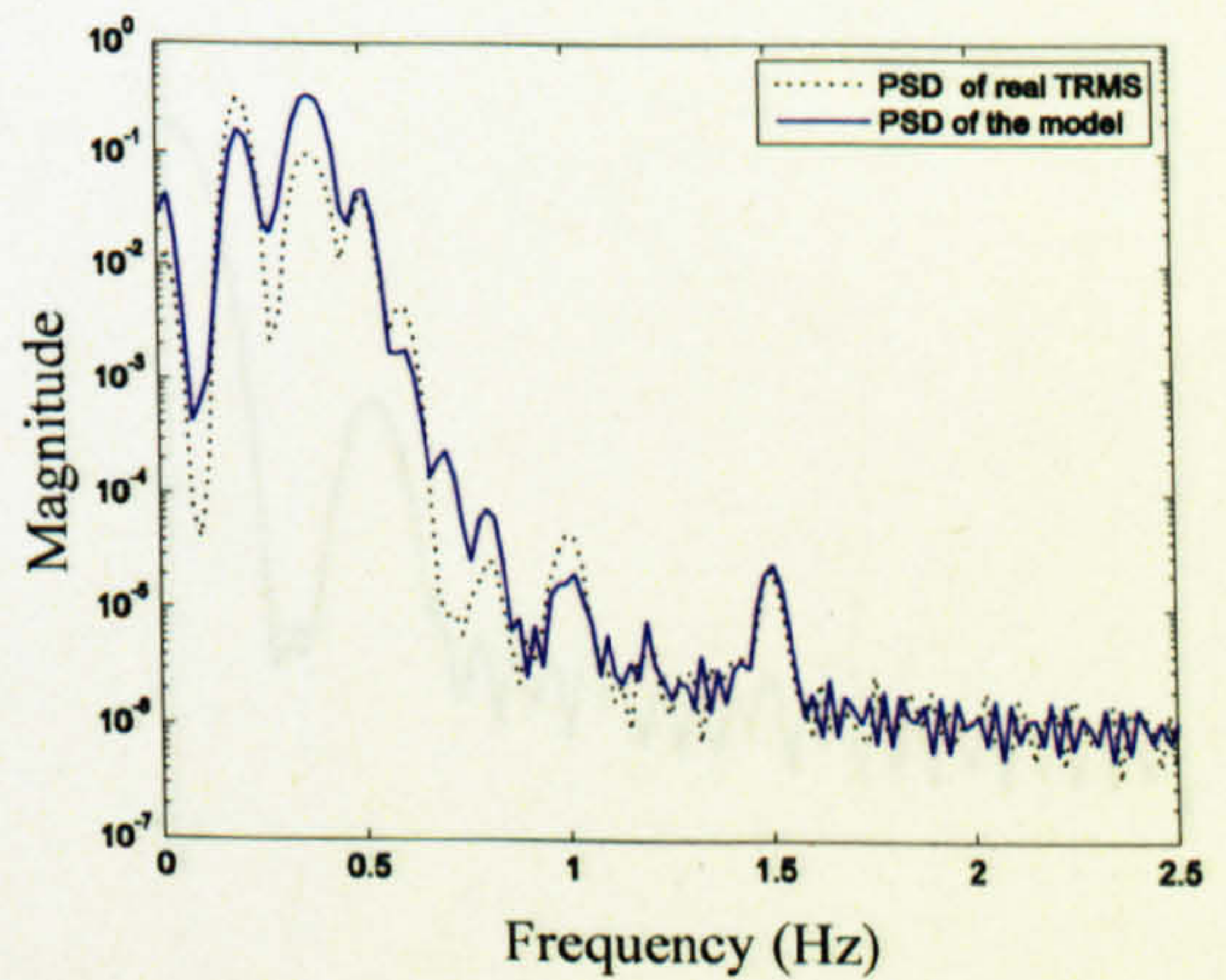
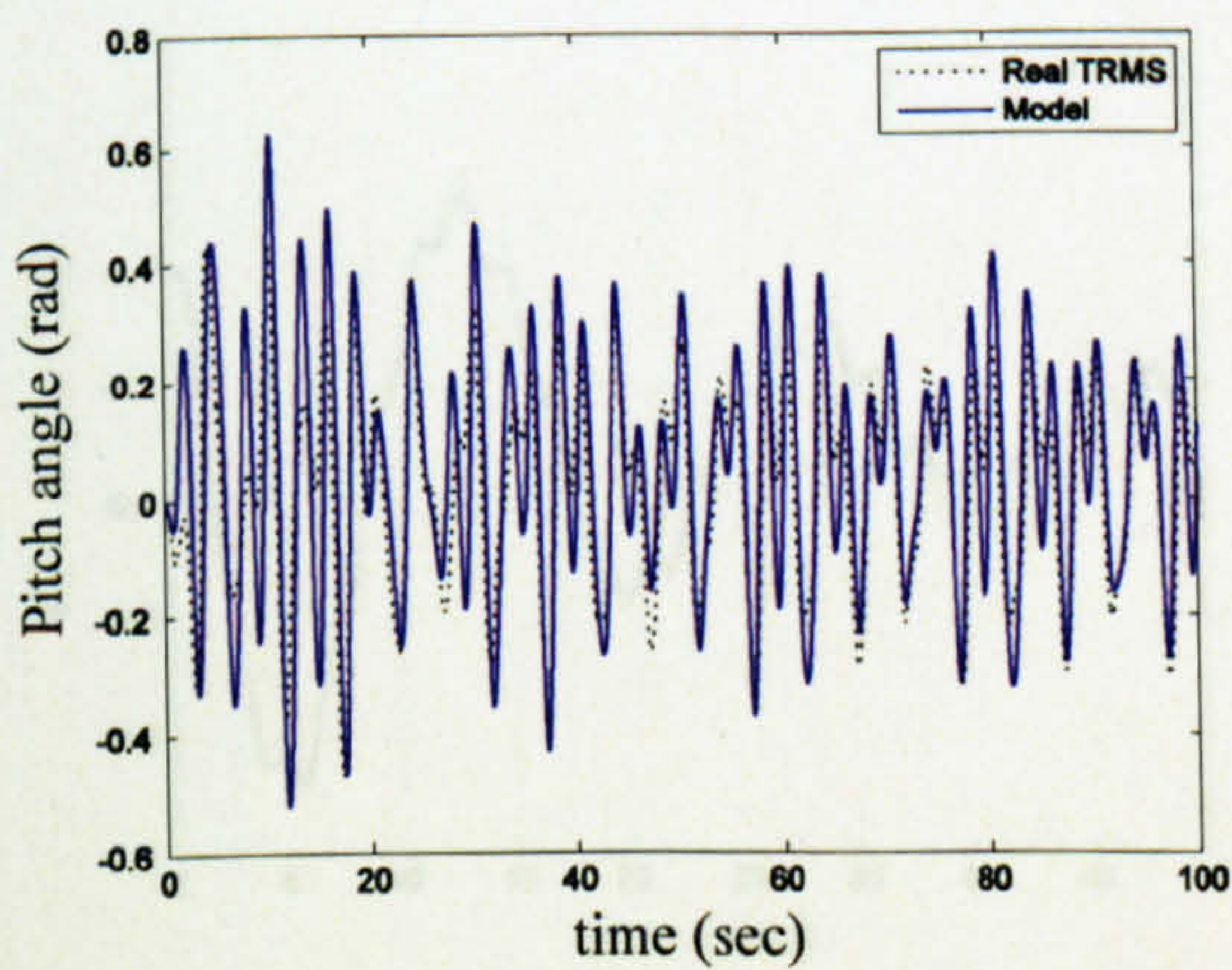


Figure 2.50 Pitch angle and PSD (case 5)

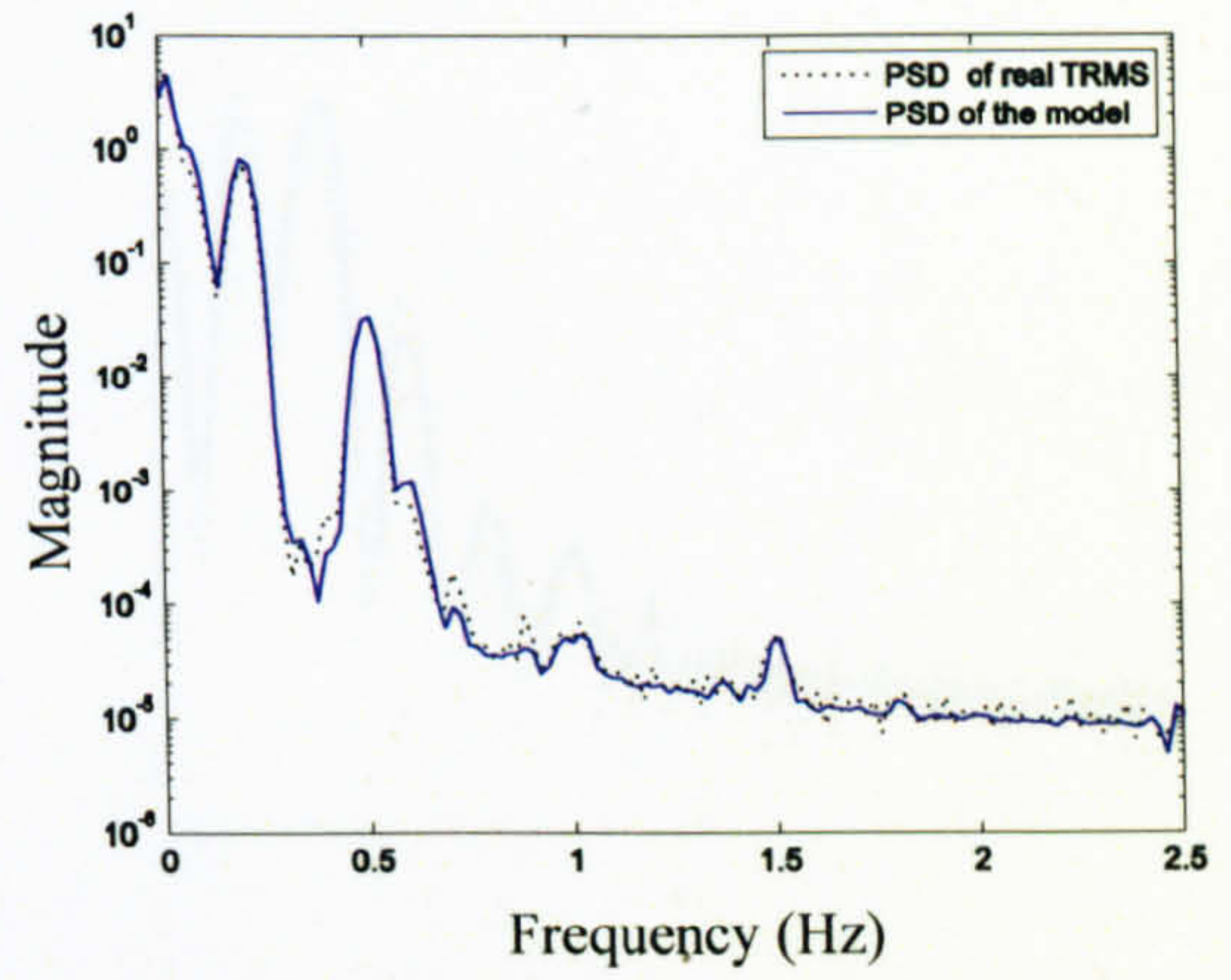
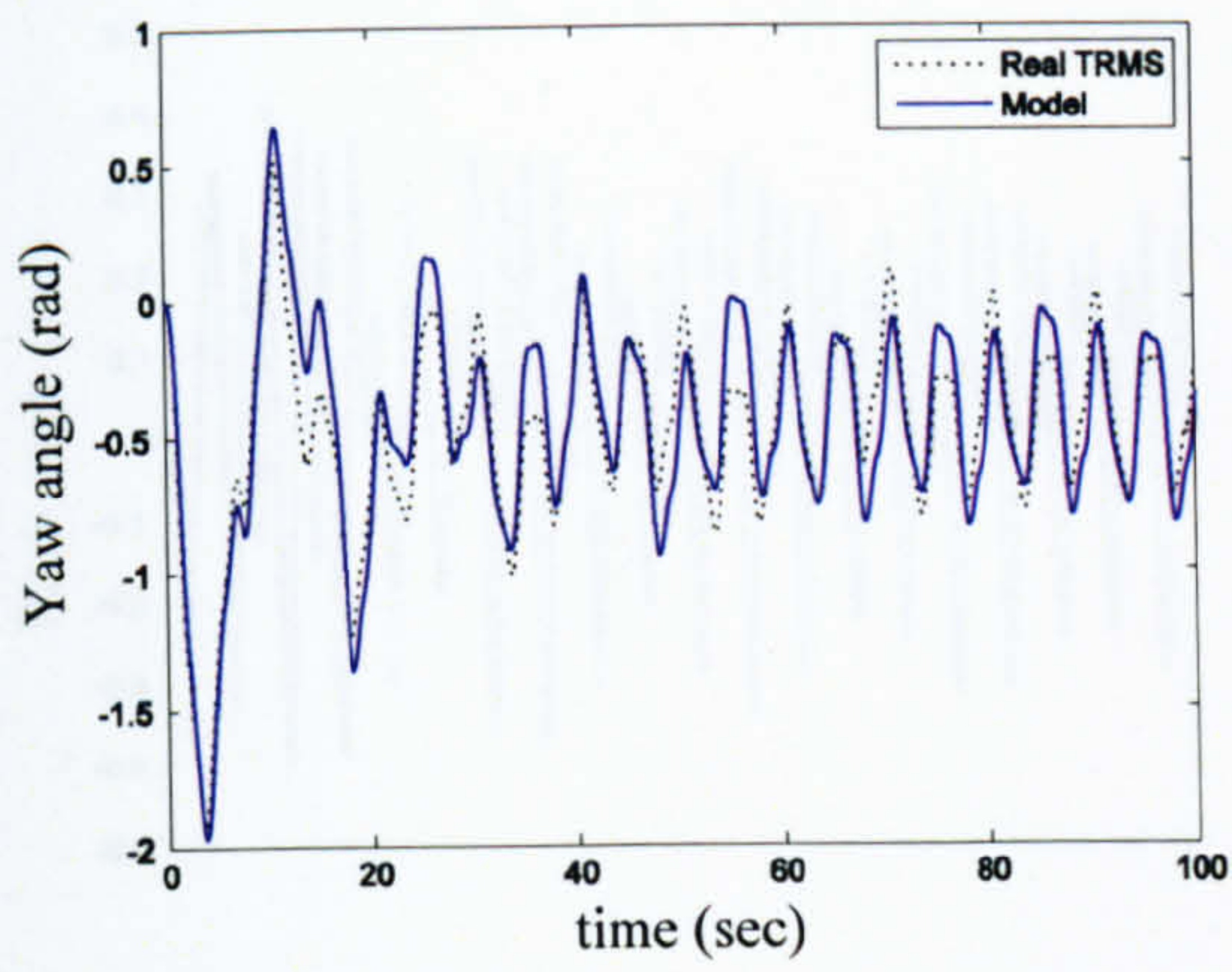


Figure 2.51 Yaw angle and PSD (case 5)

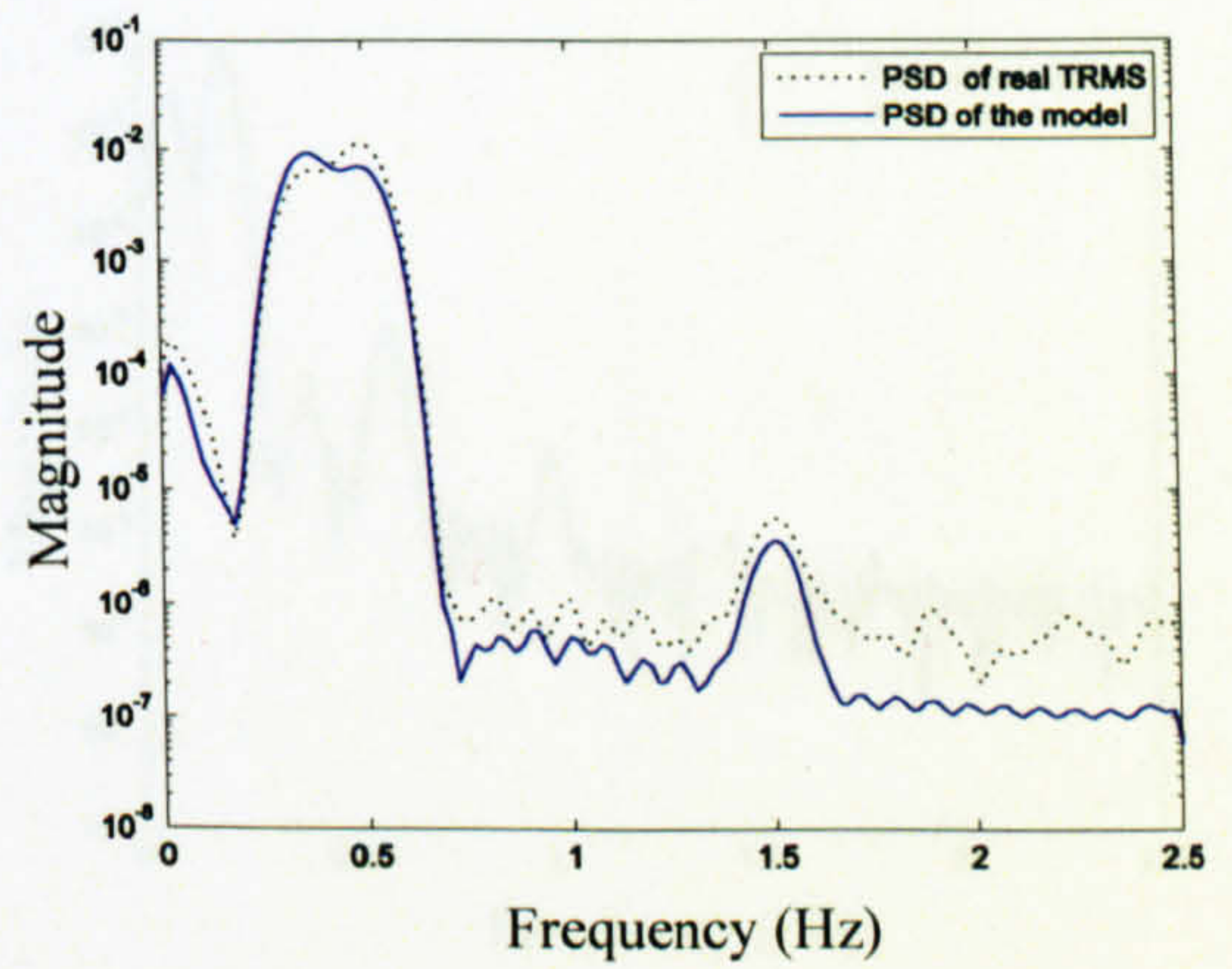
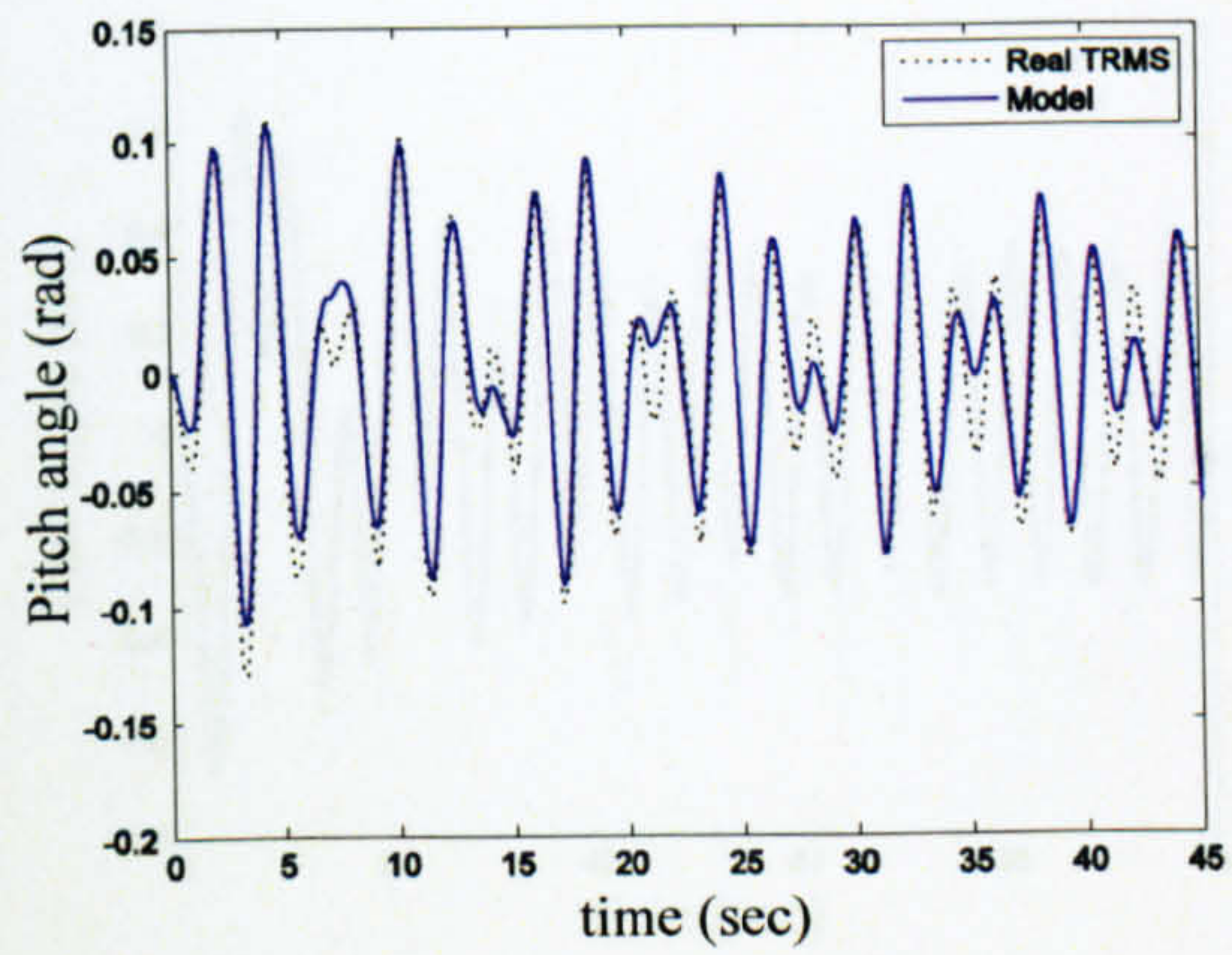


Figure 2.52 Pitch angle and PSD (case 6)

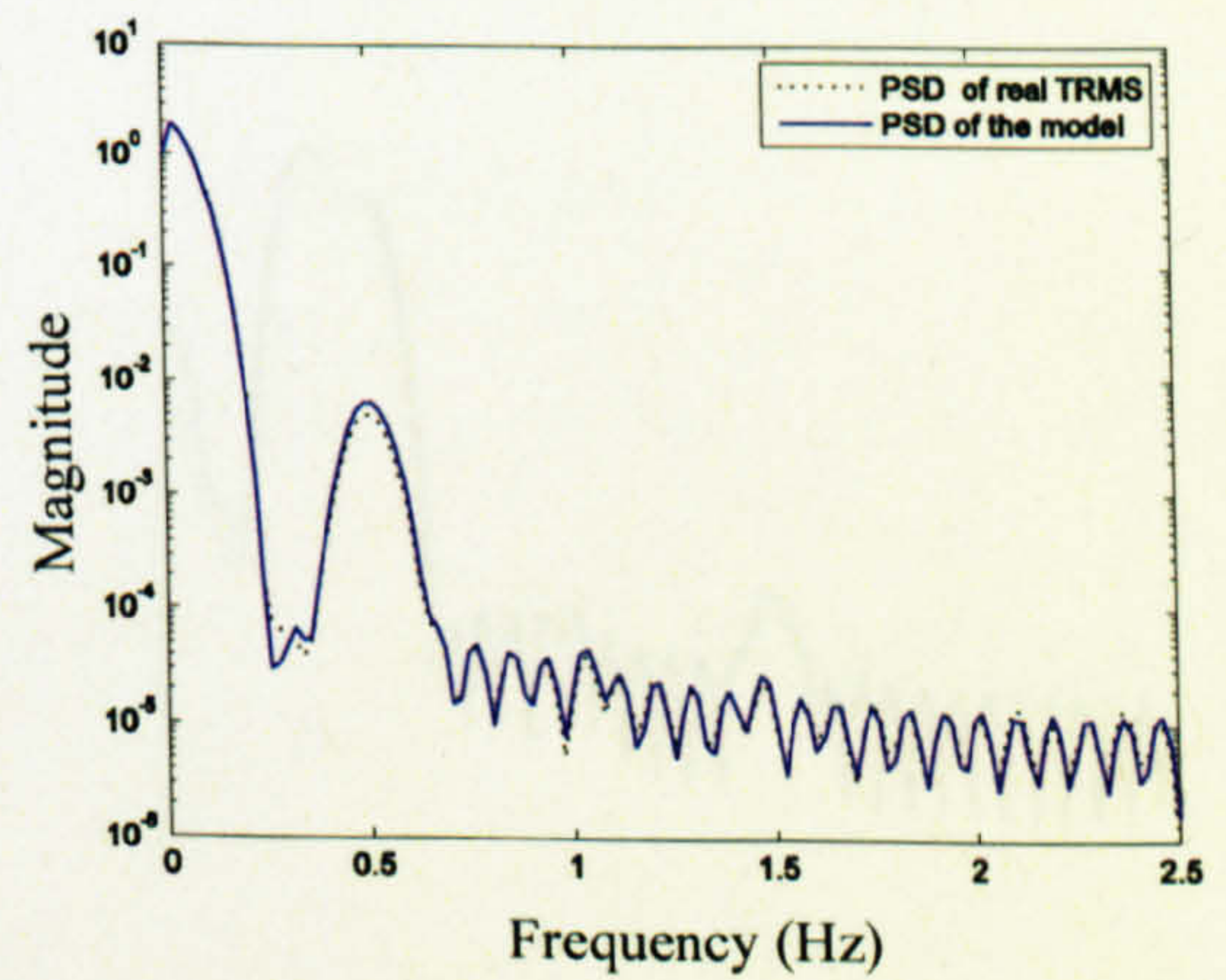
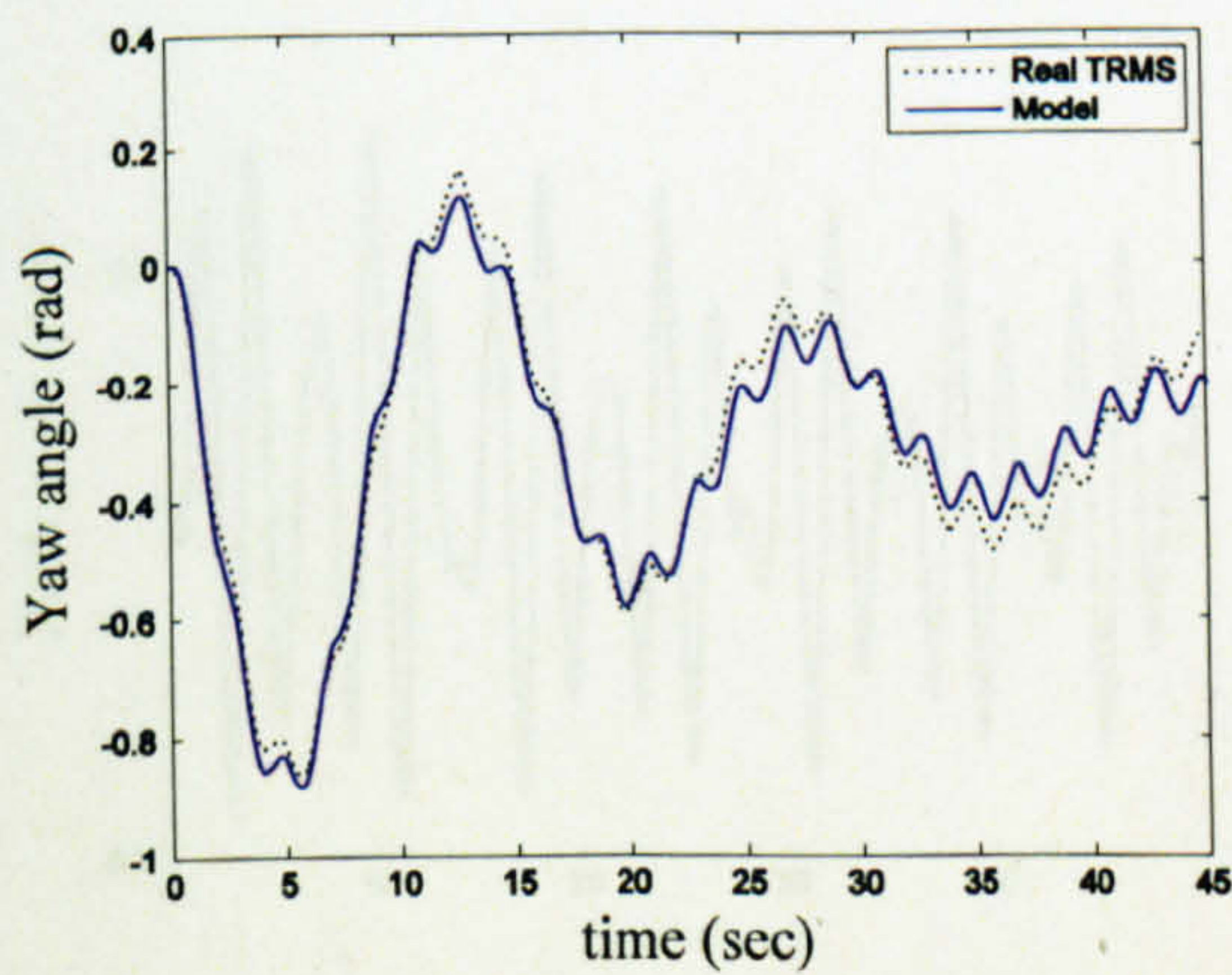


Figure 2.53 Yaw angle and PSD (case 6)

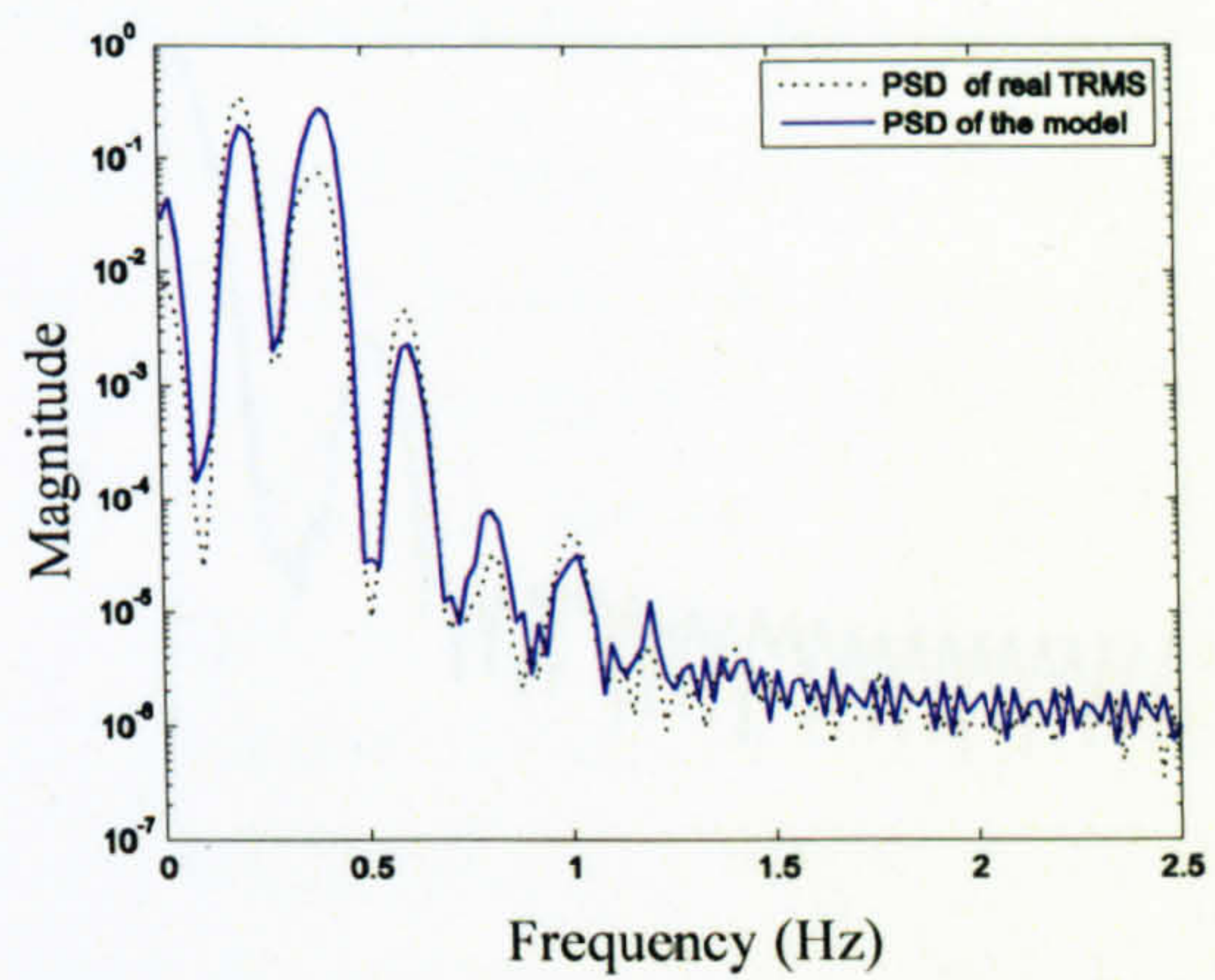
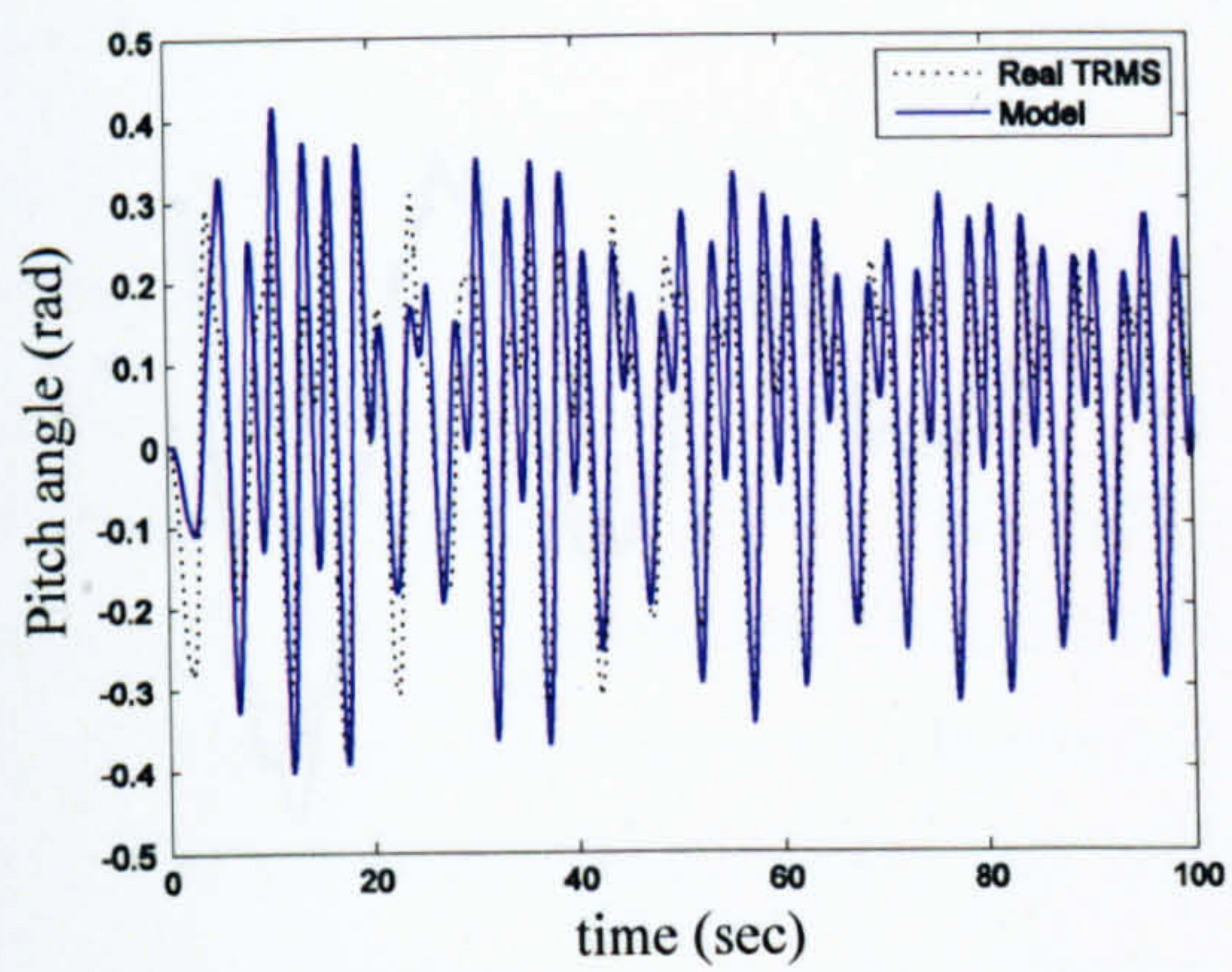


Figure 2.54 Pitch angle and PSD (case 7)

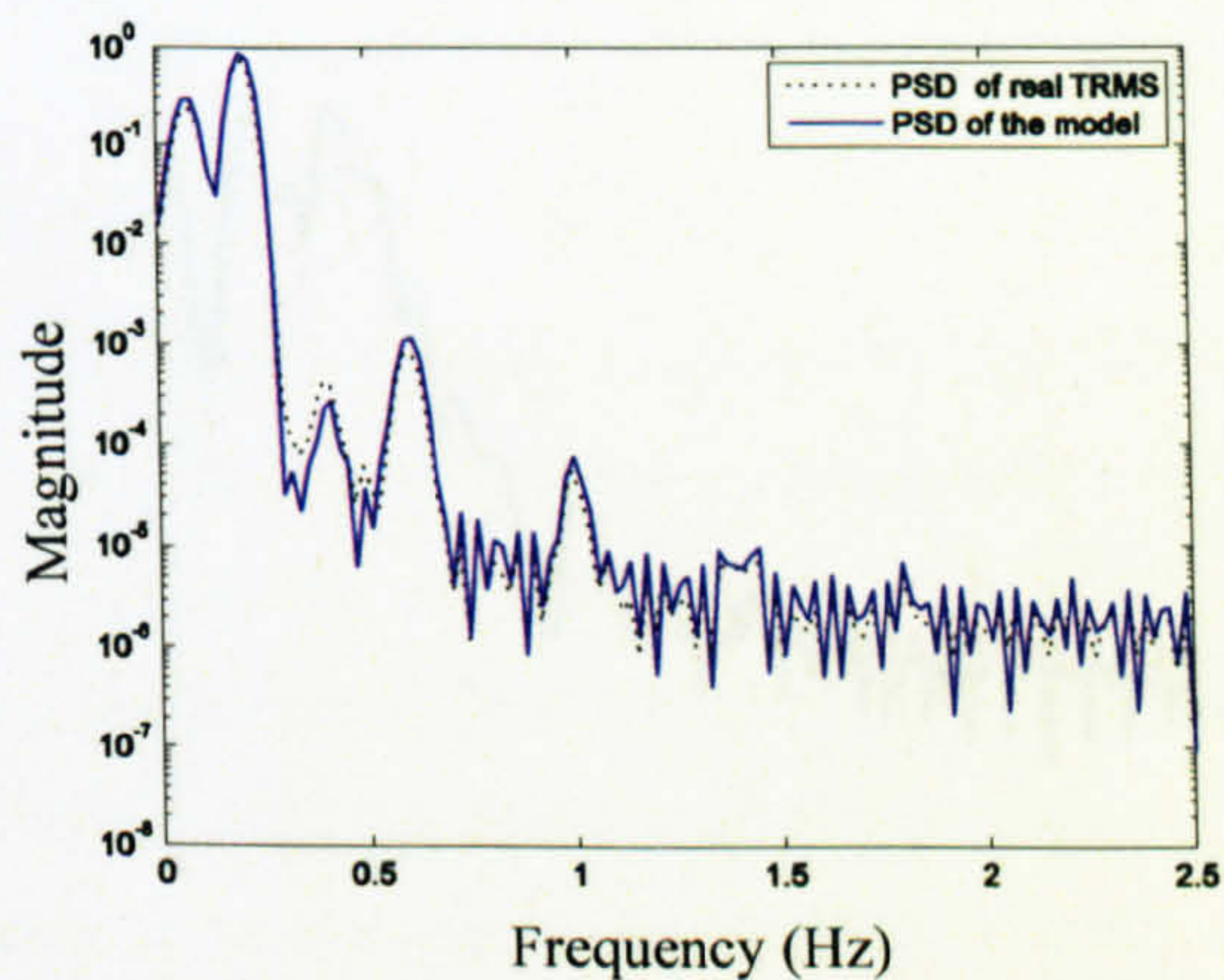
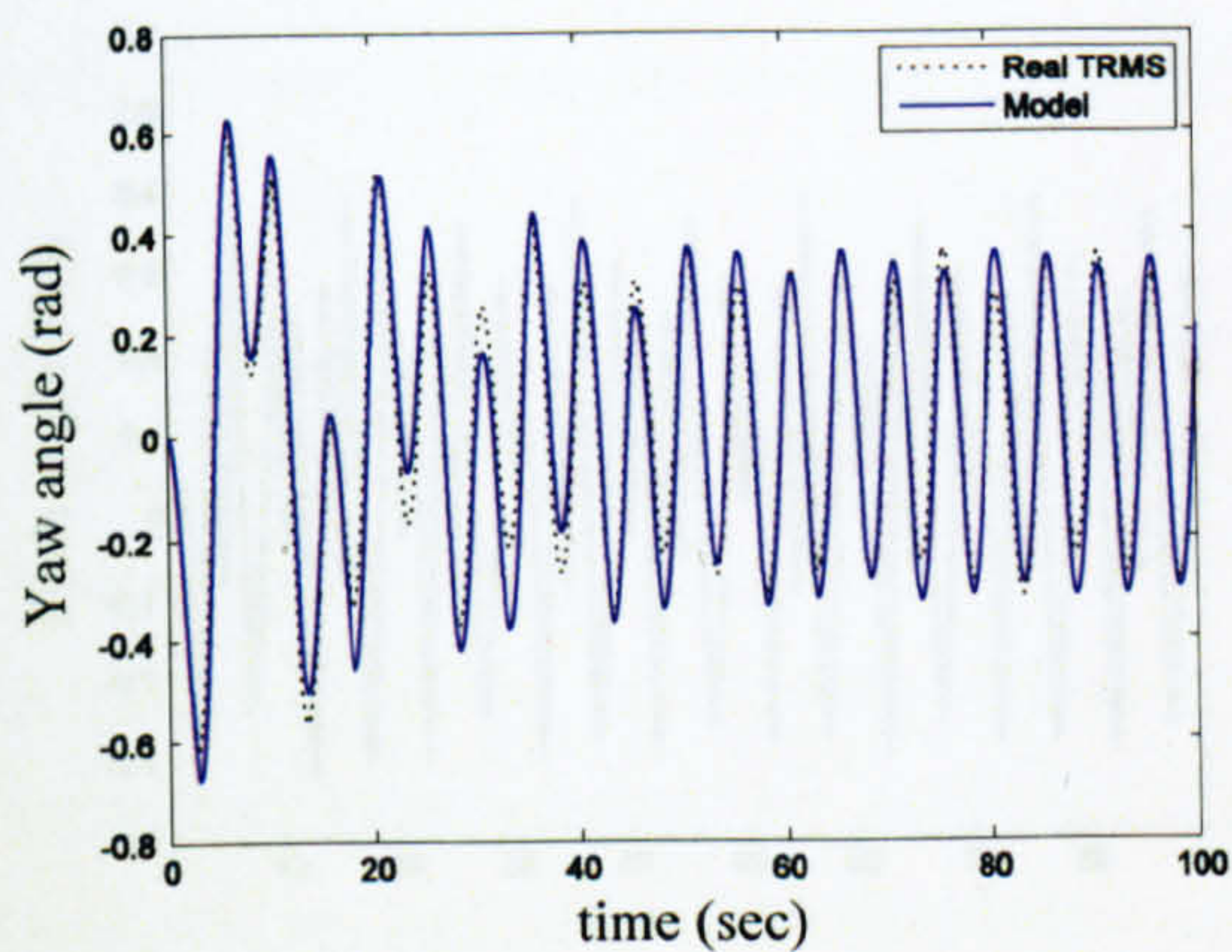


Figure 2.55 Yaw angle and PSD (case 7)

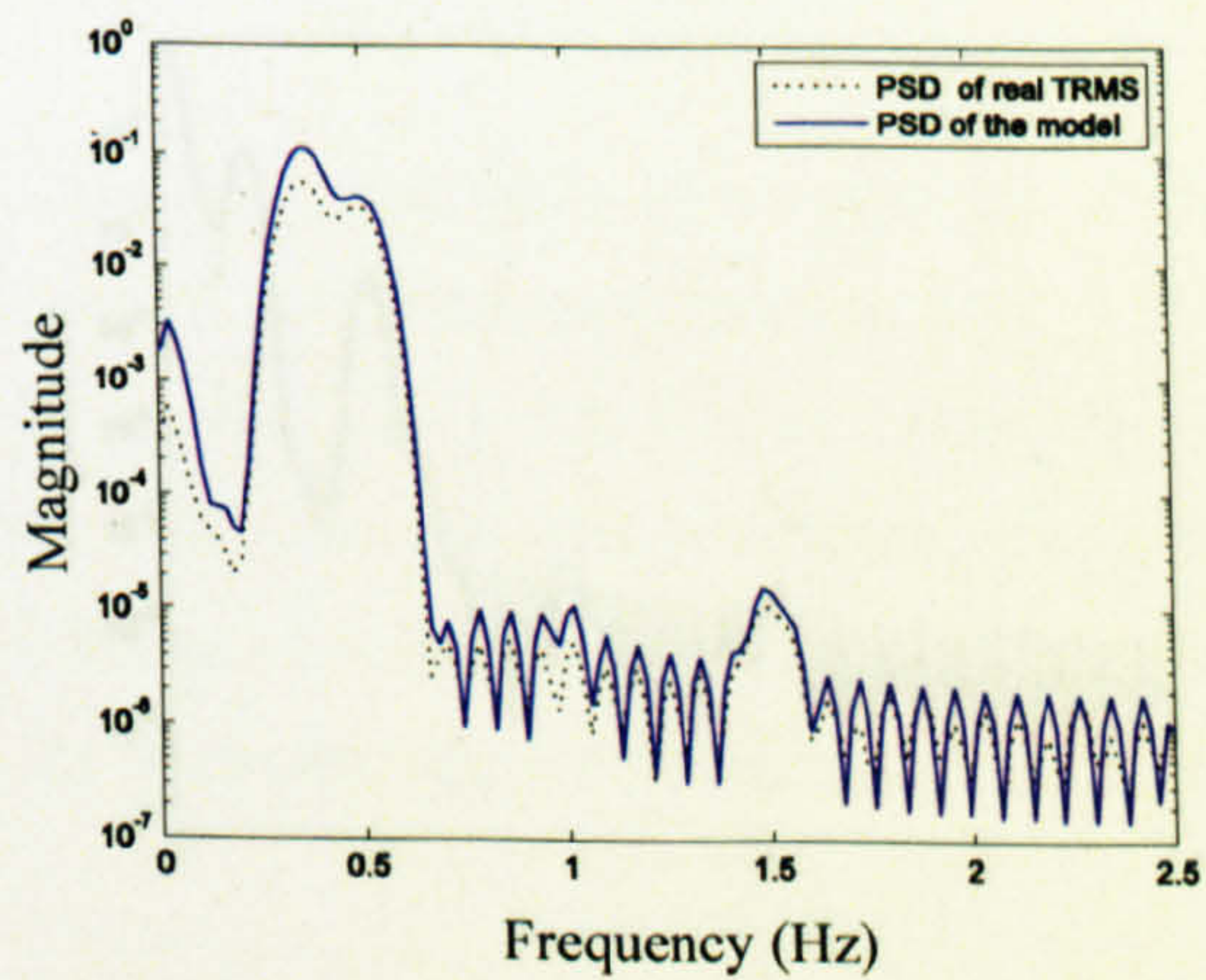
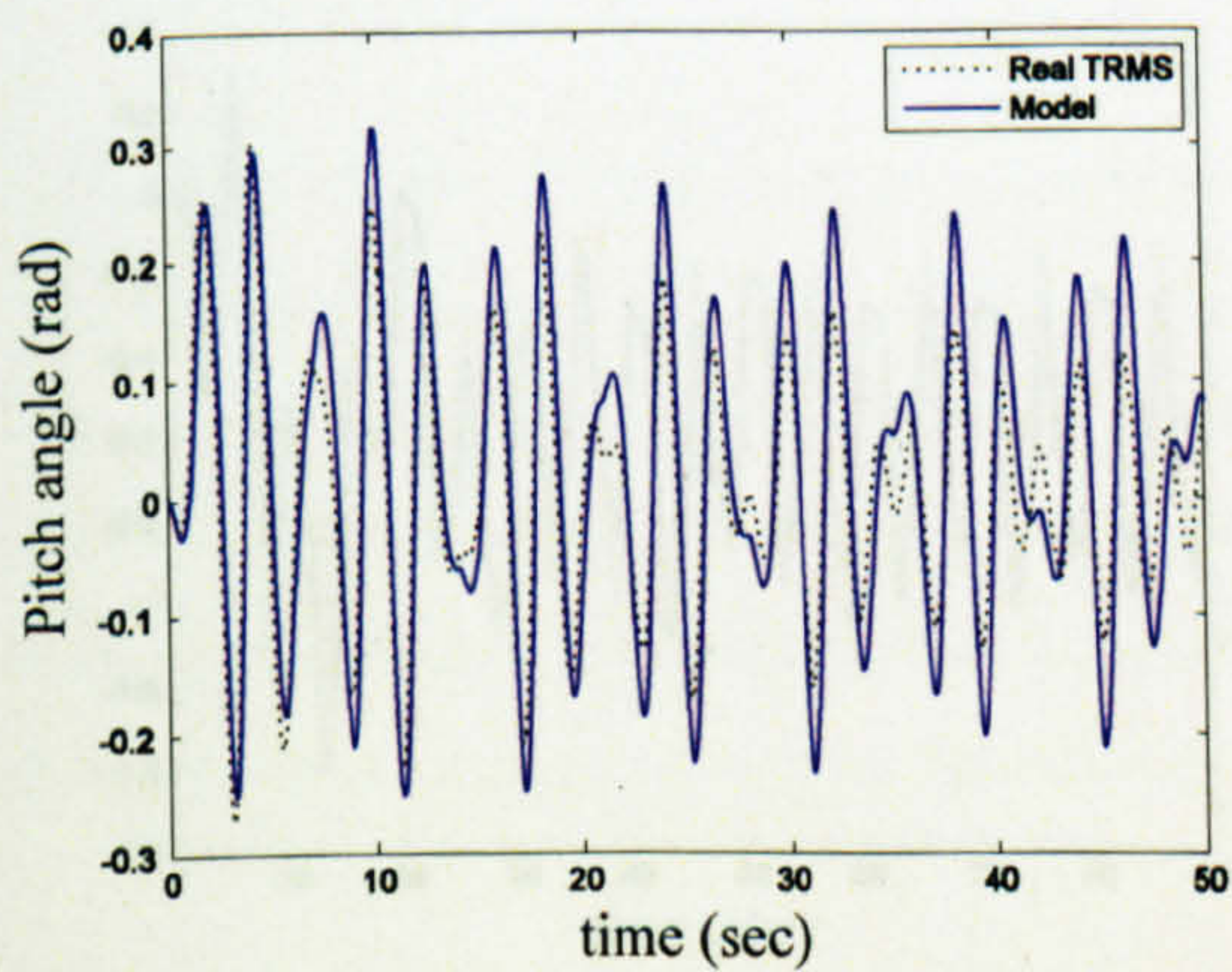


Figure 2.56 Pitch angle and PSD (case 8)

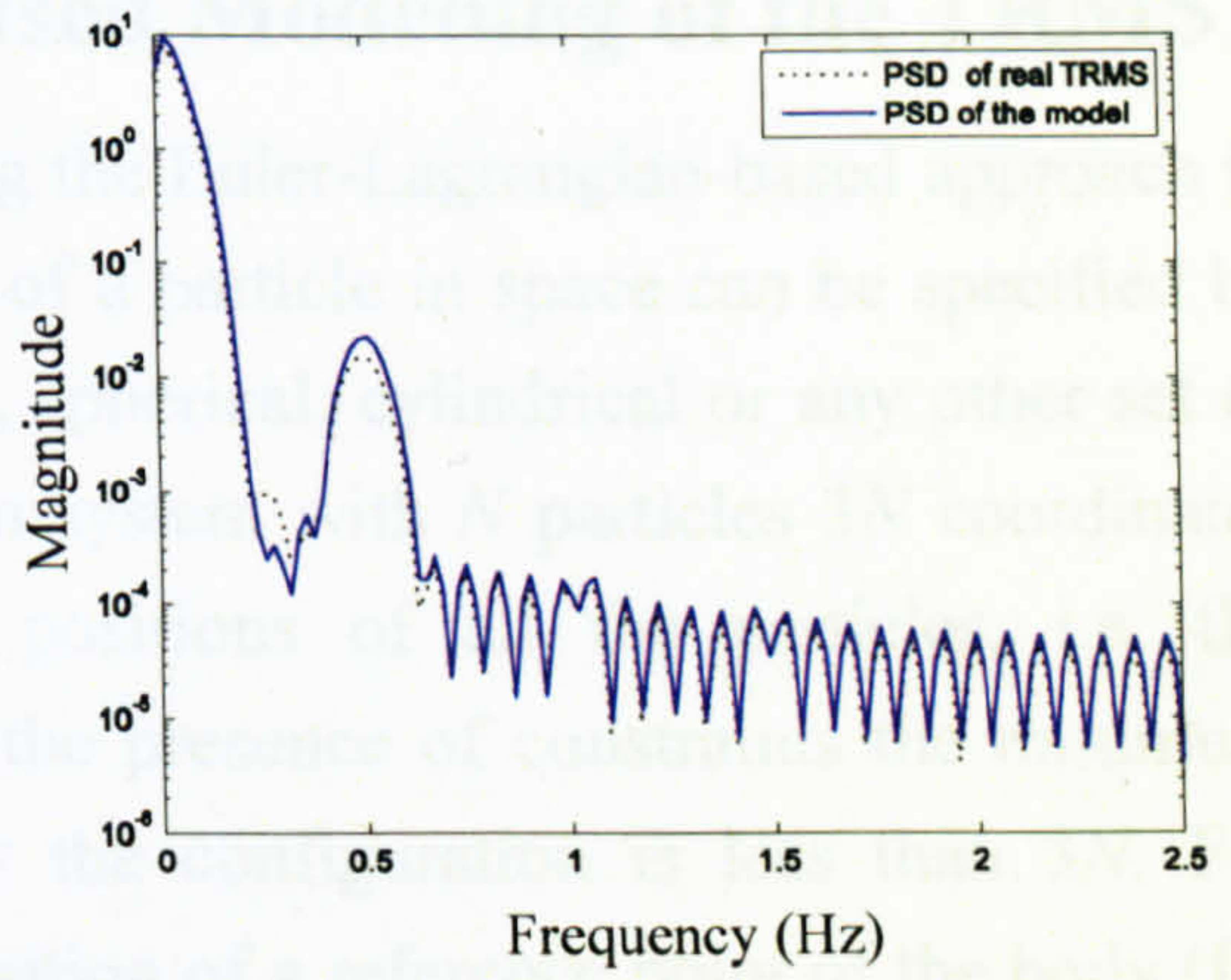
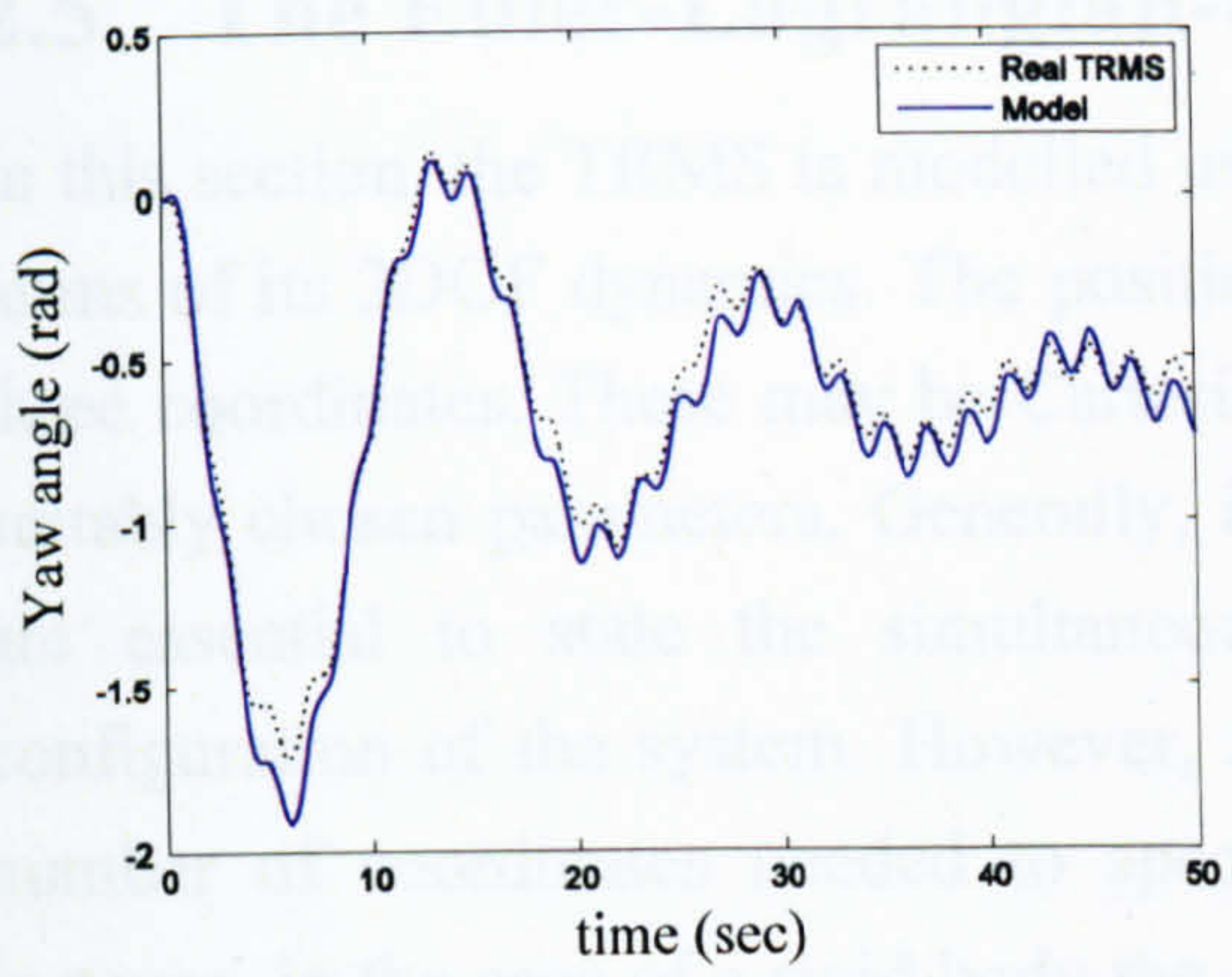


Figure 2.57 Yaw angle and PSD (case 8)

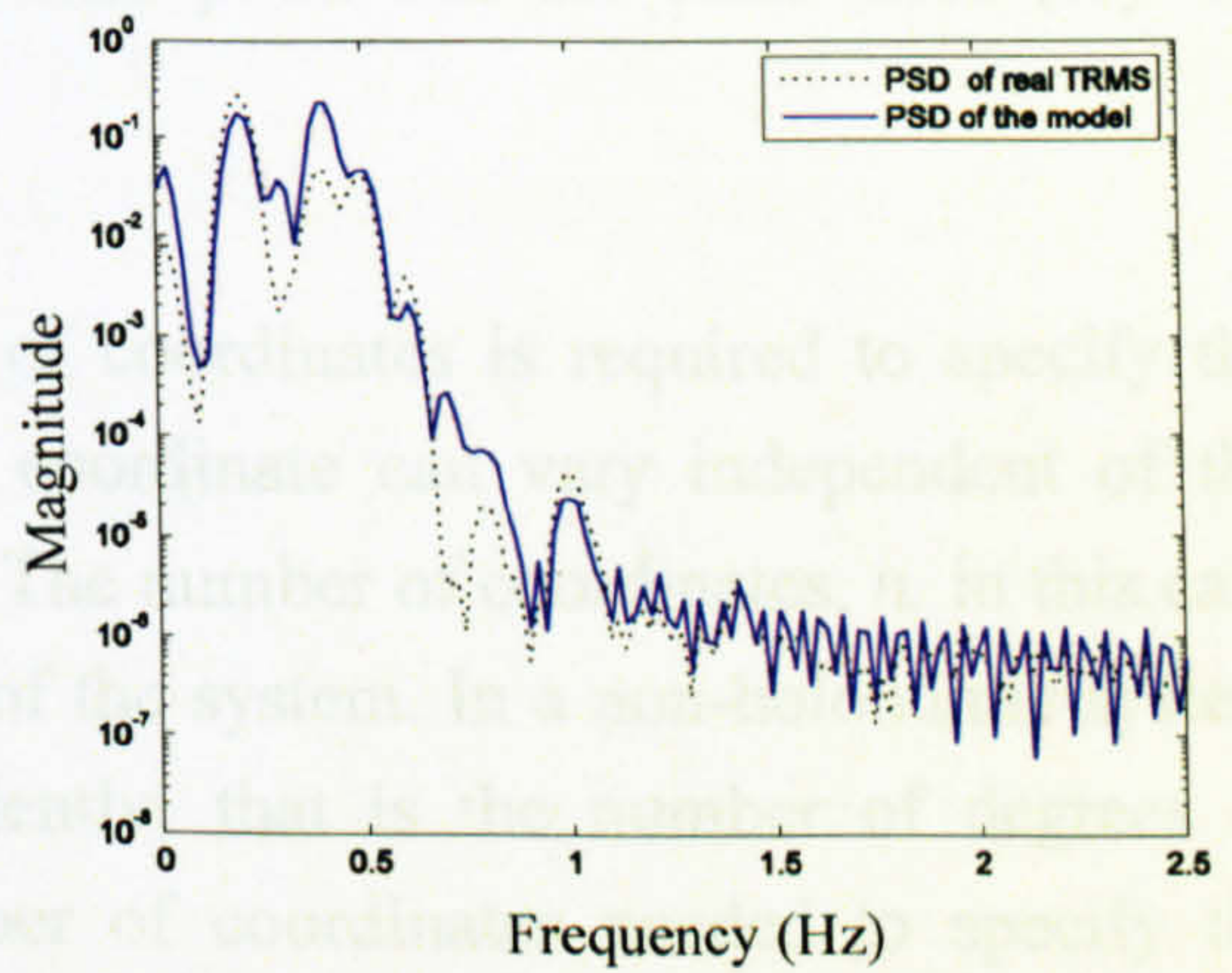
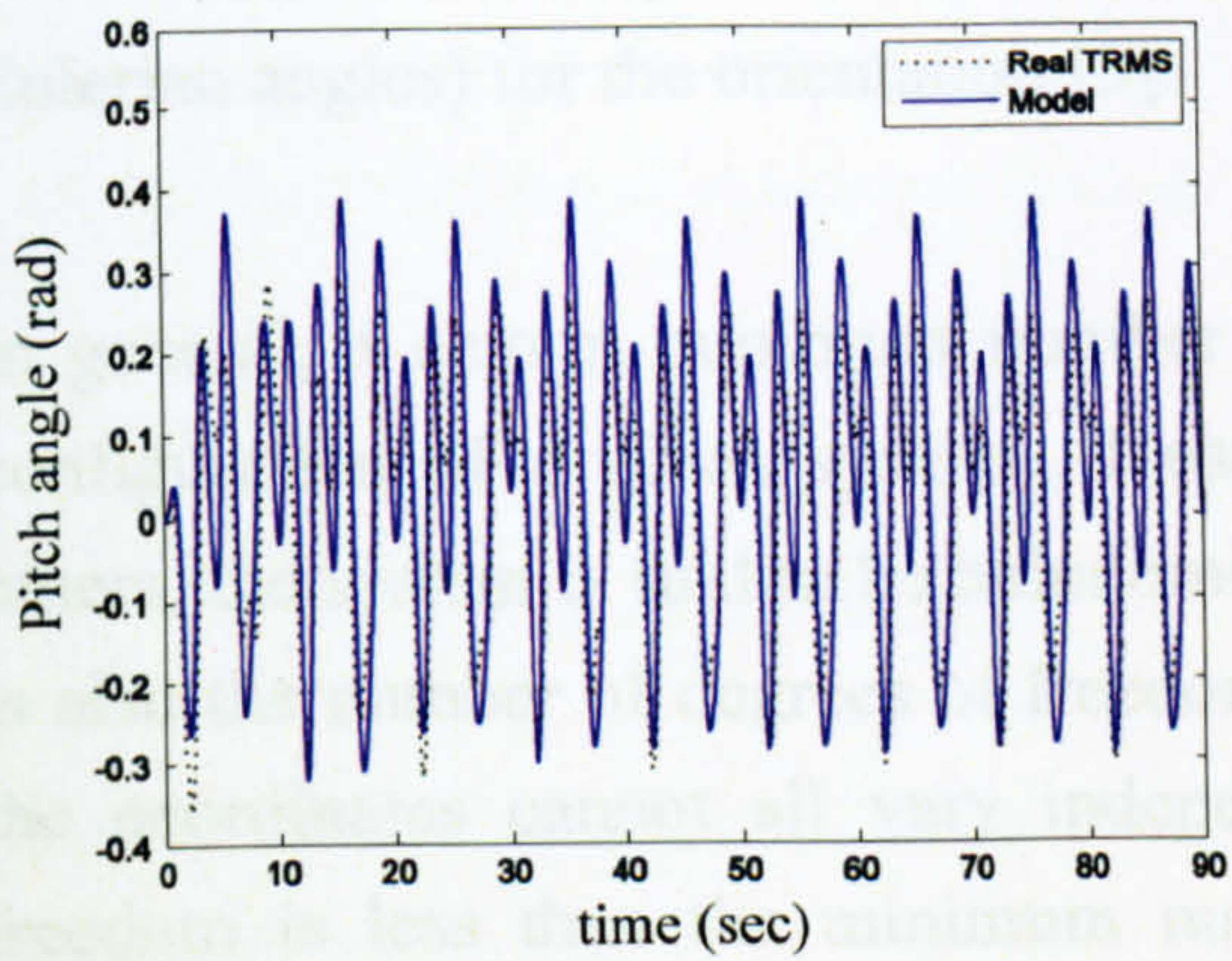


Figure 2.58 Pitch angle and PSD (case 9)

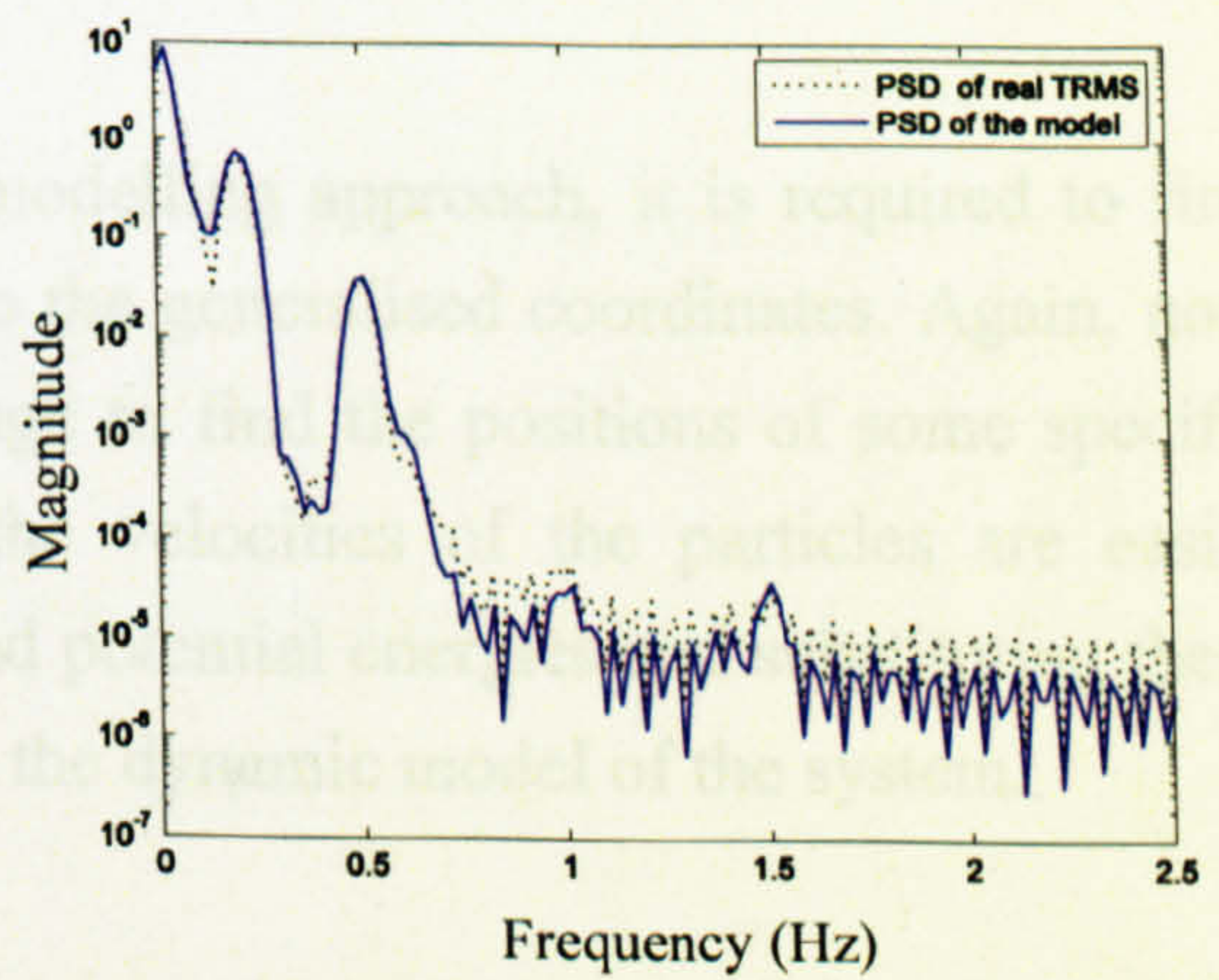
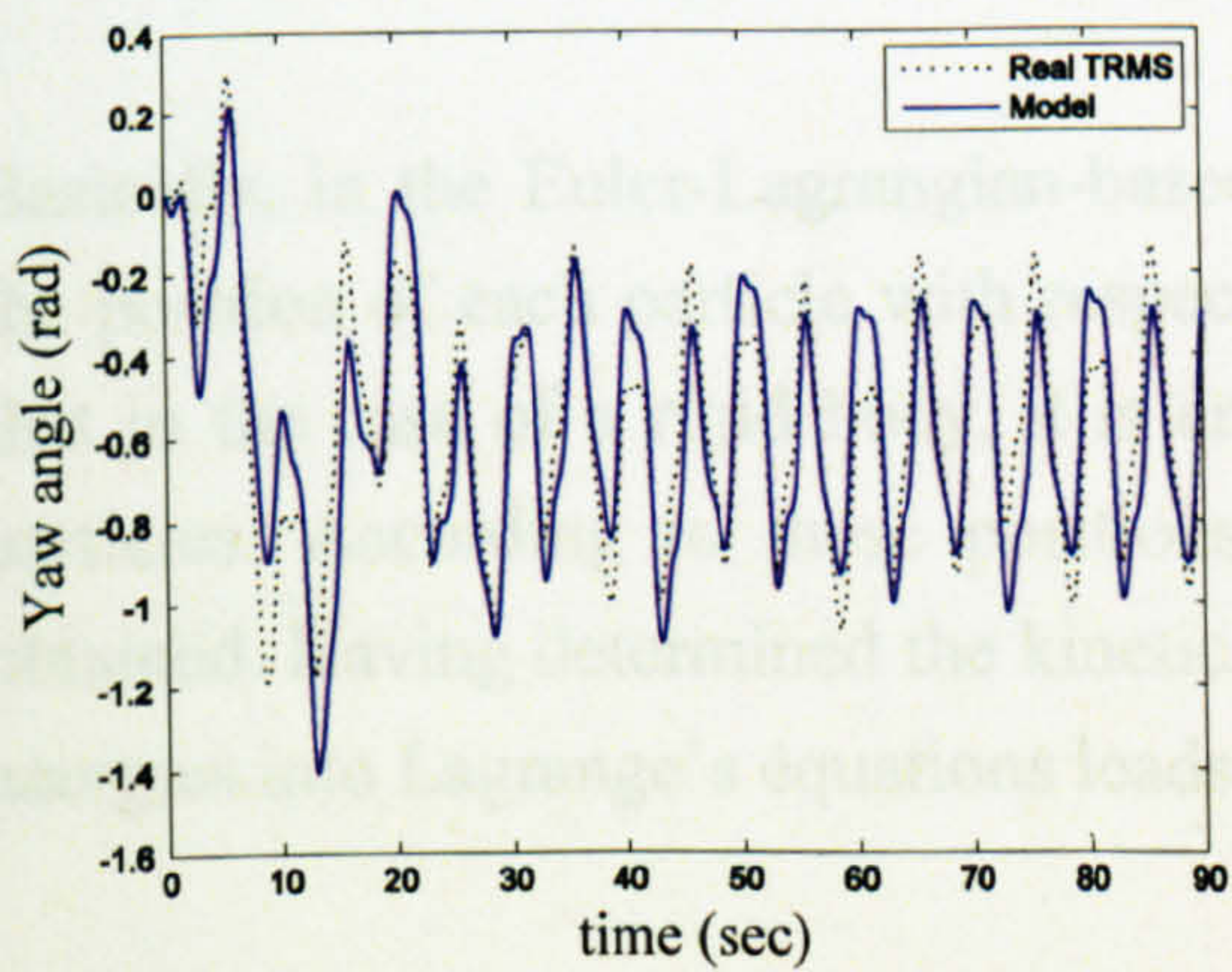


Figure 2.59 Yaw angle and PSD (case 9)

2.5 The Euler-Lagrangian-based Modelling of the TRMS

In this section, the TRMS is modelled using the Euler-Lagrangian-based approach in terms of its 2DOF dynamics. The position of a particle in space can be specified by three coordinates. These may be Cartesian, spherical, cylindrical or any other set of suitably chosen parameters. Generally, in a system with N particles $3N$ coordinates are essential to state the simultaneous positions of all the particles, i.e. the configuration of the system. However, in the presence of constraints the minimum number of coordinates needed to specify the configuration is less than $3N$. For instance, in the case of a rigid body the position of a reference point of the body (for example, the centre of mass) and the orientation of the body in space are needed for specifying the configuration. Therefore, in the case of a single rigid body, only six coordinates are enough, three for the reference point and the other three (say the Eulerian angles) for the orientation [3].

In general, a certain minimum number n of coordinates is required to specify the configuration of a given system. If each coordinate can vary independent of the others, the system is said to be holonomic. The number of coordinates, n , in this case is also the number of degrees of freedom of the system. In a non-holonomic system the coordinates cannot all vary independently; that is the number of degrees of freedom is less than the minimum number of coordinates needed to specify the configuration [3]. It is interesting to note that the TRMS is a holonomic system and in the case of its 2DOF dynamics, two coordinates, the yaw and pitch angles, are enough to derive the governing equations of the system.

Basically, in the Euler-Lagrangian-based modelling approach, it is required to find the position of each particle with respect to the generalised coordinates. Again, note that in the case of a rigid body, it is enough to find the positions of some specific particles. According to these positions, the velocities of the particles are easily obtained. Having determined the kinetic and potential energies and substituting these energies into Lagrange's equations leads to the dynamic model of the system.

2.5.1 The TRMS Particles Definition

In developing the Lagrangian-based model of the TRMS, the system configuration is divided into three subsystems (see Figure 2.60); the first one consists of the free beam (the beam connecting the main and tail rotors), the tail rotor, the main rotor, the

tail shield and the main shield; the second category comprises the counter balance beam and weight; and finally, the third division is the pivoted beam.

2.5.1.1 Free Beam

Assume $[r_x(R_1), r_y(R_1), r_z(R_1)]$ denotes the coordinate of the point P_1 on the free beam parameterised in the distance R_1 from O_1 (that means $P_1O_1 = R_1$). And also assume $OO_1 = h$ in which O is the origin of the coordinates. It is noted that in order to simplify Figure 2.60 the x and y axes have been drawn from O_2 .

According to Figures 2.60, 2.61 and 2.62 the following set of equations can be obtained. Figures 2.60 and 2.62 have been used for $r_x(R_1)$ and $r_y(R_1)$, and Figure 2.61 has been drawn to make the calculation of $r_z(R_1)$ more convenient.

$$\begin{cases} r_x(R_1) = R_1 \sin \alpha_h \cos \alpha_v + h \cos \alpha_h \\ r_y(R_1) = R_1 \cos \alpha_h \cos \alpha_v - h \sin \alpha_h \\ r_z(R_1) = R_1 \sin \alpha_v \end{cases} \quad (2.39)$$

The corresponding velocities are obtained by differentiating equations (2.39) with respect to time.

$$\begin{cases} v_x(R_1) = R_1 \dot{\alpha}_h \cos \alpha_h \cos \alpha_v - R_1 \dot{\alpha}_v \sin \alpha_h \sin \alpha_v - h \dot{\alpha}_h \sin \alpha_h \\ v_y(R_1) = -R_1 \dot{\alpha}_h \sin \alpha_h \cos \alpha_v - R_1 \dot{\alpha}_v \cos \alpha_h \sin \alpha_v - h \dot{\alpha}_h \cos \alpha_h \\ v_z(R_1) = R_1 \dot{\alpha}_v \cos \alpha_v \end{cases} \quad (2.40)$$

The square magnitude of the velocity of P_1 is given by the following expression,

$$v^2(R_1) = v_x^2(R_1) + v_y^2(R_1) + v_z^2(R_1) \quad (2.41)$$

After substitution and simplification, the following equation is achieved.

$$v^2(R_1) = R_1^2 \dot{\alpha}_h^2 \cos^2 \alpha_v + h^2 \dot{\alpha}_h^2 + R_1^2 \dot{\alpha}_v^2 + 2R_1 h \dot{\alpha}_h \dot{\alpha}_v \sin \alpha_v \quad (2.42)$$

Note that α_h has no effect on the $r_z(R)$'s and for simplicity, it can be assumed to be zero, as shown in Figure 2.61.

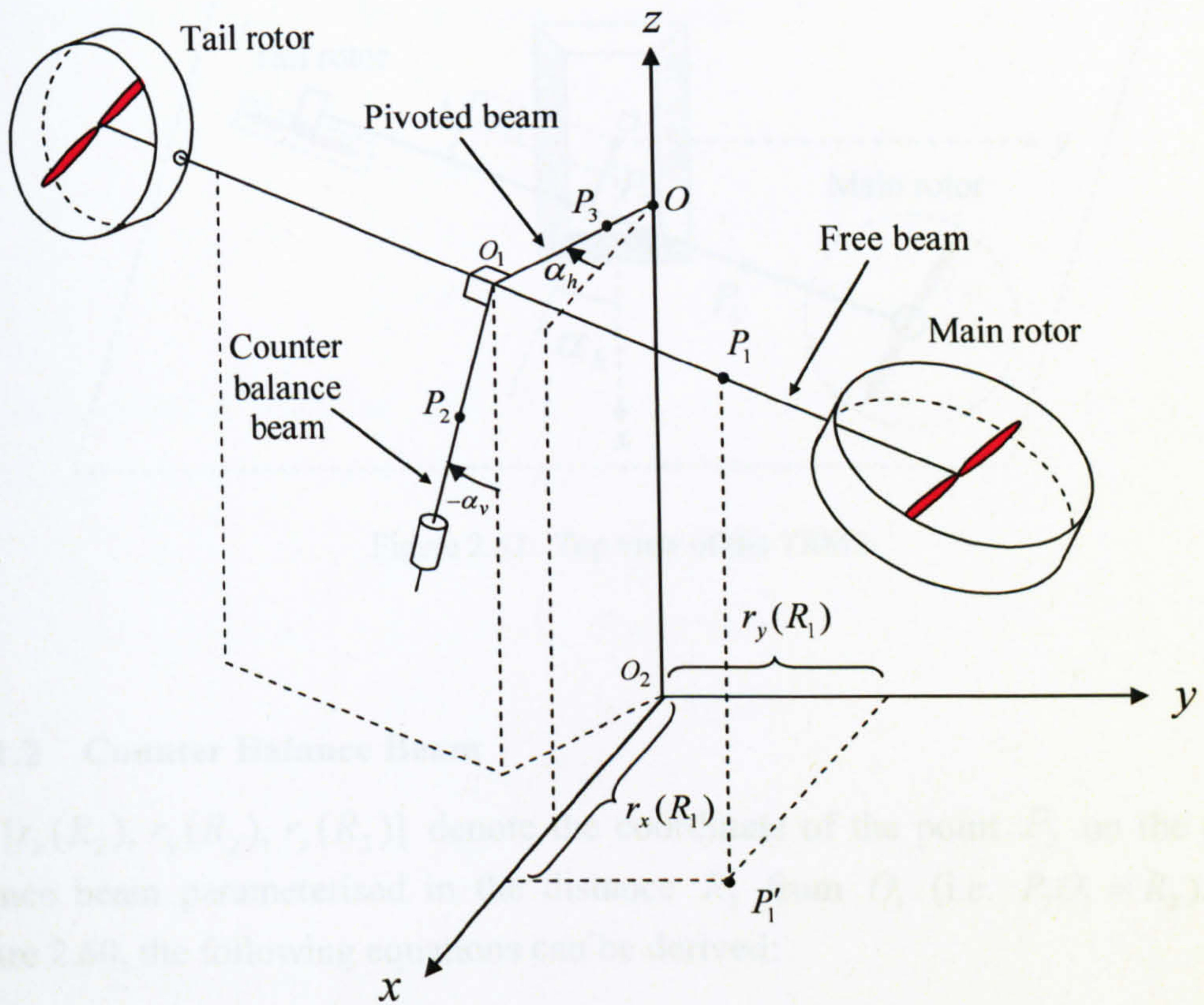


Figure 2.60 The TRMS

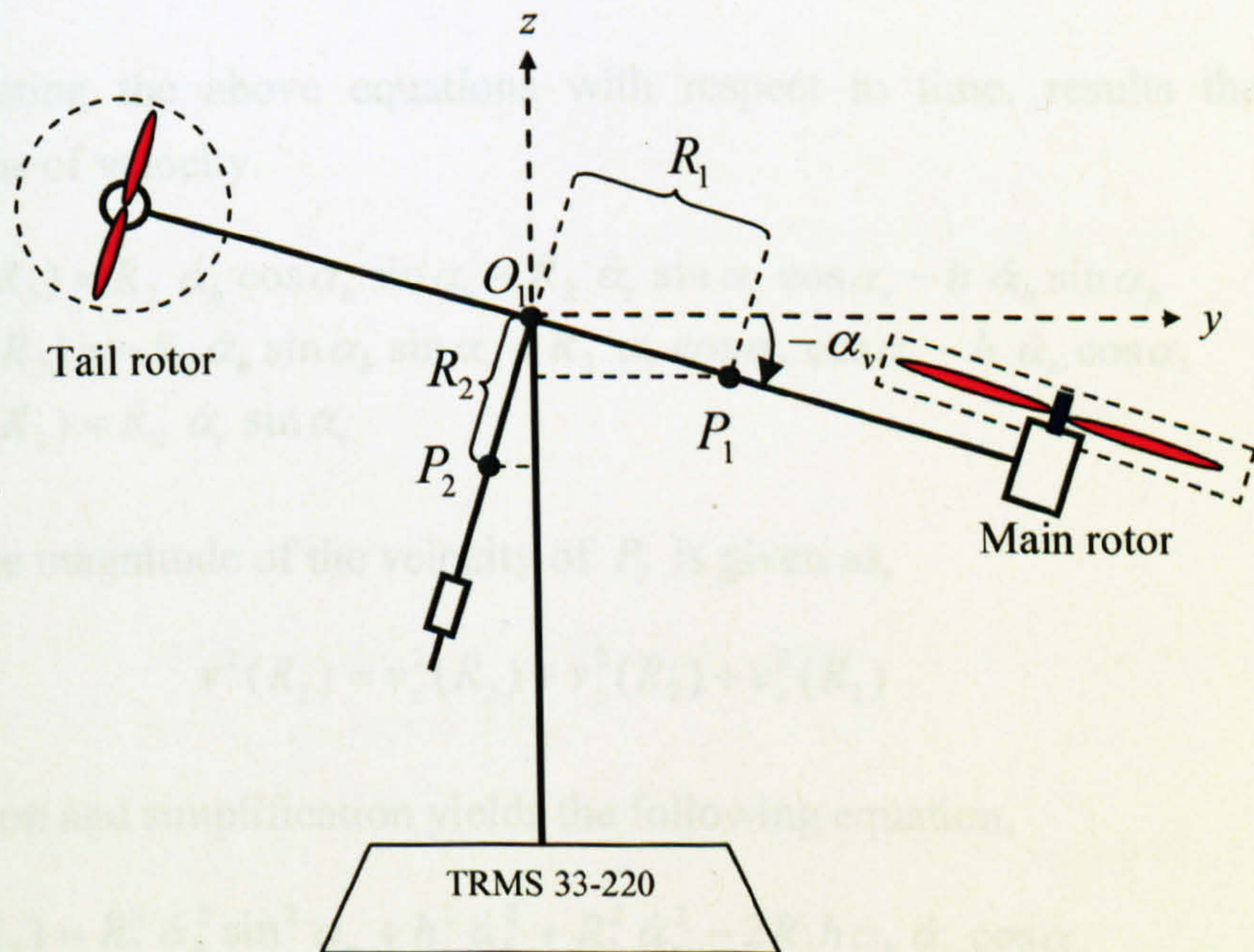


Figure 2.61 Front view of the TRMS with $\alpha_h = 0$

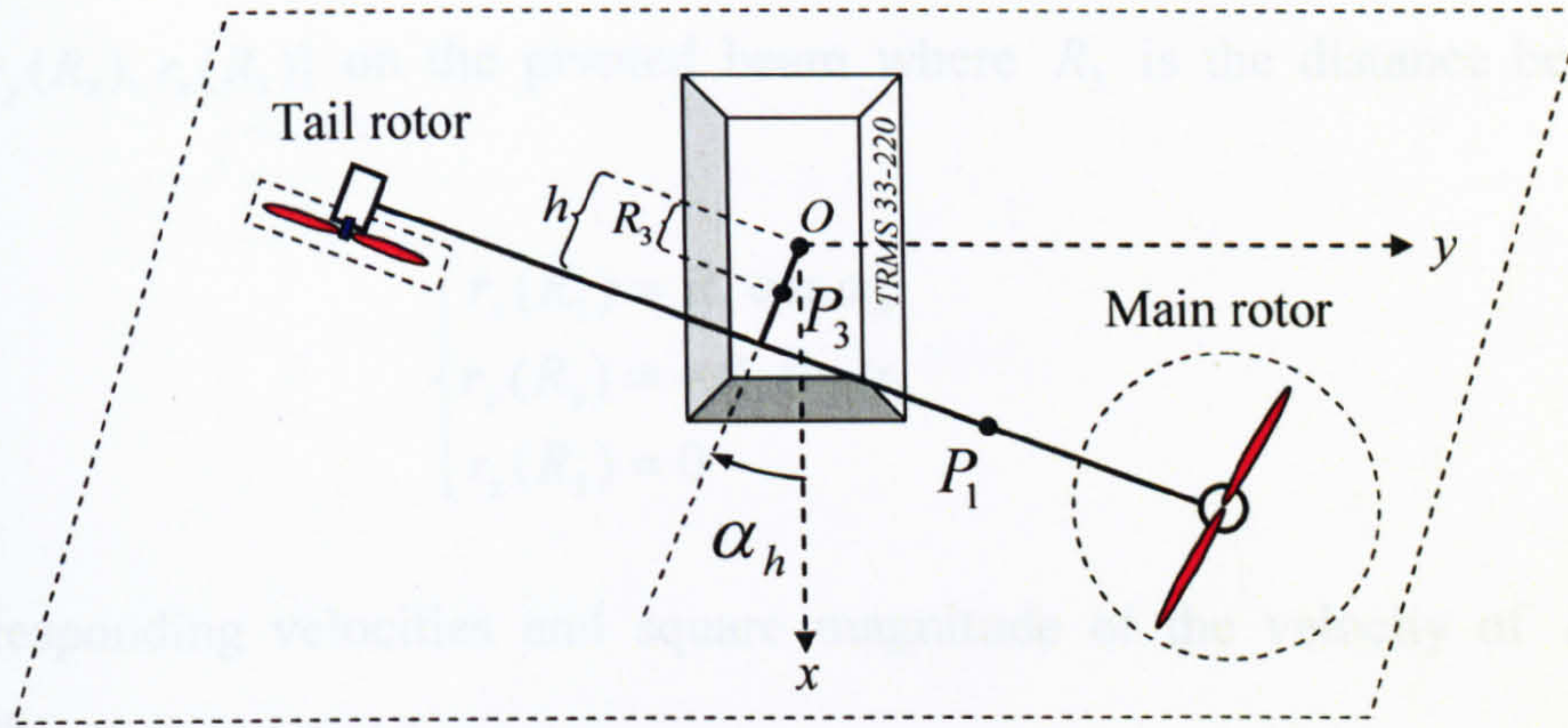


Figure 2.62 Top view of the TRMS

2.5.1.2 Counter Balance Beam

Let $[r_x(R_2), r_y(R_2), r_z(R_2)]$ denote the coordinate of the point P_2 on the counter balance beam parameterised in the distance R_2 from O_1 (i.e. $P_2O_1 = R_2$). Using Figure 2.60, the following equations can be derived:

$$\begin{cases} r_x(R_2) = R_2 \sin \alpha_h \sin \alpha_v + h \cos \alpha_h \\ r_y(R_2) = R_2 \cos \alpha_h \sin \alpha_v - h \sin \alpha_h \\ r_z(R_2) = -R_2 \cos \alpha_v \end{cases} \quad (2.43)$$

Differentiating the above equations with respect to time, results the following expressions of velocity.

$$\begin{cases} v_x(R_2) = R_2 \dot{\alpha}_h \cos \alpha_h \sin \alpha_v + R_2 \dot{\alpha}_v \sin \alpha_h \cos \alpha_v - h \dot{\alpha}_h \sin \alpha_h \\ v_y(R_2) = -R_2 \dot{\alpha}_h \sin \alpha_h \sin \alpha_v + R_2 \dot{\alpha}_v \cos \alpha_h \cos \alpha_v - h \dot{\alpha}_h \cos \alpha_h \\ v_z(R_2) = R_2 \dot{\alpha}_v \sin \alpha_v \end{cases} \quad (2.44)$$

The square magnitude of the velocity of P_2 is given as,

$$v^2(R_2) = v_x^2(R_2) + v_y^2(R_2) + v_z^2(R_2) \quad (2.45)$$

Substitution and simplification yields the following equation,

$$v^2(R_2) = R_2^2 \dot{\alpha}_h^2 \sin^2 \alpha_v + h^2 \dot{\alpha}_h^2 + R_2^2 \dot{\alpha}_v^2 - 2R_2 h \dot{\alpha}_h \dot{\alpha}_v \cos \alpha_v \quad (2.46)$$

2.5.1.3 Pivoted Beam

For more accuracy the point P_3 has been considered with the coordinate

$[r_x(R_3), r_y(R_3), r_z(R_3)]$ on the pivoted beam where R_3 is the distance between P_3 and O .

$$\begin{cases} r_x(R_3) = R_3 \cos \alpha_h \\ r_y(R_3) = -R_3 \sin \alpha_h \\ r_z(R_3) = 0 \end{cases} \quad (2.47)$$

The corresponding velocities and square magnitude of the velocity of P_3 can be written as,

$$\begin{cases} v_x(R_3) = -R_3 \dot{\alpha}_h \sin \alpha_h \\ v_y(R_3) = -R_3 \dot{\alpha}_h \cos \alpha_h \\ v_z(R_3) = 0 \end{cases} \quad (2.48)$$

$$v^2(R_3) = R_3^2 \dot{\alpha}_h^2 \quad (2.49)$$

2.5.2 Energy Expression

The kinetic and potential energies are shown in the following equations respectively.

$$T = \frac{1}{2} \int v^2(R) dm(R) \quad (2.50)$$

$$V = g \int r_z(R) dm(R) \quad (2.51)$$

Where T is the kinetic energy, V is the potential energy, m is mass and g is the acceleration of gravity.

2.5.2.1 Free Beam Energy

The kinetic and potential energies of the free beam are obtained and presented in equations (2.52) and (2.53) respectively.

$$T_1 = \frac{1}{2} (\dot{\alpha}_h^2 \cos^2 \alpha_v + \dot{\alpha}_v^2) J_1 + \frac{1}{2} h^2 \dot{\alpha}_h^2 m_{T1} + h \dot{\alpha}_h \dot{\alpha}_v m_{T1} l_{T1} \sin \alpha_v \quad (2.52)$$

where,

$$J_1 = \int R_1^2 dm(R_1) = \left(\frac{m_t}{3} + m_{tr} + m_{ts} \right) l_t^2 + \left(\frac{m_m}{3} + m_{mr} + m_{ms} \right) l_m^2 + \frac{m_{ms}}{2} r_{ms}^2 + m_{ts} r_{ts}^2$$

$$m_{T1} = \int dm(R_1) = m_t + m_{tr} + m_{ts} + m_m + m_{mr} + m_{ms}$$

$$l_{T1} = \frac{\int R_1 dm(R_1)}{\int dm(R_1)} = \frac{\left(\frac{m_m}{2} + m_{mr} + m_{ms}\right)l_m - \left(\frac{m_t}{2} + m_{tr} + m_{ts}\right)l_t}{m_{T1}}$$

$$V_1 = g m_{T1} l_{T1} \sin \alpha_v \quad (2.53)$$

where,

T_1 : Kinetic energy of the free beam (J)

J_1 : Moment of inertia of the free beam (kg m²)

m_{T1} : Total mass of the free beam (kg)

l_{T1} : Centre of gravity of the free beam (m)

V_1 : Potential energy of the free beam (J)

m_t : Mass of the tail part of the beam (kg)

m_{tr} : Mass of the tail DC motor (kg)

m_{ts} : Mass of the tail shield (kg)

m_m : Mass of the main part of the beam (kg)

m_{mr} : Mass of the main DC motor (kg)

m_{ms} : Mass of the main shield (kg)

l_t : Length of the tail part of the beam (m)

l_m : Length of the main part of the beam (m)

r_{ms} : Radius of the main shield (m)

r_{ts} : Radius of the tail shield (m).

2.5.2.2 Counter Balance Beam Energy

The kinetic and potential energies of the counter balance beam are presented in equations (2.54) and (2.55) respectively.

$$T_2 = \frac{1}{2}(\dot{\alpha}_h^2 \sin^2 \alpha_v + \dot{\alpha}_v^2)J_2 + \frac{1}{2}h^2\dot{\alpha}_h^2 m_{T2} - h \dot{\alpha}_h \dot{\alpha}_v m_{T2} l_{T2} \cos \alpha_v \quad (2.54)$$

where,

$$J_2 = \int R_2^2 dm(R_2) = \frac{m_b}{3}l_b^2 + m_{cb}l_{cb}^2$$

$$m_{T2} = \int dm(R_2) = m_b + m_{cb}$$

$$l_{T2} = \frac{\int R_2 dm(R_2)}{\int dm(R_2)} = \frac{\frac{m_b}{2} l_b + m_{cb} l_{cb}}{m_{T2}}$$

$$V_2 = -g m_{T2} l_{T2} \cos \alpha_v \quad (2.55)$$

where,

T_2 : Kinetic energy of the counter balance beam (J)

V_2 : Potential energy of the counter balance beam (J)

J_2 : Moment of inertia of the counter balance beam (kg m²)

m_b : Mass of the counter-weight beam (kg)

m_{T2} : Total mass of the counter balance beam (kg)

m_{cb} : Mass of the counter-weight (kg)

l_{T2} : Centre of gravity of the counter balance beam (m)

l_b : Length of the counter-weight beam (m)

l_{cb} : Distance between the counter-weight and the joint (m).

2.5.2.3 Pivoted Beam Energy

The kinetic and potential energies of the pivoted beam are respectively shown as,

$$T_3 = \frac{1}{2} \dot{\alpha}_h^2 J_3 \quad (2.56)$$

$$J_3 = \int R_3^2 dm(R_3) = \frac{m_h}{3} h^2$$

$$V_3 = 0 \quad (2.57)$$

where,

T_3 : Kinetic energy of the pivoted beam (J)

J_3 : Moment of inertia of the pivoted beam (kg m²)

V_3 : Potential energy of the pivoted beam (J)

m_h : Mass of the pivoted beam (kg)

h : Length of the pivoted beam (m).

2.5.3 Lagrange's Equations

Lagrange's equation defines L as,

$$L = \sum_{i=1}^3 T_i - \sum_{i=1}^3 V_i \quad (2.58)$$

and the equations of motion are given by,

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\alpha}_h} \right) - \frac{\partial L}{\partial \alpha_h} = \sum_i M_{ih} \quad (2.59)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\alpha}_v} \right) - \frac{\partial L}{\partial \alpha_v} = \sum_i M_{iv} \quad (2.60)$$

Therefore, by substituting equations (2.52) to (2.57) into equations (2.58) to (2.60) and simplifying, the governing equations of the system are obtained as,

$$\begin{aligned} & [J_1 \cos^2 \alpha_v + J_2 \sin^2 \alpha_v + h^2(m_{T1} + m_{T2}) + J_3] \ddot{\alpha}_h + h[m_{T1}l_{T1} \sin \alpha_v - m_{T2}l_{T2} \cos \alpha_v] \ddot{\alpha}_v \\ & + h[m_{T1}l_{T1} \cos \alpha_v + m_{T2}l_{T2} \sin \alpha_v] \dot{\alpha}_v^2 + 2\dot{\alpha}_h \dot{\alpha}_v [J_2 - J_1] \sin \alpha_v \cos \alpha_v = \sum_i M_{ih} \end{aligned} \quad (2.61)$$

$$\begin{aligned} & [J_1 + J_2] \ddot{\alpha}_v + h[m_{T1}l_{T1} \sin \alpha_v - m_{T2}l_{T2} \cos \alpha_v] \ddot{\alpha}_h \\ & + \dot{\alpha}_h^2 [J_1 - J_2] \sin \alpha_v \cos \alpha_v + g[m_{T1}l_{T1} \cos \alpha_v + m_{T2}l_{T2} \sin \alpha_v] = \sum_i M_{iv} \end{aligned} \quad (2.62)$$

where, $\sum_i M_{ih}$ is the sum of the applied torques in the horizontal movement, and $\sum_i M_{iv}$ is the sum of the applied torques in the vertical movement. Equations (2.61) and (2.62) can be expressed in a matrix form as,

$$\begin{aligned} & \begin{bmatrix} J_1 \cos^2 \alpha_v + J_2 \sin^2 \alpha_v + h^2(m_{T1} + m_{T2}) + J_3 & h(m_{T1}l_{T1} \sin \alpha_v - m_{T2}l_{T2} \cos \alpha_v) \\ h(m_{T1}l_{T1} \sin \alpha_v - m_{T2}l_{T2} \cos \alpha_v) & J_1 + J_2 \end{bmatrix} \begin{bmatrix} \ddot{\alpha}_h \\ \ddot{\alpha}_v \end{bmatrix} \\ & + \begin{bmatrix} h(m_{T1}l_{T1} \cos \alpha_v + m_{T2}l_{T2} \sin \alpha_v) \dot{\alpha}_v^2 + 2\dot{\alpha}_h \dot{\alpha}_v (J_2 - J_1) \sin \alpha_v \cos \alpha_v \\ \dot{\alpha}_h^2 (J_1 - J_2) \sin \alpha_v \cos \alpha_v + g(m_{T1}l_{T1} \cos \alpha_v + m_{T2}l_{T2} \sin \alpha_v) \end{bmatrix} = \begin{bmatrix} \sum_i M_{ih} \\ \sum_i M_{iv} \end{bmatrix} \end{aligned} \quad (2.63)$$

Considering equation (2.61), the 1DOF horizontal TRMS model can be obtained under the conditions of constant α_v which implies $\dot{\alpha}_v = \ddot{\alpha}_v = 0$.

$$[J_1 \cos^2 \alpha_v + J_2 \sin^2 \alpha_v + h^2(m_{T1} + m_{T2}) + J_3] \ddot{\alpha}_h = \sum_i M_{ih} \quad (2.64)$$

As it can be seen, the Lagrangian model of the 1DOF horizontal TRMS is very similar to the Newtonian one; however, the assumptions behind these two modelling

approaches are not exactly the same. This difference originates from the fact that the pivoted beam has been taken into consideration in the Lagrangian-based approach but it has been neglected in the Newtonian-based method; because in the case of the Newtonian-based model the consideration of the pivoted beam renders a complexity in modelling but, it is really easy to consider it in the Lagrangian approach.

The 1DOF vertical TRMS model can be expressed as follow, by considering the conditions $\alpha_h = 0 \Rightarrow \dot{\alpha}_h = \ddot{\alpha}_h = 0$ in equation (2.62).

$$[J_1 + J_2] \ddot{\alpha}_v + g [m_{T1} l_{T1} \cos \alpha_v + m_{T2} l_{T2} \sin \alpha_v] = \sum_i M_{iv} \quad (2.65)$$

It is evident that the Lagrangian model of the 1DOF vertical TRMS is exactly the same as the Newtonian one; however, the assumptions behind the two methods are different. Therefore, for the sake of brevity, only the responses of the Lagrangian model in terms of the 2DOF TRMS have been presented here. The 1DOF Lagrangian-based dynamic models of the TRMS are the same as the Newtonian models in terms of both the horizontal and vertical channels.

According to equation (2.63), the angle between the tower and the counterweight beam in the equilibrium point can be calculated as,

$$[m_{T1} l_{T1} \cos \alpha_{v0} + m_{T2} l_{T2} \sin \alpha_{v0}] = 0 \Rightarrow \alpha_{v0} = \tan^{-1} \left(\frac{-m_{T1} l_{T1}}{m_{T2} l_{T2}} \right) \quad (2.66)$$

2.5.4 Implementation of the Lagrange-based Model

According to equation (2.63), it can be rewritten that,

$$\begin{bmatrix} \ddot{\alpha}_h \\ \ddot{\alpha}_v \end{bmatrix} = \begin{bmatrix} J_1 \cos^2 \alpha_v + J_2 \sin^2 \alpha_v + h^2(m_{T1} + m_{T2}) + J_3 & h(m_{T1} l_{T1} \sin \alpha_v - m_{T2} l_{T2} \cos \alpha_v) \\ h(m_{T1} l_{T1} \sin \alpha_v - m_{T2} l_{T2} \cos \alpha_v) & J_1 + J_2 \end{bmatrix}^{-1} \left\{ \begin{bmatrix} \sum_i M_{ih} \\ \sum_i M_{iv} \end{bmatrix} - \begin{bmatrix} h(m_{T1} l_{T1} \cos \alpha_v + m_{T2} l_{T2} \sin \alpha_v) \dot{\alpha}_v^2 + 2\dot{\alpha}_h \dot{\alpha}_v (J_2 - J_1) \sin \alpha_v \cos \alpha_v \\ \dot{\alpha}_h^2 (J_1 - J_2) \sin \alpha_v \cos \alpha_v + g(m_{T1} l_{T1} \cos \alpha_v + m_{T2} l_{T2} \sin \alpha_v) \end{bmatrix} \right\} \quad (2.67)$$

where, $\sum_i M_{ih}$ is the sum of the applied torques in the horizontal movement, and can be summarised as follows,

$$\sum_i M_{ih} = M_{prop,h} - M_{fric,h} - M_{cable} + k_m \dot{\alpha}_v \cos \alpha_v \quad (2.68)$$

$$M_{prop,h} = l_t F_h(\omega_h) \cos \alpha_v$$

where,

$M_{prop,h}$: Propulsive torque due to the tail propeller rotation (Nm)

$M_{fric,h}$: Friction torque of the horizontal movement (Nm)

M_{cable} : Flat cable torque (Nm)

$\dot{\omega}_v$: Acceleration of the main rotor (rad/s²).

Again, in equation (2.67) $\sum_i M_{iv}$ is the sum of the applied torques in the vertical movement, and can be expressed in terms of the following equation,

$$\sum_i M_{iv} = M_{prop,v} - M_{fric,v} + k_t \dot{\omega}_h + M_{gyro} \quad (2.69)$$

$$M_{prop,v} = l_m F_v(\omega_v)$$

$$M_{gyro} = k_g F_v(\omega_v) \Omega_h \cos \alpha_v$$

where,

$M_{prop,v}$: Propulsive torque due to the main propeller rotation (Nm)

$M_{fric,v}$: Friction torque of the vertical movement (Nm)

M_{gyro} : Gyroscopic torque (Nm)

$\dot{\omega}_h$: Acceleration of the tail rotor (rad/s²).

The governing equations of two DC motors and their propulsive forces are exactly the same as in the case of the Newtonian method. Figure 2.63 shows the complete block diagram of the Lagrangian-based TRMS model and Figure 2.64 depicts the inside of the nonlinear Lagrangian equations block. The overall Lagrangian-based model is a nonlinear system of the order of 8 since, each motor model contains two integrators and the nonlinear Lagrangian equations subsystem has four integrators. In order to model the effects of both the channels on each other, one needs to have access to the acceleration of tail and main rotors. It is clear from Figure 2.63 that these accelerations come from the motors blocks directly to prevent any derivative which may render instability of the model in the face of noisy signals.

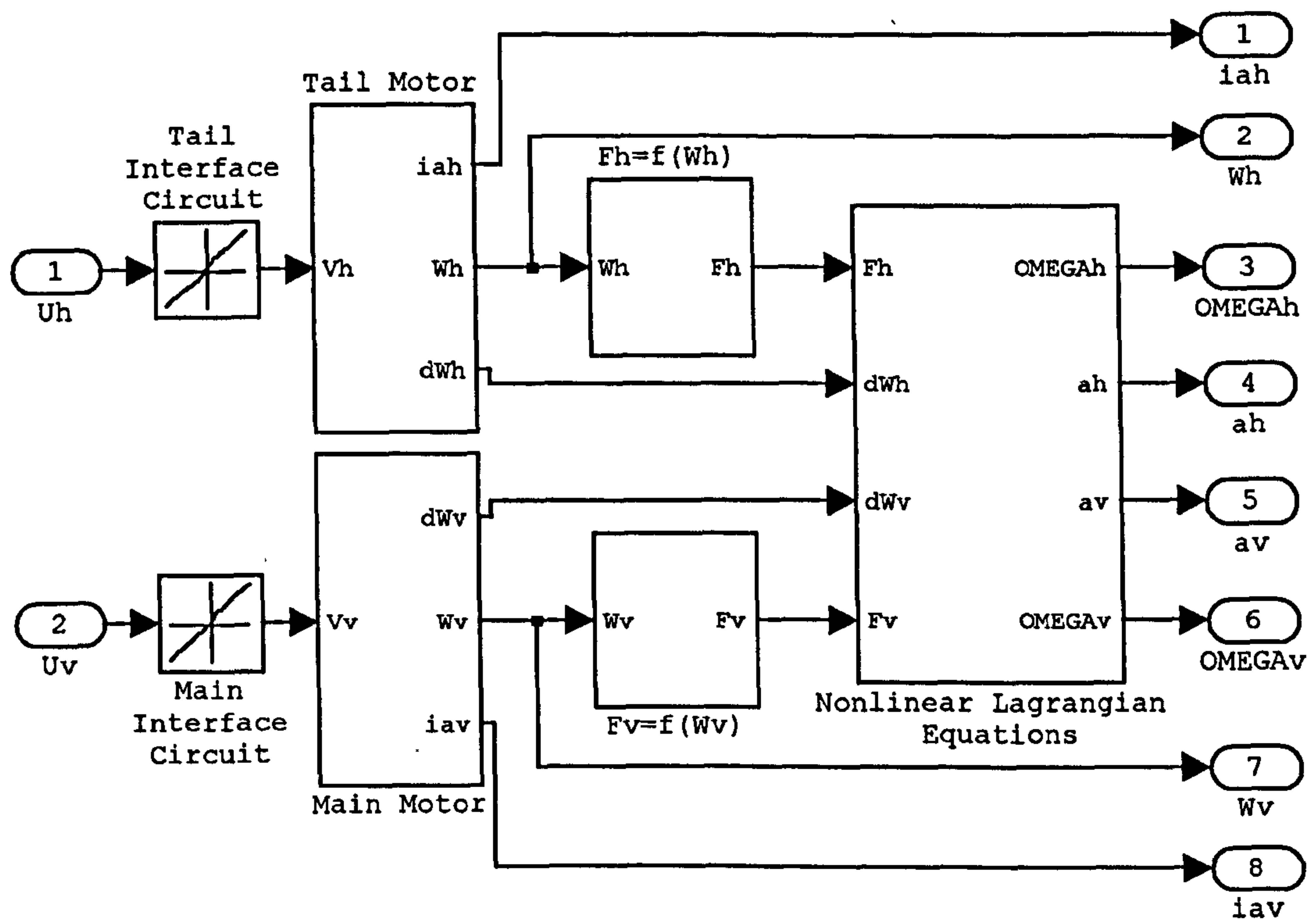


Figure 2.63 The complete block diagram of the Lagrangian-based TRMS model

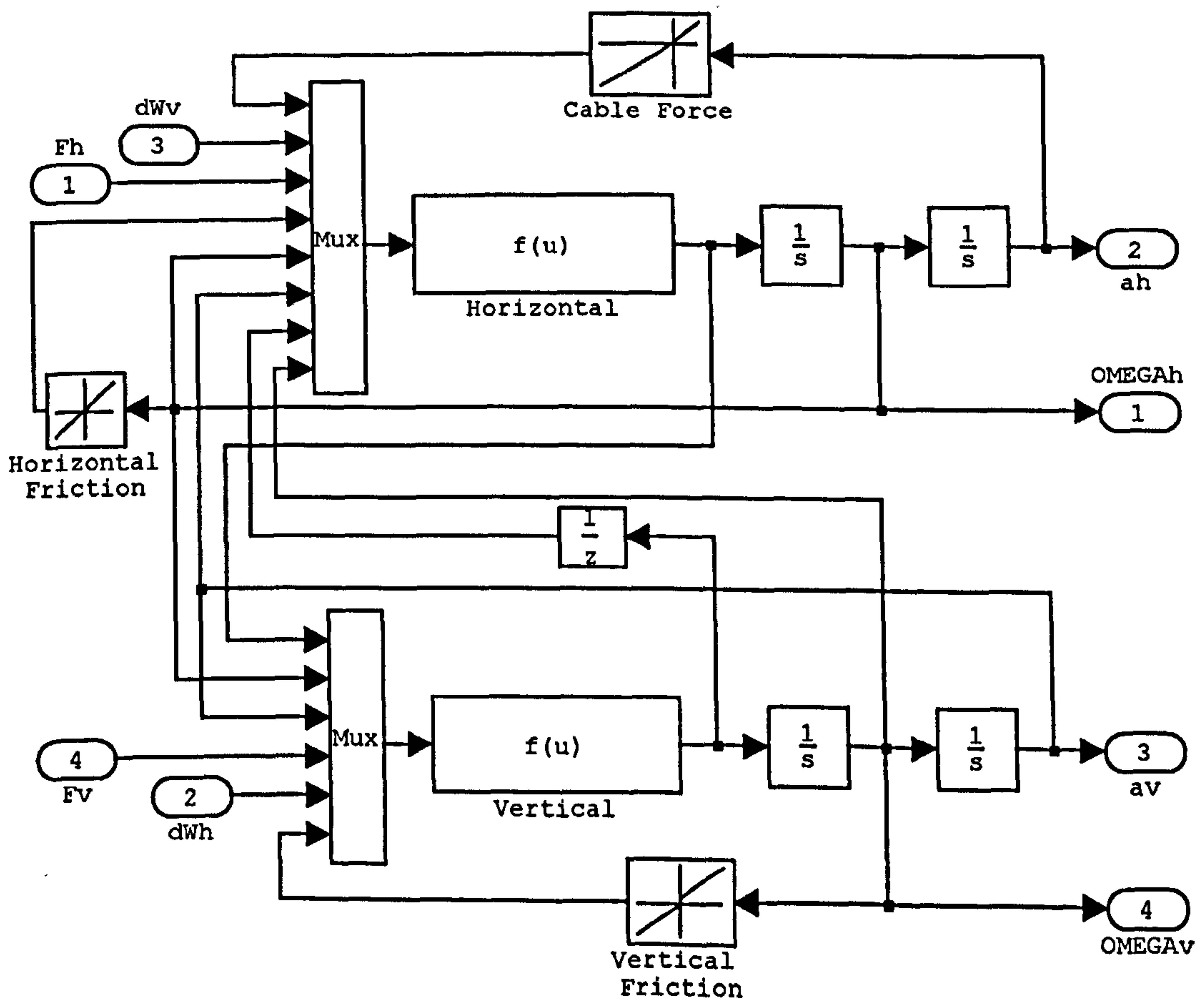


Figure 2.64 The inside of the nonlinear Lagrangian equations block

2.6 Results of the 2DOF TRMS Model (Lagrangian-based)

The responses of the developed 2DOF Lagrangian-based model and their corresponding PSD tests, with the input signals presented in Table 2.3, are shown in Figures 2.65 to 2.82. Again, it can be perceived from the figures that the model responses closely match the real TRMS responses. In order to have a logical comparison between the Newtonian and Lagrangian approaches, the models have been evaluated using an identical set of input signals. Figures 2.65 and 2.66 respectively show the pitch and yaw angles responses, when the main and tail motors are excited with two sine waves. The frequency and the amplitude of the main channel are 0.5 Hz and 1.5 V, and the tail channel has the same frequency but different amplitude of 1 V. Similarly, the PSD test is used to validate the model in the frequency domain besides the overlapping assessment utilised to show the performance of the model in the time domain.

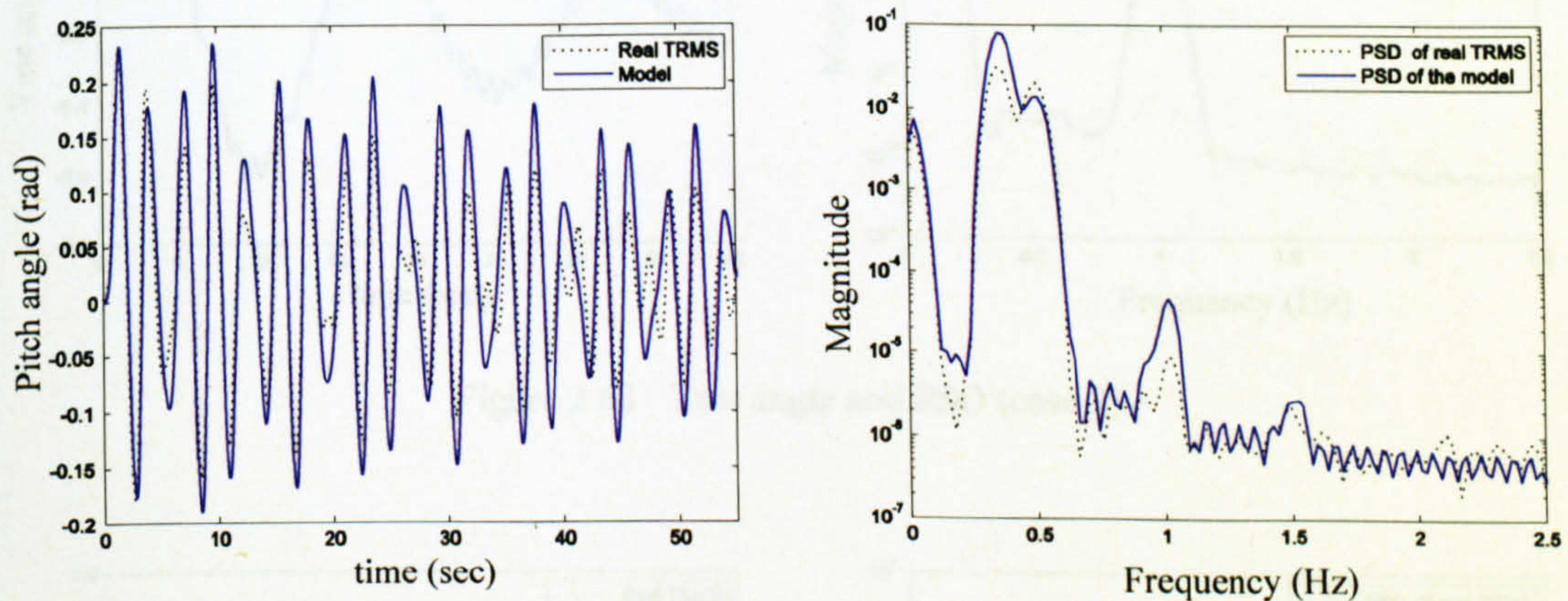


Figure 2.65 Pitch angle and PSD (case 1)

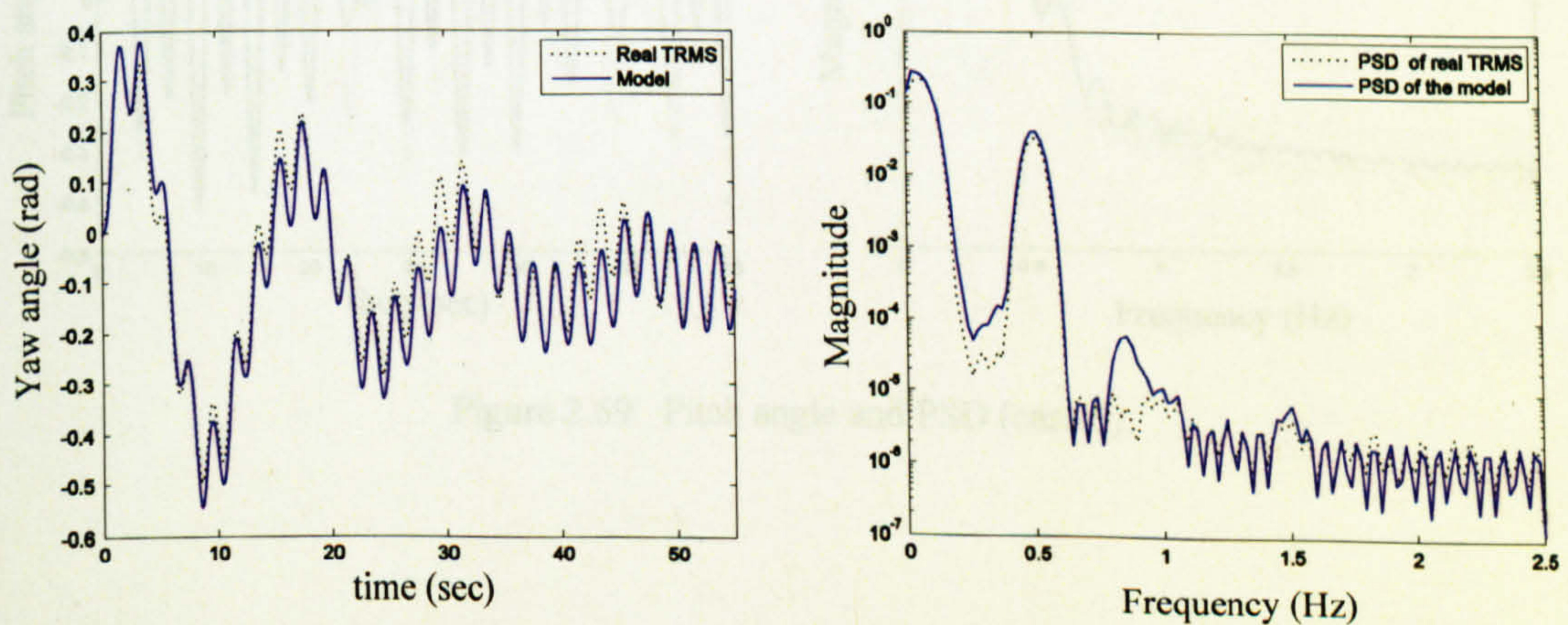


Figure 2.66 Yaw angle and PSD (case 1)

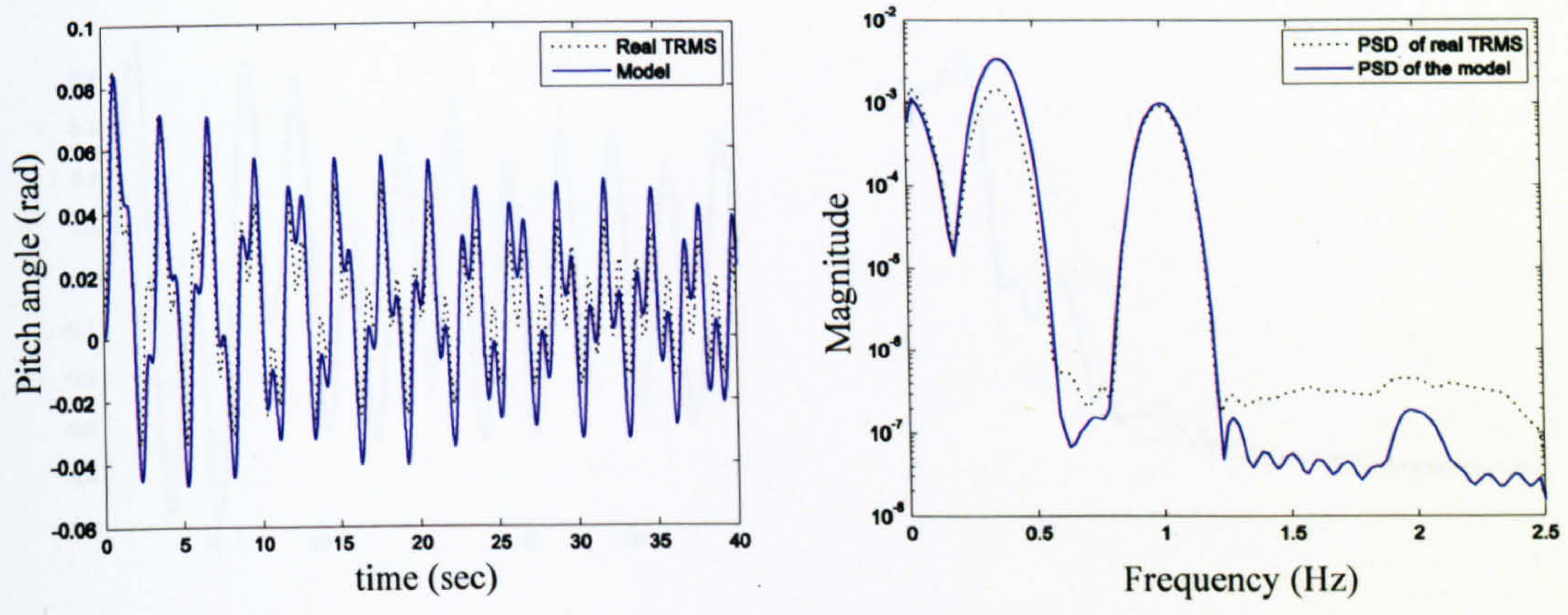


Figure 2.67 Pitch angle and PSD (case 2)

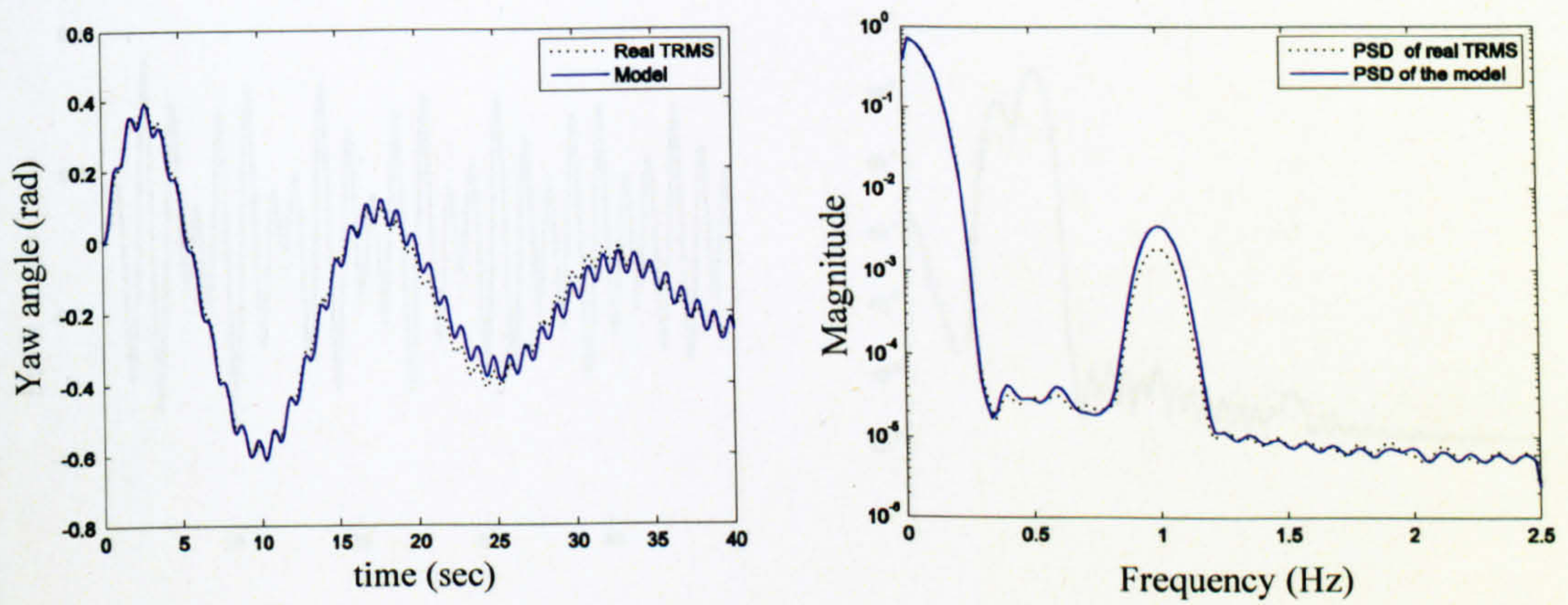


Figure 2.68 Yaw angle and PSD (case 2)

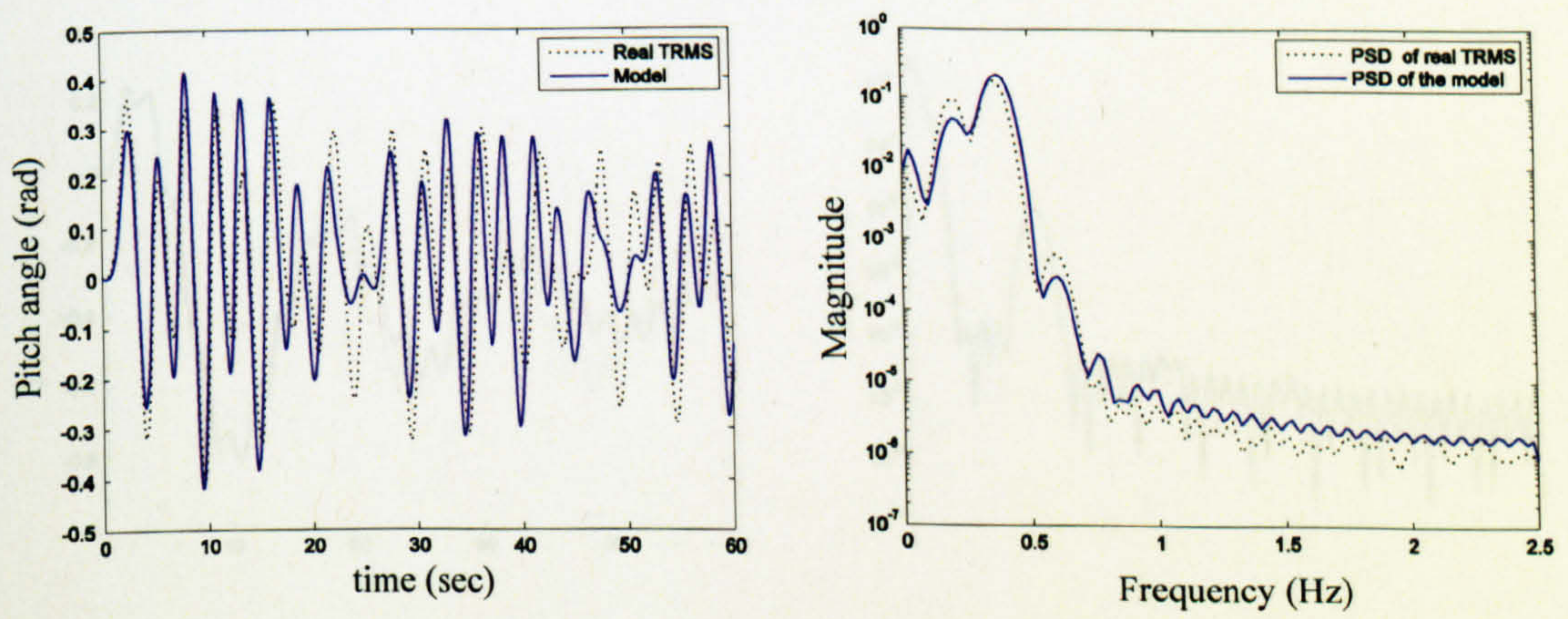


Figure 2.69 Pitch angle and PSD (case 3)

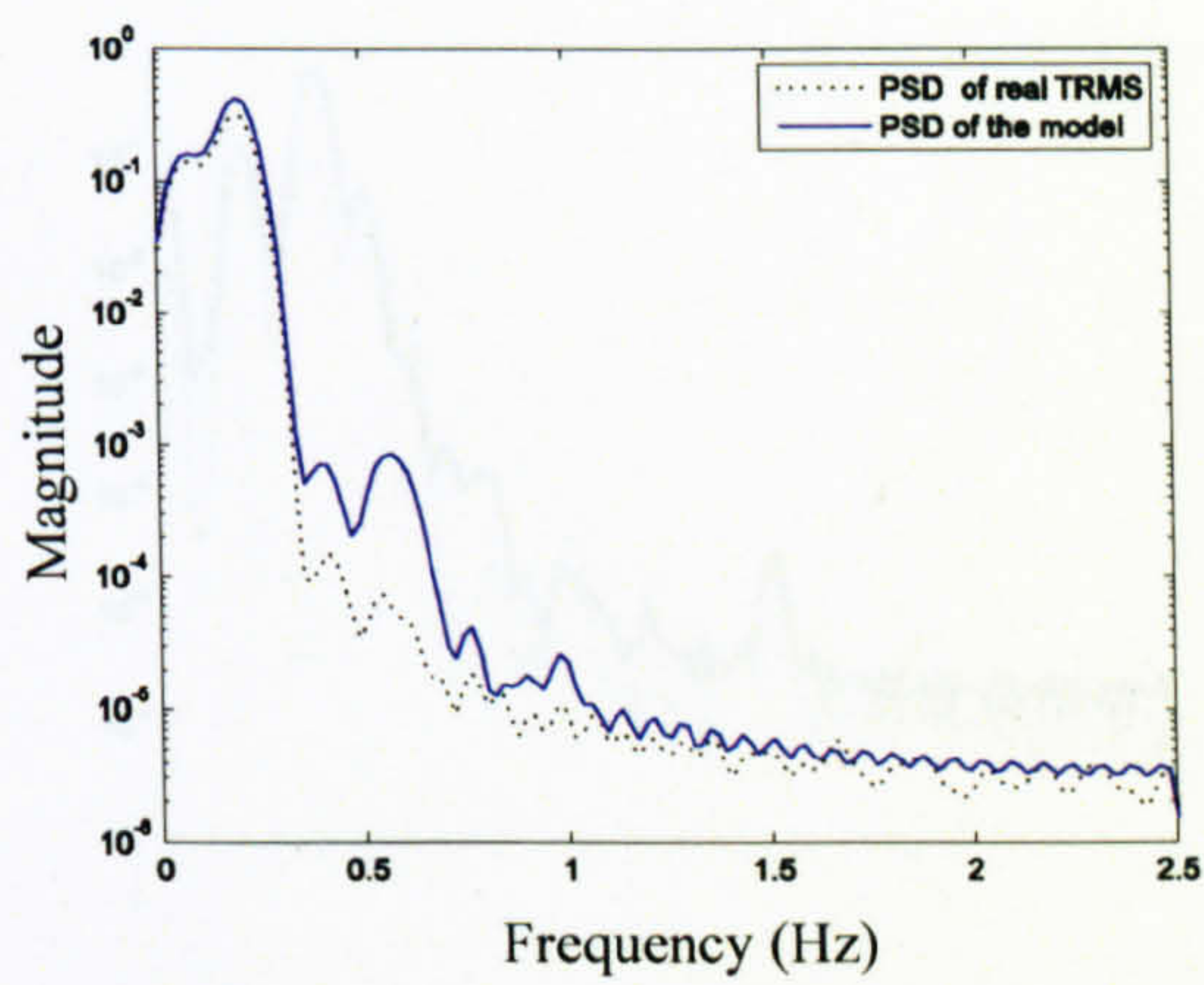
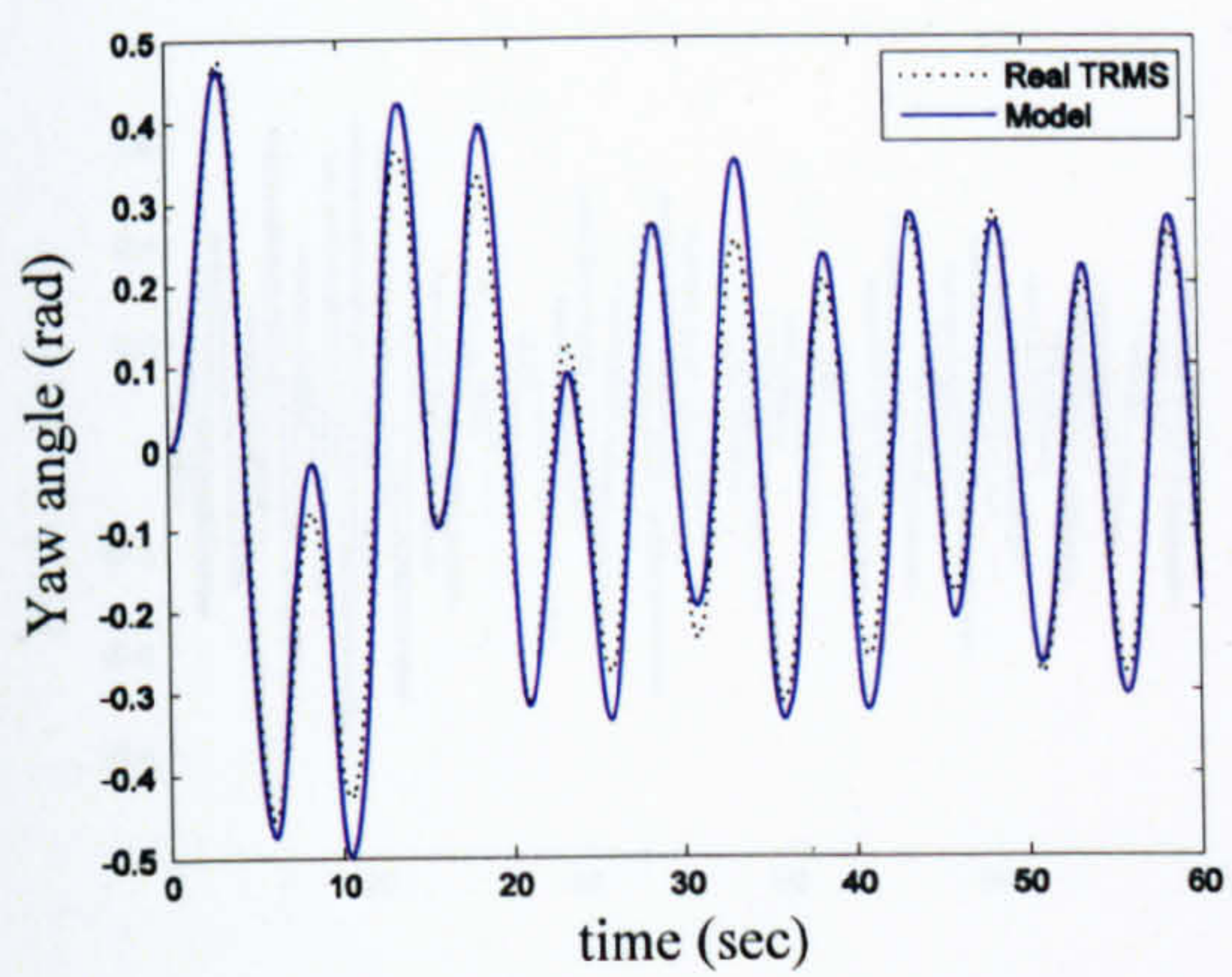


Figure 2.70 Yaw angle and PSD (case 3)

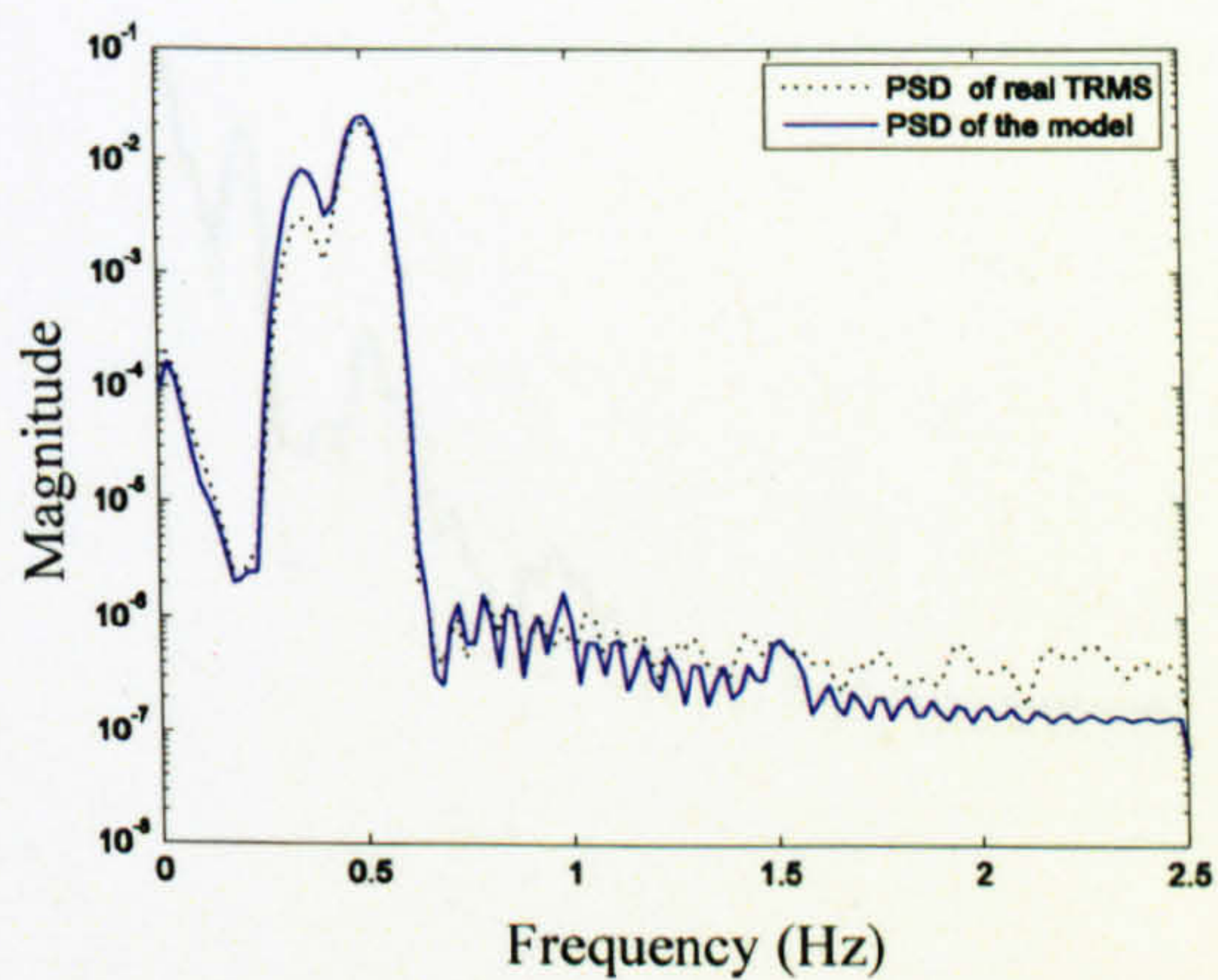
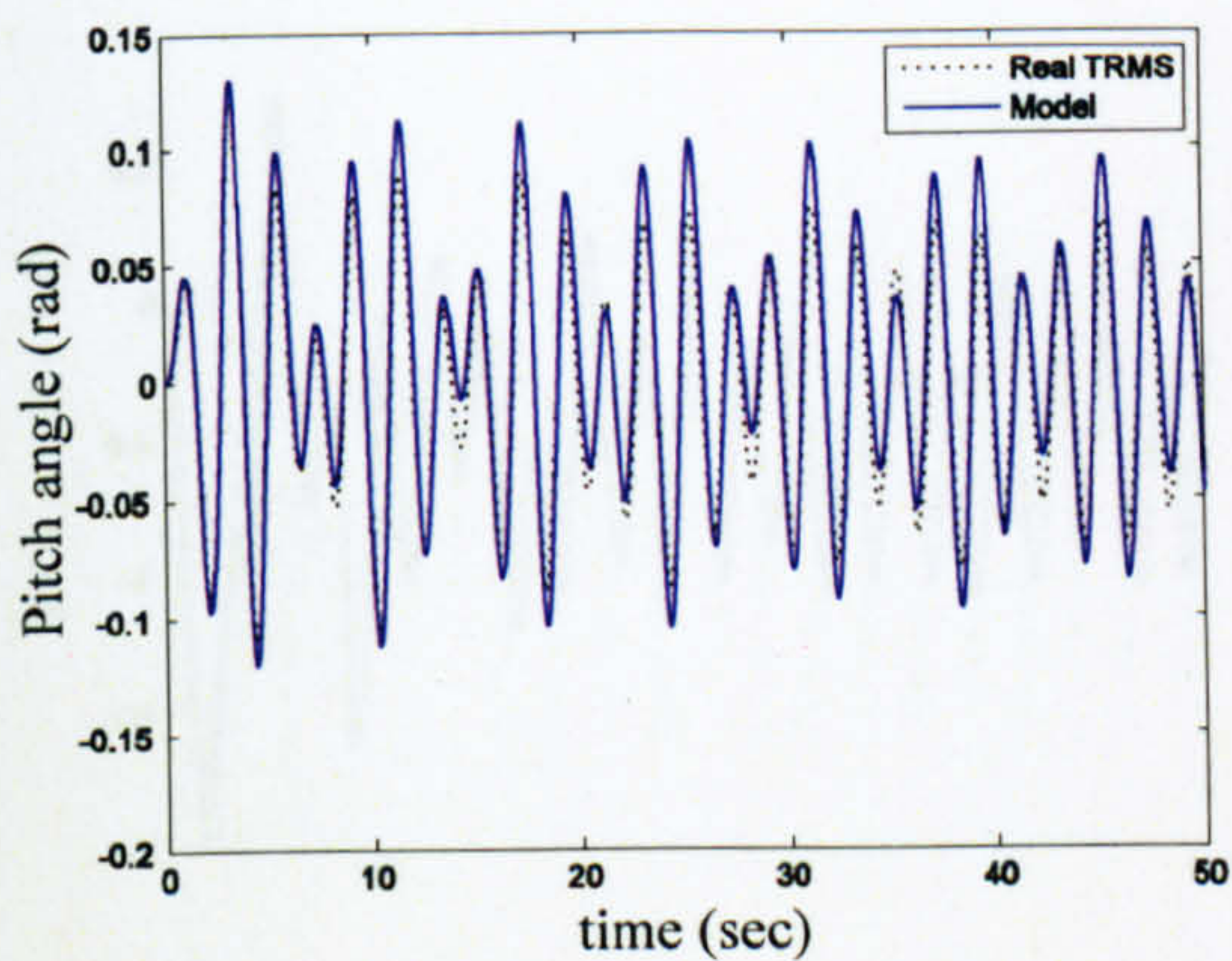


Figure 2.71 Pitch angle and PSD (case 4)

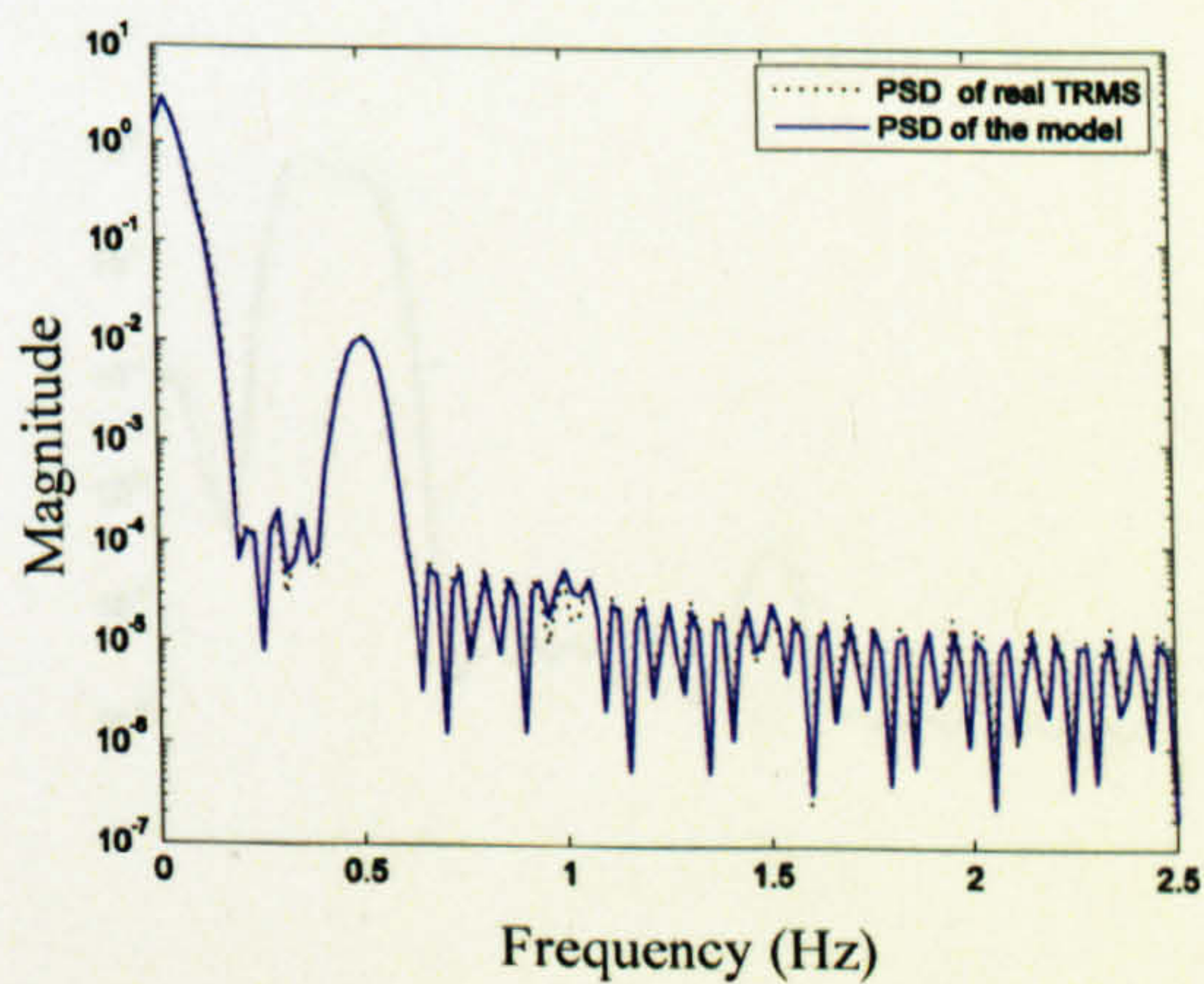
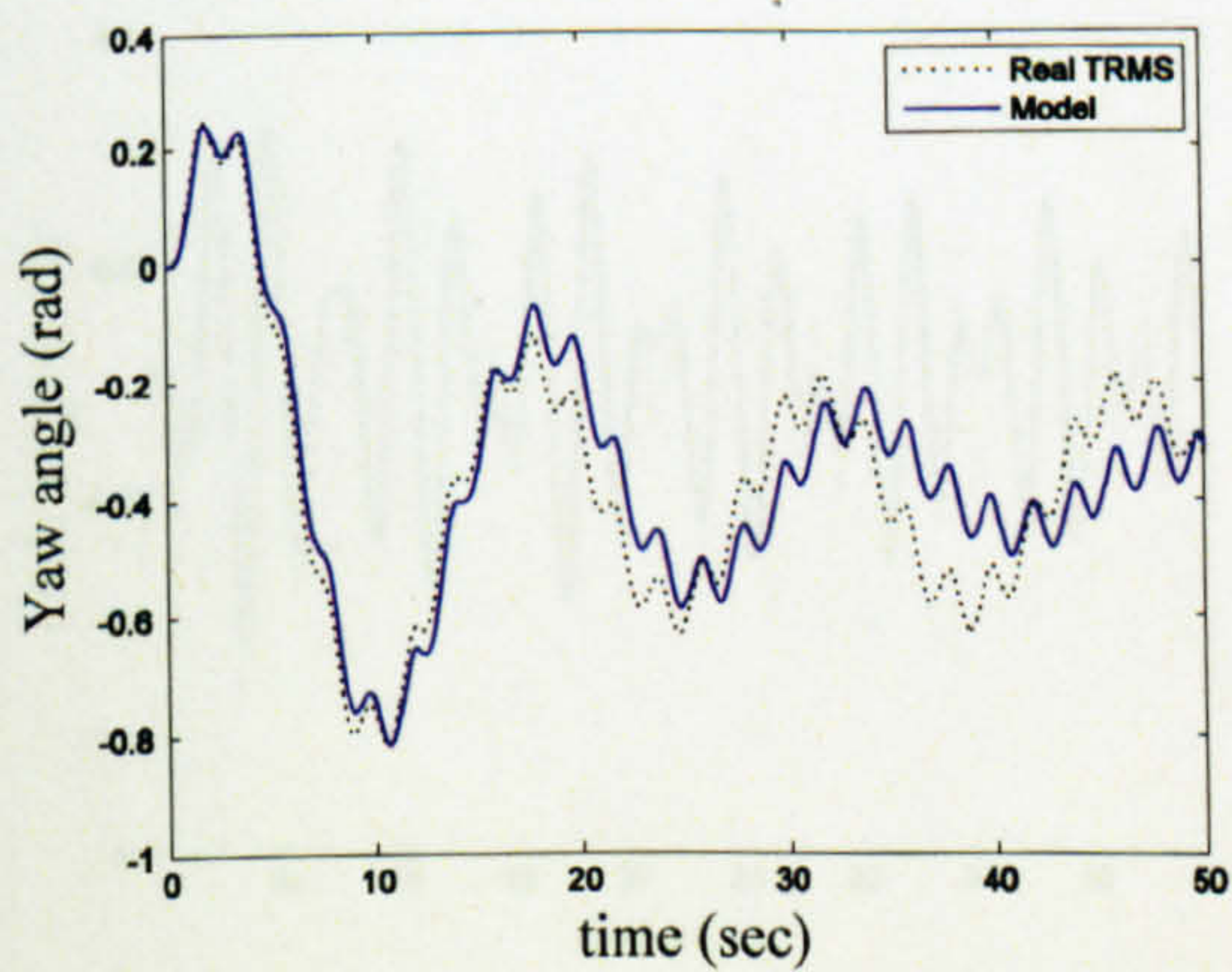


Figure 2.72 Yaw angle and PSD (case 4)

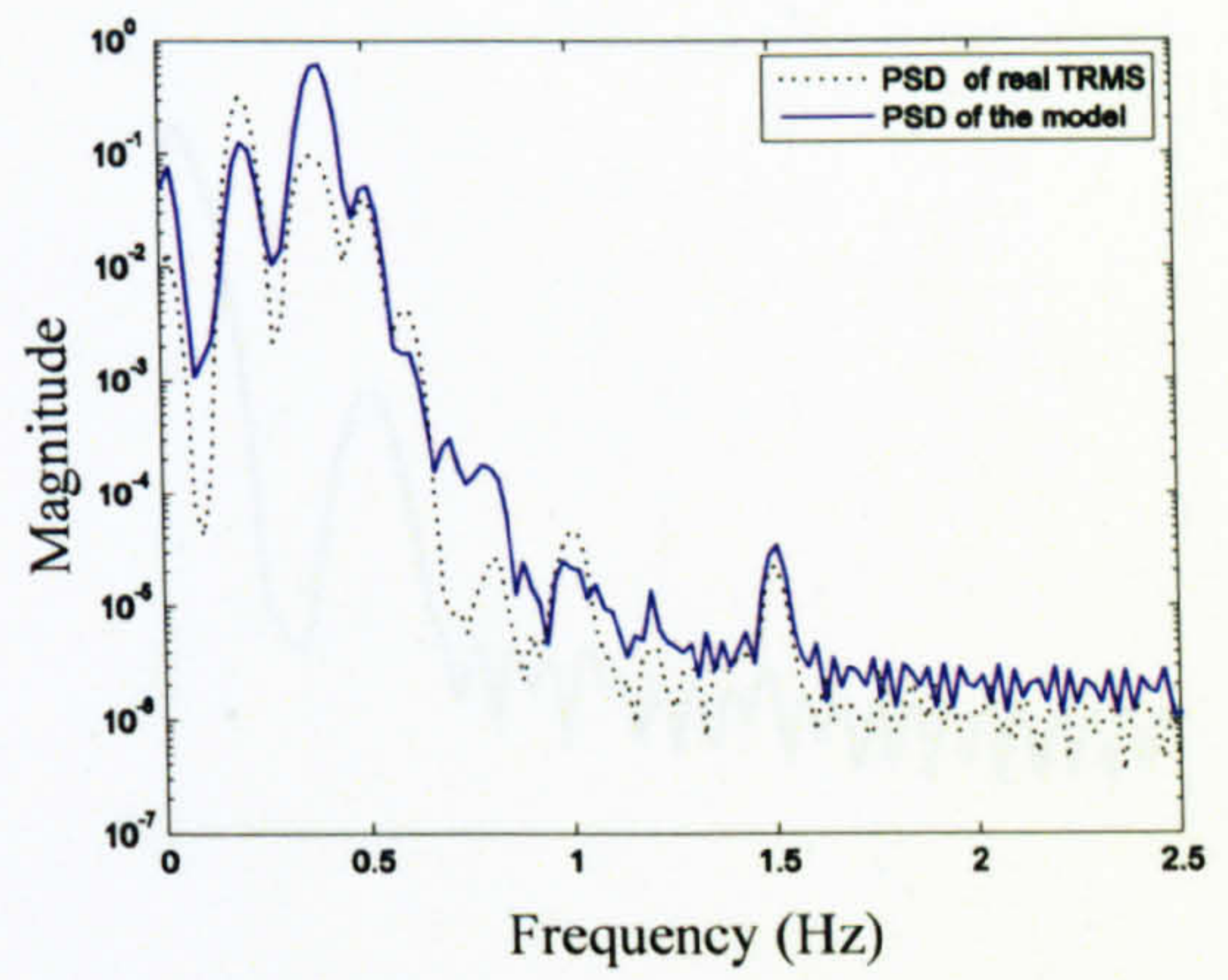
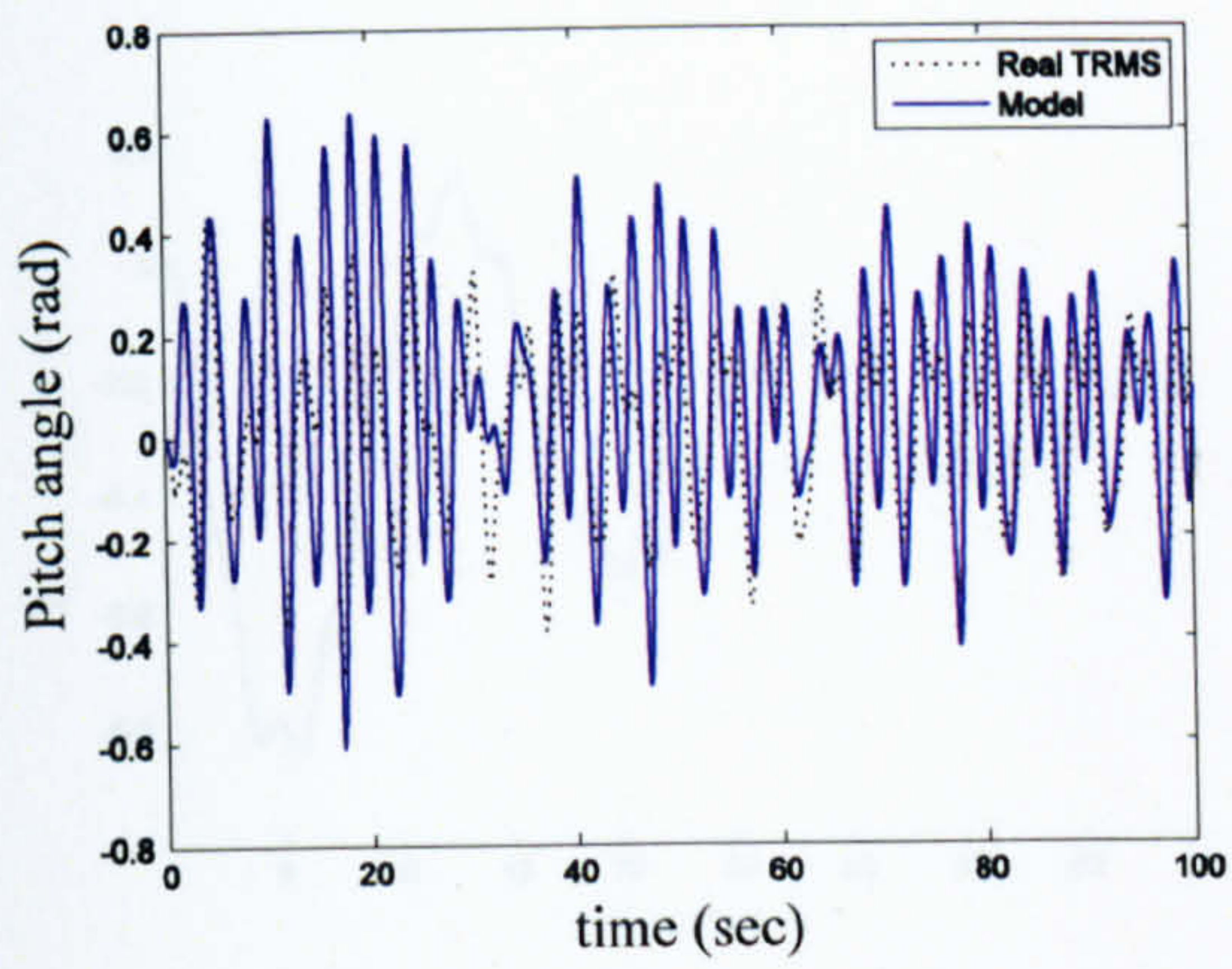


Figure 2.73 Pitch angle and PSD (case 5)

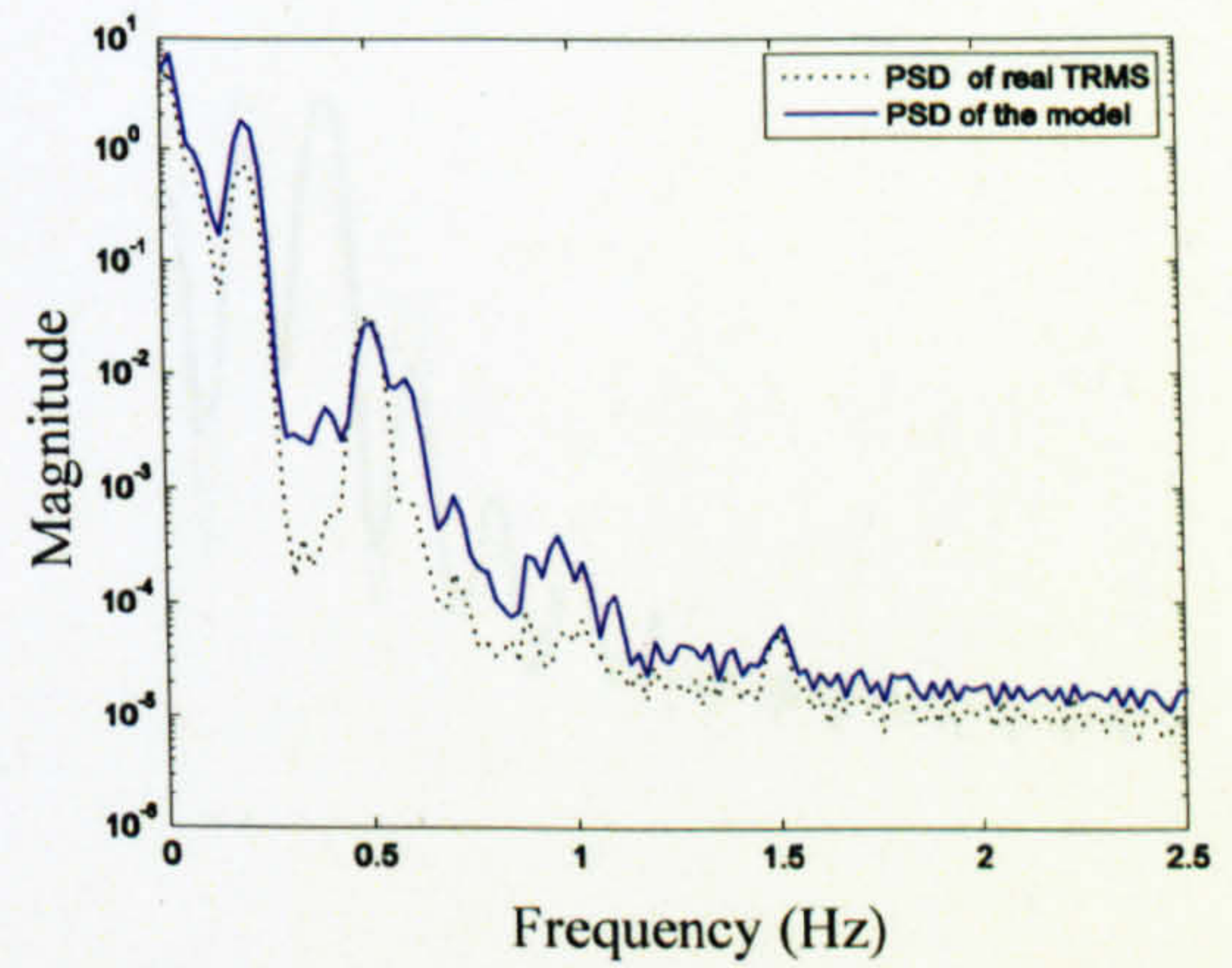
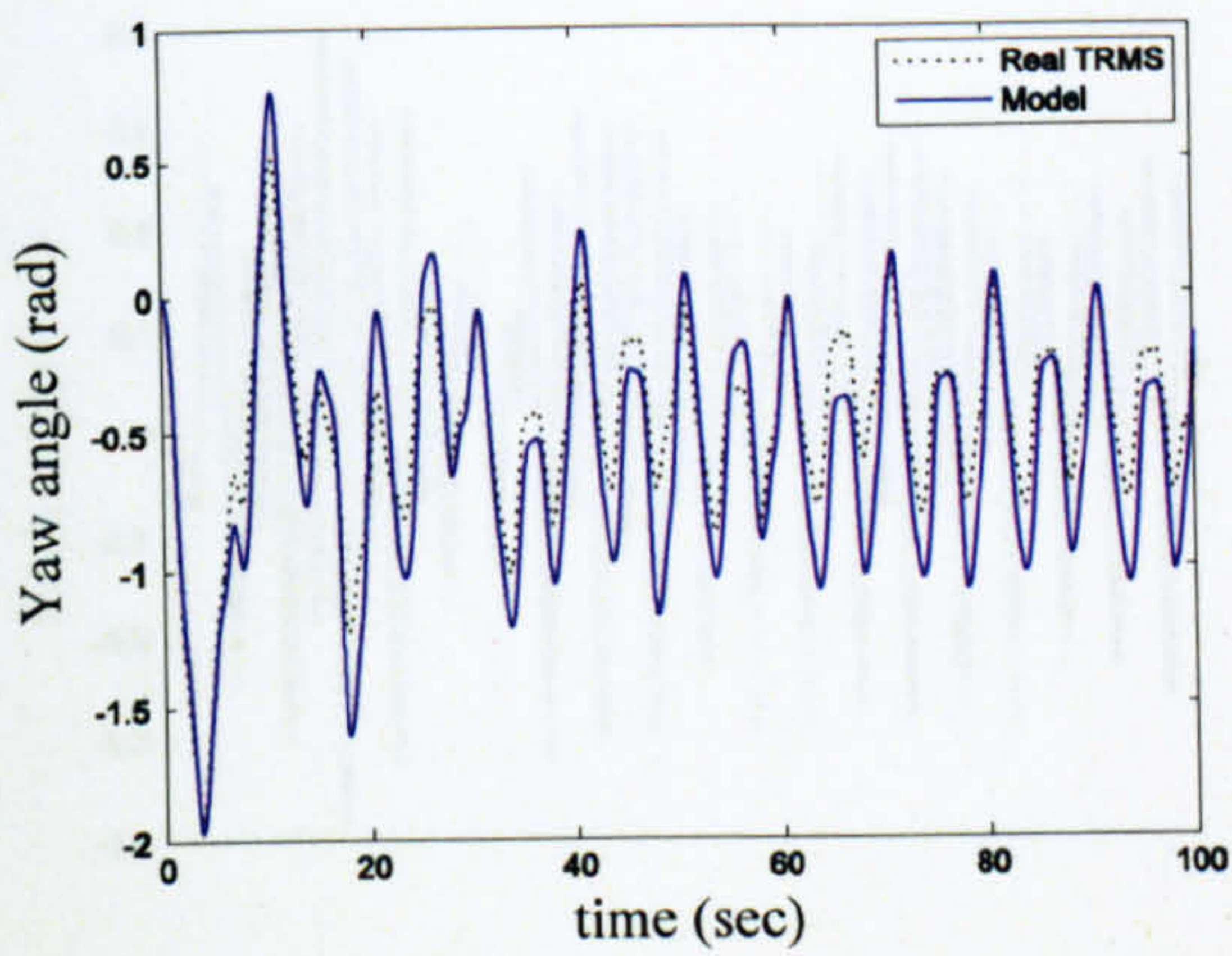


Figure 2.74 Yaw angle and PSD (case 5)

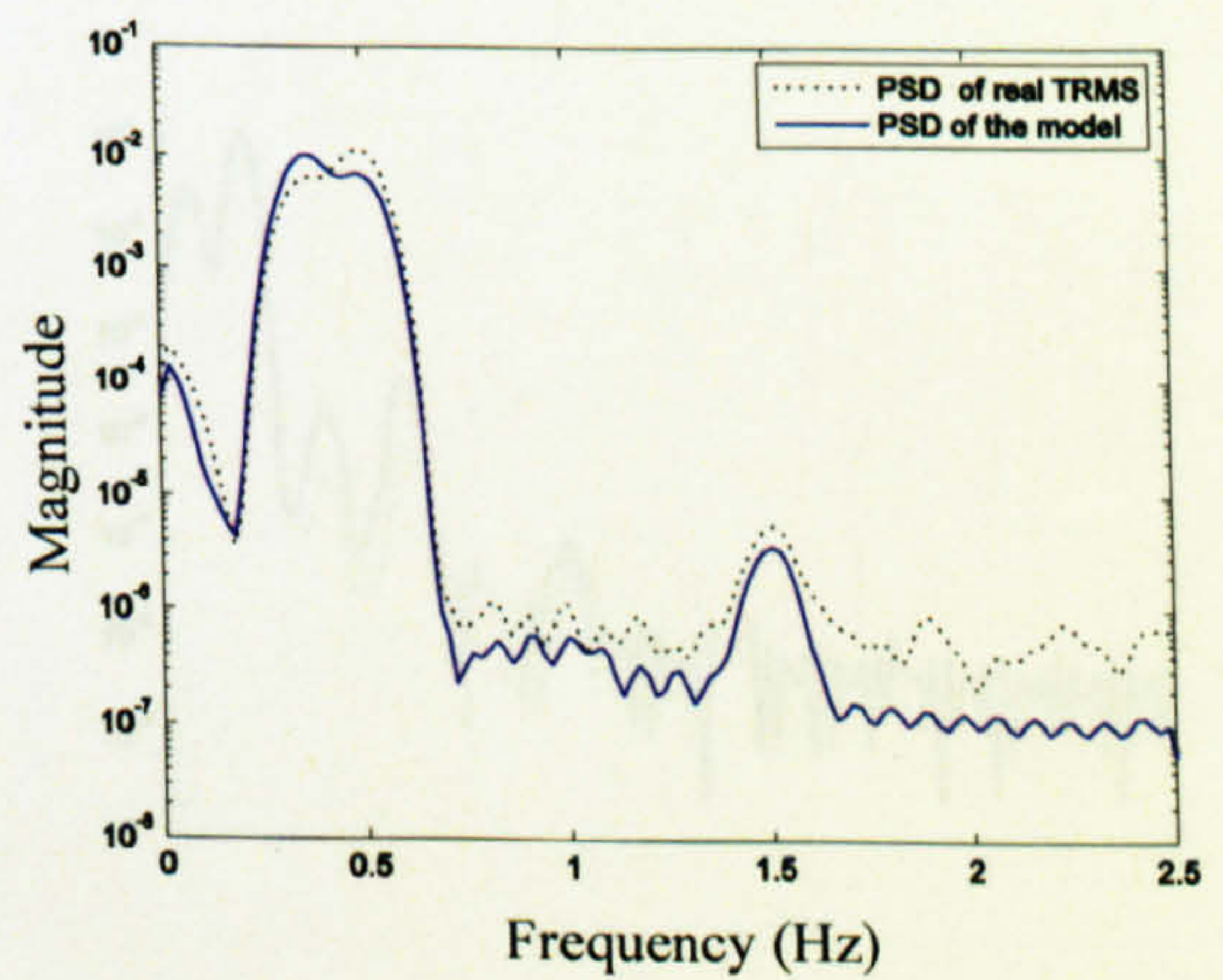
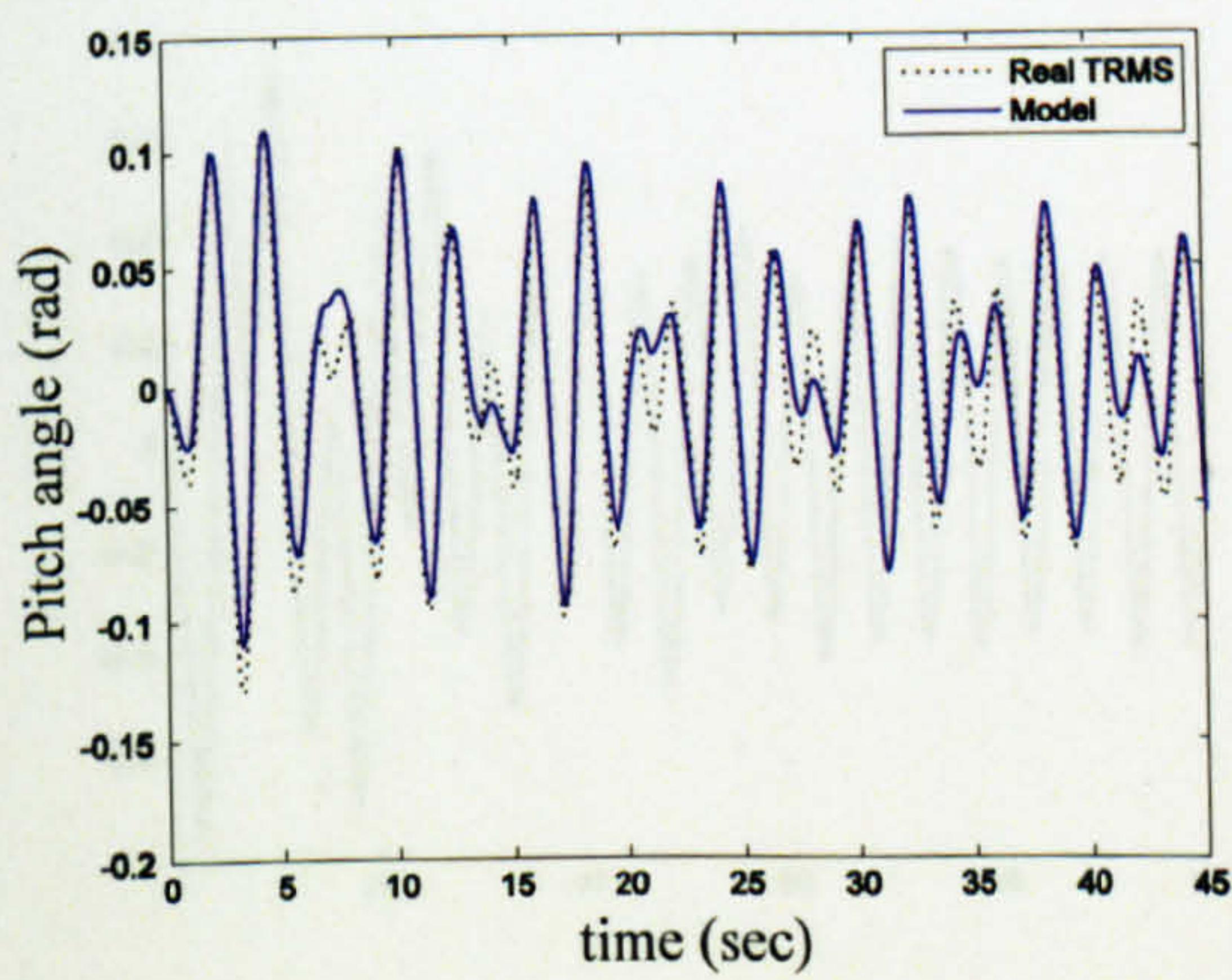


Figure 2.75 Pitch angle and PSD (case 6)

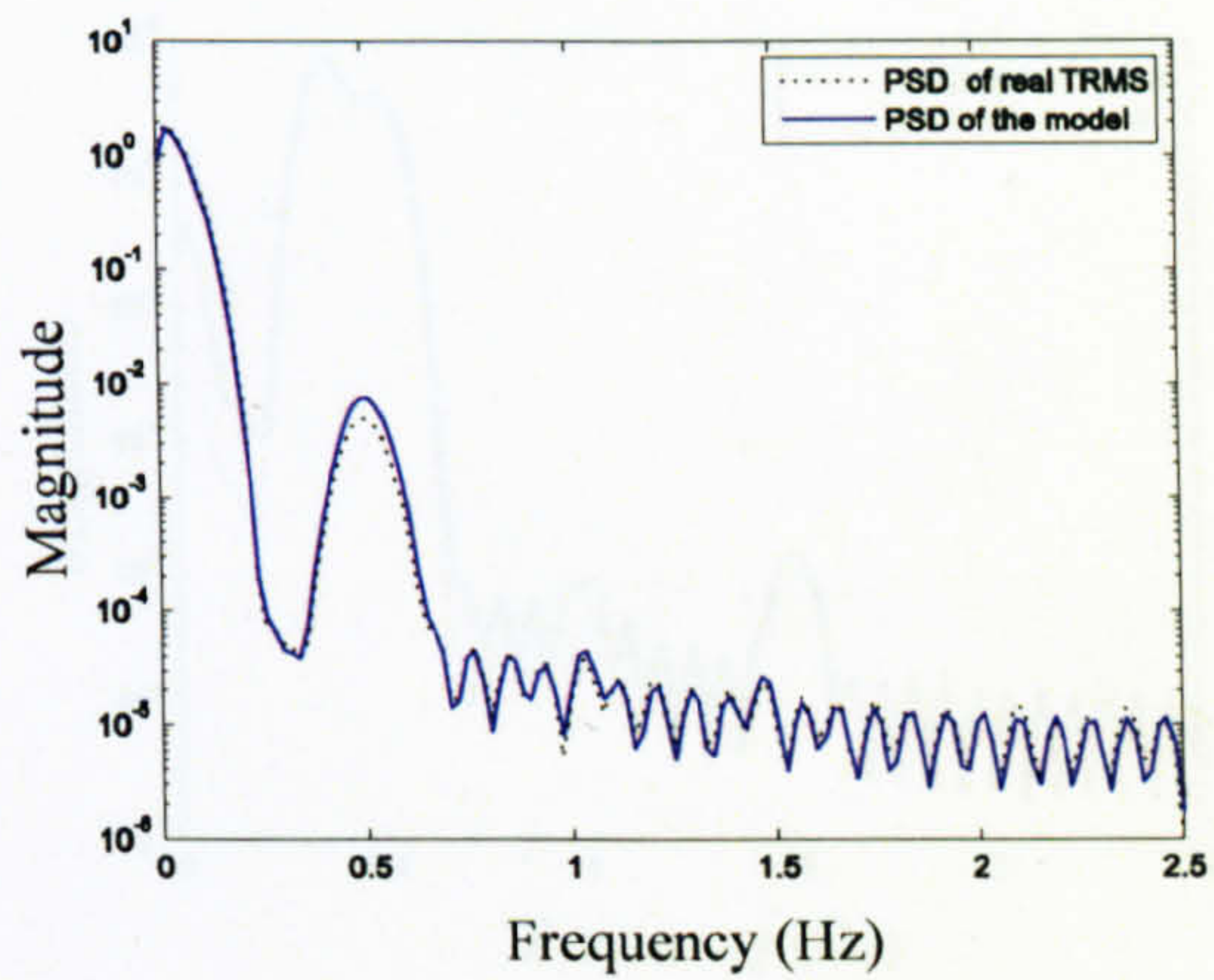
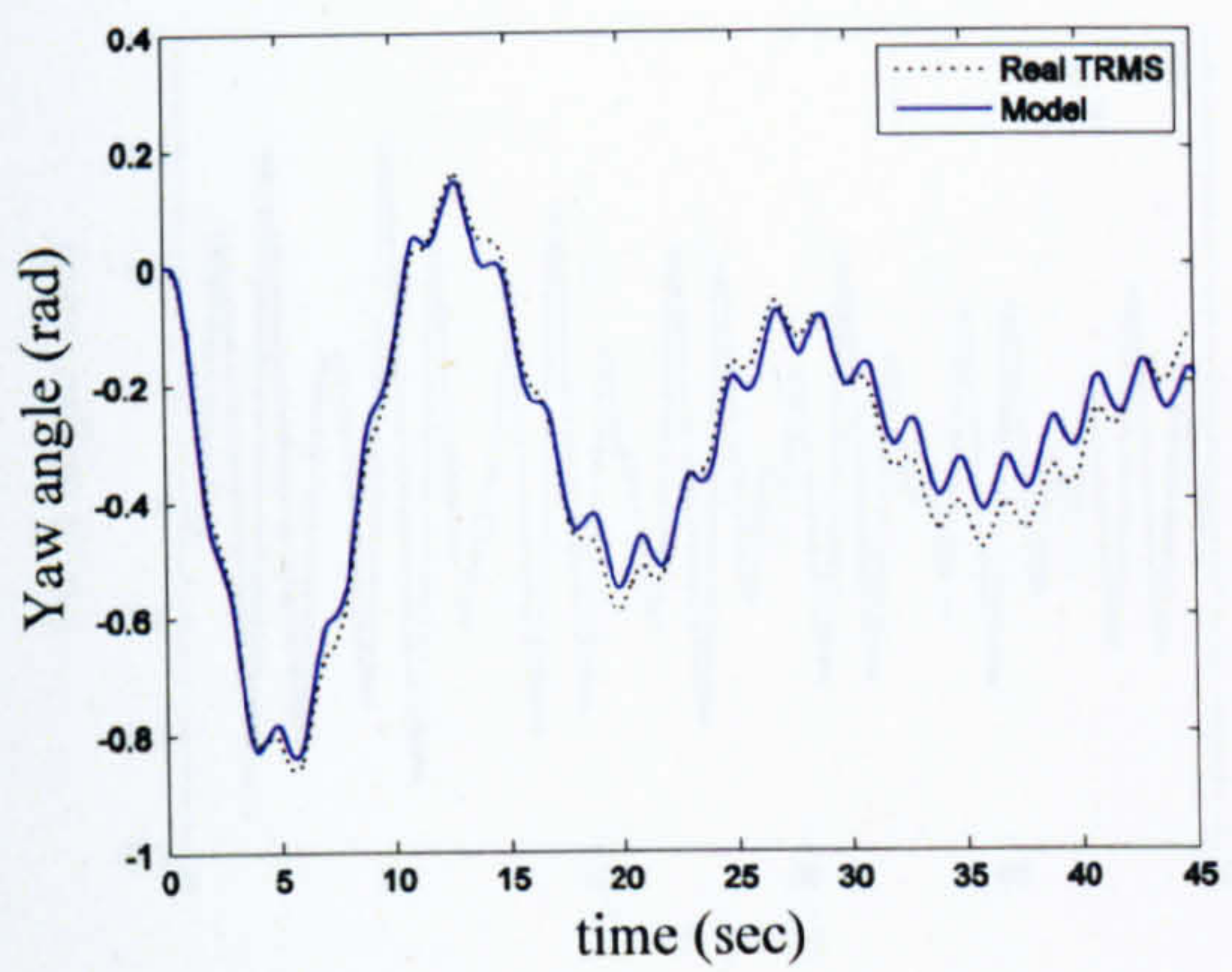


Figure 2.76 Yaw angle and PSD (case 6)

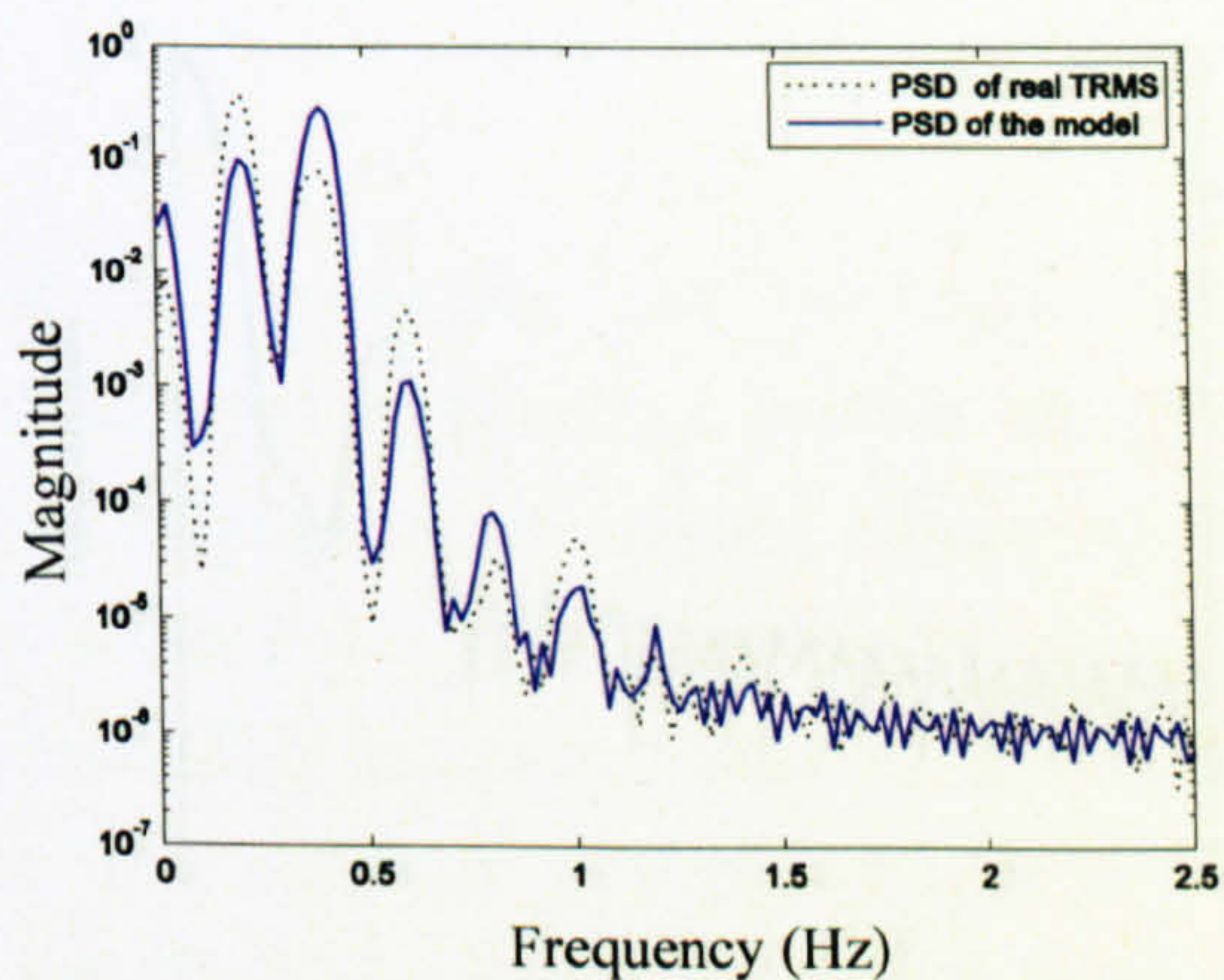
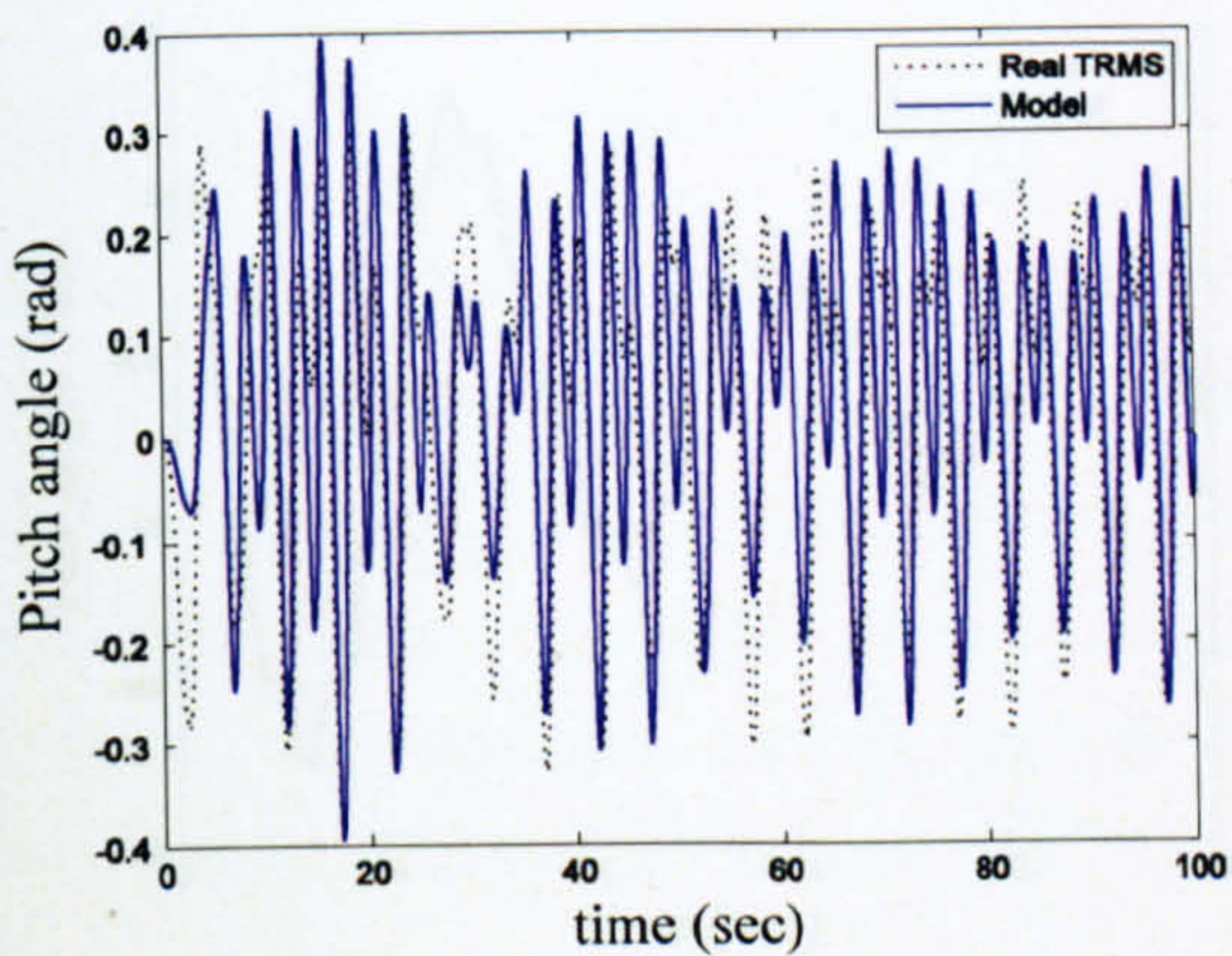


Figure 2.77 Pitch angle and PSD (case 7)

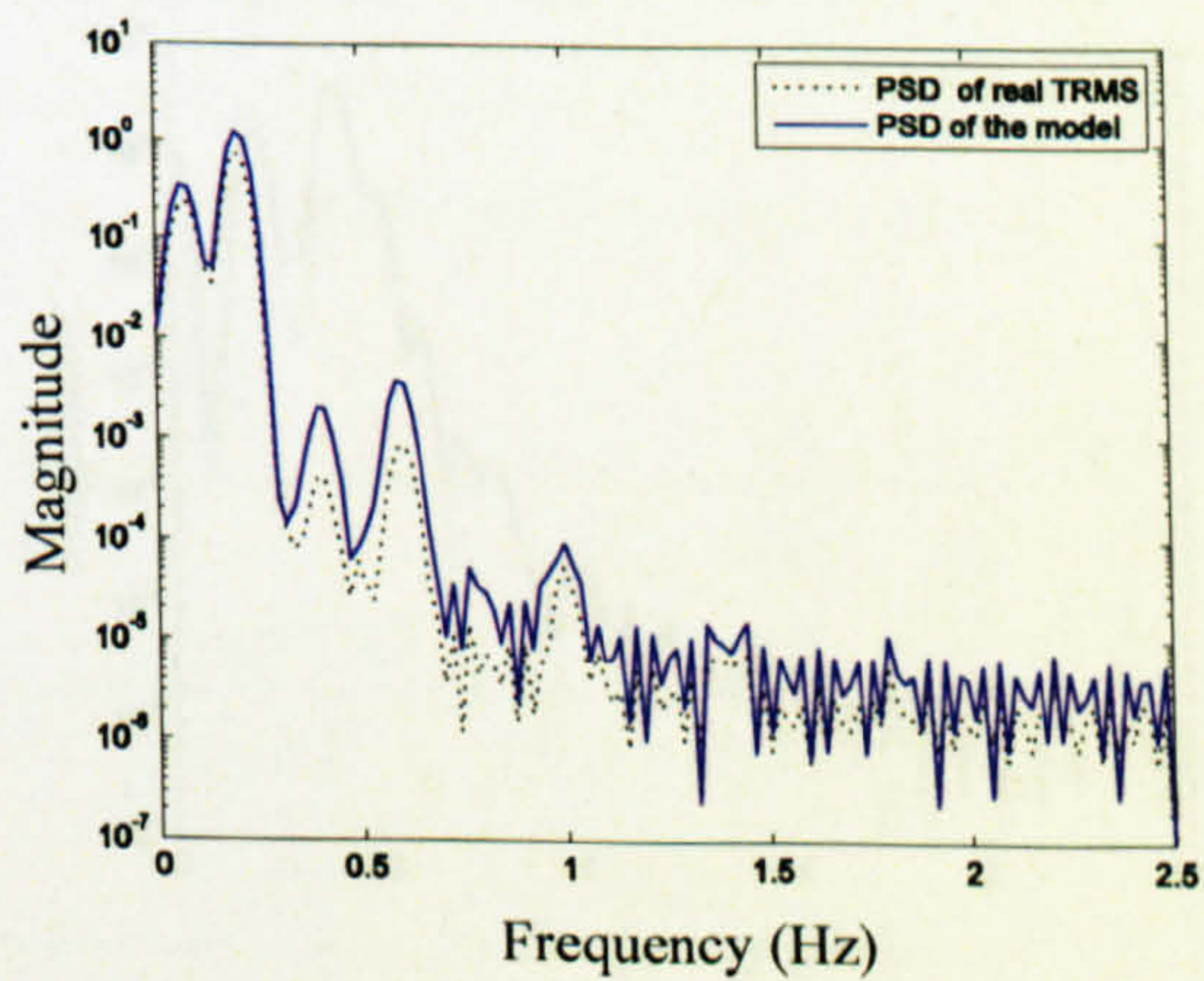
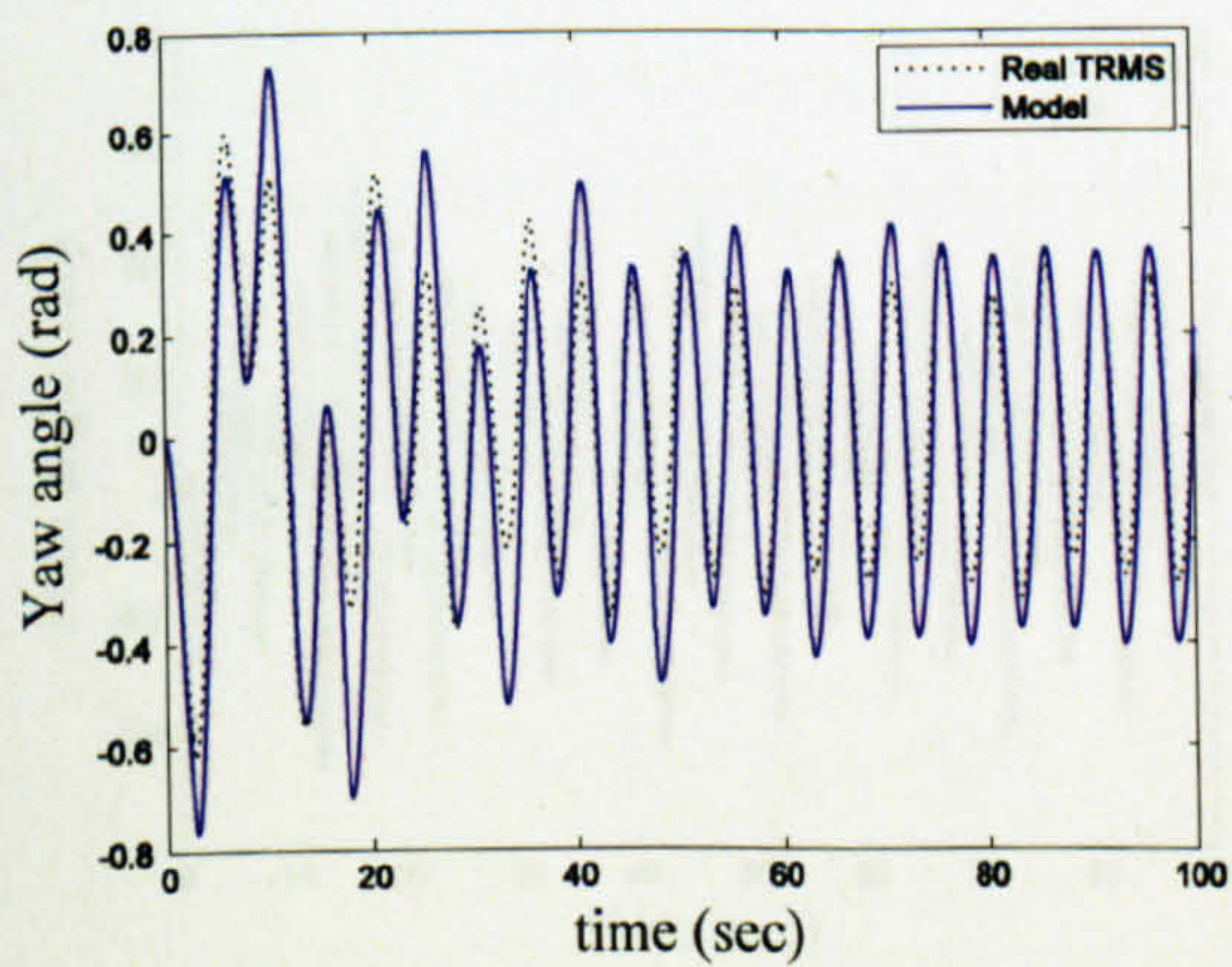


Figure 2.78 Yaw angle and PSD (case 7)

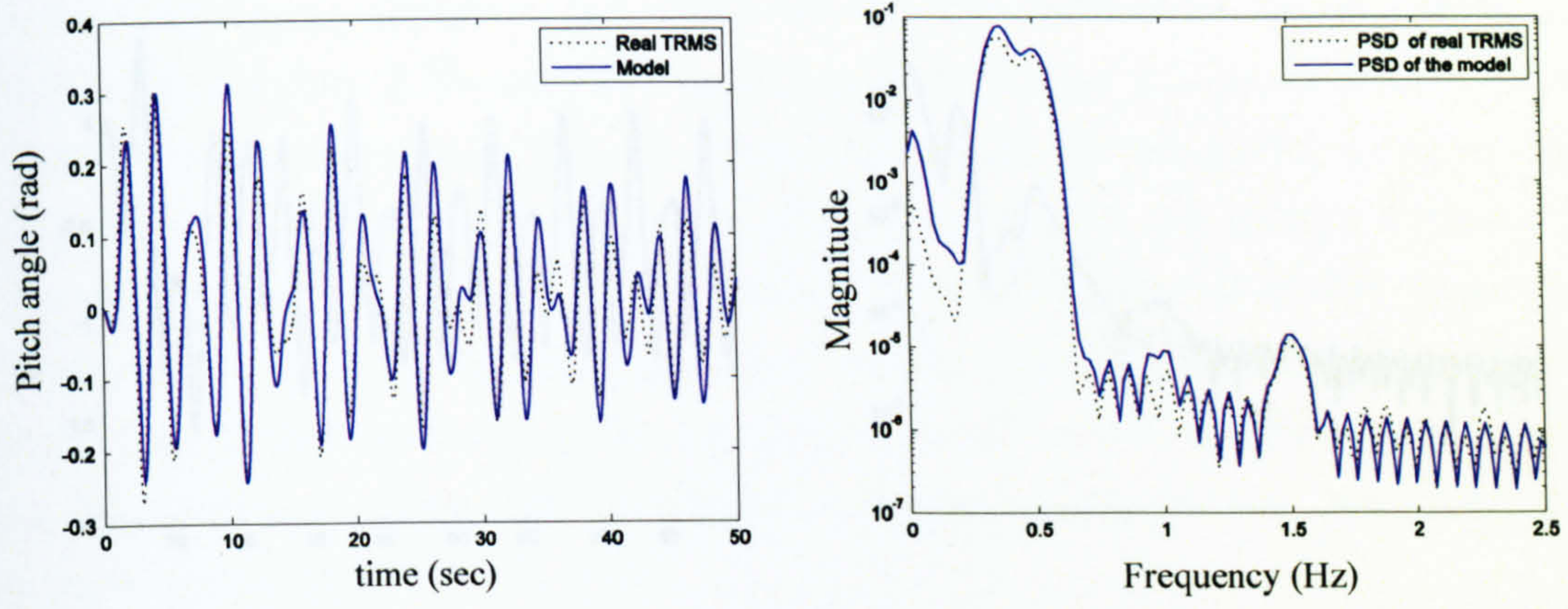


Figure 2.79 Pitch angle and PSD (case 8)

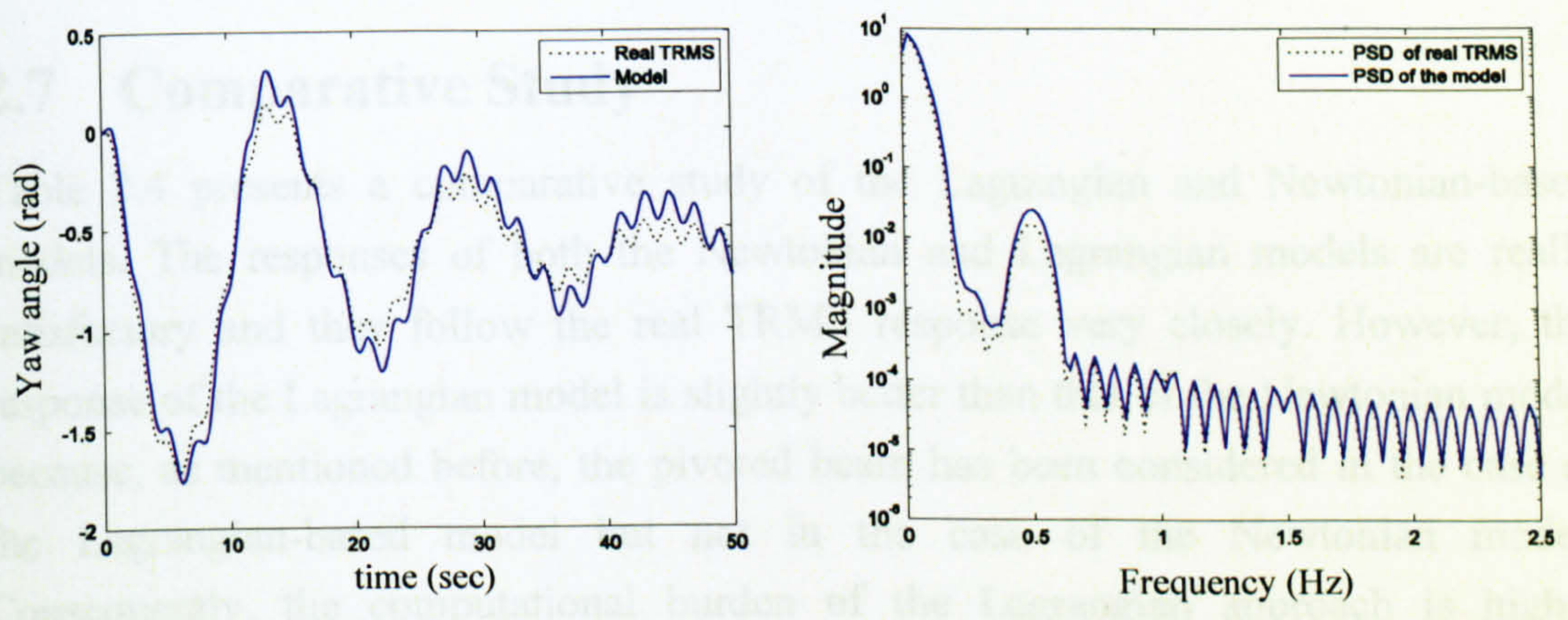


Figure 2.80 Yaw angle and PSD (case 8)

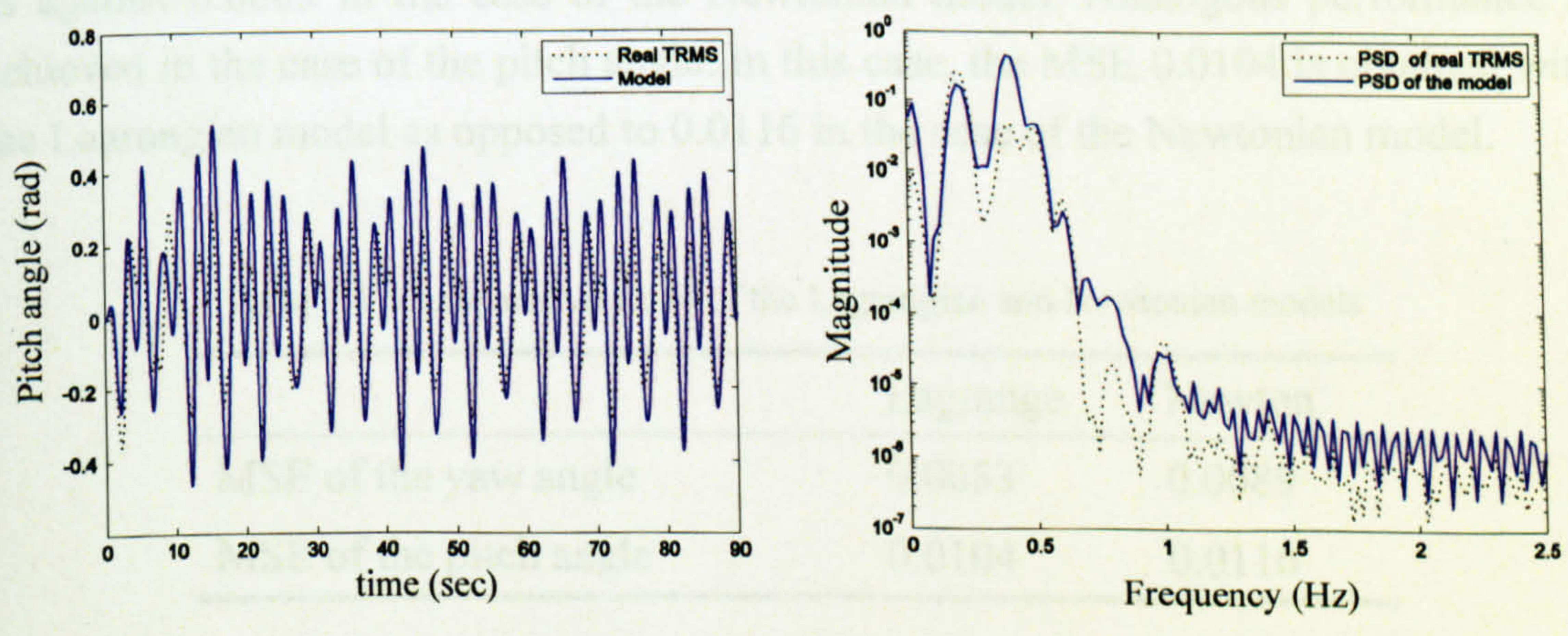


Figure 2.81 Pitch angle and PSD (case 9)

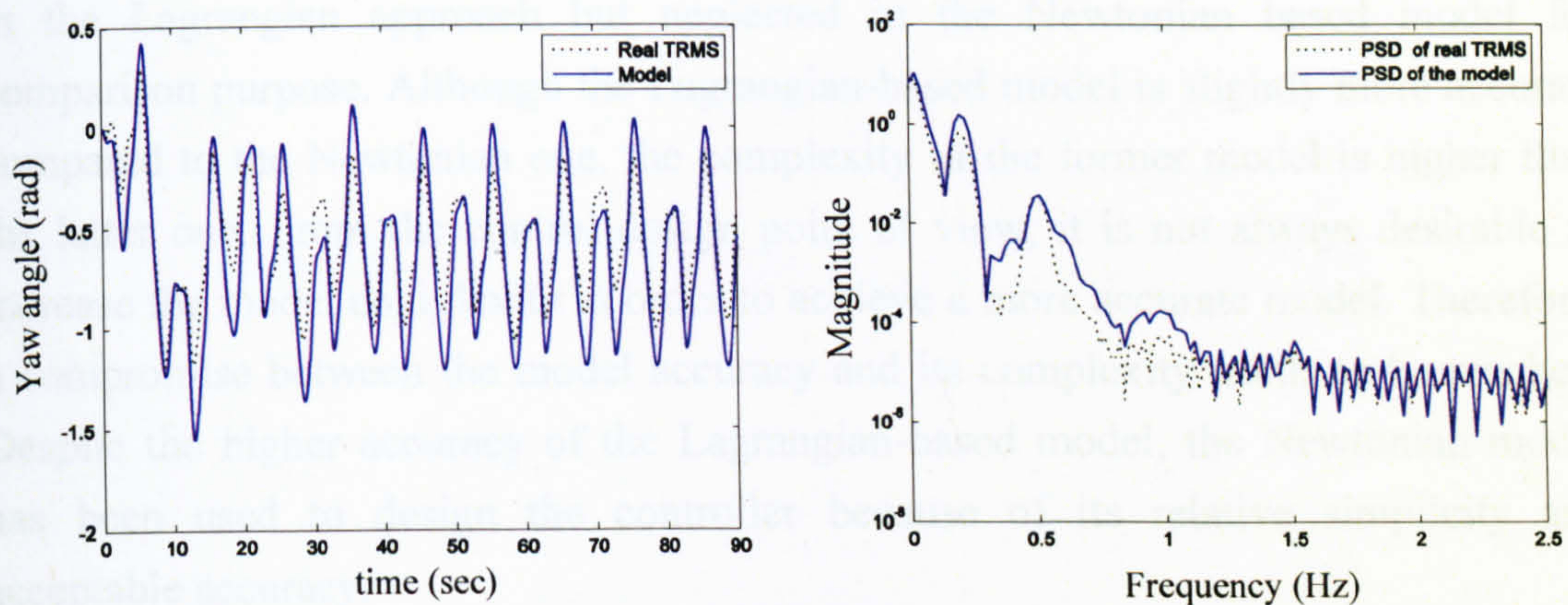


Figure 2.82 Yaw angle and PSD (case 9)

2.7 Comparative Study

Table 2.4 presents a comparative study of the Lagrangian and Newtonian-based models. The responses of both the Newtonian and Lagrangian models are really satisfactory and they follow the real TRMS response very closely. However, the response of the Lagrangian model is slightly better than that of the Newtonian model because, as mentioned before, the pivoted beam has been considered in the case of the Lagrangian-based model but not in the case of the Newtonian model. Consequently, the computational burden of the Lagrangian approach is higher compared to that of the Newtonian method. As presented in Table 2.4, the mean squared error (MSE) of the yaw angle in the case of the Lagrangian model is 0.0053 as against 0.0089 in the case of the Newtonian model. Analogous performance is achieved in the case of the pitch angle. In this case, the MSE 0.0104 is obtained with the Lagrangian model as opposed to 0.0116 in the case of the Newtonian model.

Table 2.4 Comparative study of the Lagrangian and Newtonian models

	Lagrange	Newton
MSE of the yaw angle	0.0053	0.0089
MSE of the pitch angle	0.0104	0.0116

2.8 Conclusions

In this chapter, the models of the TRMS developed using the Newtonian and Lagrangian approaches have been presented. The pivoted beam has been considered

in the Lagrangian approach but neglected in the Newtonian based model for comparison purpose. Although the Lagrangian-based model is slightly more accurate compared to the Newtonian one, the complexity of the former model is higher than the latter one. From the control design point of view, it is not always desirable to increase the model complexity in order to achieve a more accurate model. Therefore, a compromise between the model accuracy and its complexity needs to be reached. Despite the higher accuracy of the Lagrangian-based model, the Newtonian model has been used to design the controller because of its relative simplicity and acceptable accuracy.

CHAPTER 3

Nonlinear Black-box Modelling

This chapter presents the NN-based empirical modelling of the TRMS in terms of its vertical 1DOF, horizontal 1DOF and 2DOF. The model response and the real TRMS response are compared with each other to validate the accuracy of the model. A study has also been carried out to compare the performance of the analytical and empirical models.

3.1 Introduction

ANNs are intended to mimic the behaviour of biological NNs. As a matter of fact, there are varieties of NNs and their applications are also different. As such, it is difficult to find a unique definition of NNs, but the majority of NNs can be covered by this definition: "A system of simple processing elements, neurons that are connected into a network by a set of (synaptic) weight" [2]. Other names such as parallel distributed processing (PDP) systems [97] are also used in the literature. Over the years, several models of ANNs and the learning algorithms that associate with networks have been developed. In terms of structure, there are non-recurrent [98] and recurrent networks [99]. On the basis of functionality of the networks, there are associative memory networks [100], feedforward NNs [98], self-organising NNs [101], adaptive resonance theory networks [102], [103], and optimisation networks [104]. These networks need to be trained using appropriate learning algorithms for a particular application. A learning algorithm is an optimisation procedure in which the synaptic weights between the neurons are found in order to obtain an optimal mapping between a set of inputs and the corresponding desired outputs. In this investigation, the MLP networks have been chosen for modelling with various training methods. Therefore, the MLP networks and the related learning algorithms have briefly been presented here. Note that although other types of NNs networks may sometimes show a better performance from the perspective of training results, an MLP network usually outperforms its rivals from test results viewpoint.

3.2 The Fundamentals of MLP Networks

The function of a network is determined by the architecture of the network, its parameters (weights and biases) and the type of its processing elements. The architecture of a typical MLP network is defined by the number of layers and the number of neurons in each layer. Generally, the structure of an MLP network is designed based on an iterative process. It is needed to achieve a desirable performance with the lowest possible computational burden, i.e. the minimum number of layers and the least number of neurons in each layer. The synaptic weights and biases of a network are calculated during the so-called learning process in order to minimise the network error. Some of the most common learning algorithms that can be used to train MLP networks include: Levenberg-Marquardt back-propagation, gradient descent back-propagation, quasi-Newton back-propagation, Bayesian regularisation back-propagation, conjugate gradient back-propagation, one step secant back-propagation, resilient back-propagation, and scaled conjugate gradient back-propagation. All of these are employed in this investigation to train the NN-based model of the TRMS.

3.2.1 A Neuron

A neuron is a processing element, the McCulloch-Pitts (MP) model [105], takes a number of inputs, weighs them, sums them up, and uses the resultant as the argument of a single variable function, an activation function. Figure 3.1 shows a neuron with a bias and R inputs. As shown in equation (3.1), the output of the neuron is the output of a single variable activation function that takes the sum of the bias and the weighted inputs as its input argument.

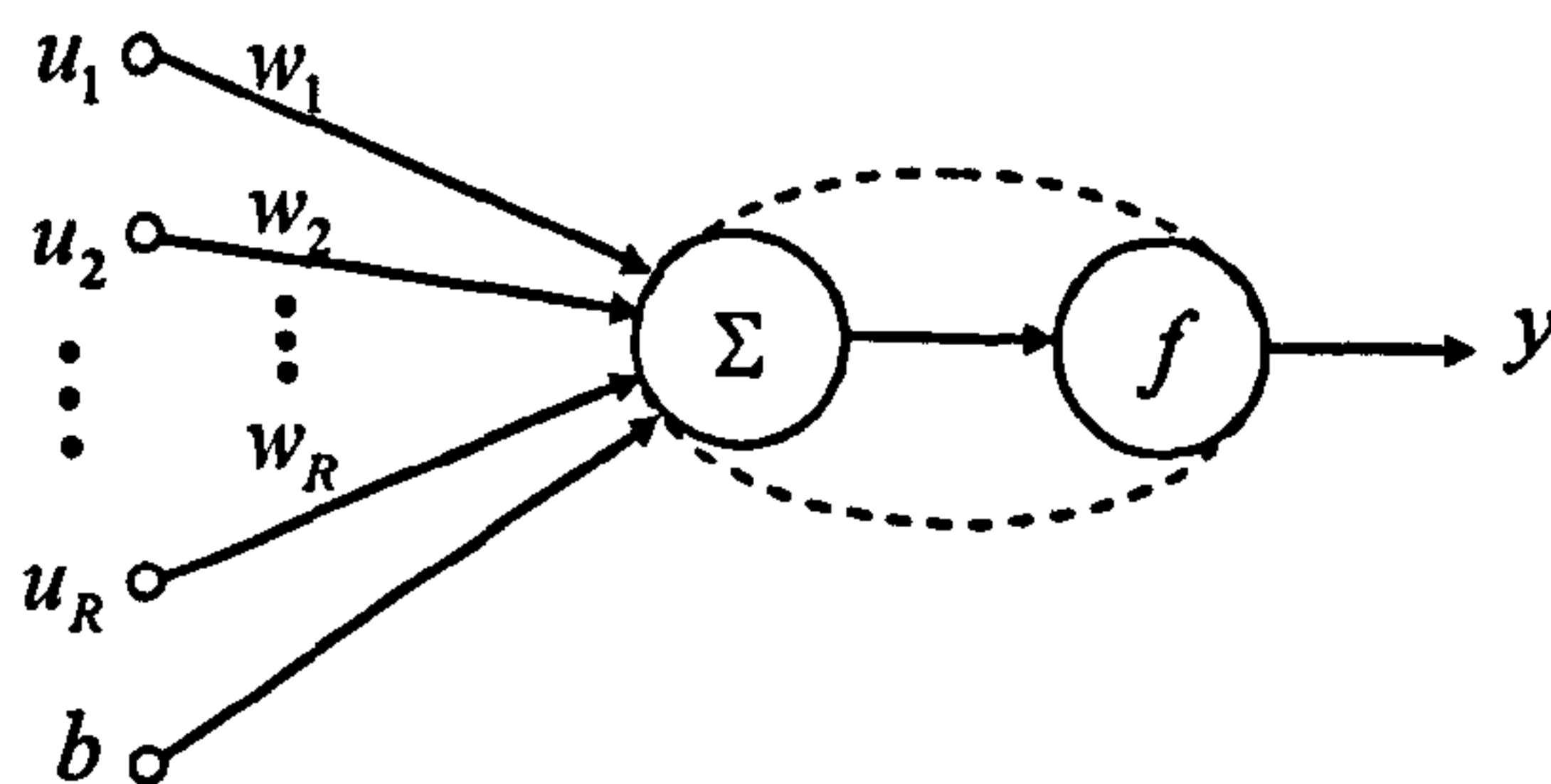


Figure 3.1 A neuron with a bias and R inputs

$$y = f\left(\left[\sum_{j=1}^R u_j w_j\right] + b\right) = f(\mathbf{u} \mathbf{w}^T + b) \quad (3.1)$$

Where $\mathbf{u} = [u_1, u_2, \dots, u_R]$ is the input vector, $\mathbf{w} = [w_1, w_2, \dots, w_R]$ is the weight vector, b is the bias, f is the activation function of the neuron, and y is the output of the neuron. The inputs to a neuron can be either the outputs of other neurons or external inputs. The activation function, f , can take any form; however, most often it is preferably a monotonic function. Figure 3.2 depicts some common activation functions. It is clear from Figure 3.2 that some of the functions such as (a), (b), (c) and (d) are monotonic; however, the others, i.e. (e), (f), and (g) are strictly monotonic. As pointed out before, differentiability of activation functions is one of the necessary conditions for most broadly used training algorithms. The first two functions in Figure 3.2 are not differentiable due to their discontinuity and moreover, the differentiations of the third and fourth functions are undefined at some points. Overall, only the last three functions are *smooth*, i.e. they have continuous derivatives of any required order.

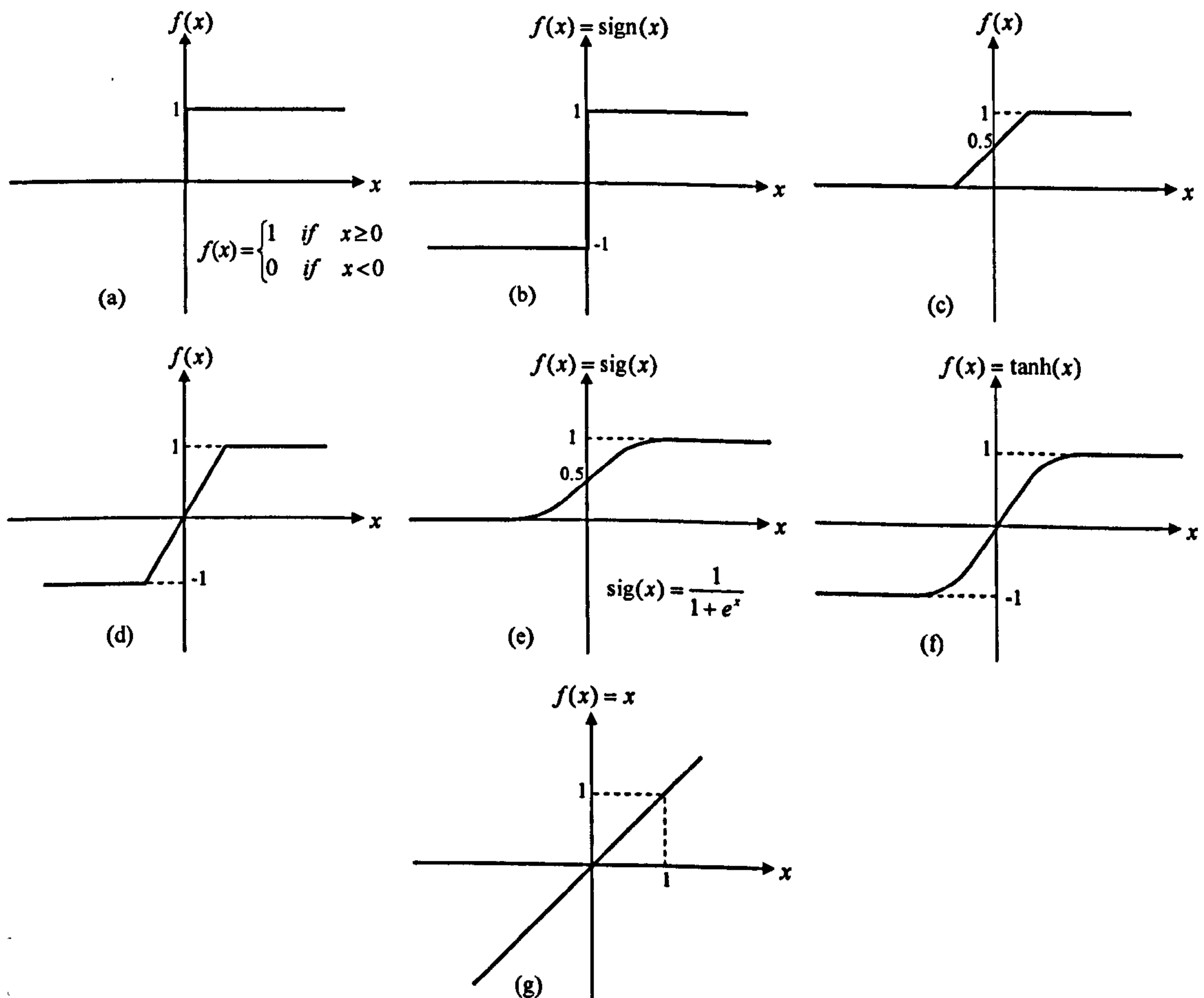


Figure 3.2 Some commonly used activation functions

3.2.2 The Multilayer Perceptron

The combination of a number of neurons with the same input signals, but most probably with different weights and biases, constructs a layer of neurons. Connection of two or more such layers forms a multilayer network. The neurons can be combined into a network in numerous fashions. Figure 3.3 shows a typical feedforward MLP network with R inputs and three layers with a configuration of $3 \times 5 \times 1$, i.e. three neurons in the first layer, five neurons in the second layer and one in the final layer. The final layer is called the output layer referring to the fact that it produces the output of the network. The other layers are hidden layers, e.g. in Figure 3.3 the network has two hidden layers. It is common to augment biases with weights by considering an extra input with the constant value of 1 for each layer, as shown in Figure 3.3 just for the first layer. Note that although each neuron has a bias, for the sake of simplicity, the biases of the second and final layers are not depicted in Figure 3.3.

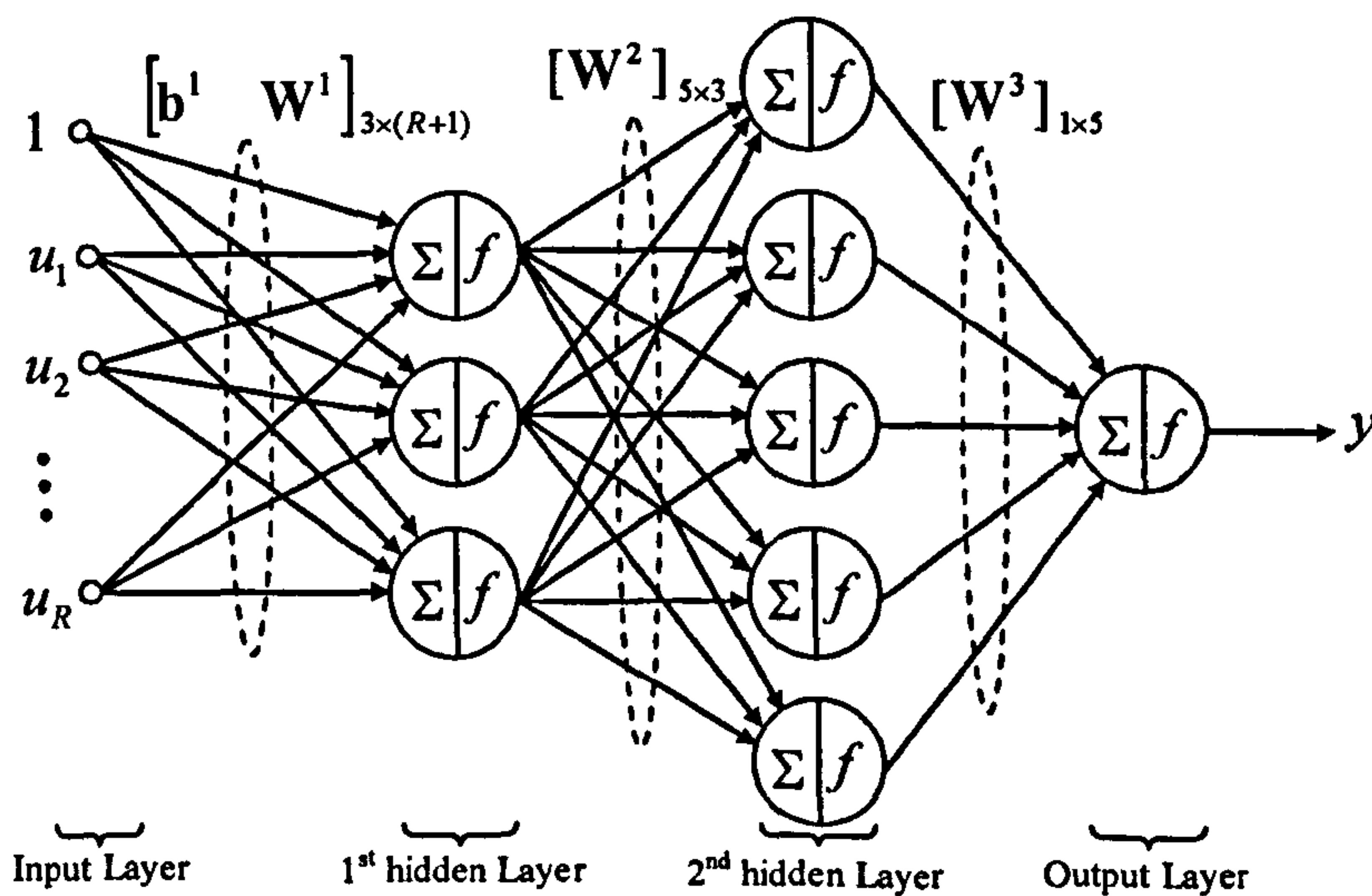


Figure 3.3 A typical MLP network with R inputs and three layers

3.2.3 Designing the Excitation Signal

Before an input signal is selected, it is important to identify the operating range of the system. Special care must be taken not to excite dynamics that one does not intend to incorporate in the model; however, in this investigation all the necessary frequencies have been considered at the training stage. In the case of identification of a linear system, it is customary to apply a signal consisting of a number of sinusoids of different amplitudes. Also the so-called pseudo random binary sequence (PRBS)

signals are popular. However, when working with nonlinear model structures, it is of great importance to represent all necessary amplitudes and frequencies [2].

In this study, the TRMS has been excited with various input signals of different shapes (e.g. sine and square waves), different possible amplitudes and different frequencies. The frequency of the training data ranges from 0.01 to 1 Hz and cover amplitude between -2.5 and 2.5 V, the minimum and maximum applicable voltages to the system respectively. According to the dynamics of the system the input signals with higher frequencies cause no considerable movement in the TRMS beam due to the fact that the approximate time constant of the system is significantly higher than the period of the signals with higher frequencies. So it is not necessary to excite the system with higher frequencies. However, in order to improve the model performance at high frequencies, which is important in control design, several PRBS and composite PRBS data sets have been utilised in addition to the data sets mentioned before.

3.2.4 Model Structure Selection

When widening the focus to black-box identification of nonlinear dynamic systems, the problem of selecting model structure becomes increasingly difficult. Some of the most common NN-based models include: neural network finite impulse response model structure (NNFIR), neural network autoregressive external input model structure (NNARX), neural network autoregressive moving average with external input model structure (NNARMAX), neural network output error model structure (NNOE), neural network state-space innovation form (NNSSIF) [2]. In this study, the NNARX approach has been selected in order to model the system, due to the fact that the input-output data set of the real system is available. As stated before, the TRMS possesses two shaft encoders to measure the outputs of the system, positions of the beam in both the horizontal and vertical planes.

In an NNARX model structure, the inputs to the NN-based model are the past control inputs of the real system, $[u(t-d), u(t-d-1), \dots, u(t-d-m)]$, and also the past observed outputs, $[y(t-1), y(t-2), \dots, y(t-n)]$, where d is some multiple of the sampling period and set to one here, and m and n are input and output lag spaces which are assumed to be two and three respectively. Figure 3.4 shows a SISO NNARX model structure. It is noted that the output of the SISO model, $\hat{y}(t)$, can be expressed as,

$$\hat{y}(t) = f_{NN}(u(t-d), u(t-d-1), \dots, u(t-d-m), y(t-1), y(t-2), \dots, y(t-n)) \quad (3.2)$$

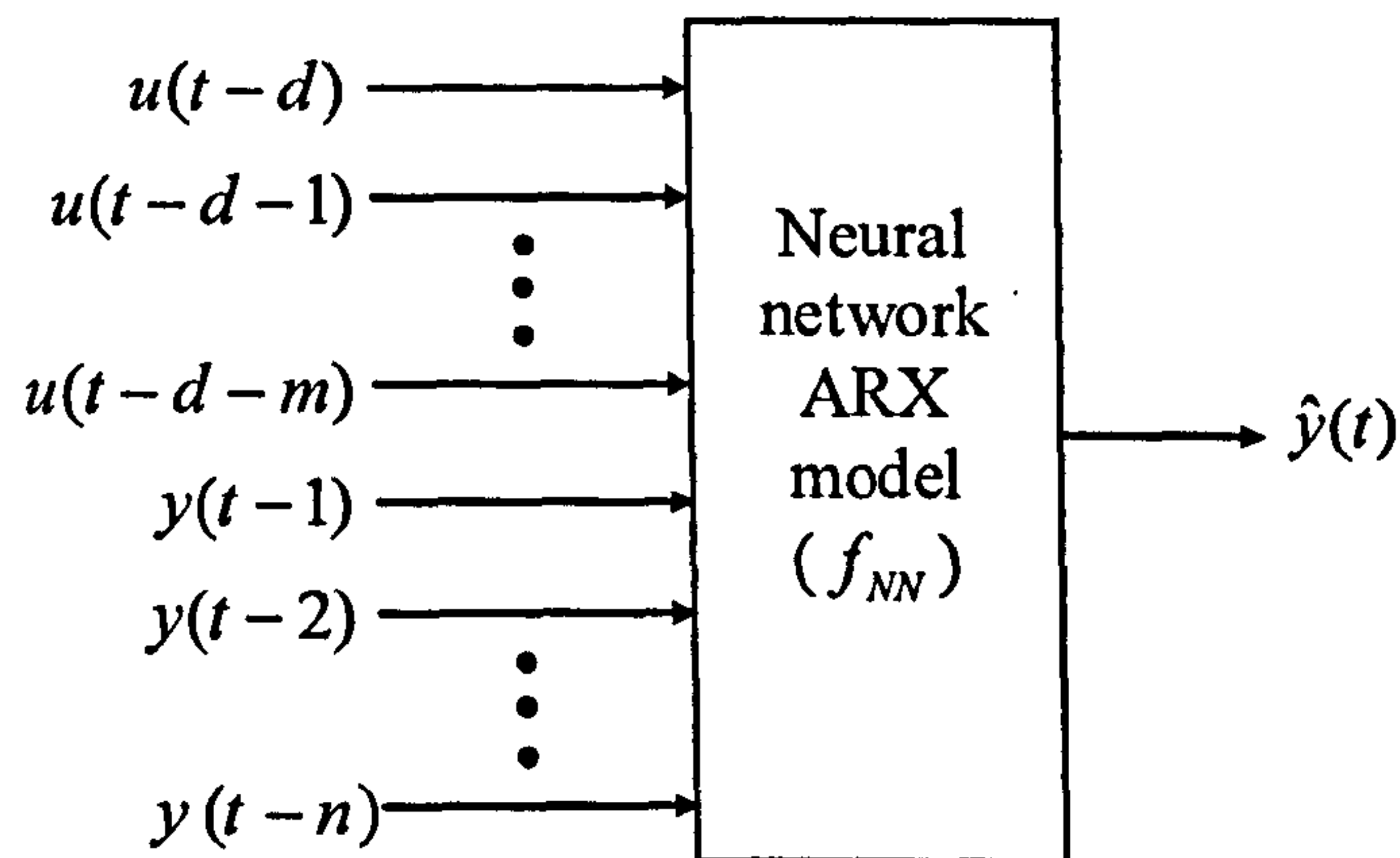


Figure 3.4 The NNARX model structure of a SISO system

3.2.5 Training algorithm

Assume that a data set has been acquired and the model structure has been selected. The next step is then to apply the data set to achieve the best model among the candidates by tuning the network parameters, i.e. weights and biases. This stage is called training or learning and its aim is to find a suitable set of weights and biases. Although training process is computationally intensive, it is generally one of the easiest stages in the identification of a system.

Some of the most common learning algorithms used in this study include:

- Gradient (steepest) descent (GD) back-propagation [106], [107],
- Gradient descent with momentum (GDM) back-propagation [107],
- Gradient descent with adaptive learning rate (GDA) back-propagation [107],
- Gradient descent with adaptive learning rate and momentum (GDX) back-propagation [107],
- Resilient back-propagation (RB) [108],
- Conjugate gradient back-propagation with Fletcher-Reeves (CGF) [109],
- Conjugate gradient back-propagation with Polak-Ribiere updates (CGP) [106],
- Conjugate gradient back-propagation with Powell-Beale restart (CGB) [110], [111],
- Scaled conjugate gradient (SCG) back-propagation [106],

- BFGS (Broyden Fletcher Goldfarb Shanno) quasi-Newton back-propagation [112]-[114],
- One step secant (OSS) back-propagation [113], [114],
- Levenberg-Marquardt (LM) back-propagation [115],
- Bayesian regularisation (BR) back-propagation [116], [117].

All of these algorithms are based on the so-called MSE method. In MSE paradigms the objective is to determine the weights and the biases of an NN by minimising the following criterion,

$$V = \frac{1}{2Q} \sum_{q=1}^Q \mathbf{e}_q^T \mathbf{e}_q \quad (3.3)$$

where,

$$\mathbf{e}_q = \mathbf{t}_q - \mathbf{y}_q \quad q = 1, 2, \dots, Q \quad (3.4)$$

Q : Number of patterns or number of data set for training

\mathbf{t}_q : Vector of target output (observed output) of pattern q

\mathbf{y}_q : Vector of network output (output prediction) of pattern q

\mathbf{e}_q : Error of pattern q

V : Mean squared error.

3.3 Model Validation

Model validation is a procedure in which a model is tested to clarify whether the model adequately represents the characteristics of the corresponding real system. Note that each model is tested in accordance with its future application. However, it is usual to test a model using some standard test methods such as: overlapping, PSD, and correlation tests. Overlapping represents, in the time domain, a process response and its model response in one figure and it focuses on the discrepancy between them. The PSD result illustrates the accuracy of a model in the frequency domain. Some other useful features of the model can be envisaged from the PSD. Clearly, the data used to test a model should be different from those used in the training stage of the model. Such a data set is called test or validation data set. Correlation test is described as follows.

If the residuals (prediction errors) of a model contain no or negligible information about the past residuals or about the dynamics of the system, it is likely that all information has been extracted from the training set, and conclusively the model approximates the system well. Hence, one needs, in principle, to check whether the residuals are uncorrelated with all the linear and nonlinear combinations of the past data. Such a test is of course completely unrealistic to be carried out in practice; thus, it is common to consider only a few wisely chosen auto-correlation and cross-correlation functions. Some of the correlation functions recommended by Billings *et al.* [118], [119] are listed below. Equation (3.5) is the auto-correlation of prediction errors, and the cross-correlation between the inputs and the residuals of a model is presented in equation (3.6). The cross-correlation of the squared inputs and the squared errors is mentioned in equation (3.7). Equation (3.8) presents the cross-correlation between the squared inputs and the residuals of the model and finally the cross-correlation between the residuals and the multiplication of the residuals and the inputs is expressed in equation (3.9).

$$r_{ee}(\tau) = \frac{\sum_{i=1}^{N-\tau} (e_i - \bar{e})(e_{i+\tau} - \bar{e})}{\sum_{i=1}^N (e_i - \bar{e})^2} = \begin{cases} 1 & \text{if } \tau = 0 \\ 0 & \text{if } \tau \neq 0 \end{cases} \quad (3.5)$$

$$r_{ue}(\tau) = \frac{\sum_{i=1}^{N-\tau} (u_i - \bar{u})(e_{i+\tau} - \bar{e})}{\sqrt{\sum_{i=1}^N (u_i - \bar{u})^2 \sum_{i=1}^N (e_i - \bar{e})^2}} = 0 \quad \forall \tau \quad (3.6)$$

$$r_{u^2e^2}(\tau) = \frac{\sum_{i=1}^{N-\tau} (u_i^2 - \bar{u}^2)(e_{i+\tau}^2 - \bar{e}^2)}{\sqrt{\sum_{i=1}^N (u_i^2 - \bar{u}^2)^2 \sum_{i=1}^N (e_i^2 - \bar{e}^2)^2}} = 0 \quad \forall \tau \quad (3.7)$$

$$r_{u^2e}(\tau) = \frac{\sum_{i=1}^{N-\tau} (u_i^2 - \bar{u}^2)(e_{i+\tau} - \bar{e})}{\sqrt{\sum_{i=1}^N (u_i^2 - \bar{u}^2)^2 \sum_{i=1}^N (e_i - \bar{e})^2}} = 0 \quad \forall \tau \quad (3.8)$$

$$r_{e\beta}(\tau) = \frac{\sum_{i=1}^{N-\tau} (e_i - \bar{e})(\beta_{i+\tau} - \bar{\beta})}{\sqrt{\sum_{i=1}^N (e_i - \bar{e})^2 \sum_{i=1}^N (\beta_i - \bar{\beta})^2}} = 0 \quad \tau \geq 0 \quad (3.9)$$

Where the bar over the symbols represents the average of a signal as,

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (3.10)$$

N is the total number of data and,

$$\beta_i = u_i e_i \quad (3.11)$$

It is common to check whether the functions for lags in the interval $\tau \in [-20, 20]$ are zero within an asymptotical 95% confidence interval, i.e. if the following condition is held. Details can be found in Appendix 3.

$$-1.96/\sqrt{N} < r < 1.96/\sqrt{N} \quad (3.12)$$

The first two tests are commonly used to validate the identification of conventional systems, especially in the case of linear systems. It should be mentioned that all the tests apply equally well to MIMO systems. The correlation functions must then be calculated for each combination of the input and the output.

3.4 Results

In this section, the TRMS is empirically modelled in terms of its 1DOF horizontal, 1DOF vertical and 2DOF dynamics. The three following sub-sections present the results of these NN-based models. As mentioned before, in this research, the models are developed using NN-based autoregressive external input approach. The MATLAB NN toolbox [120] is utilised to estimate the parameters, i.e. the weights and biases, of the networks.

3.4.1 The 1DOF Horizontal Model

In order to model the TRMS in terms of its 1DOF horizontal dynamics, an MLP neural network with a configuration of $6 \times 3 \times 1$ has been used. In other words, the NN-based model has 6 inputs, 3 neurons in the hidden layer and 1 neuron in the output layer. The configuration is started with a simple single layer network and the number of neurons is, gradually, increased until the desired satisfactory performance is reached. Note that if increasing the number of neurons causes no satisfactory result, the number of hidden layers needs to be expanded. The network inputs are the tail rotor voltage at the previous sample time, $V_h(t-1)$, the tail rotor voltage at two

samples before, $V_h(t-2)$, the tail rotor voltage at three samples before, $V_h(t-3)$, the yaw angle of the beam at the previous instant, $\alpha_h(t-1)$, the yaw angle of the beam at two samples before, $\alpha_h(t-2)$, and the yaw angle of the beam at three samples before, $\alpha_h(t-3)$. This configuration has been selected using the method mentioned before in order to have a satisfactory result without any unnecessary computational burden, i.e. with a minimum network structure. Figure 3.5 shows the structure of the NN-based model in terms of the 1DOF horizontal dynamics. The activation functions used in the hidden and output layers are logarithmic and pure linear respectively. The numbers of patterns for training and test processes used are 8290 and 500 sets respectively. It is noted that the sample time is set to be 0.2 seconds.

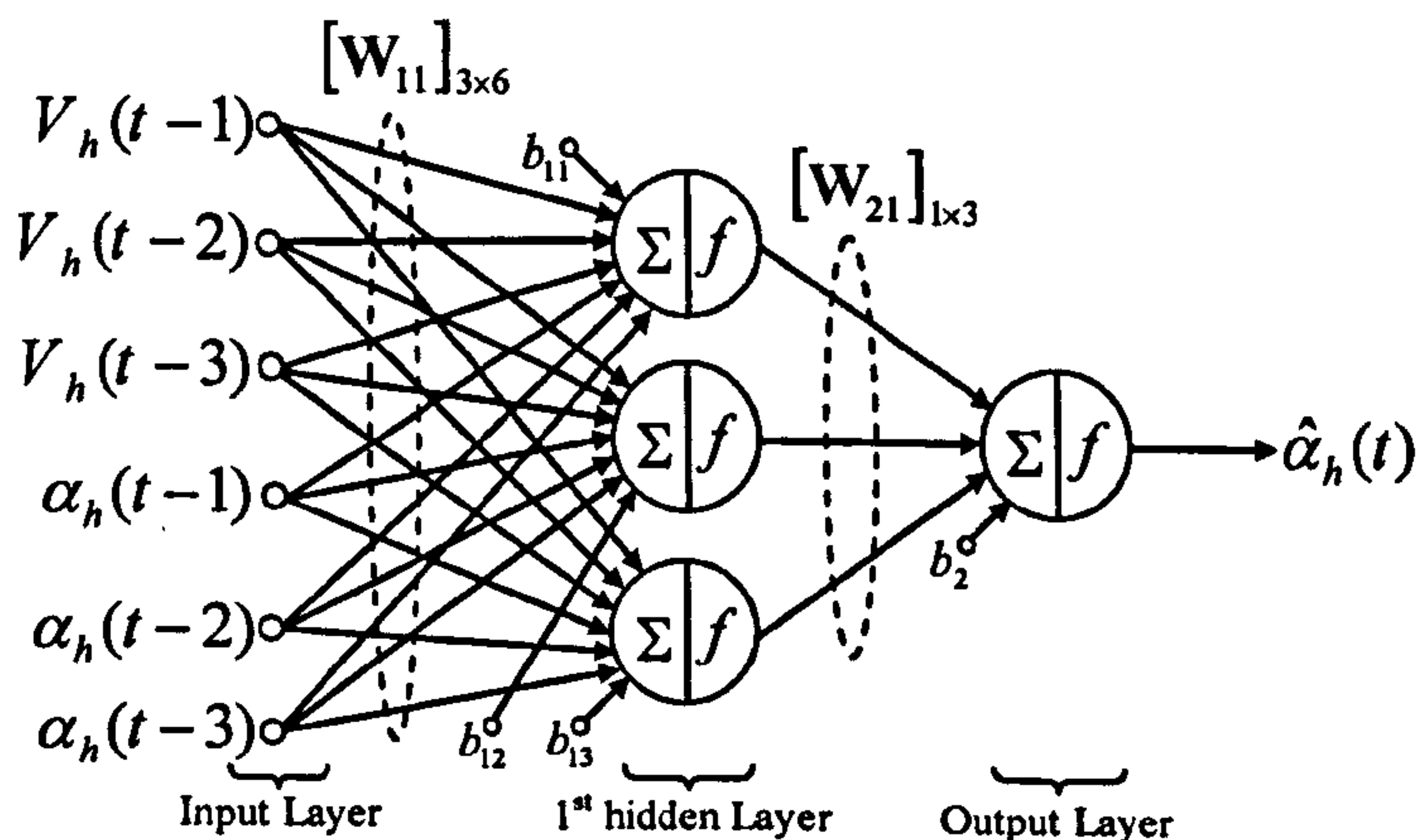


Figure 3.5 The structure of the NN-based model in terms of the 1DOF horizontal dynamics

The NN has been trained with 13 different training algorithms. The trained NNs have been tested with a set of data not used in the training phase. The response of the neural models trained with various types of algorithm and the corresponding validation test results are depicted in Figures 3.6 to 3.31.

Figure 3.6 shows the overlapping and the PSD test results of the NN trained with the gradient descent back-propagation algorithm. It is clear from the results that the model is not adequate to predict the output of the system accurately. Also, correlation tests, as shown in Figure 3.7, illustrate that the model is insufficient since, the correlation results are out of the confidence interval. Note that the gradient descent learning algorithm, the simplest form of back-propagation learning method, updates the weights and biases in the direction in which the objective function (see equation 3.2) decreases most rapidly. This direction can be found using the negative of the gradient.

The gradient descent with momentum learning algorithm converges faster than the simple gradient descent. A momentum term uses the recent trends of network parameters (weights and biases) besides the gradient and prevents from getting stuck in a local minimum. The performance of this learning algorithm in modelling the TRMS in terms of its 1DOF horizontal dynamics is presented in Figures 3.8 and 3.9. Although the overlapping and the PSD tests show that the results are quite satisfactory, it is evident from the more convincing correlation tests that the model cannot represent the real TRMS adequately.

Standard gradient descent techniques employ a fixed learning rate during the training process. Changing the learning rate improves the performance of the algorithm, as this parameter significantly affects the stability and the convergence time of the training algorithm. Figures 3.10 and 3.11 depict the test results of the trained network using the gradient descent with adaptive learning rate algorithm. The performance is comparable to that of the previous training method, and thus it needs to be improved. Sometimes employing the advantages of the momentum term and the adaptive learning rate simultaneously can improve the performance of neural models; however, in the case of the TRMS the results, Figures 3.12 and 3.13, show no considerable improvement.

Sigmoid functions are the most popular types of transfer functions used in the hidden layers of multilayer networks. They are also called squashing functions, due to the fact that they squash an infinite input range into a finite output range. Sigmoid functions are characterised by the fact that their gradient must approach zero as the input becomes large. This fact renders a difficulty when the gradient descent learning algorithm is used to train a multilayer network with sigmoid functions. The gradient may take a very small magnitude when the modulus of the input is large and therefore, cause small changes in the weights and biases, even though the weights and biases are far from their optimal values.

In order to eliminate these harmful effects, the resilient back-propagation training algorithm can be used. In this case, the direction of the weight update is determined using the sign of the derivative, i.e. the magnitude of the derivative has no effect on the weight update. A separate update value is defined to determine the size of the weight change. Figures 3.14 and 3.15 show the model developed using resilient back-propagation learning approach. Compared to the aforementioned training

algorithms, this training approach performs better, although from the perspective of correlation tests, the performance is not good enough and needs further enhancement.

As mentioned before, the standard back-propagation algorithm updates the network parameters in the steepest descent direction. The conjugate gradient algorithms search along conjugate directions, which lead to a faster convergence in comparison with the steepest descent directions. In this investigation, four types of conjugate gradient learning algorithms have been used to train the NN-based model. Amongst them, the scaled conjugate and the conjugate Powell-Beale learning algorithms outperform the other two in terms of the correlation tests and MSE. Figures 3.16 to 3.23 illustrate the performance of the conjugate gradient learning approaches.

The one step secant and the BFGS quasi-Newton learning algorithms are the simplified version of Newton's optimisation method in which an approximate Hessian matrix (second derivatives) is updated in each iteration. Therefore, these algorithms reduce the computational burden compared to Newton's method. Figures 3.24 to 3.27 show the results obtained using these two learning algorithms. It is clear that the performance of the BFGS learning algorithm is satisfactory; however, the one step secant learning approach could not properly model the TRMS in terms of its 1DOF horizontal dynamics.

The Levenberg-Marquardt learning algorithm can be considered to be a modified version of the Gauss-Newton approach. It has been designed to approximate the second-order training speed without having to compute the Hessian matrix. Figures 3.28 and 3.29 depict the overlapping, PSD and correlation tests of the network trained with the LM training algorithm. The LM learning algorithm has outperformed its rivals in terms of the correlation tests and MSE, and can be considered the best learning approach to model the 1DOF horizontal TRMS.

The Bayesian regularisation method modifies the performance function by adding a term consisting of the mean squares of the network weights and biases and then uses one of the aforementioned learning algorithms (e.g. LM). The results of NN-based model trained with Bayesian regularisation are presented in Figures 2.30 and 2.31. The results are satisfactory in terms of all the performance criteria.

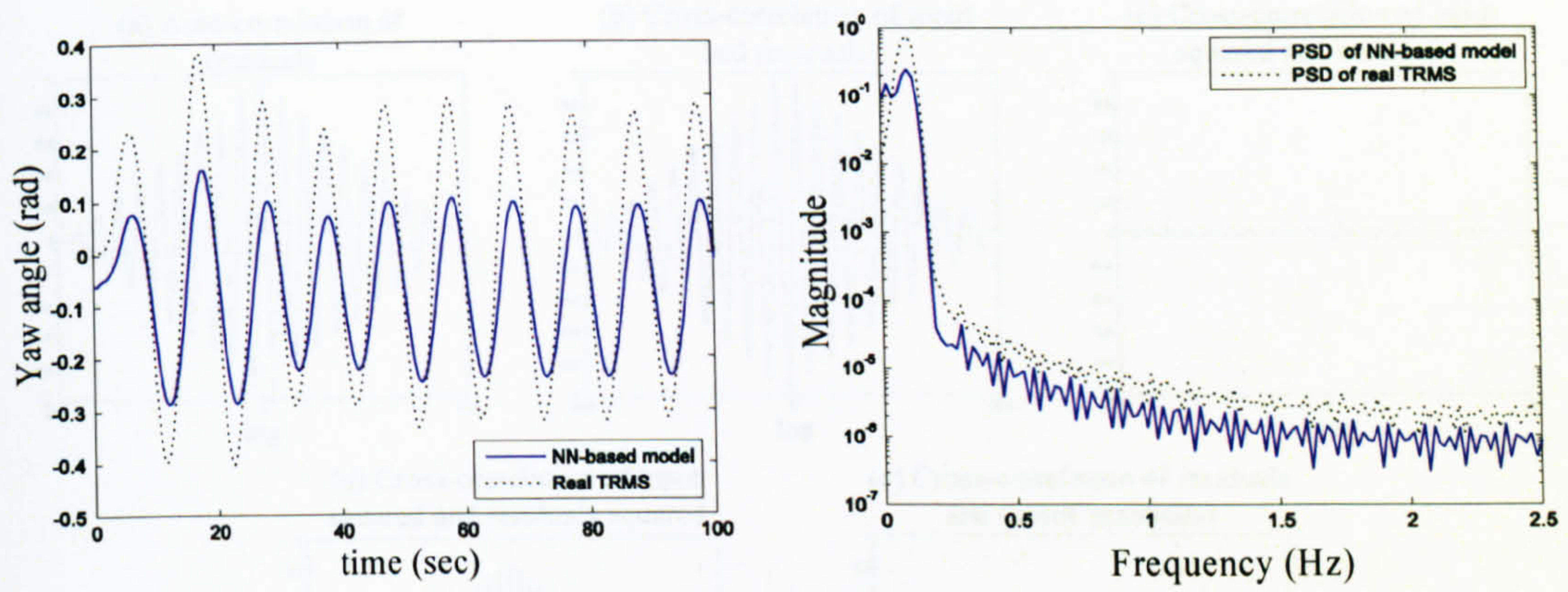


Figure 3.6 Model response with the gradient descent method

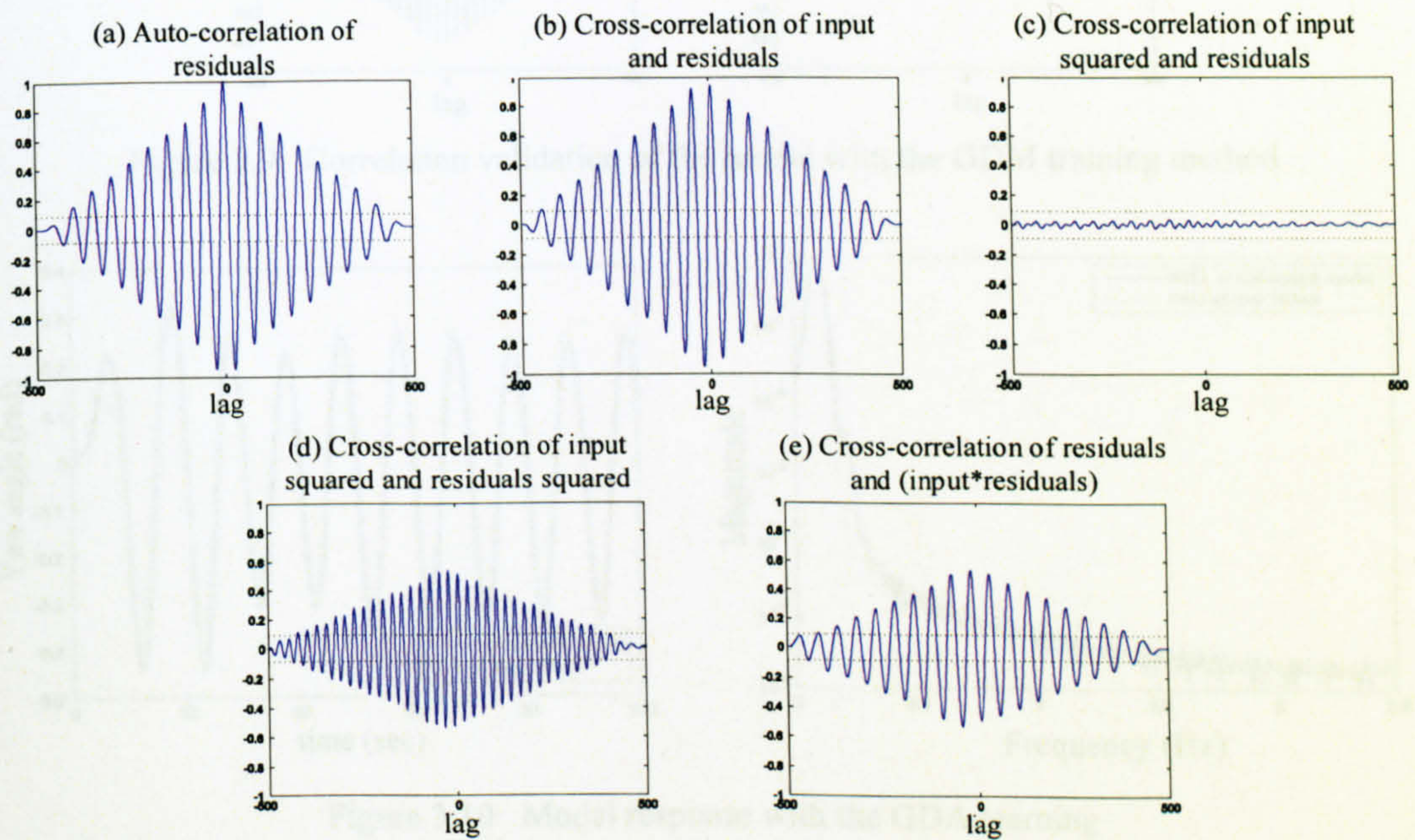


Figure 3.7 Correlation validation of the neural model with the GD training method

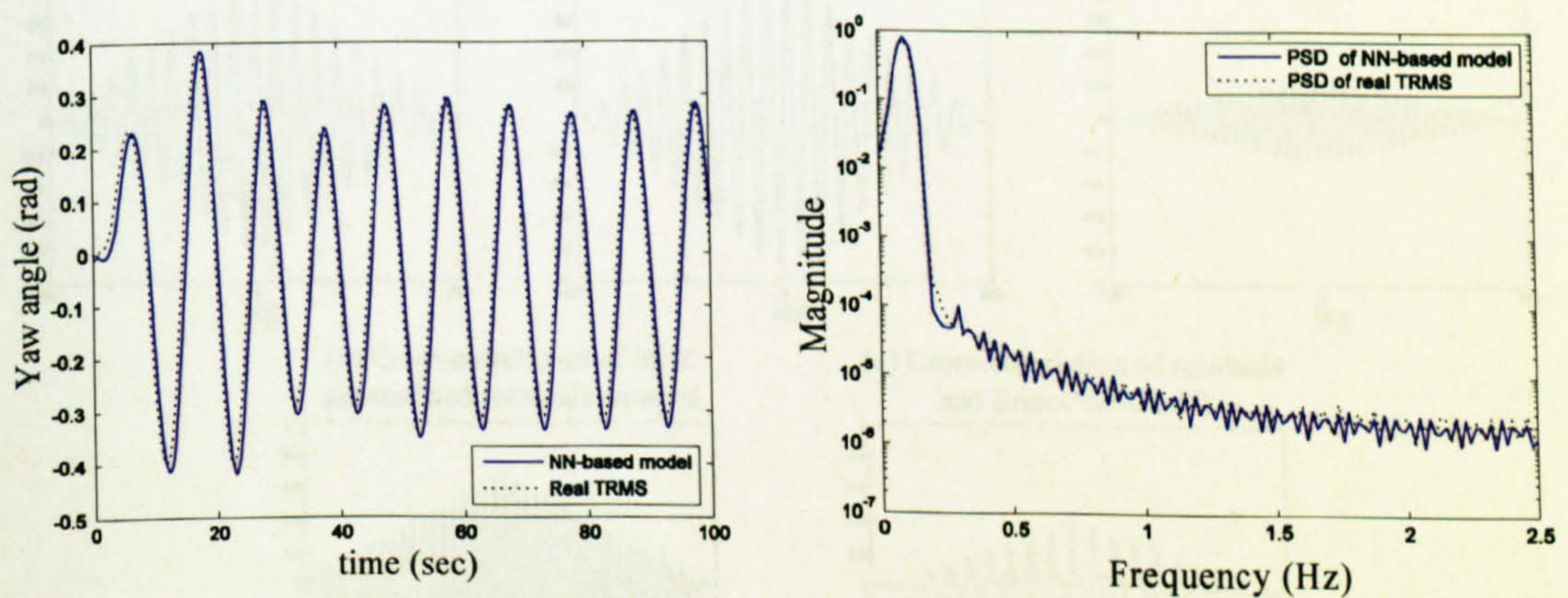


Figure 3.8 Model response with the GDM training method

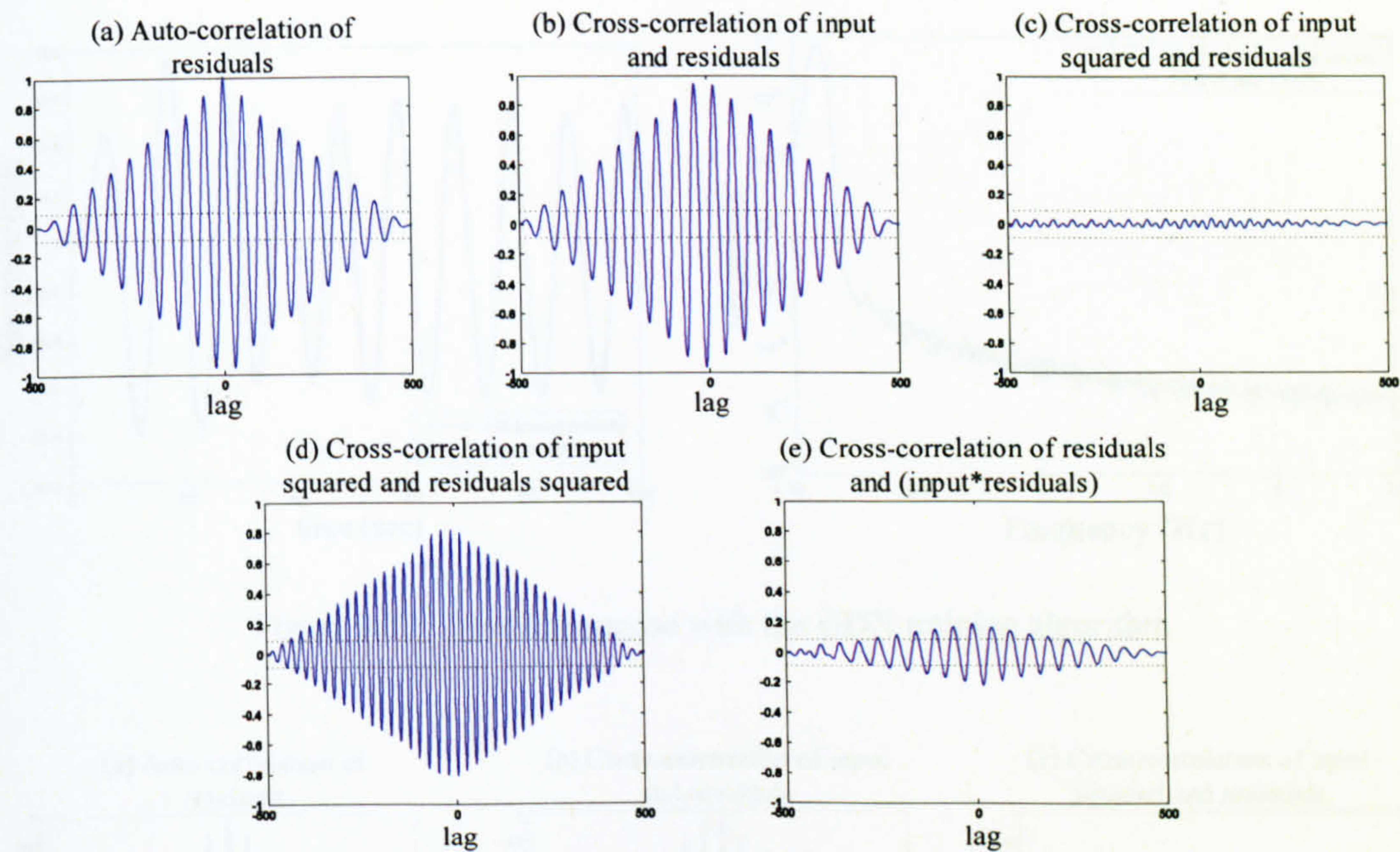


Figure 3.9 Correlation validation of the model with the GDM training method

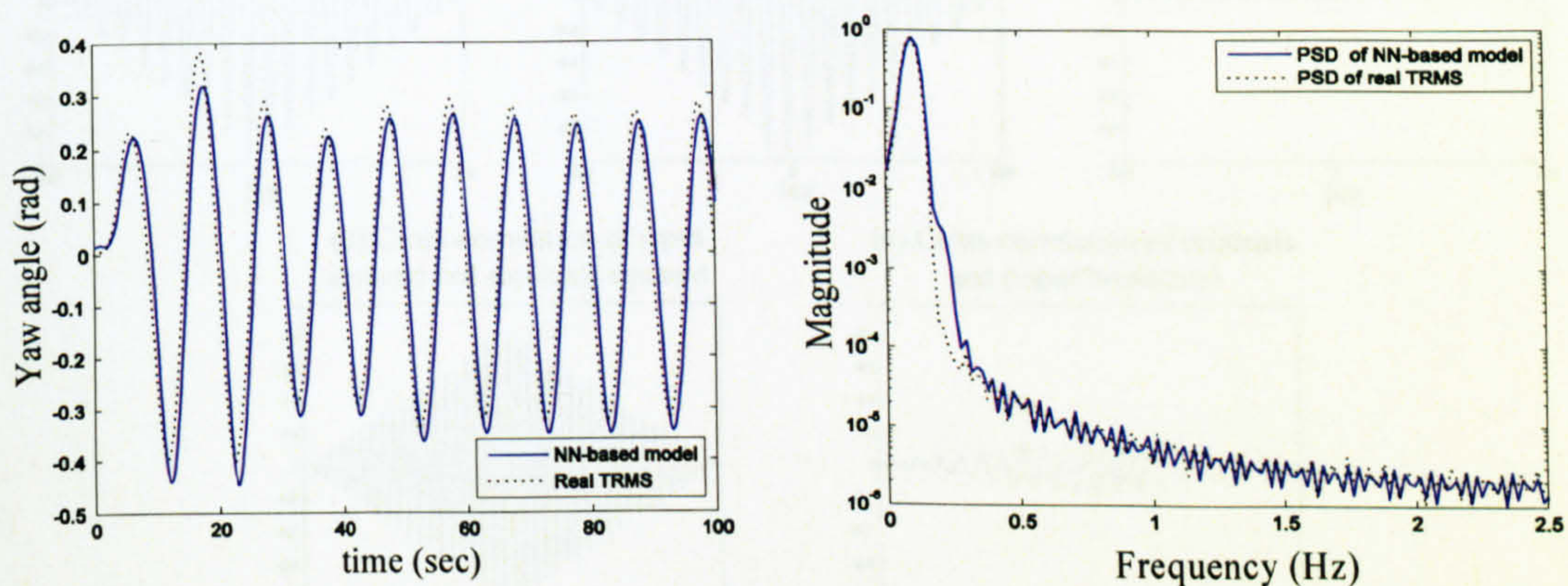


Figure 3.10 Model response with the GDA learning

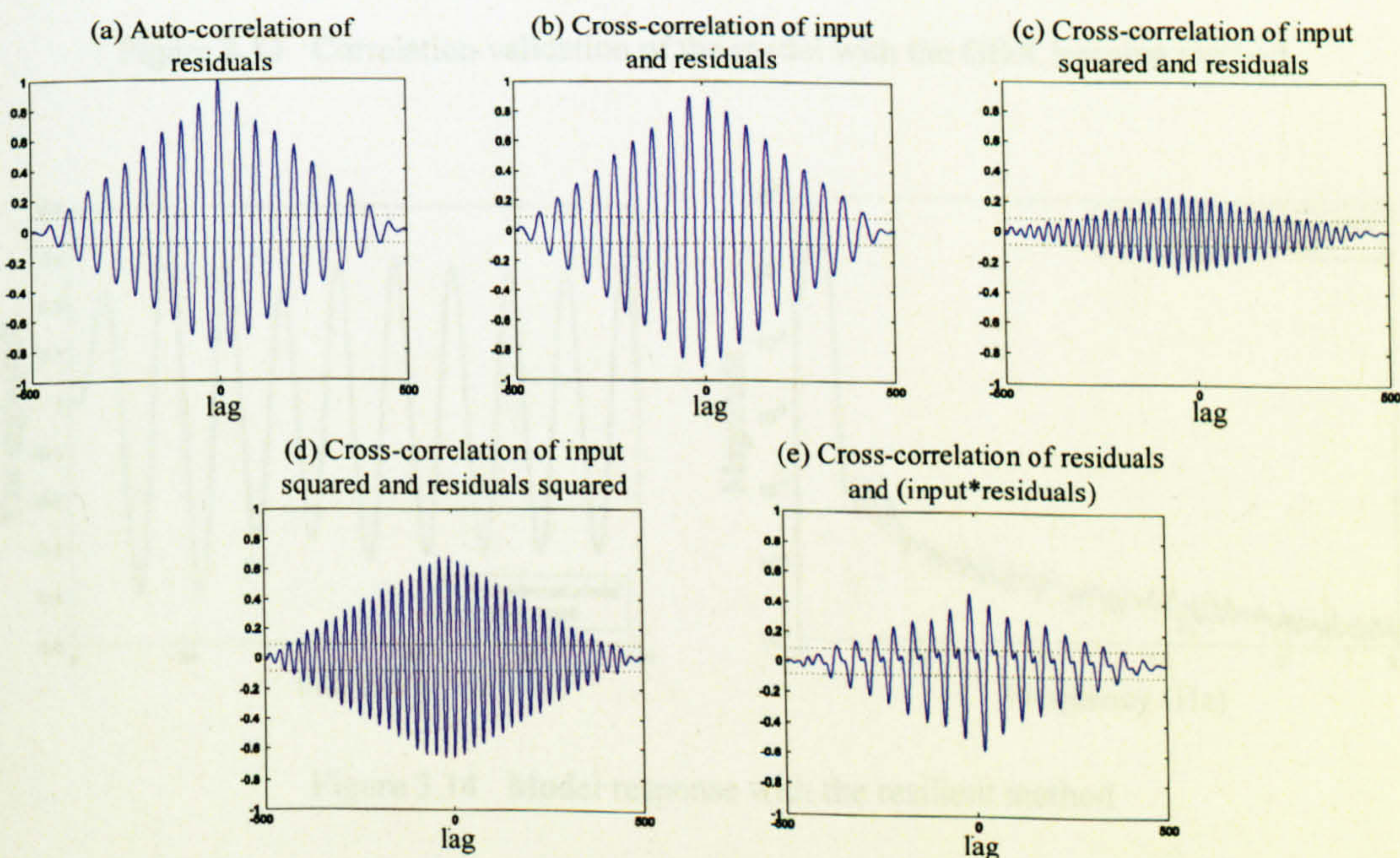


Figure 3.11 Correlation validation of the model with the GDA training method

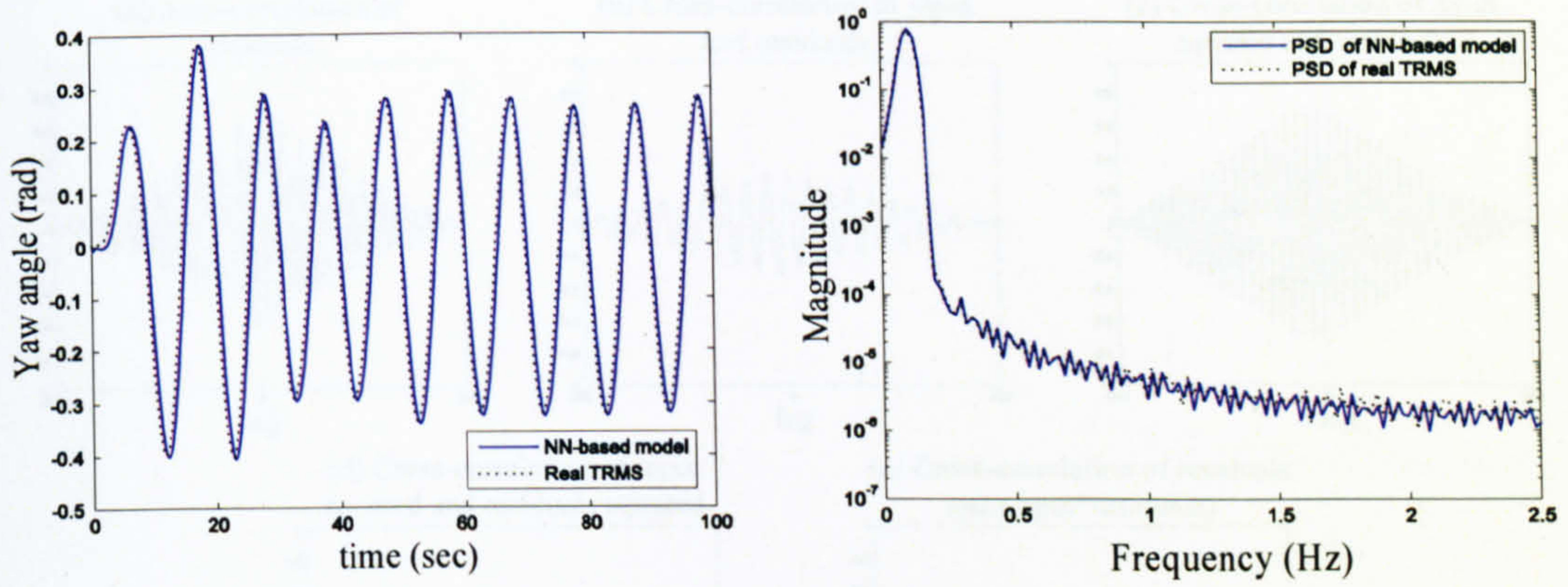


Figure 3.12 Model response with the GDX training algorithm

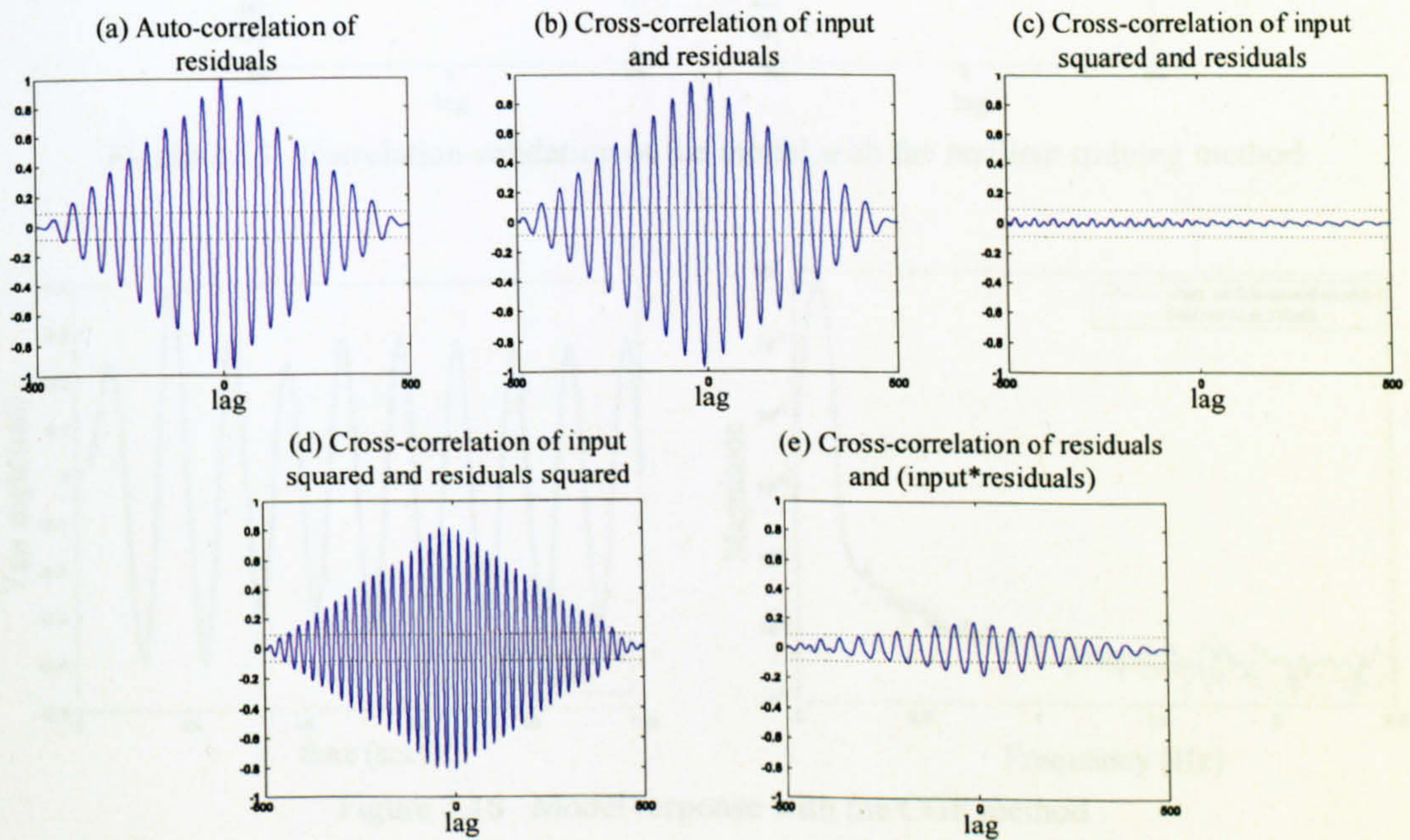


Figure 3.13 Correlation validation of the model with the GDX learning method

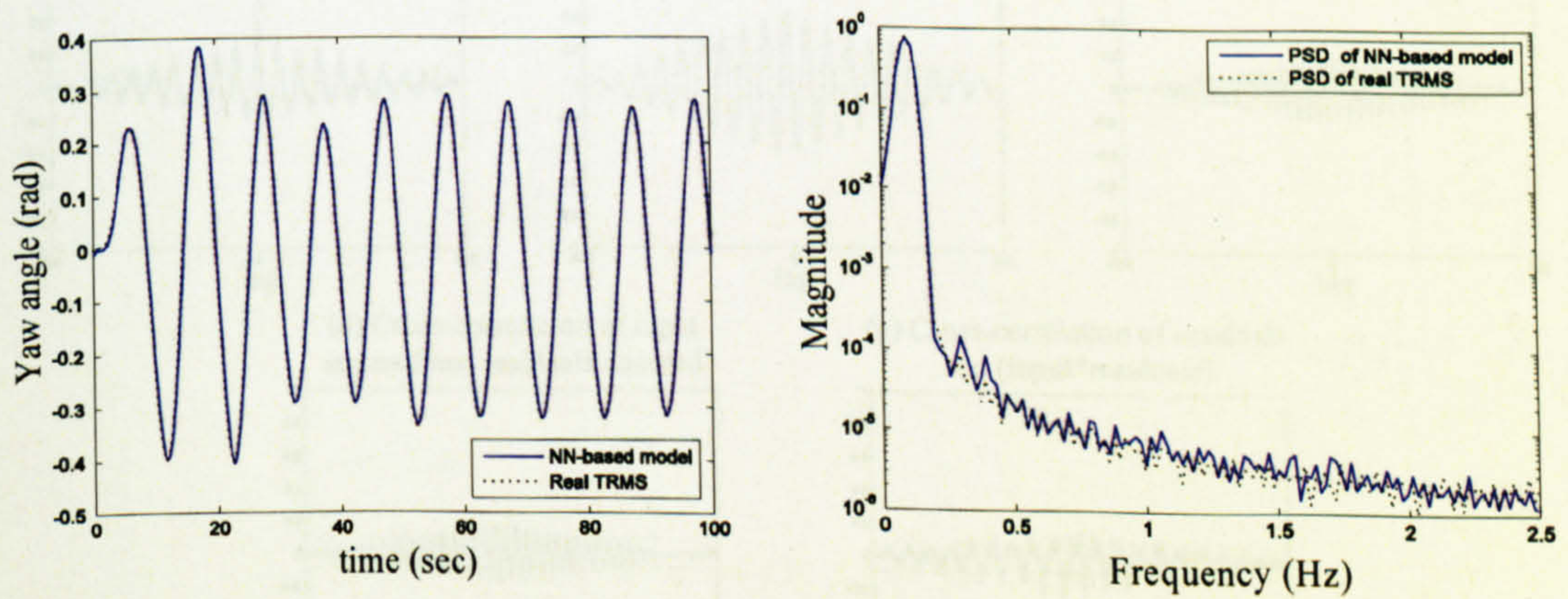


Figure 3.14 Model response with the resilient method

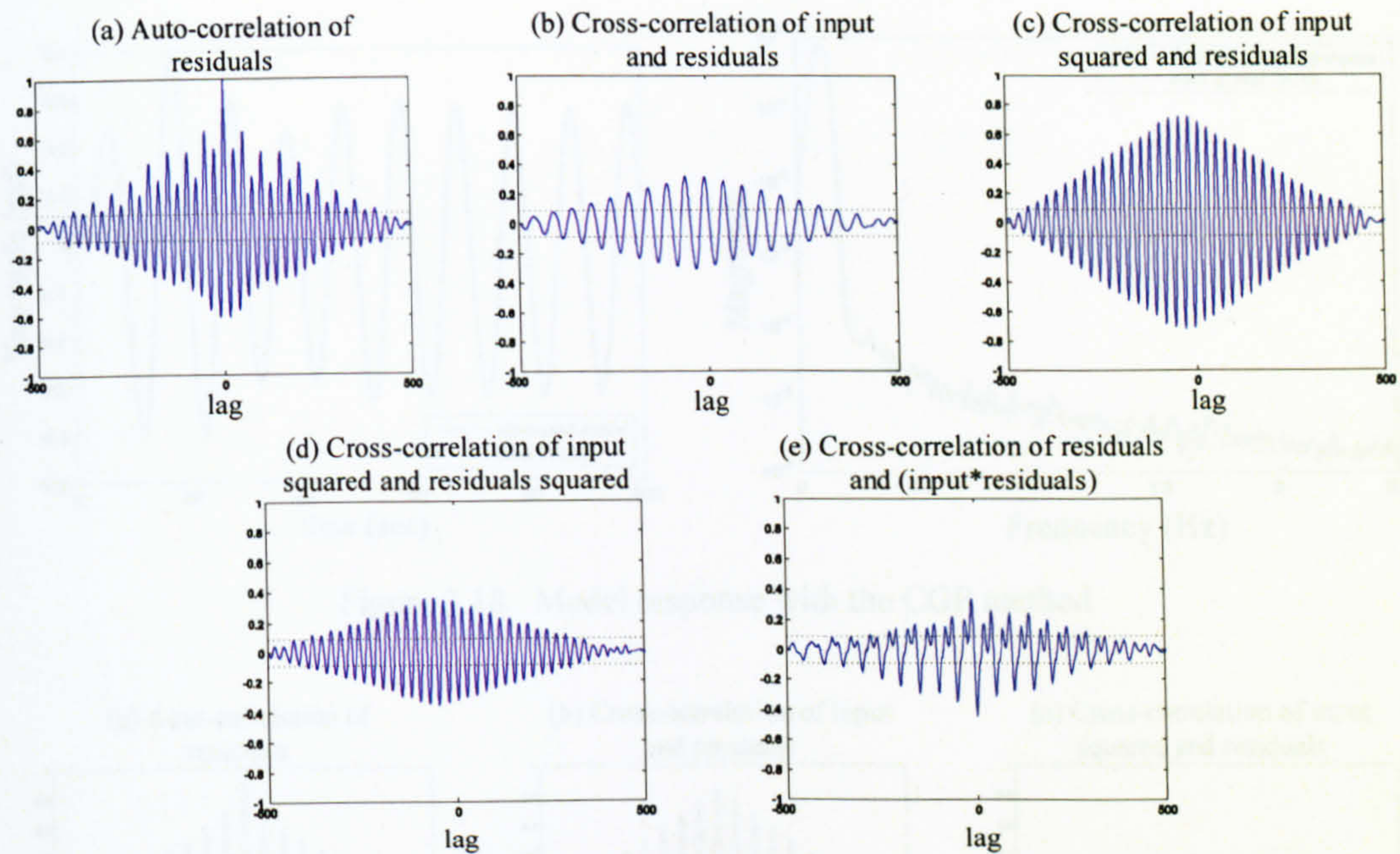


Figure 3.15 Correlation validation of the model with the resilient training method

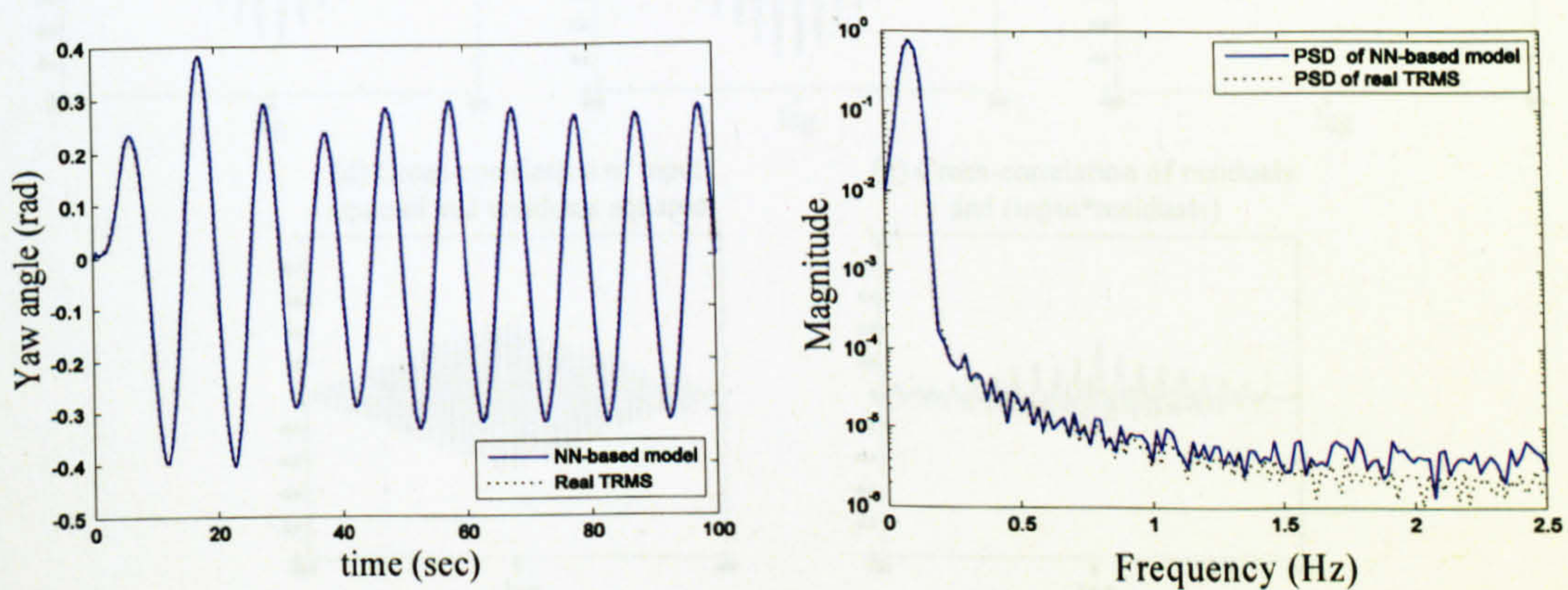


Figure 3.16 Model response with the CGF method

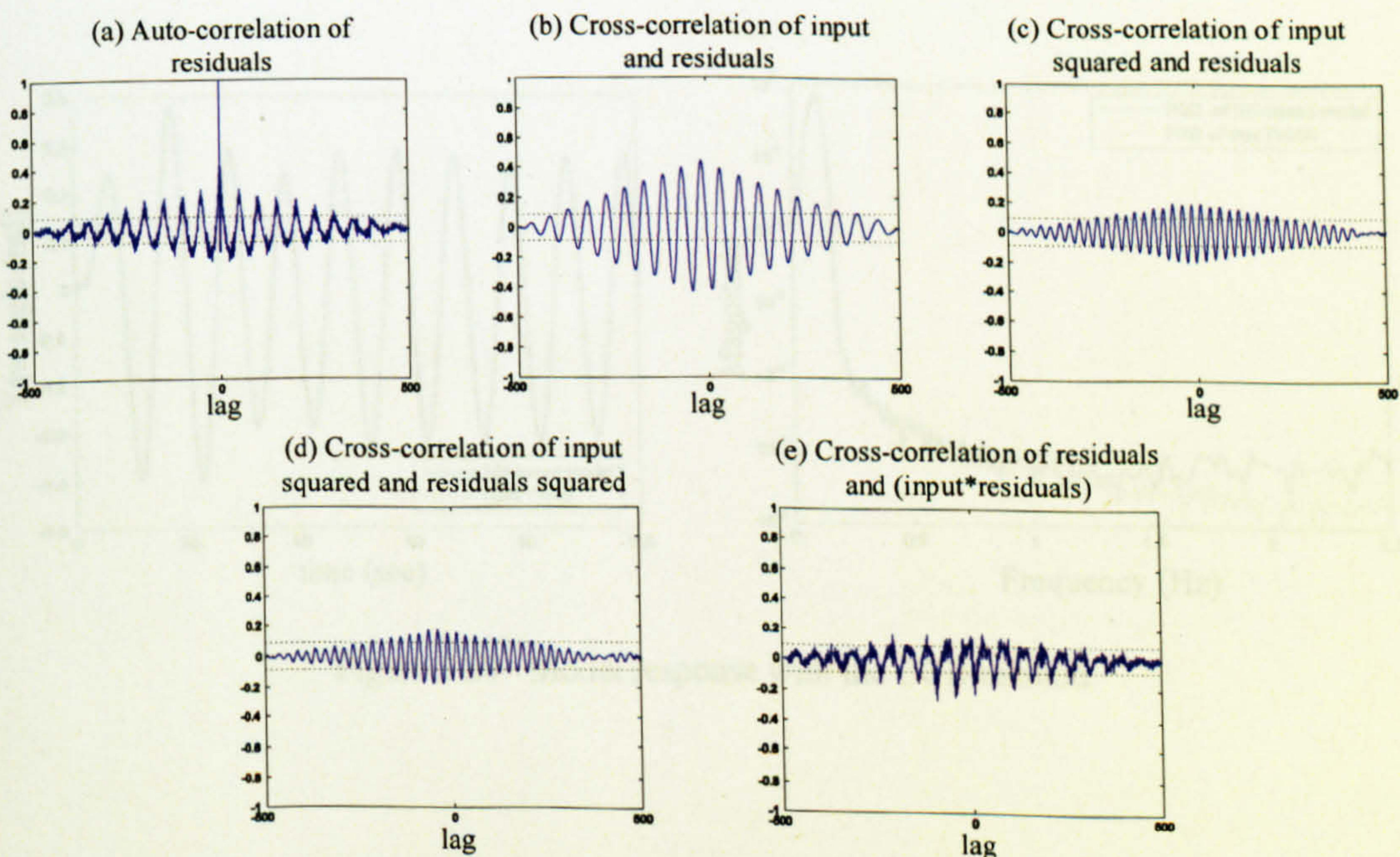


Figure 3.17 Correlation validation of the model with the CGF training method

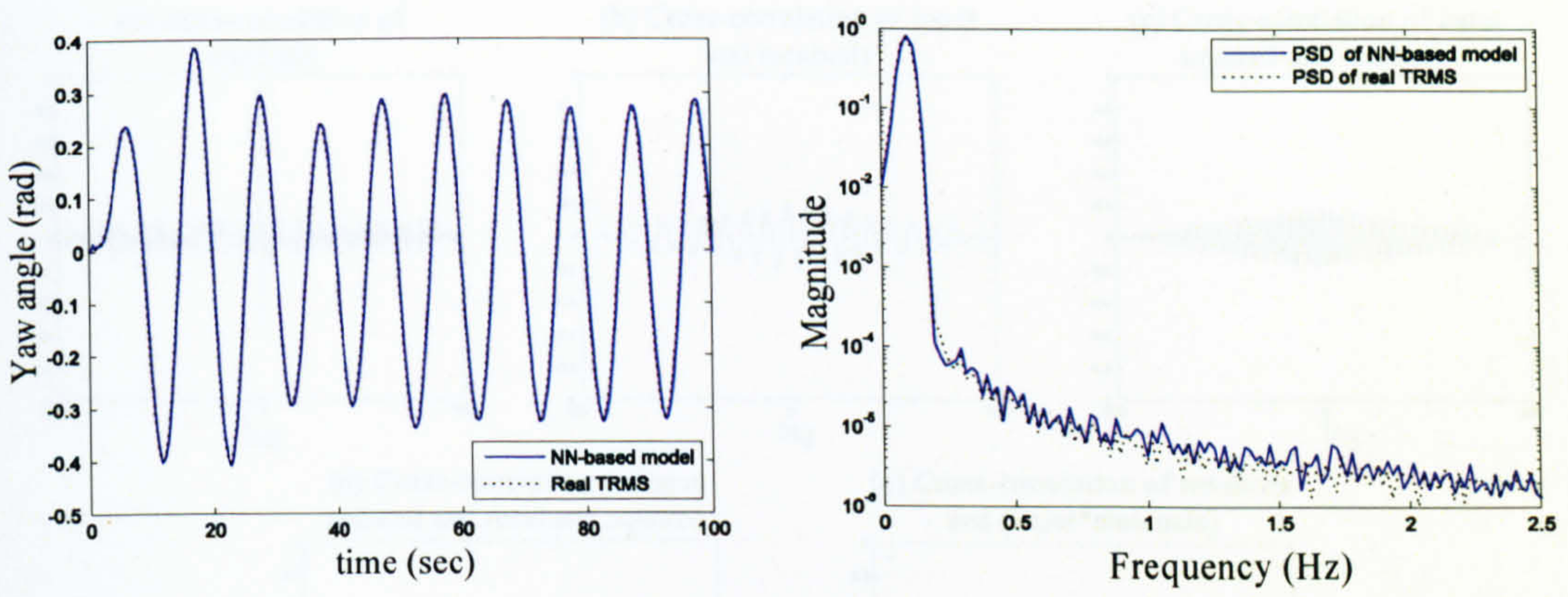


Figure 3.18 Model response with the CGP method

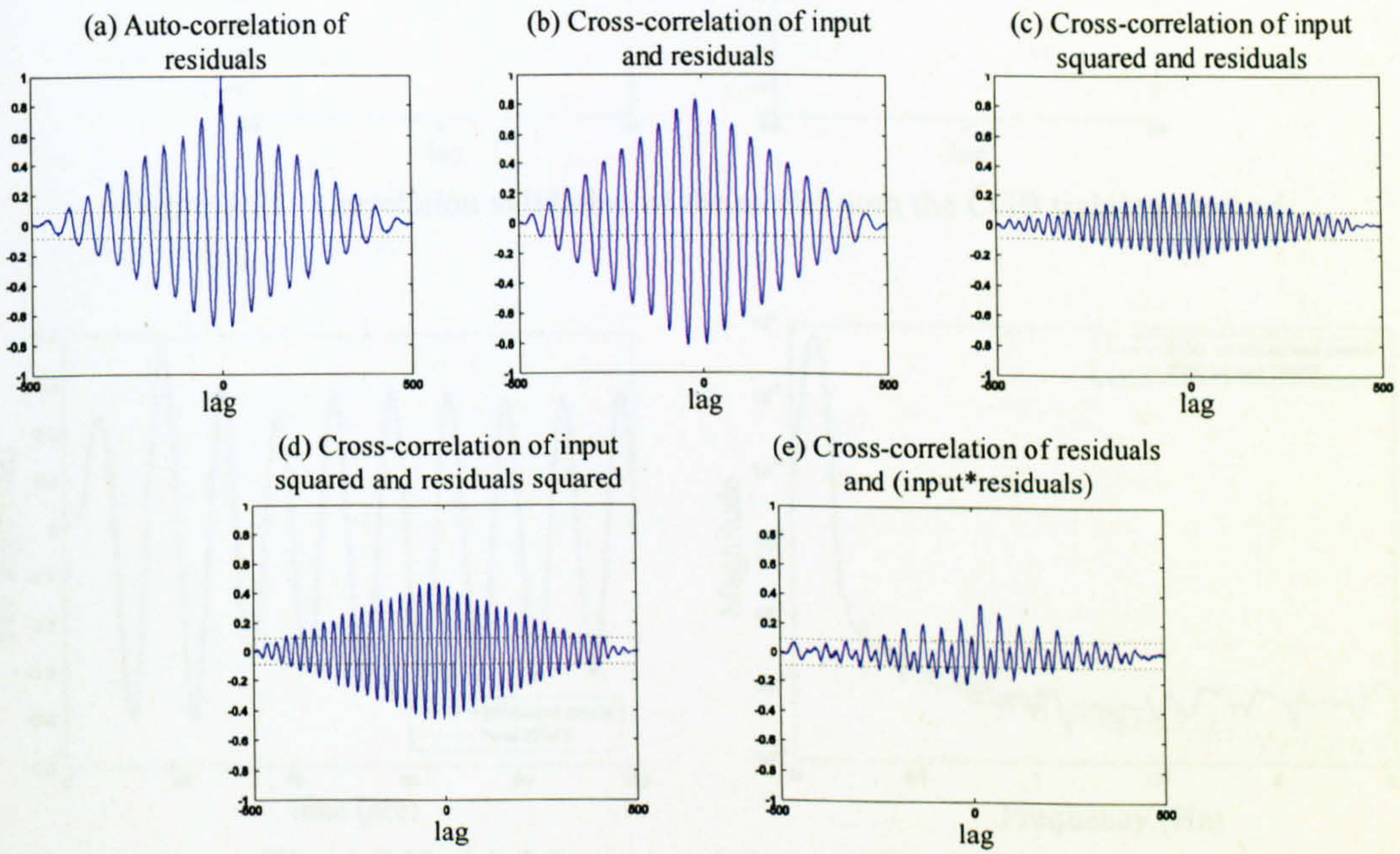


Figure 3.19 Correlation validation of the model with the CGP training method

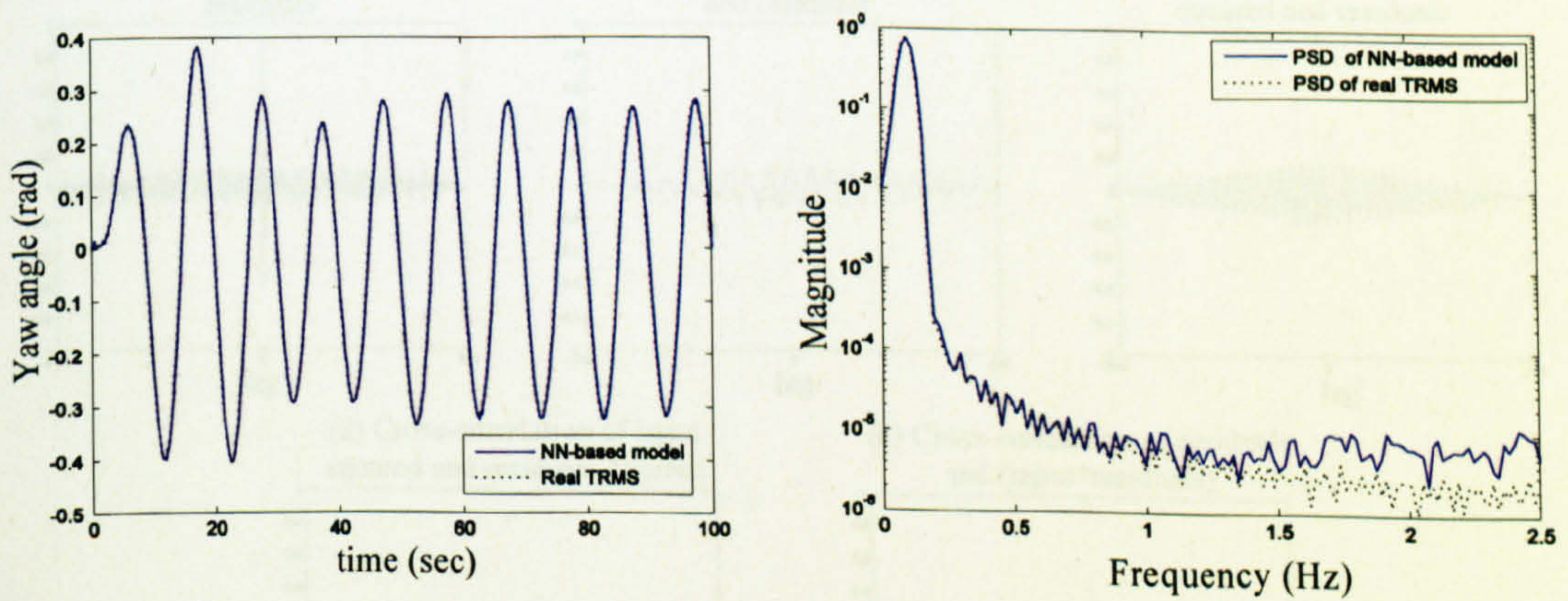


Figure 3.20 Model response with the CGB method

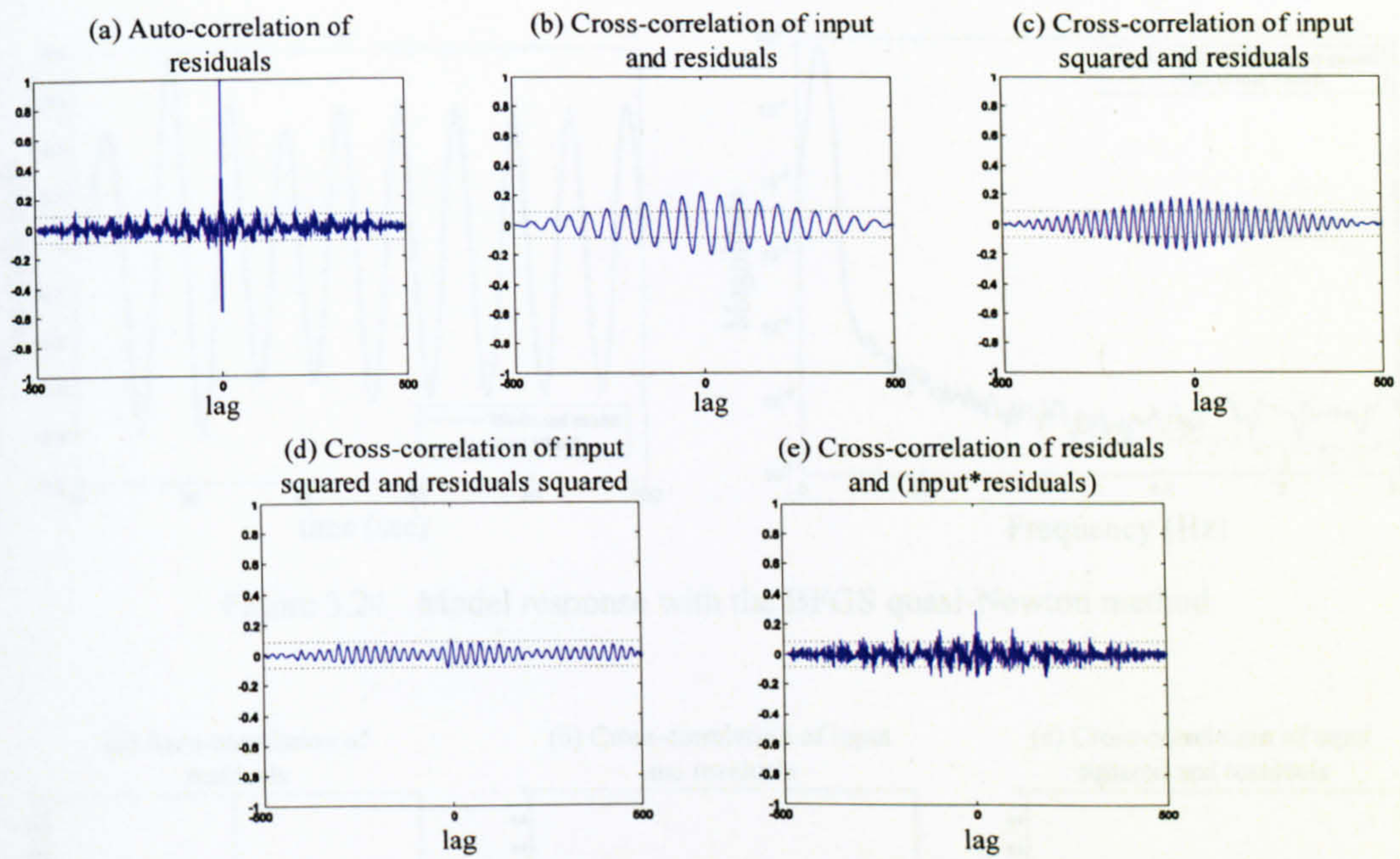


Figure 3.21 Correlation validation of the model with the CGB training method

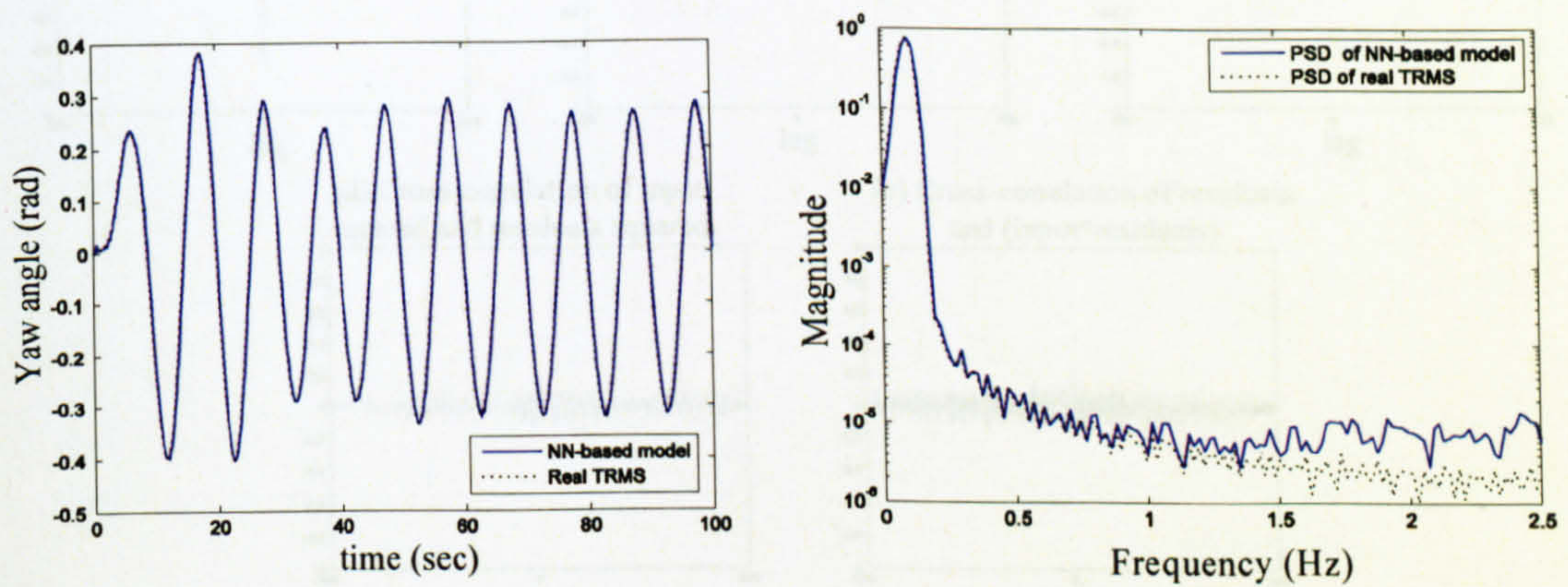


Figure 3.22 Model response with the SCG training method

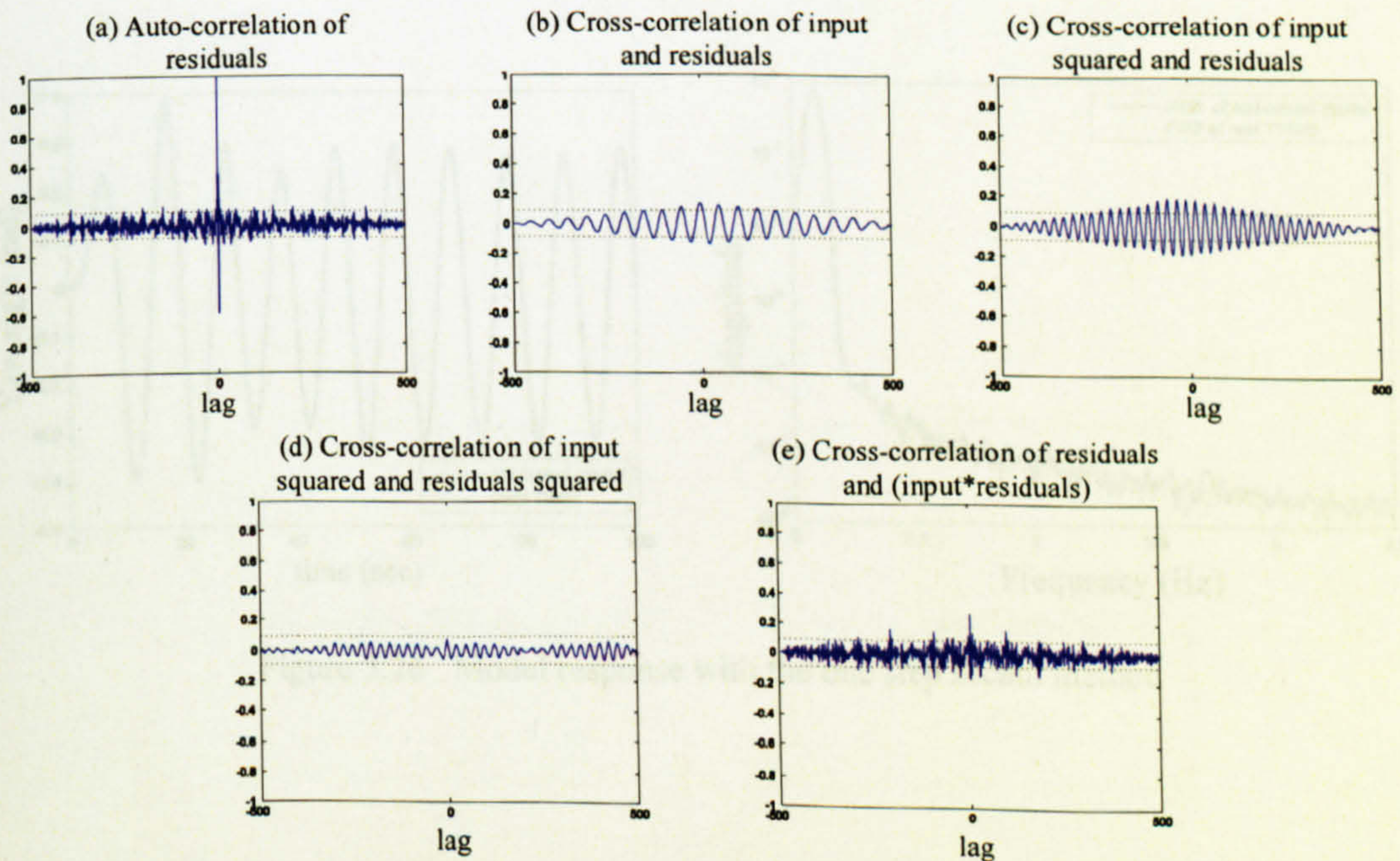


Figure 3.23 Correlation validation of the model with the SCG training method

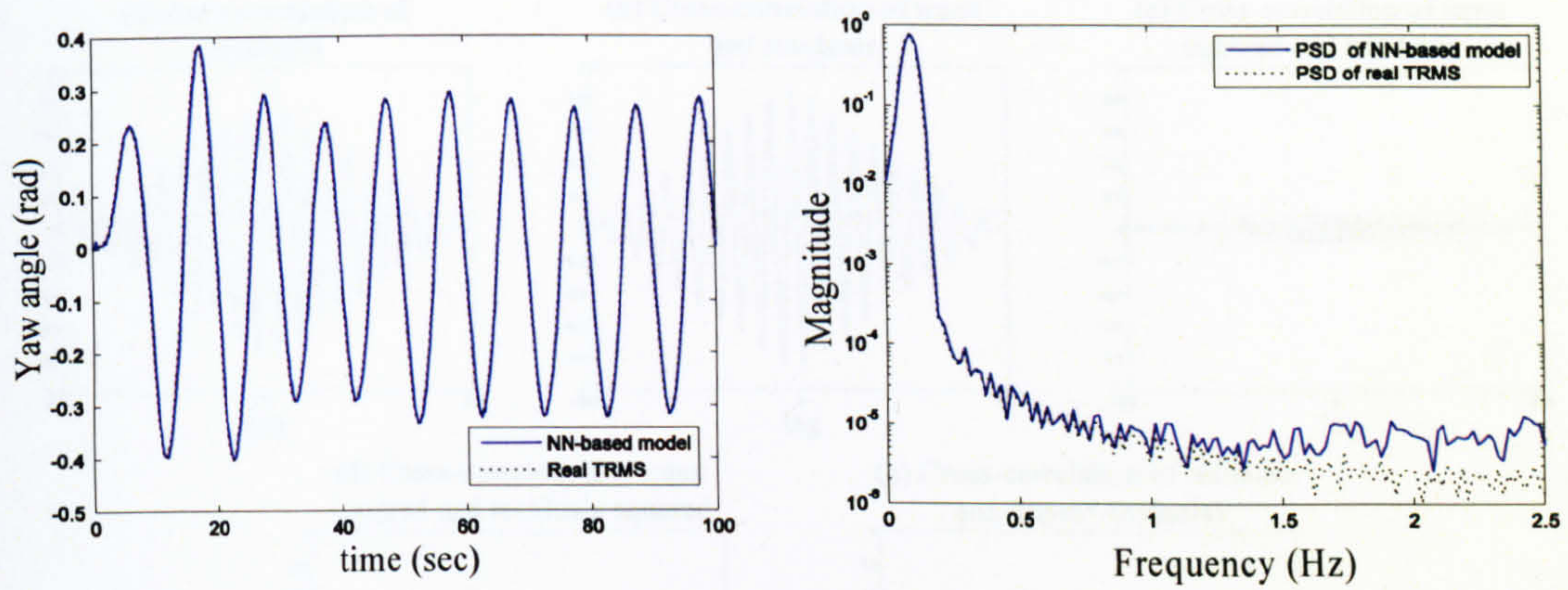


Figure 3.24 Model response with the BFGS quasi-Newton method

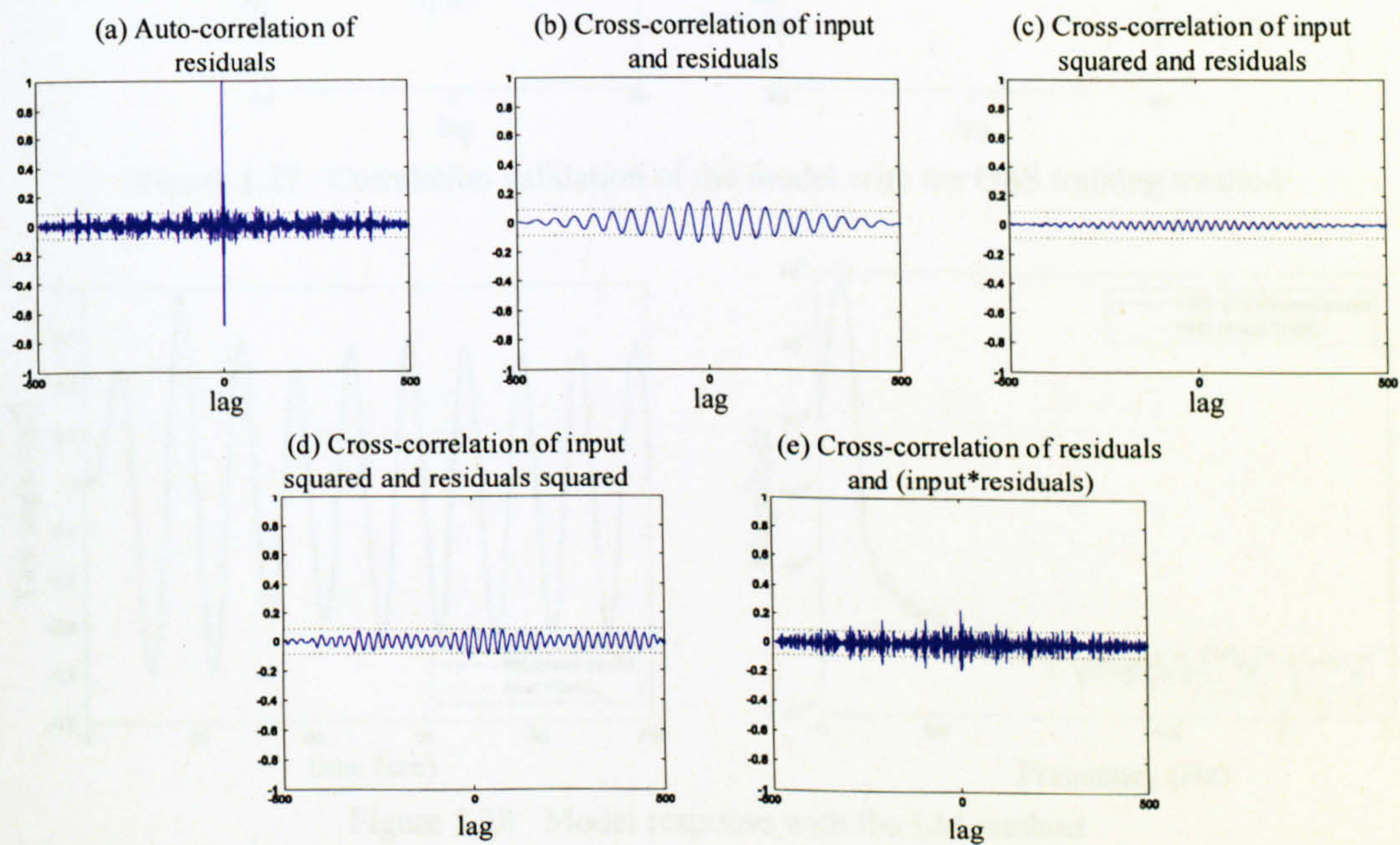


Figure 3.25 Correlation validation of the model with the BFGS learning method

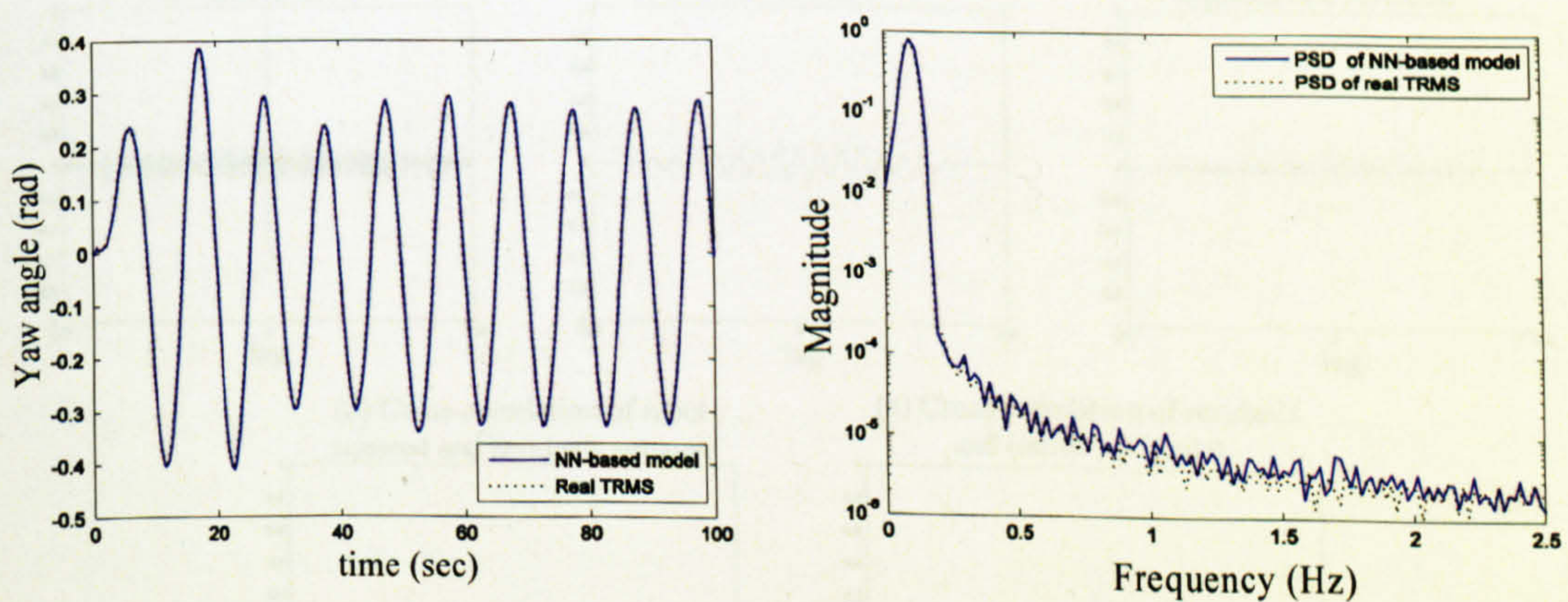


Figure 3.26 Model response with the one step secant method

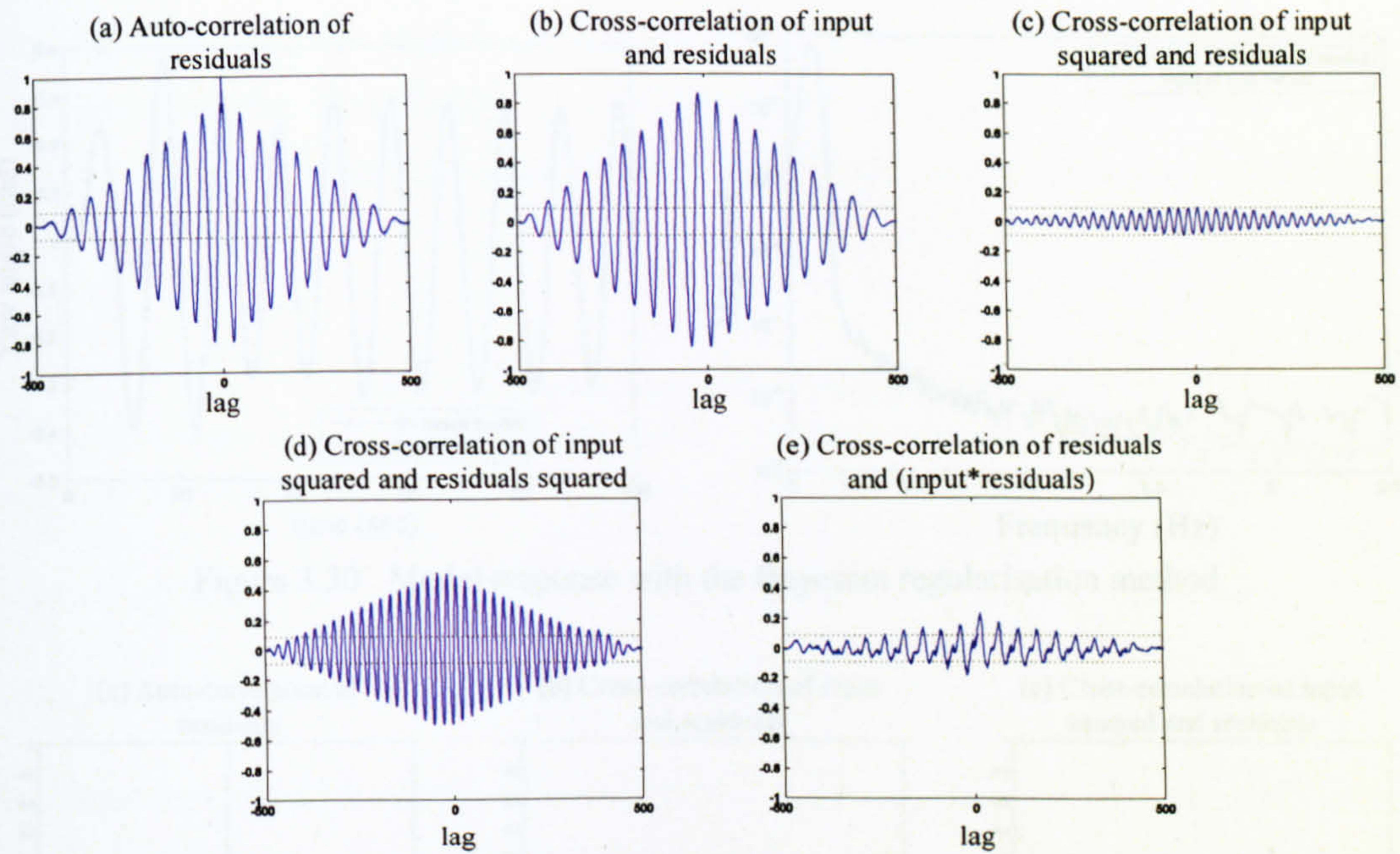


Figure 3.27 Correlation validation of the model with the OSS training method

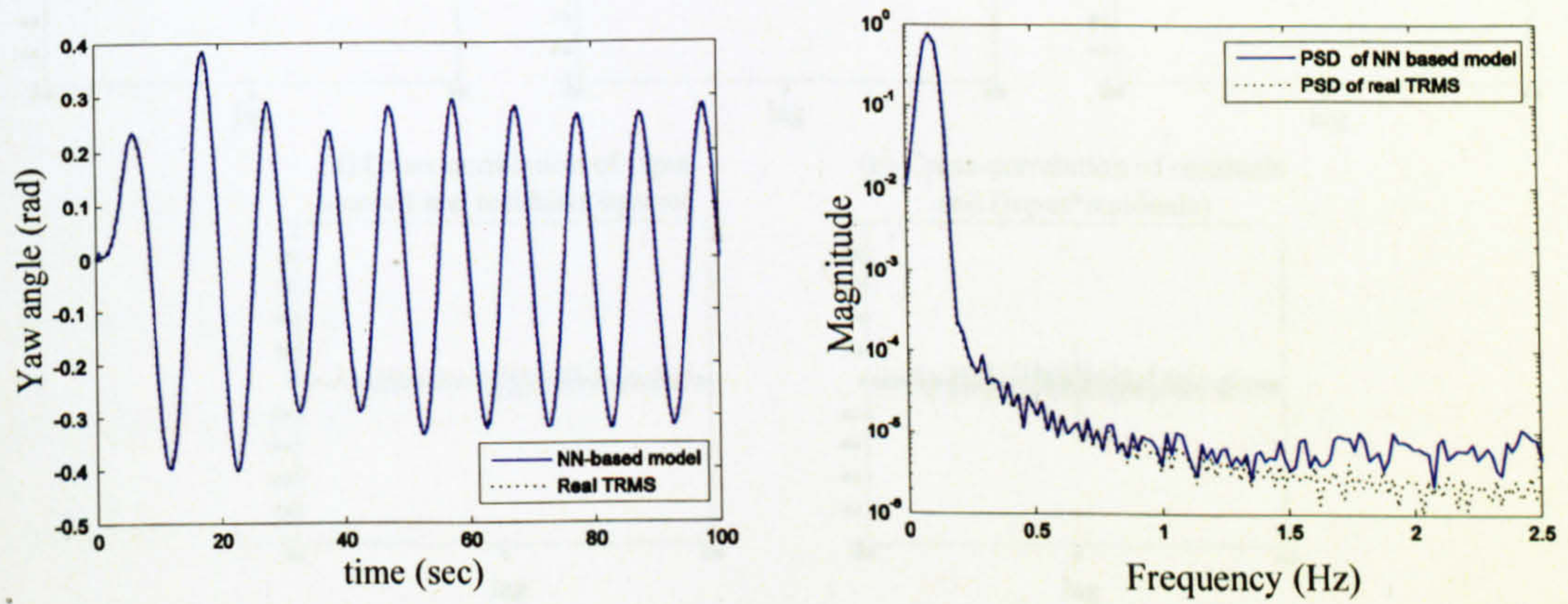


Figure 3.28 Model response with the LM method

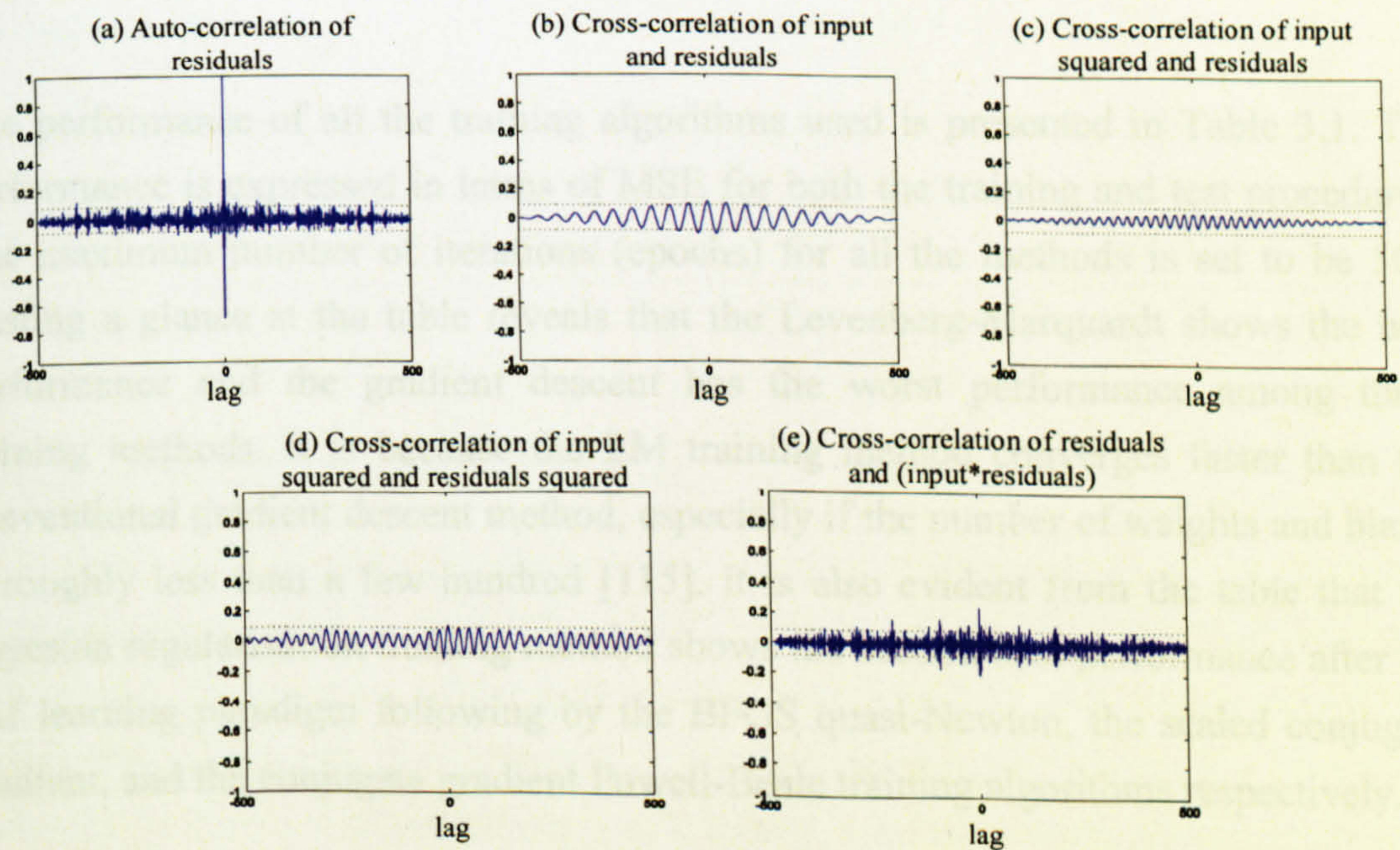


Figure 3.29 Correlation validation of the model with the LM training algorithm

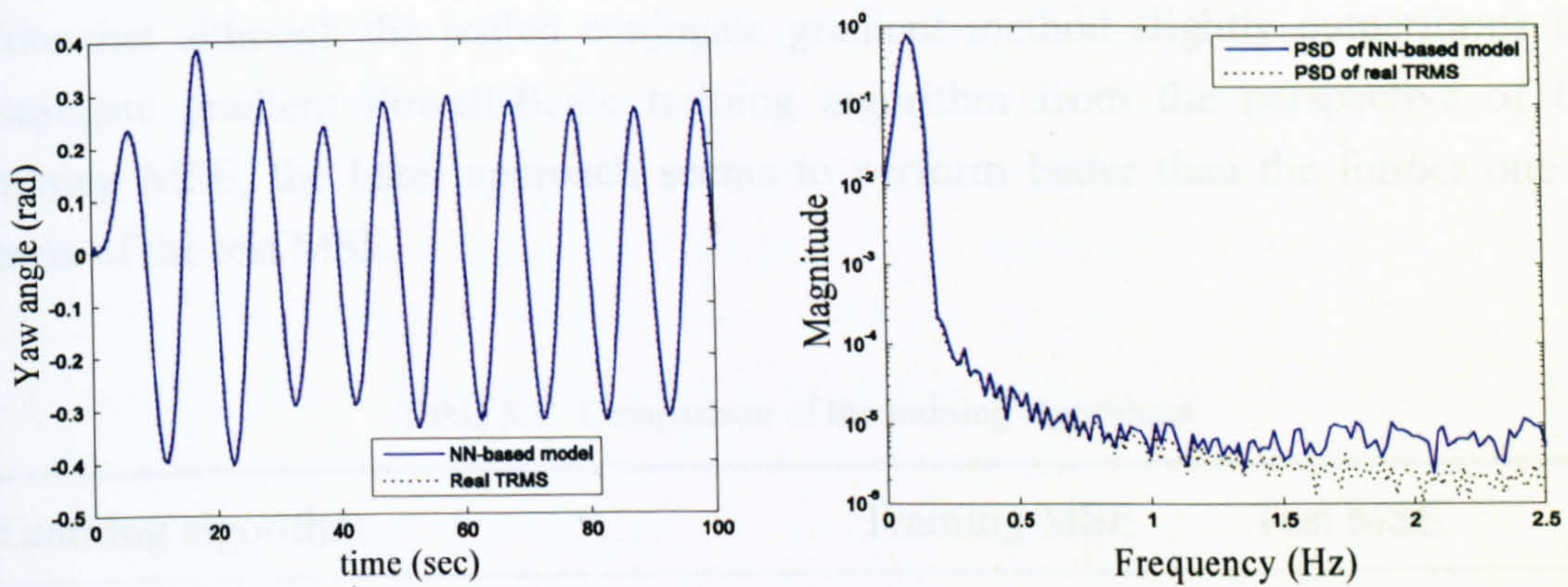


Figure 3.30 Model response with the Bayesian regularisation method

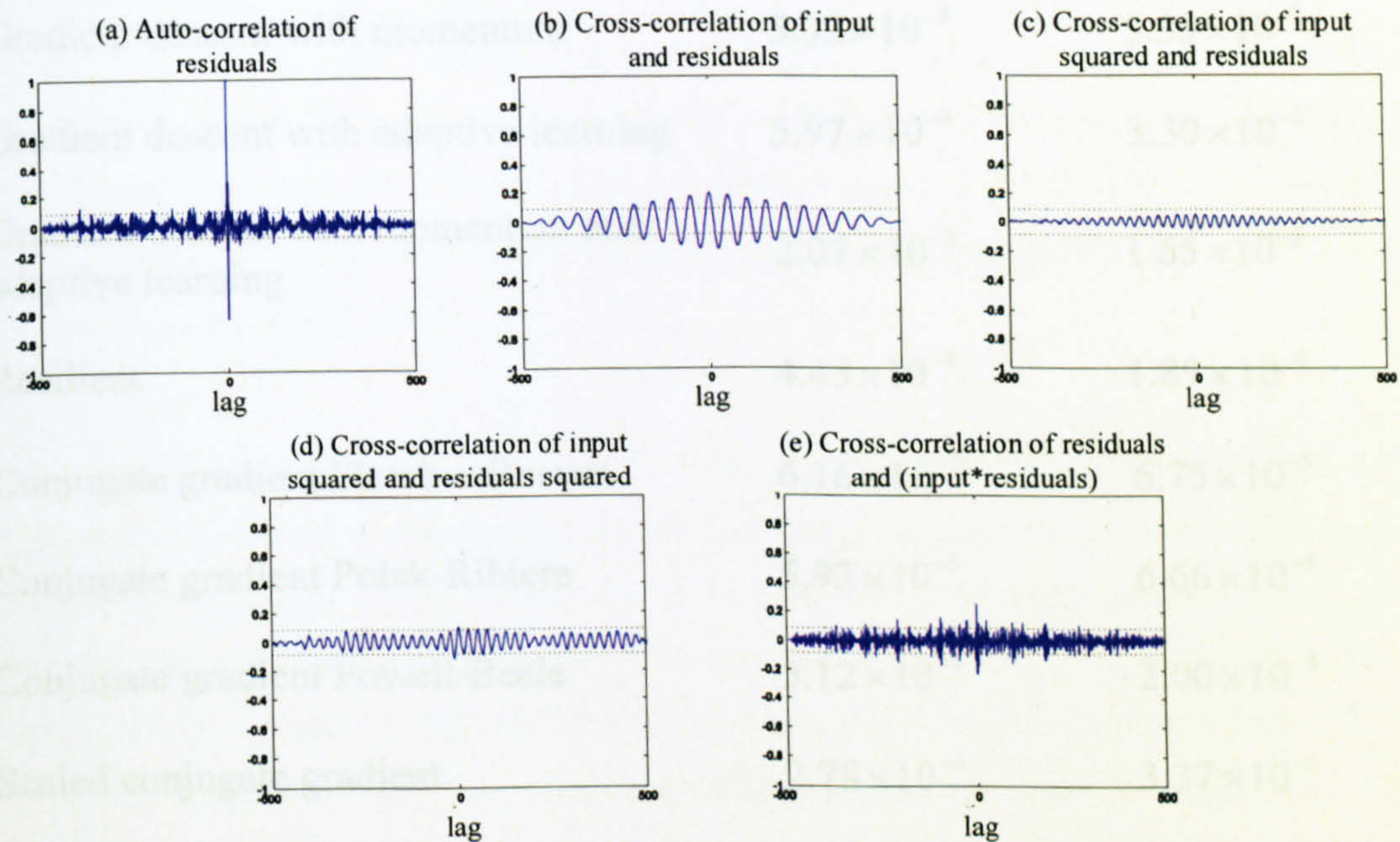


Figure 3.31 Correlation validation of the model with the BR training method

The performance of all the training algorithms used is presented in Table 3.1. The performance is expressed in terms of MSE for both the training and test procedures. The maximum number of iterations (epochs) for all the methods is set to be 500. Casting a glance at the table reveals that the Levenberg-Marquardt shows the best performance and the gradient descent has the worst performance among these training methods. It is because the LM training method converges faster than the conventional gradient descent method, especially if the number of weights and biases is roughly less than a few hundred [115]. It is also evident from the table that the Bayesian regularisation training method shows the second best performance after the LM learning paradigm following by the BFGS quasi-Newton, the scaled conjugate gradient, and the conjugate gradient Powell-Beale training algorithms respectively.

Note that although the scaled conjugate gradient method slightly outperforms the conjugate gradient Powell-Beale training algorithm from the perspective of the training MSE, the latter approach seems to perform better than the former one in terms of the test MSE.

Table 3.1 Comparison of the training algorithms

Learning algorithm	Training MSE	Test MSE
Gradient descent	1.13×10^{-2}	1.74×10^{-2}
Gradient descent with momentum	5.32×10^{-3}	5.35×10^{-3}
Gradient descent with adaptive learning	5.97×10^{-3}	3.30×10^{-3}
Gradient descent with momentum and adaptive learning	2.07×10^{-3}	1.65×10^{-3}
Resilient	4.45×10^{-4}	1.89×10^{-4}
Conjugate gradient Fletcher-Reeves	6.16×10^{-5}	6.75×10^{-5}
Conjugate gradient Polak-Ribiere	5.93×10^{-5}	6.66×10^{-5}
Conjugate gradient Powell-Beale	3.12×10^{-5}	2.00×10^{-5}
Scaled conjugate gradient	2.78×10^{-5}	3.37×10^{-5}
BFGS quasi-Newton	2.09×10^{-5}	1.71×10^{-5}
One step secant	1.47×10^{-4}	7.27×10^{-5}
Levenberg-Marquardt	1.57×10^{-5}	1.35×10^{-5}
Bayesian regularisation	1.94×10^{-5}	1.62×10^{-5}

3.4.2 The 1DOF Vertical Model

A feedforward MLP neural network with the same configuration as the horizontal model has been used to model the TRMS in terms of its 1DOF vertical dynamics. Again, the configuration of the network has been found to prevent any superfluous computational burden. In other words, a single-hidden-layer network has been found to be the best choice which consists of 6 inputs, 3 neurons in the hidden layer and 1 neuron in the output layer. The inputs are the main rotor voltage at the previous

instant, $V_v(t-1)$, the main rotor voltage at two samples before, $V_v(t-2)$, the main rotor voltage at three samples before, $V_v(t-3)$, the pitch angle of the beam at the previous time, $\alpha_v(t-1)$, the pitch angle of the beam at two samples before, $\alpha_v(t-2)$, and the pitch angle of the beam at three samples before, $\alpha_v(t-3)$. Figure 3.32 shows the structure of the network of the 1DOF vertical model. Similar to the previous case, the activation functions used in the hidden layer and the output layer are logarithmic sigmoid and pure linear respectively. The numbers of patterns used for the training and test processes are 10700 and 500 sets respectively.

Various training methods have been used to train the constructed network. The trained NNs have been tested with several sets of data that are not used in the training procedure. However, for the sake of brevity, only a single test result has been shown for each of the training approaches. Figures 3.33 to 3.58 show the results of the overlapping test, corresponding PSD, and correlation validation of the network with the training algorithms.

It is clear from Figures 3.33 to 3.40 that none of the gradient descent learning algorithm has shown a satisfactory result and therefore cannot be considered to be an adequate model. According to Figures 3.41 and 3.42, the resilient back-propagation learning method provides a better performance compared to the gradient descent algorithms, but some of the correlation tests are out of the confidence interval. Four types of conjugate gradient learning algorithms have been used to model the TRMS in terms of its 1DOF vertical dynamics (see Figures 3.43 to 3.50); however, only two of them, the Fletcher-Reeves and the Powell-Beale, show acceptable results.

Figures 3.51 to 3.54 depict the results related to the quasi-Newton learning algorithms. The BFGS learning algorithm can satisfactorily model the system; however, the performance of the one step secant learning approach to model the TRMS in terms of its 1DOF vertical dynamics is not acceptable. Figures 3.55 and 3.56 show the overlapping, PSD and correlation tests of the network trained using the LM training algorithm. The LM learning algorithm outperforms its rivals in terms of both the MSE and the correlation tests and can be considered the best learning approach to model the 1DOF vertical TRMS. The response of the network model trained by the Bayesian regularisation algorithm is found in Figures 2.57 and 2.58. The result seems to be adequate in terms of all the criteria, i.e. overlapping, PSD and correlation tests.

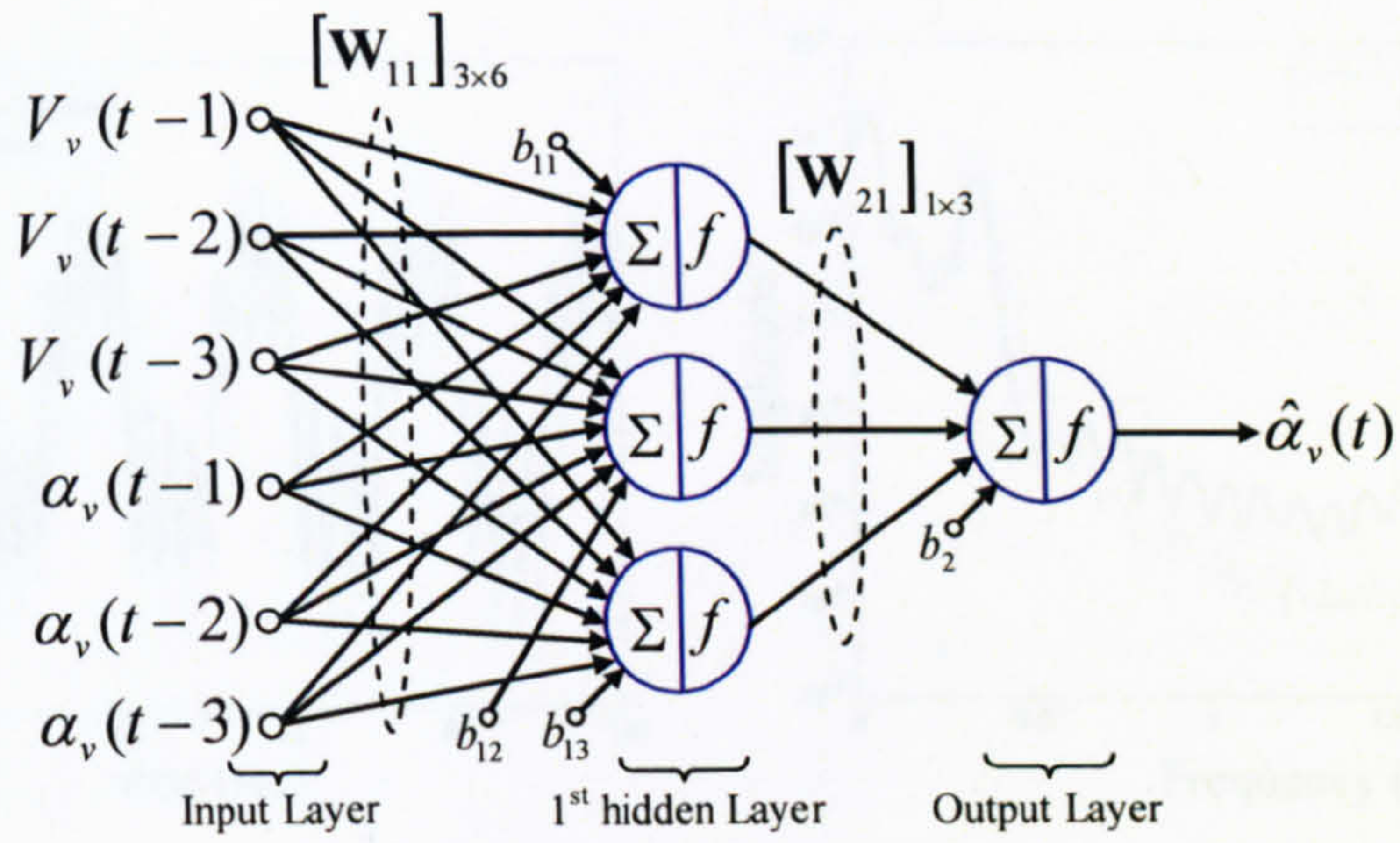


Figure 3.32 The structure of the NN-based model in terms of its 1DOF vertical dynamics

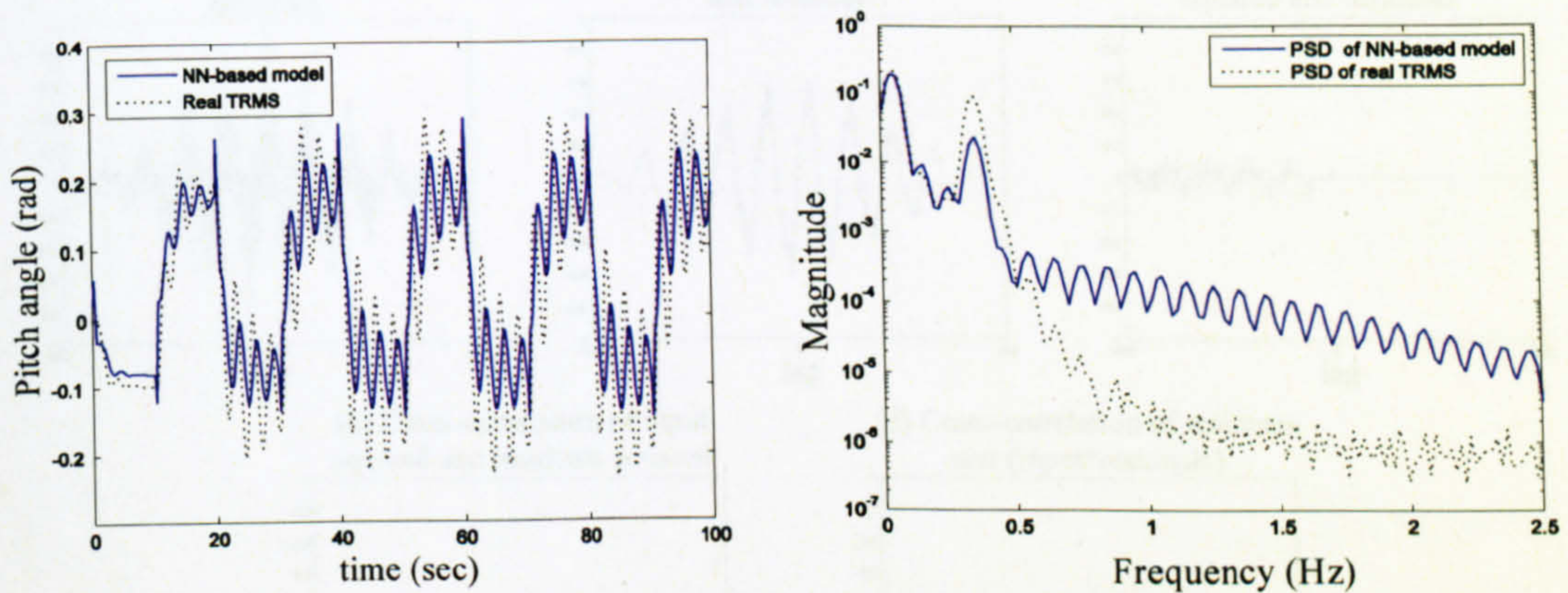


Figure 3.33 Model response with the gradient descent method

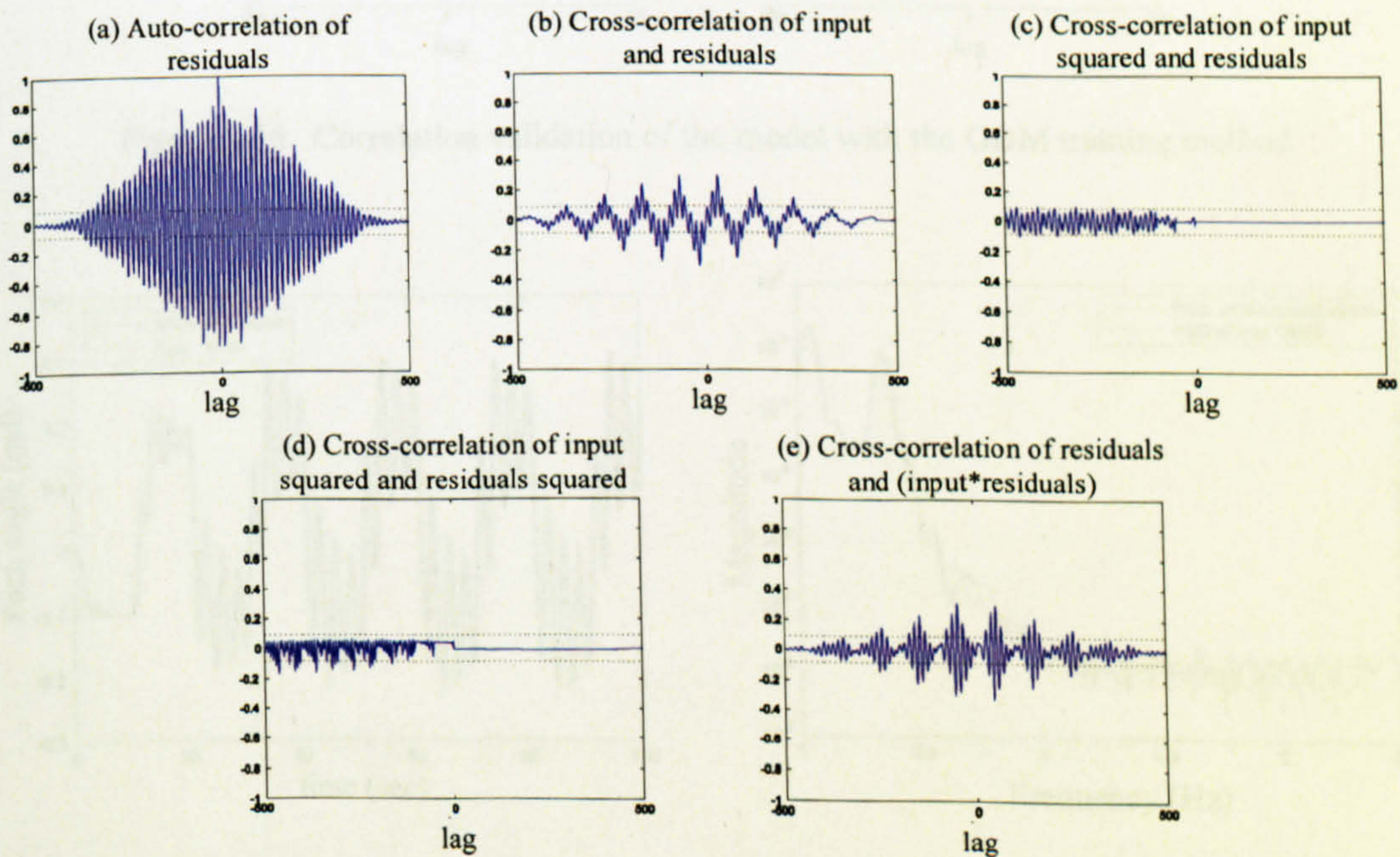


Figure 3.34 Correlation validation of the model with the GD training method

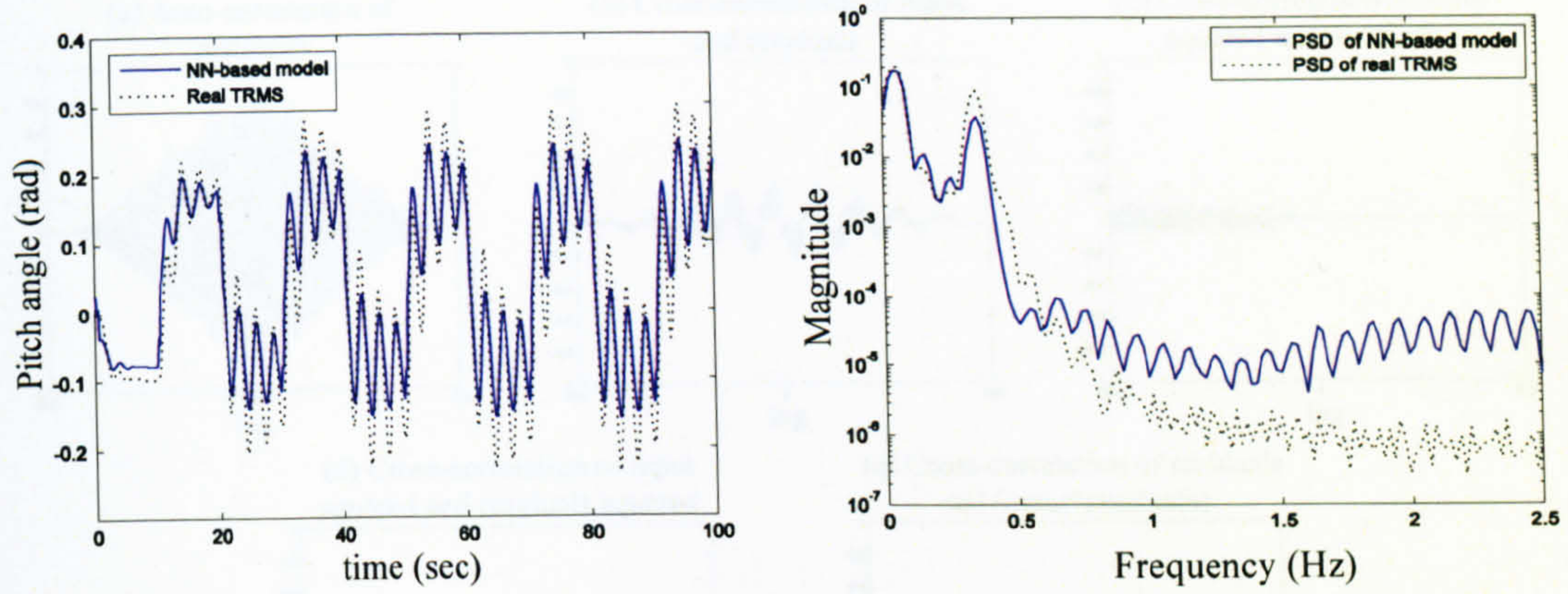


Figure 3.35 Model response with the GDM method

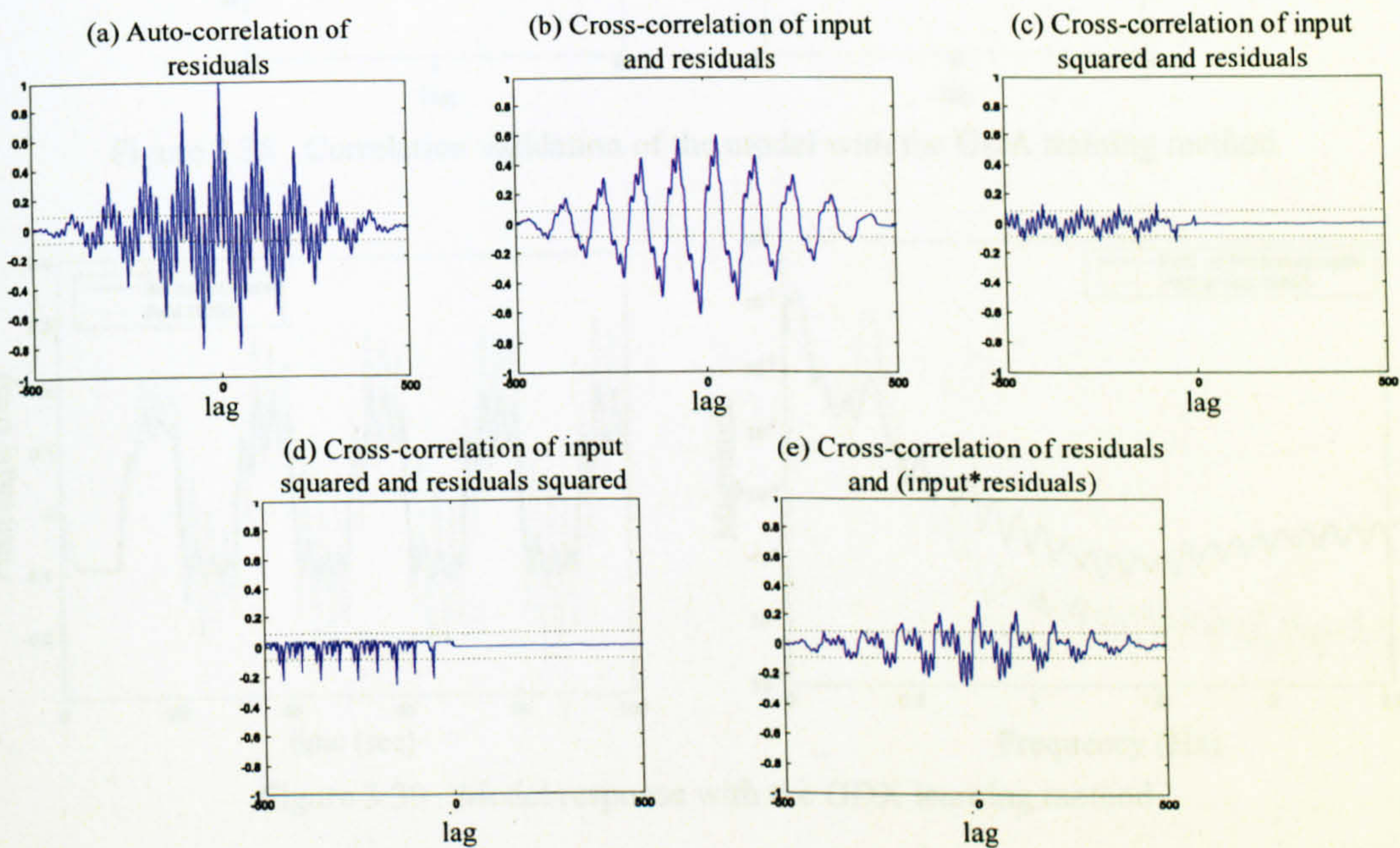


Figure 3.36 Correlation validation of the model with the GDM training method

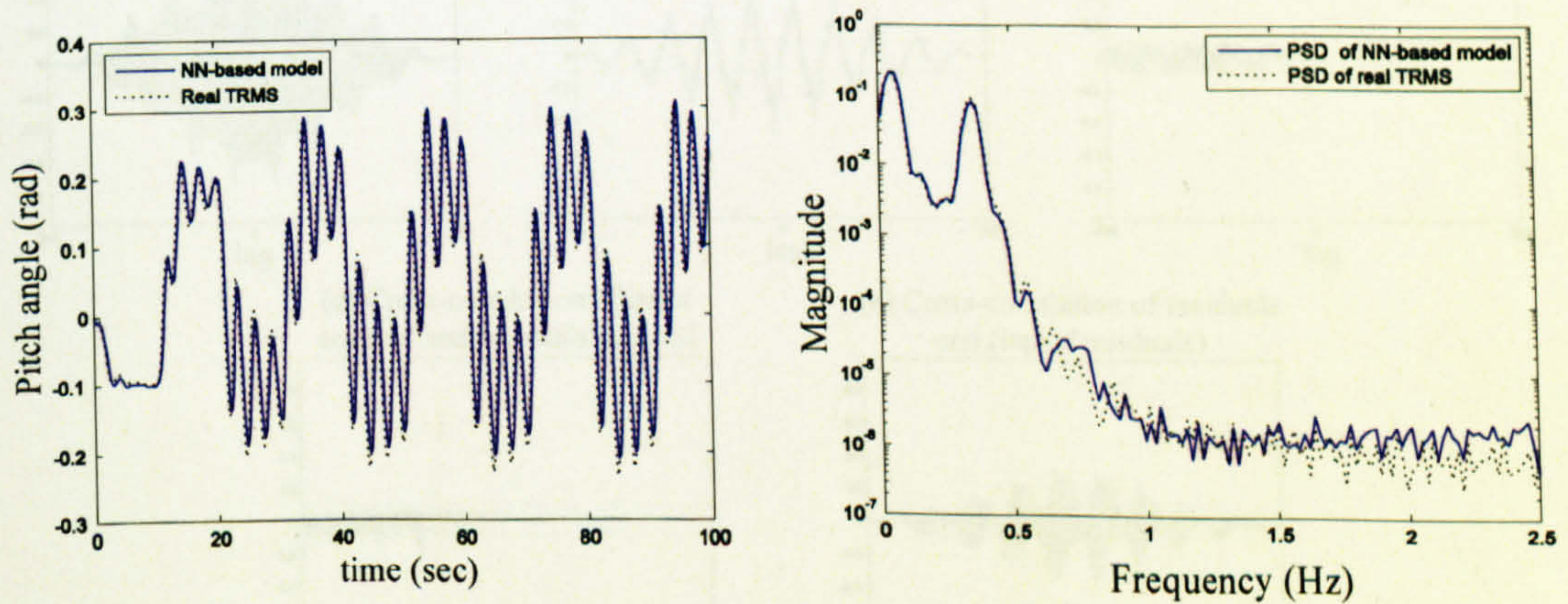


Figure 3.37 Model response with the GDA learning algorithm

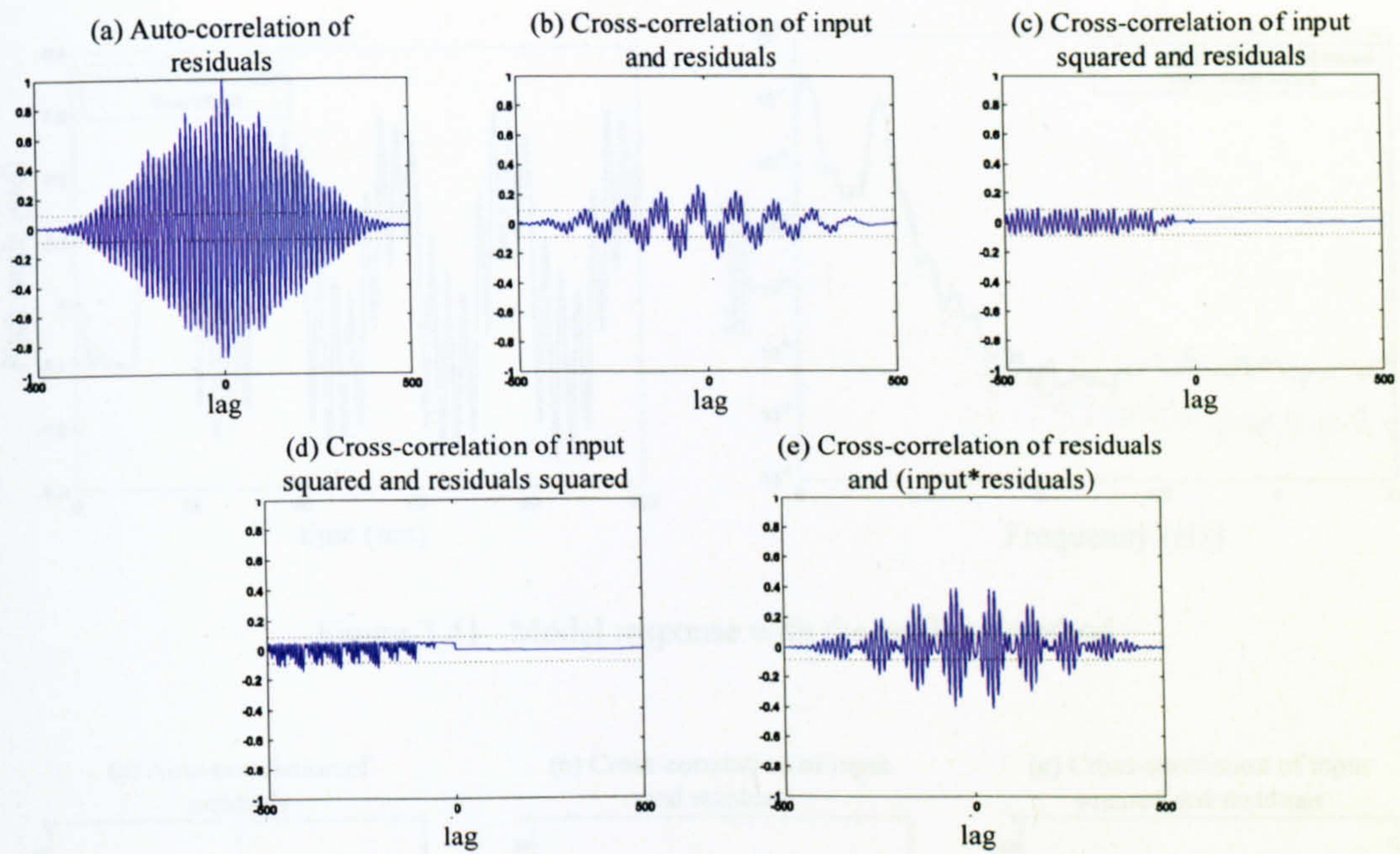


Figure 3.38 Correlation validation of the model with the GDA training method

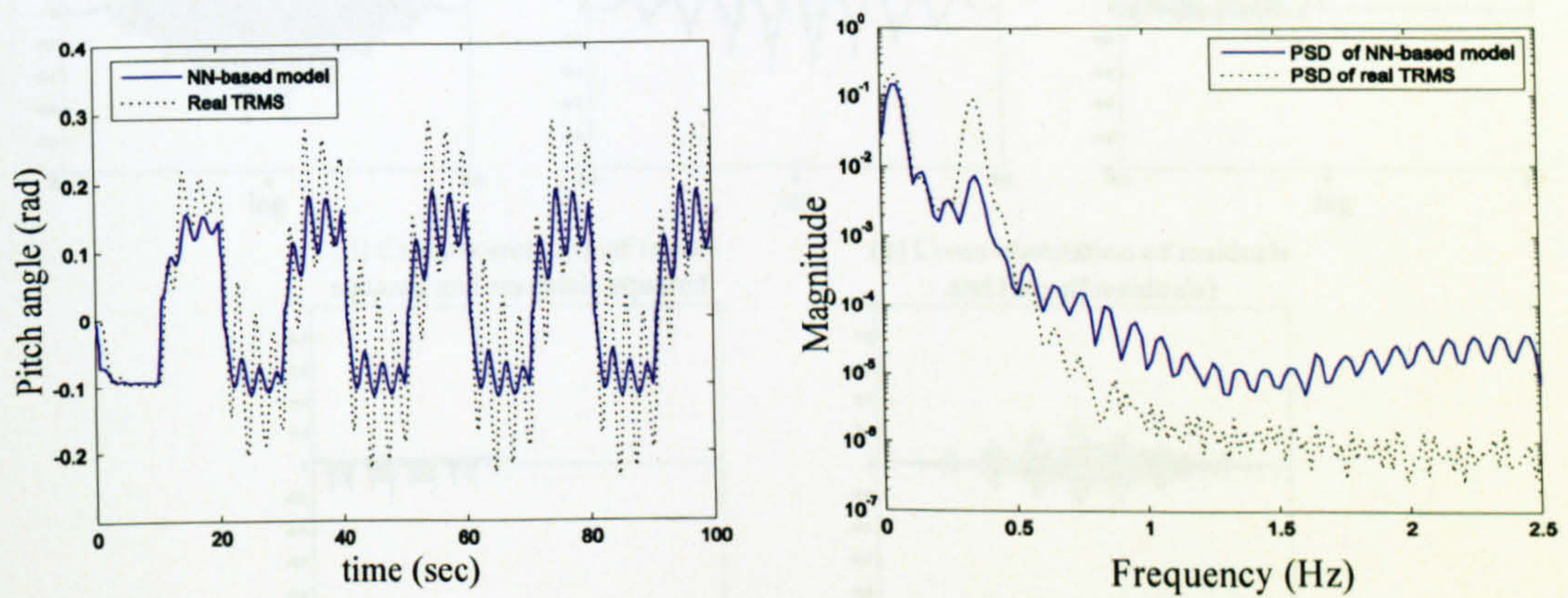


Figure 3.39 Model response with the GDX learning method

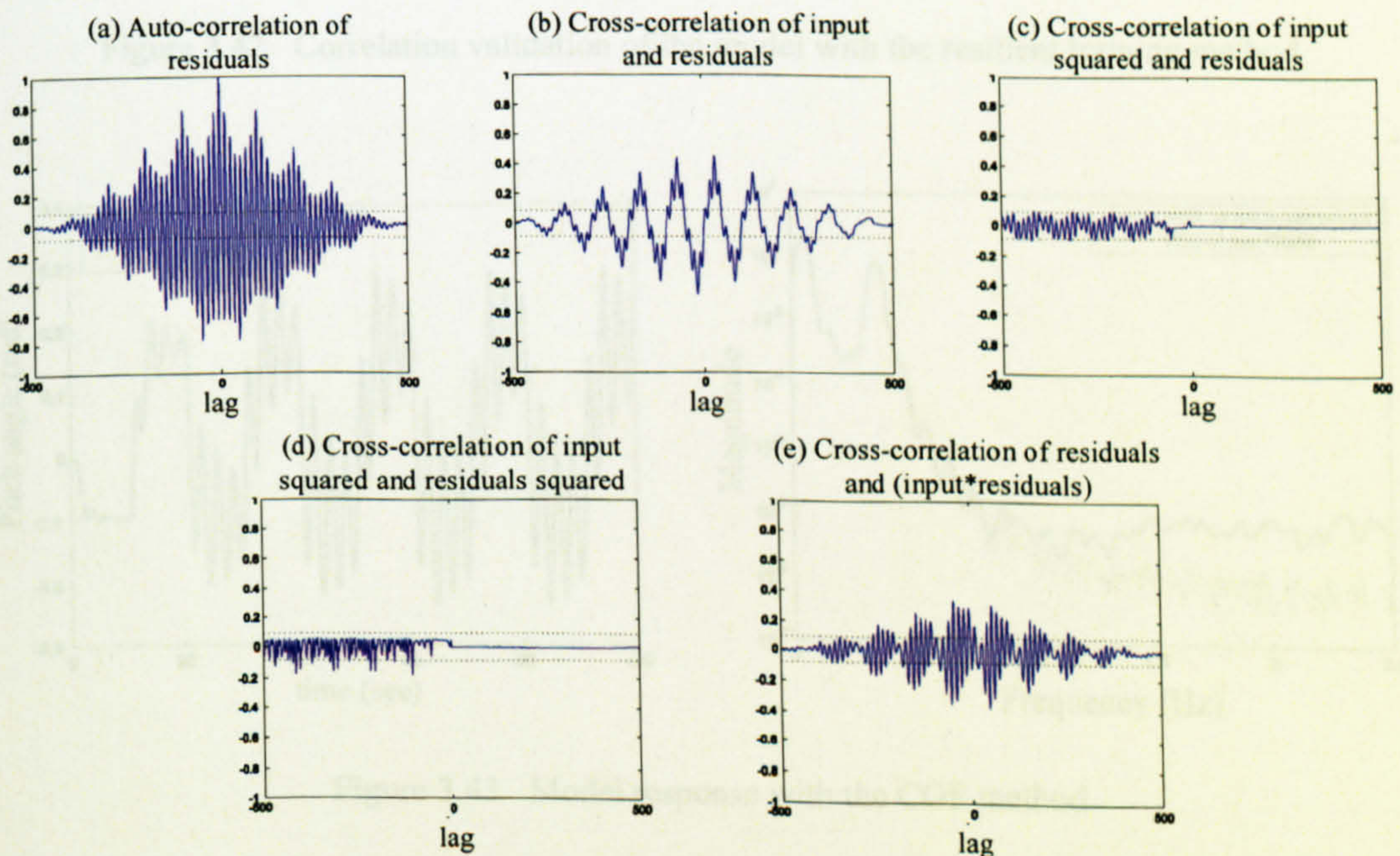


Figure 3.40 Correlation validation of the model with the GDX learning method

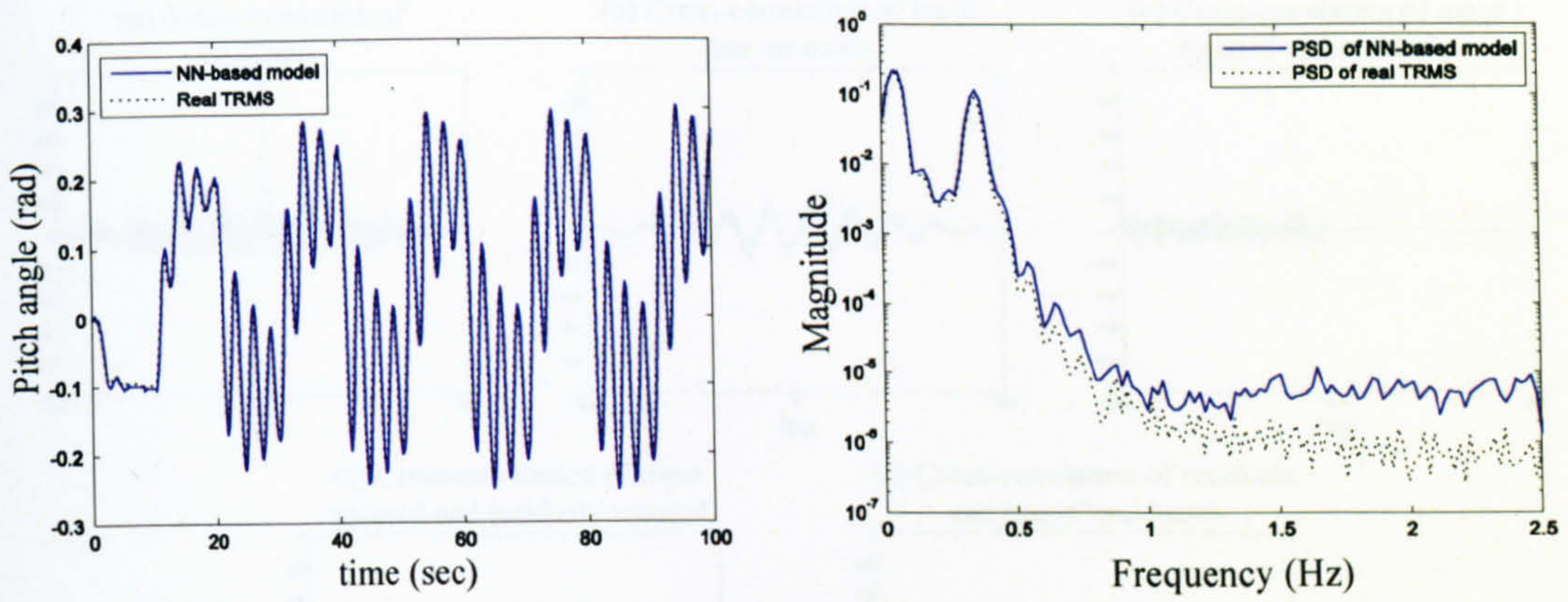


Figure 3.41 Model response with the resilient method

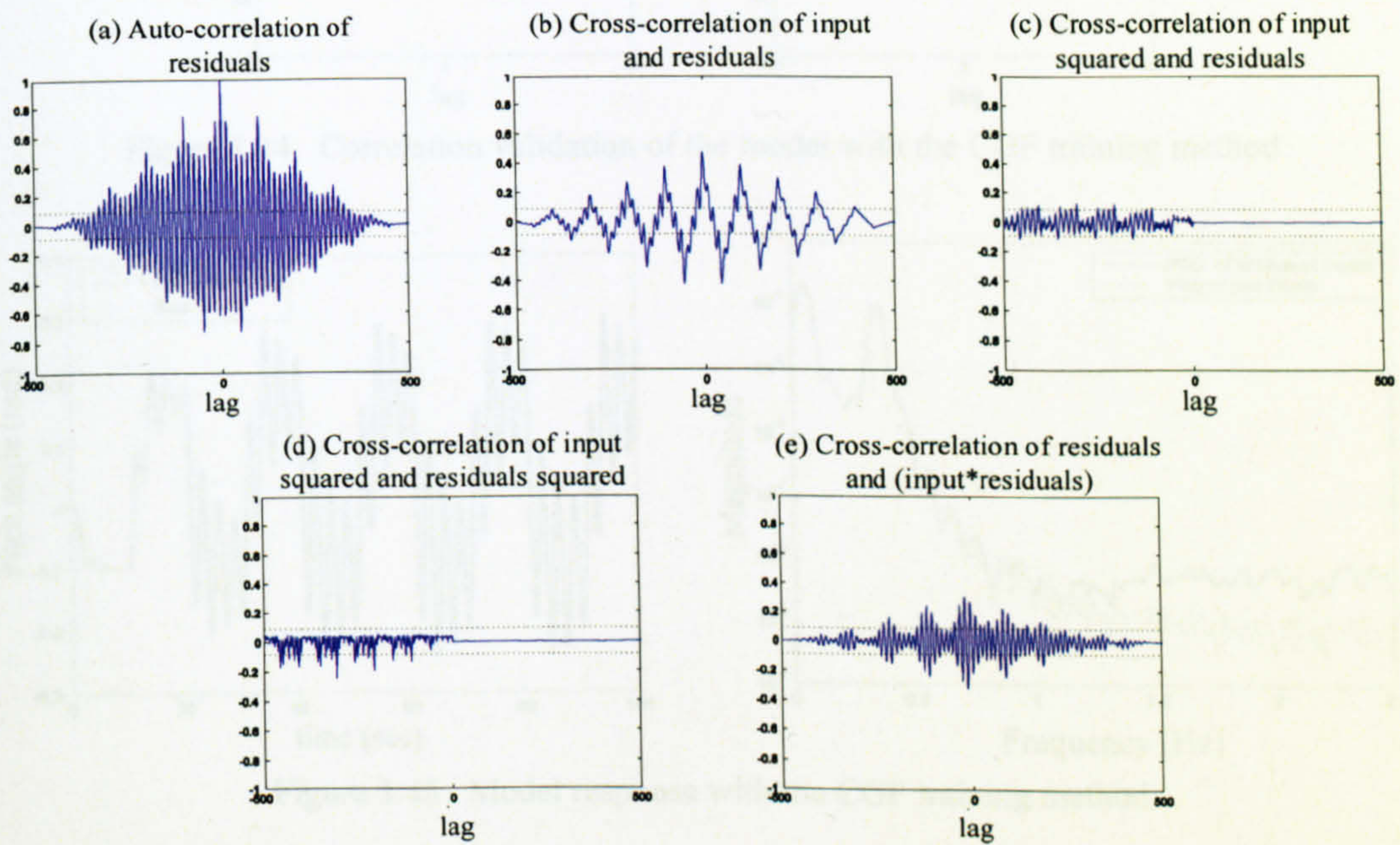


Figure 3.42 Correlation validation of the model with the resilient training method

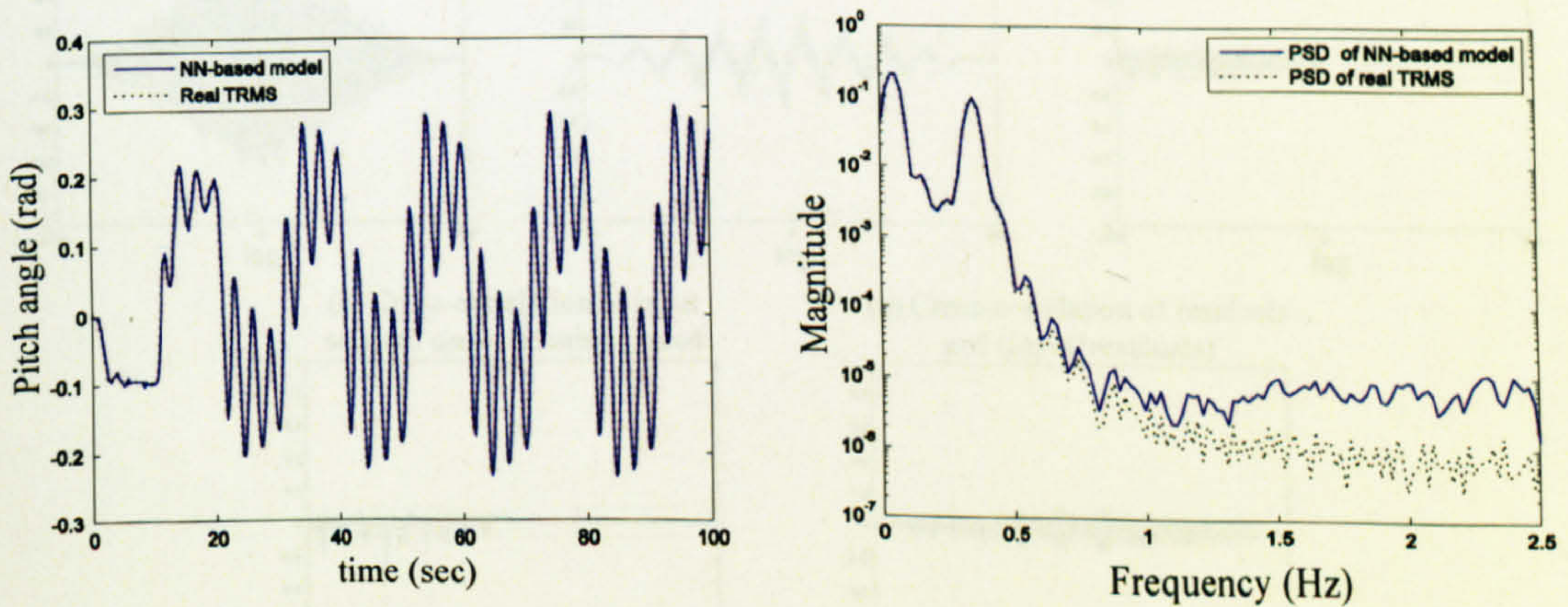


Figure 3.43 Model response with the CGF method

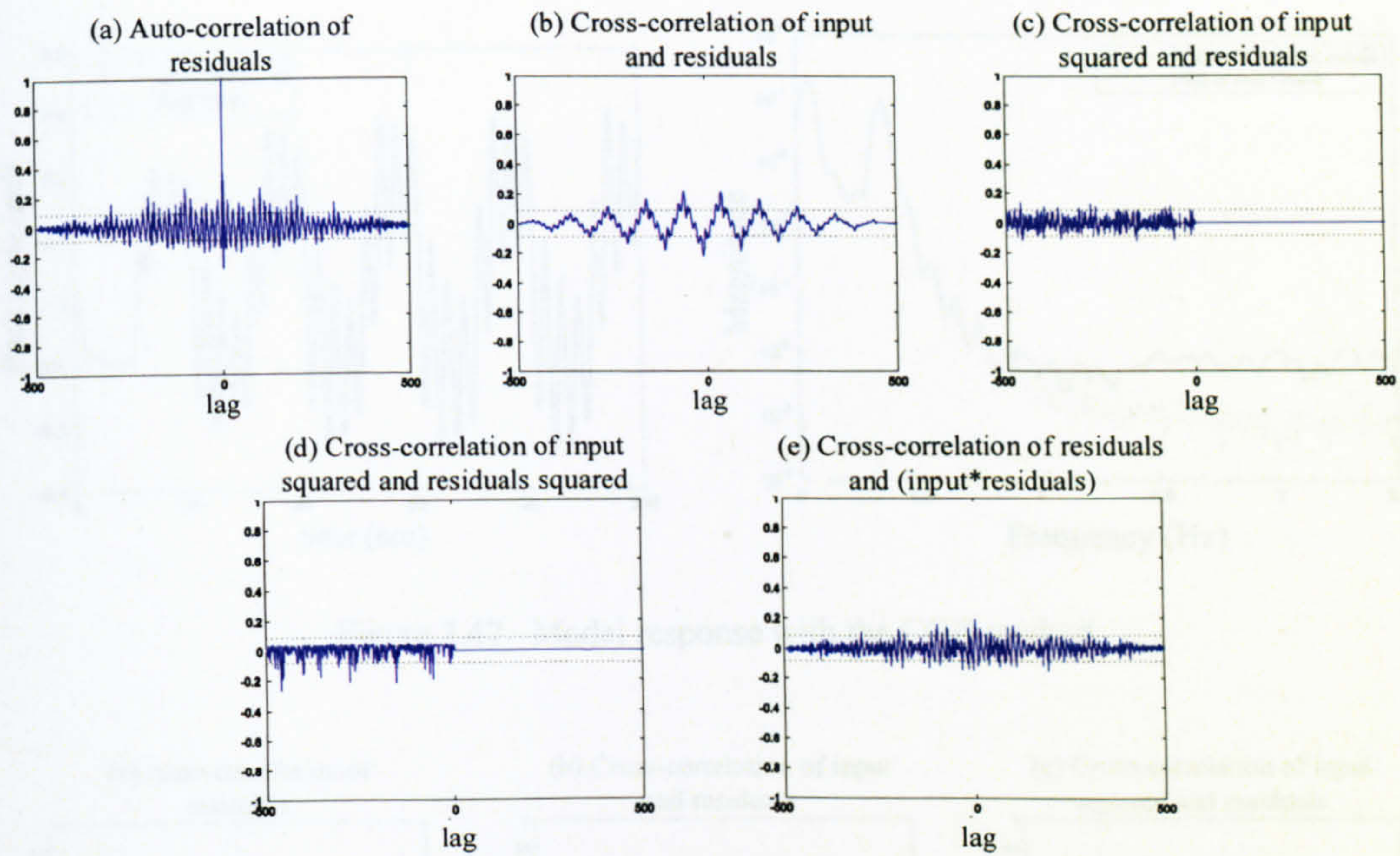


Figure 3.44 Correlation validation of the model with the CGF training method

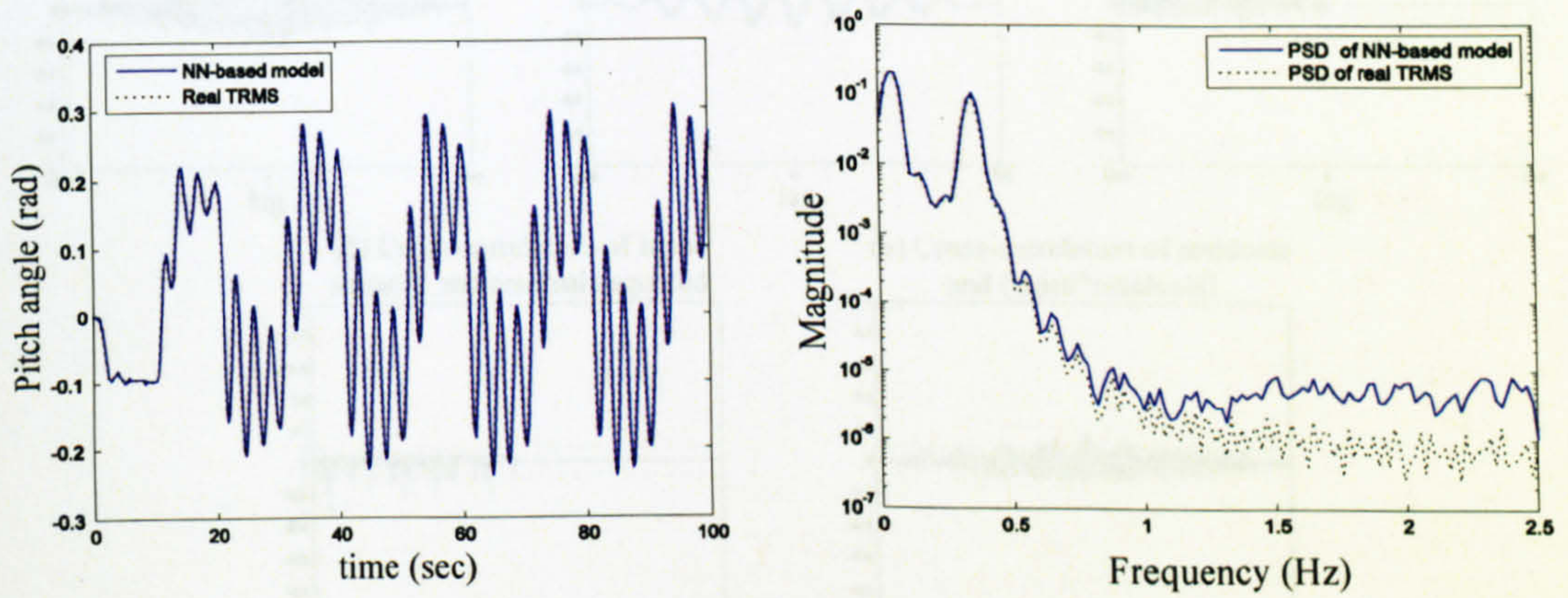


Figure 3.45 Model response with the CGP training method

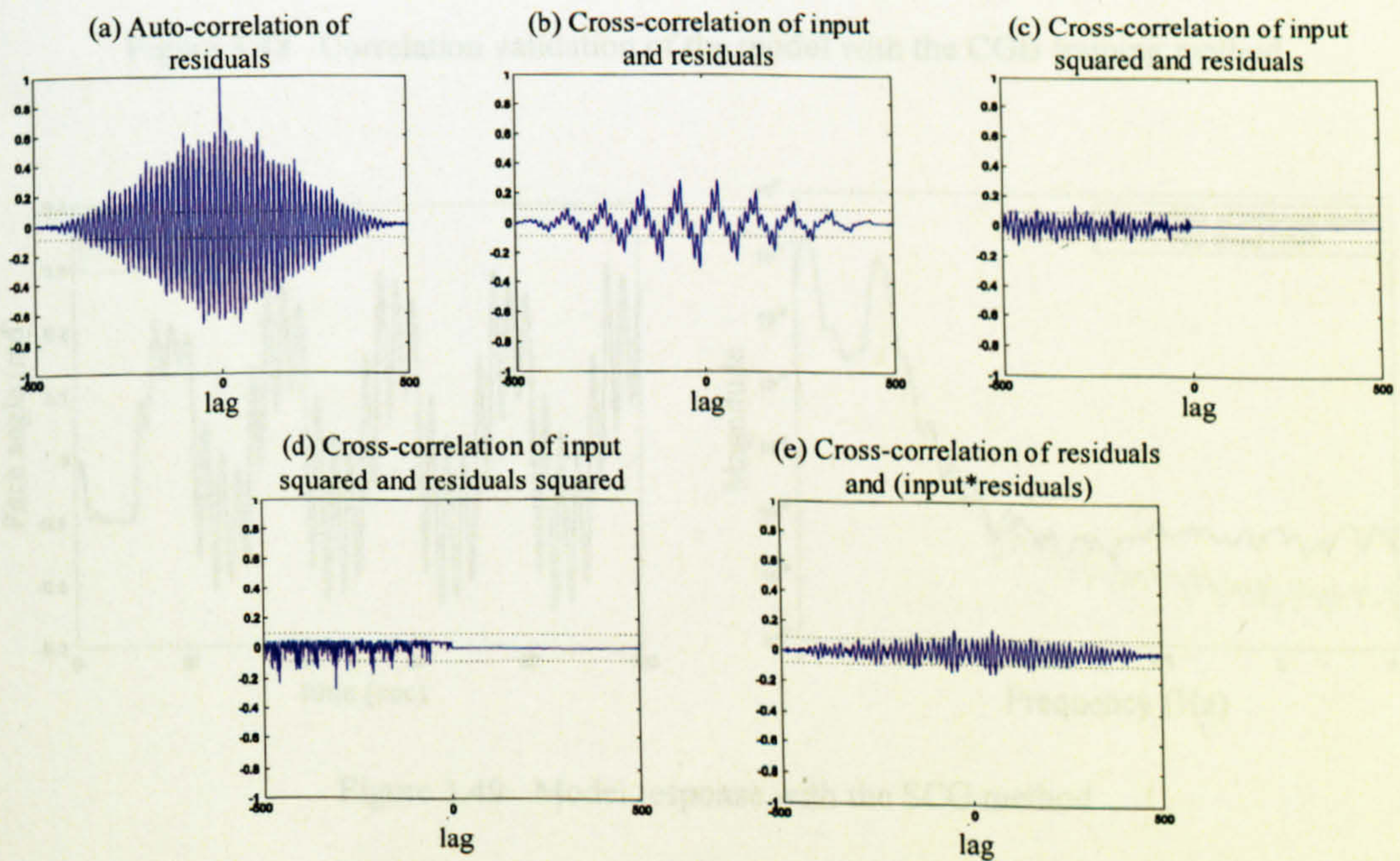


Figure 3.46 Correlation validation of the model with the CGP training method

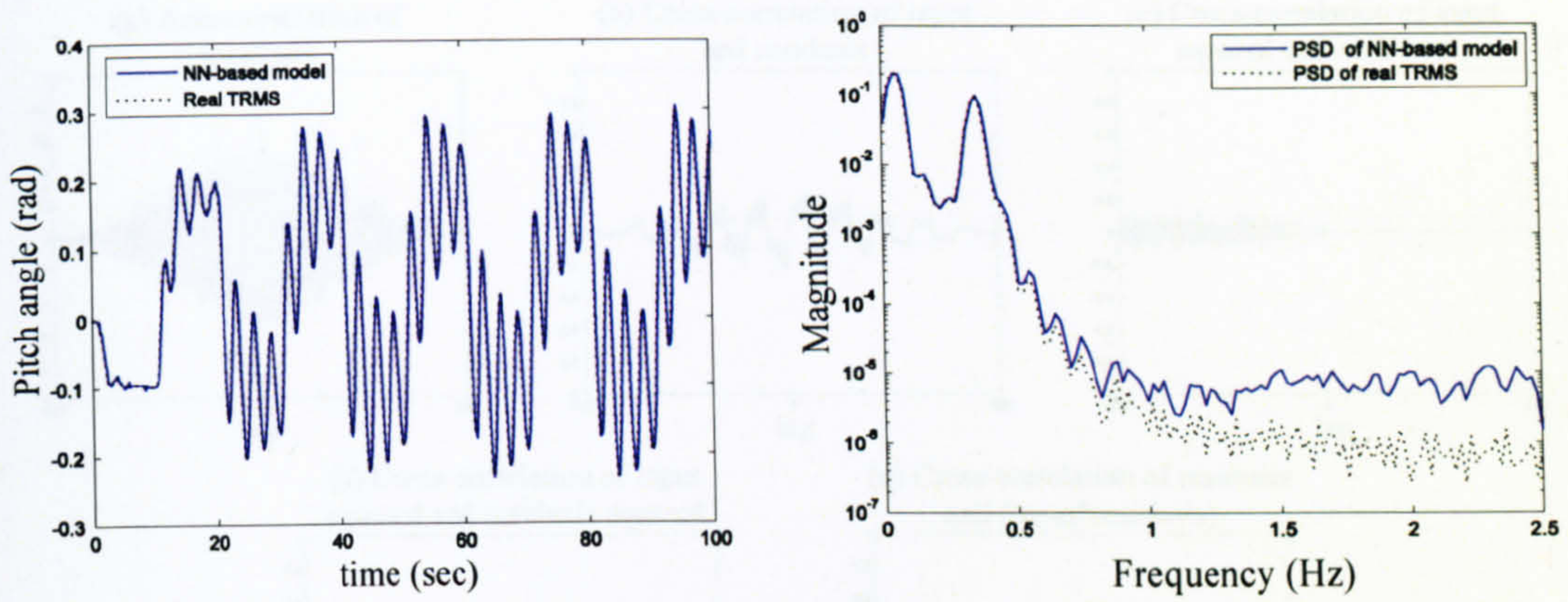


Figure 3.47 Model response with the CGB method

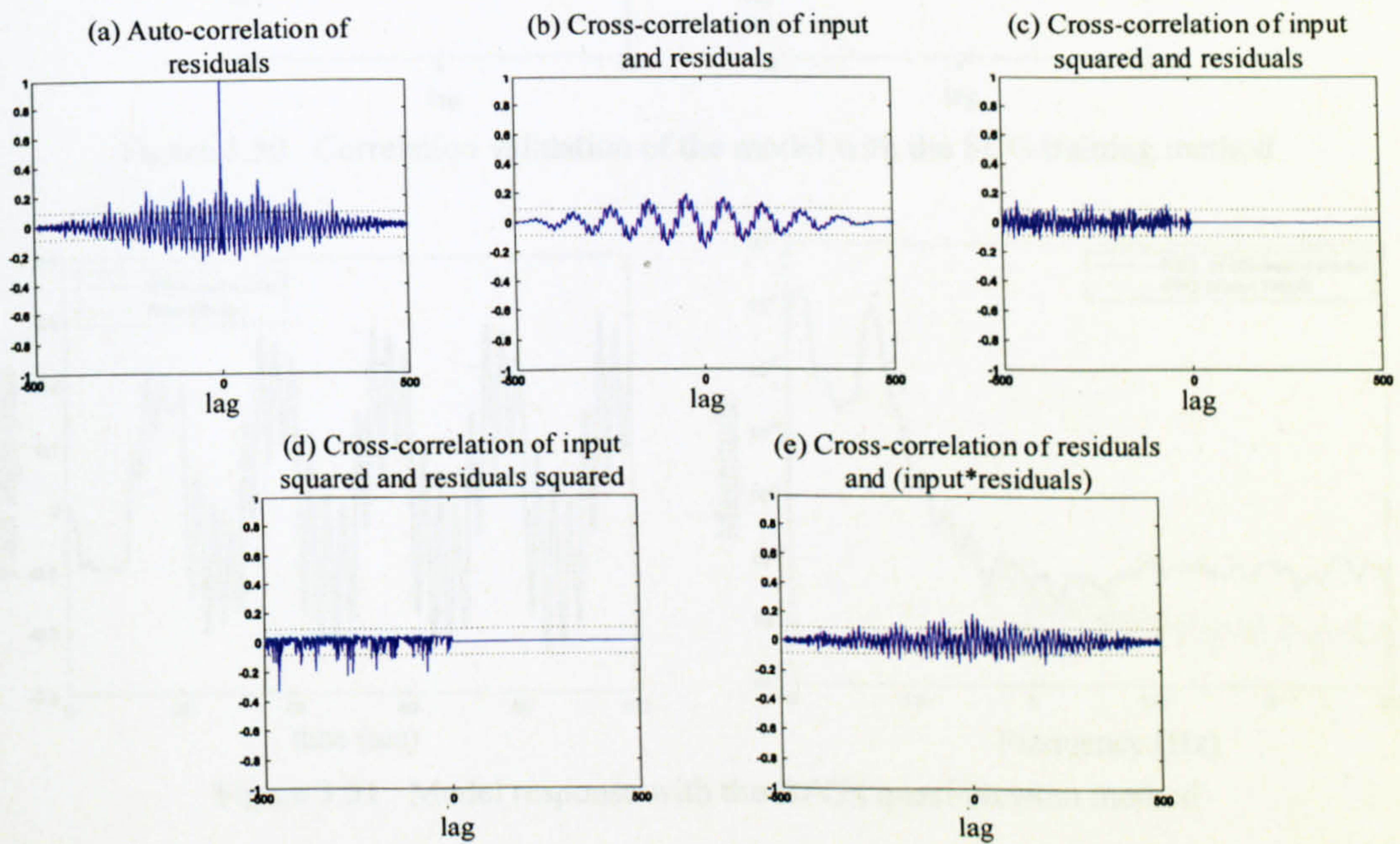


Figure 3.48 Correlation validation of the model with the CGB training method

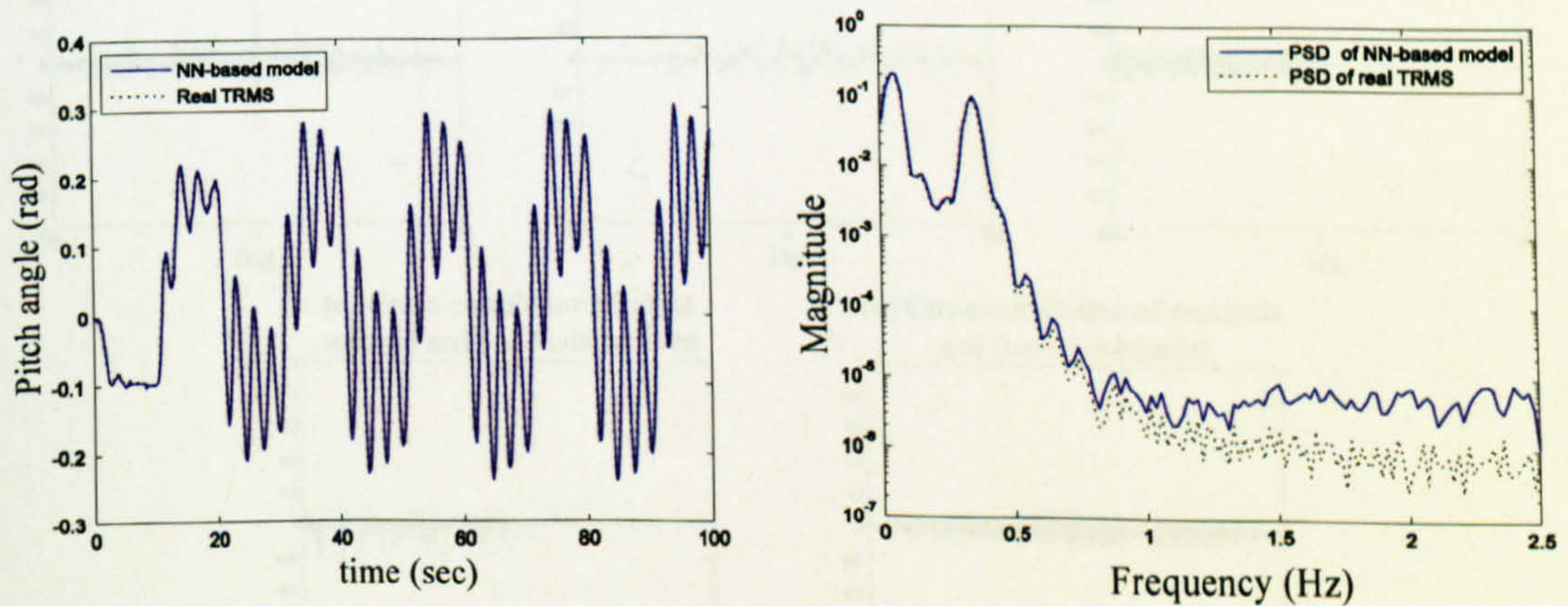


Figure 3.49 Model response with the SCG method

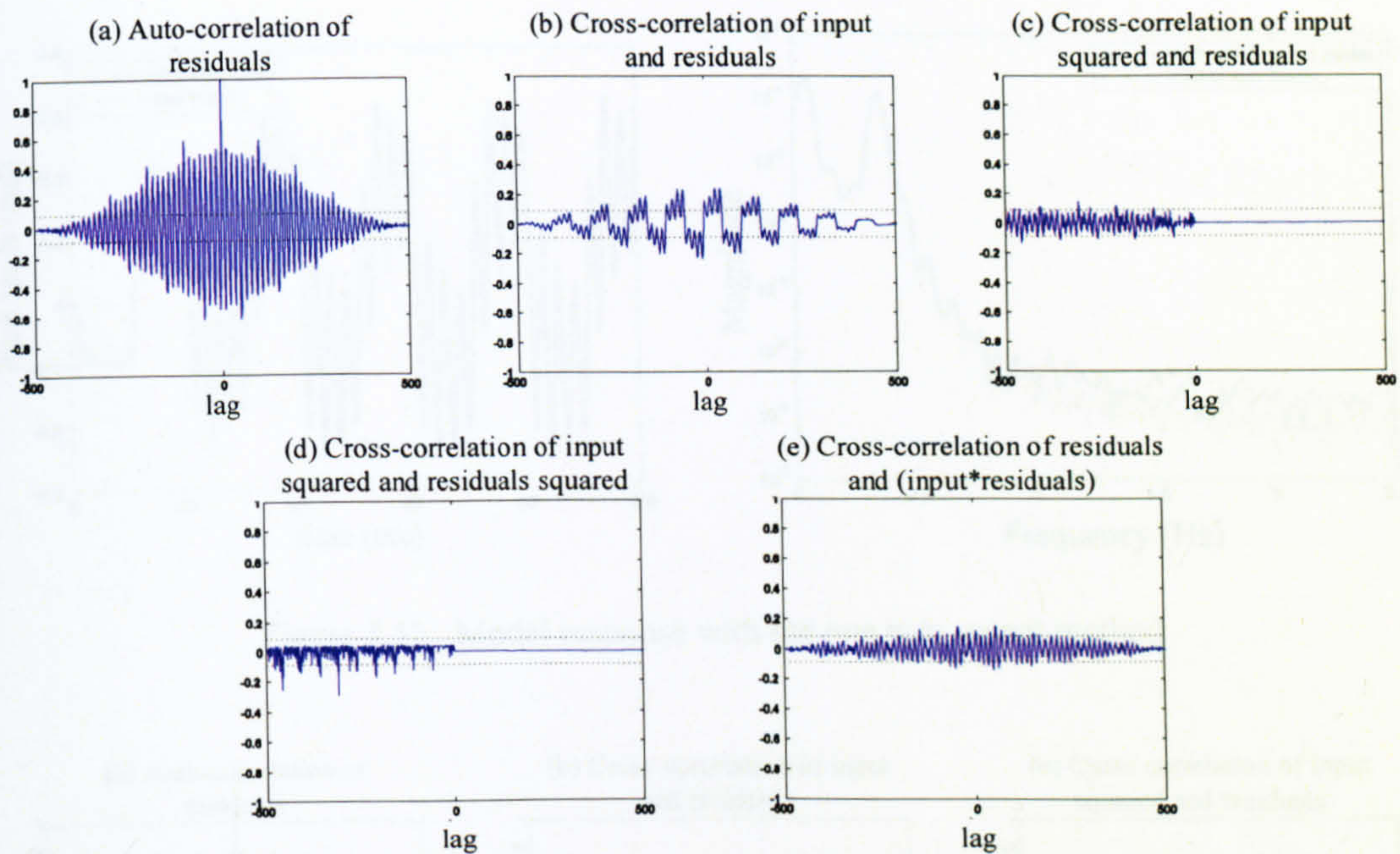


Figure 3.50 Correlation validation of the model with the SCG training method

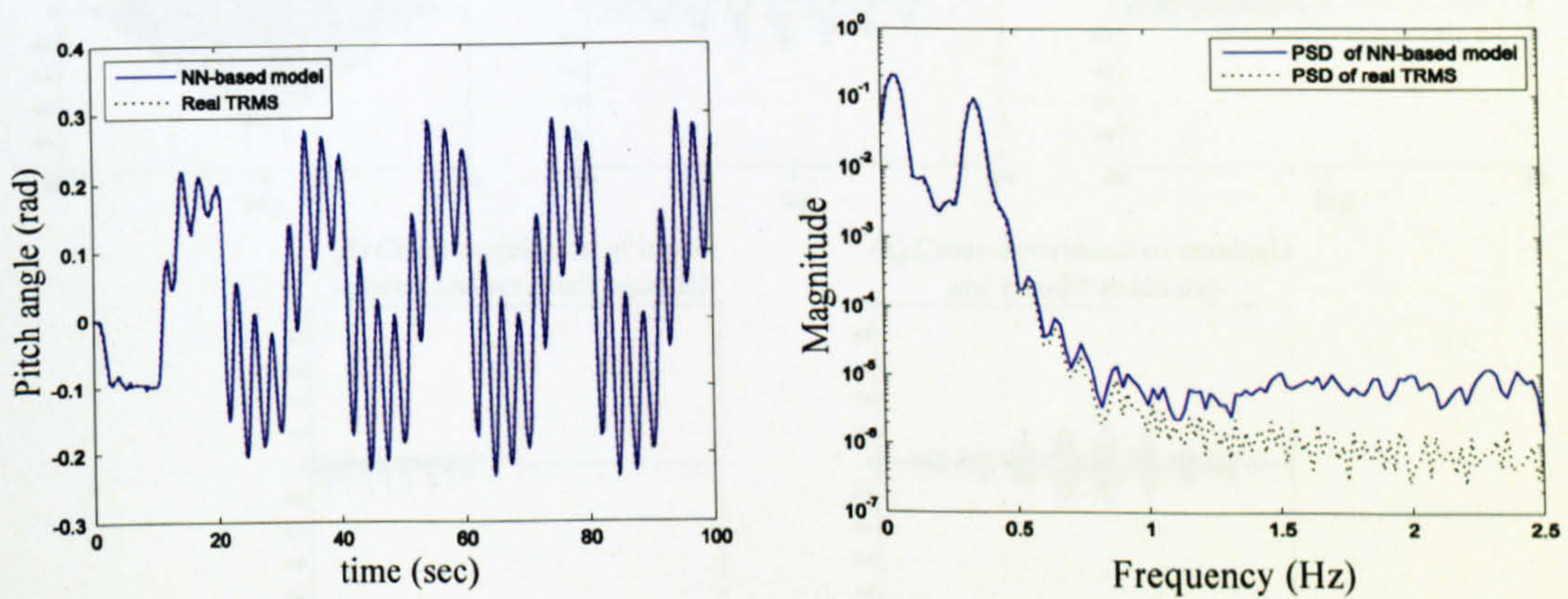


Figure 3.51 Model response with the BFGS quasi-Newton method

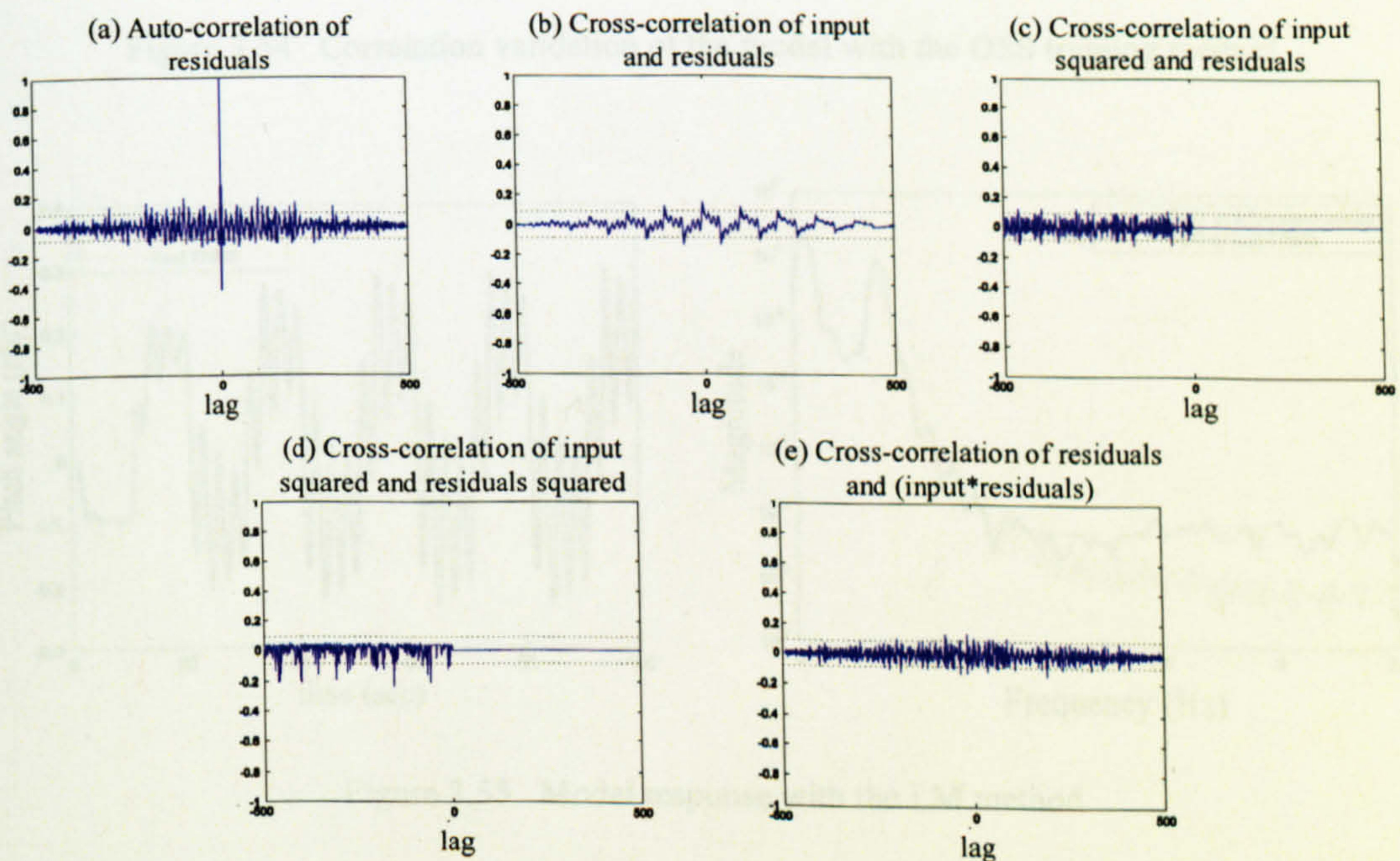


Figure 3.52 Correlation validation of the model with the BFGS learning method

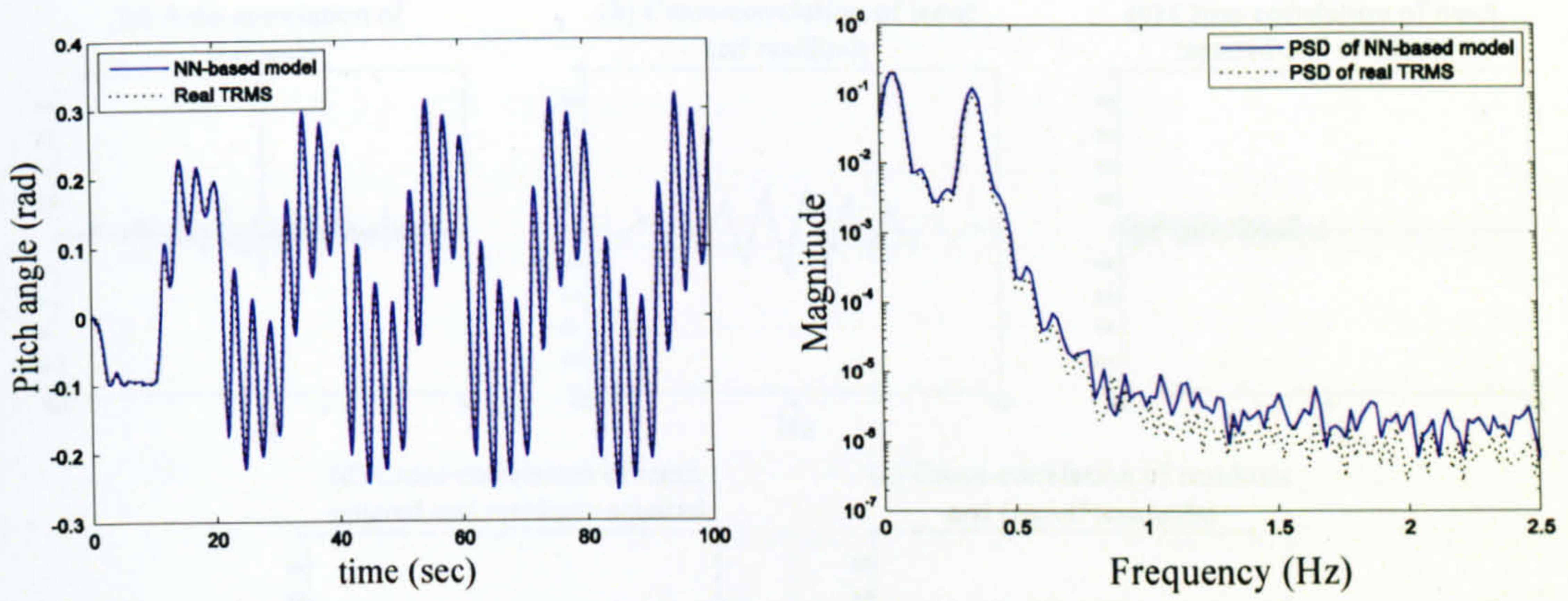


Figure 3.53 Model response with the one step secant method

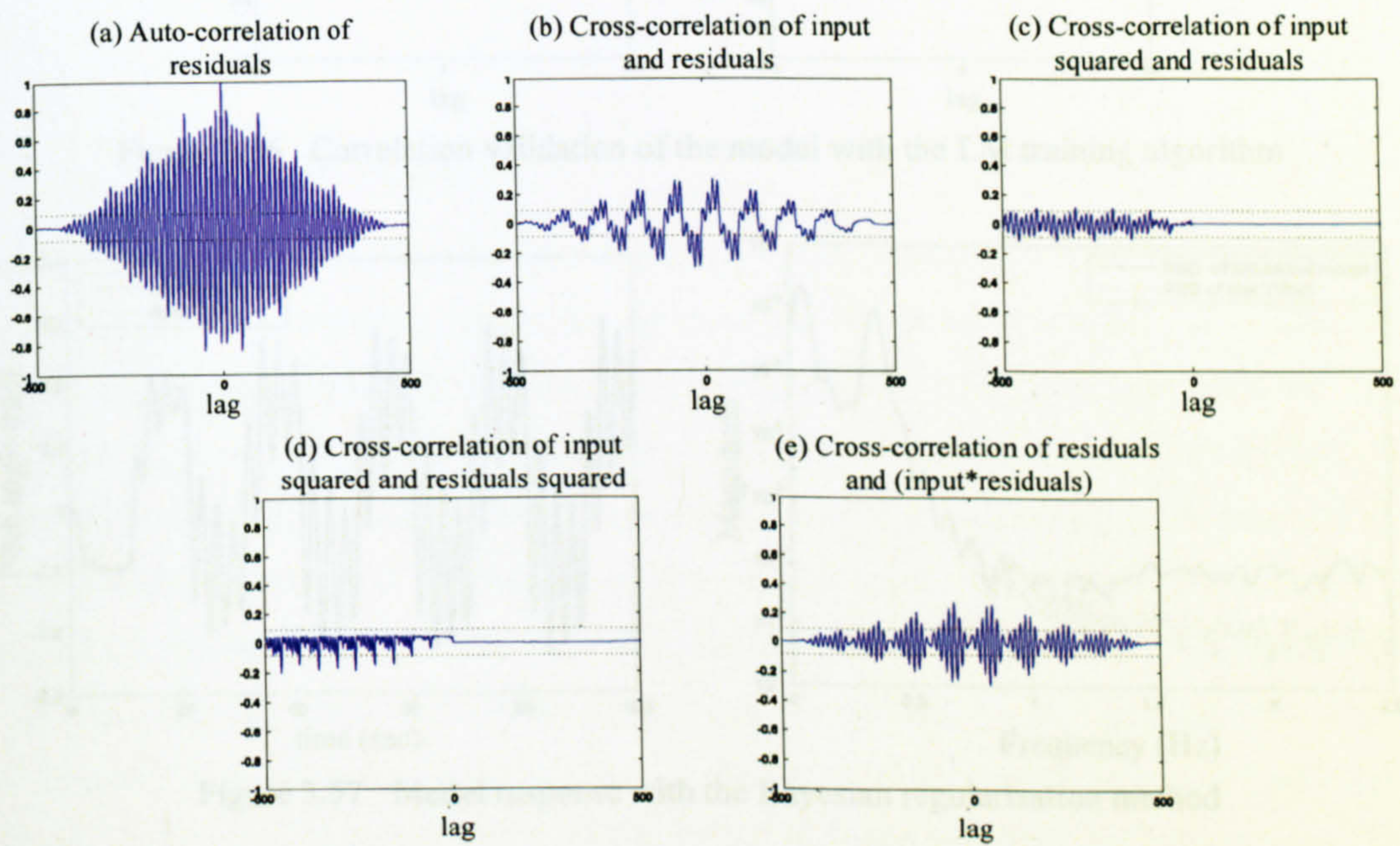


Figure 3.54 Correlation validation of the model with the OSS training method

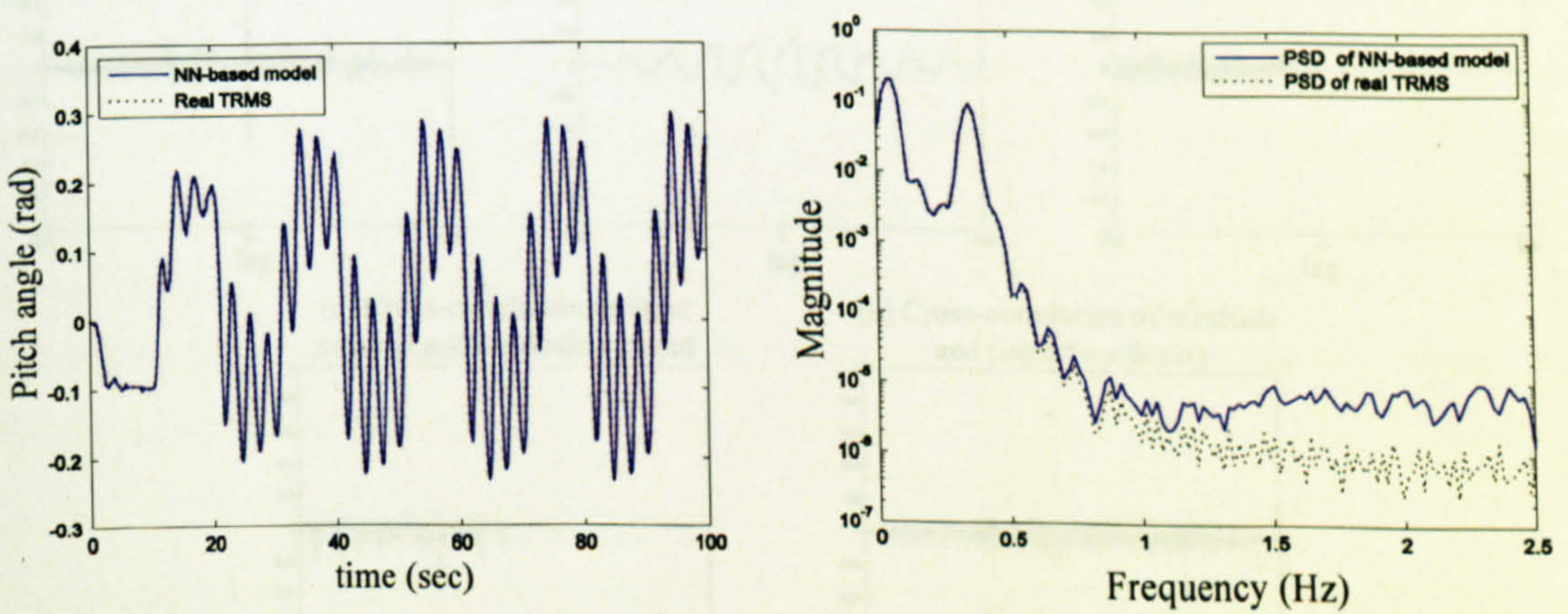


Figure 3.55 Model response with the LM method

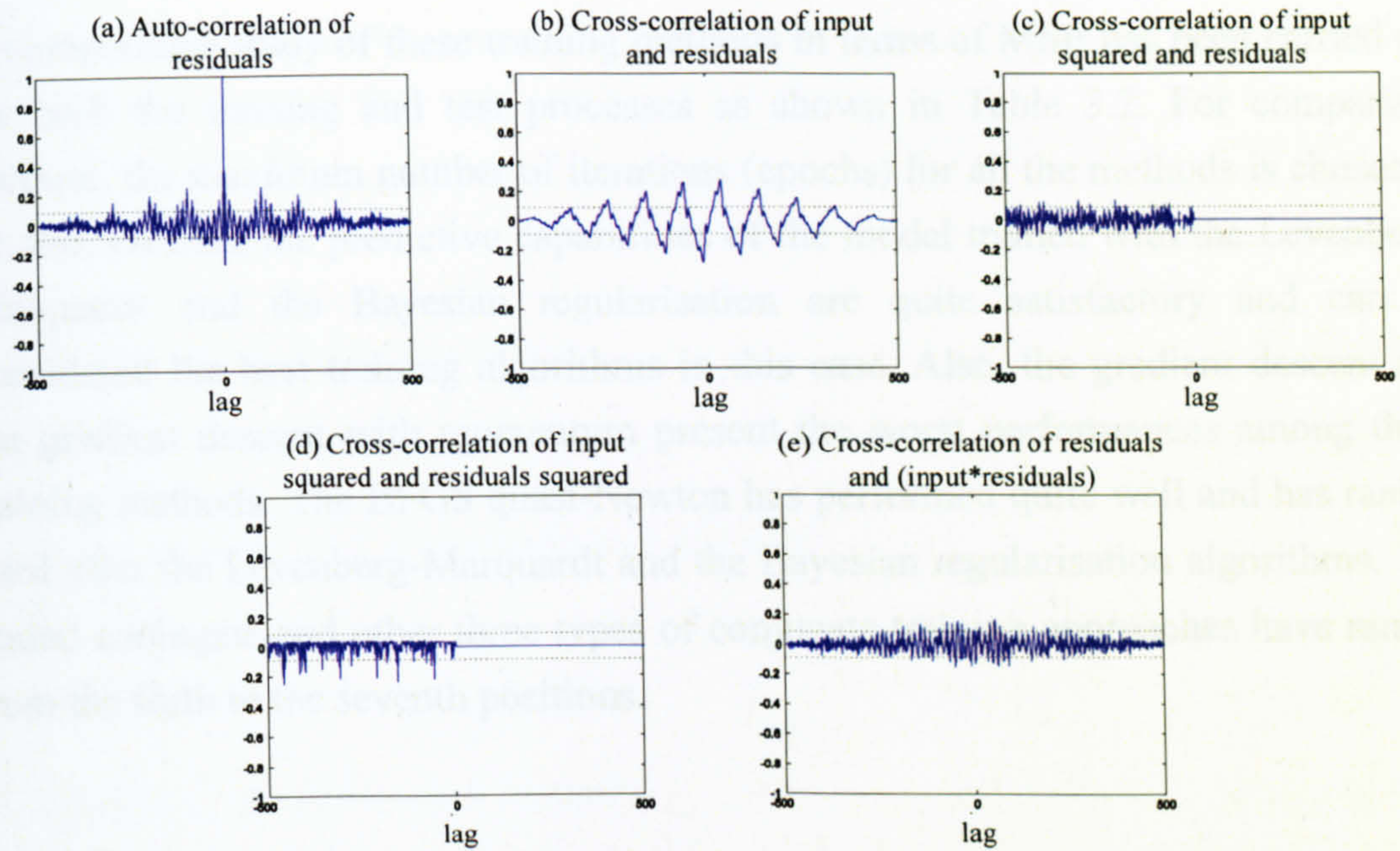


Figure 3.56 Correlation validation of the model with the LM training algorithm

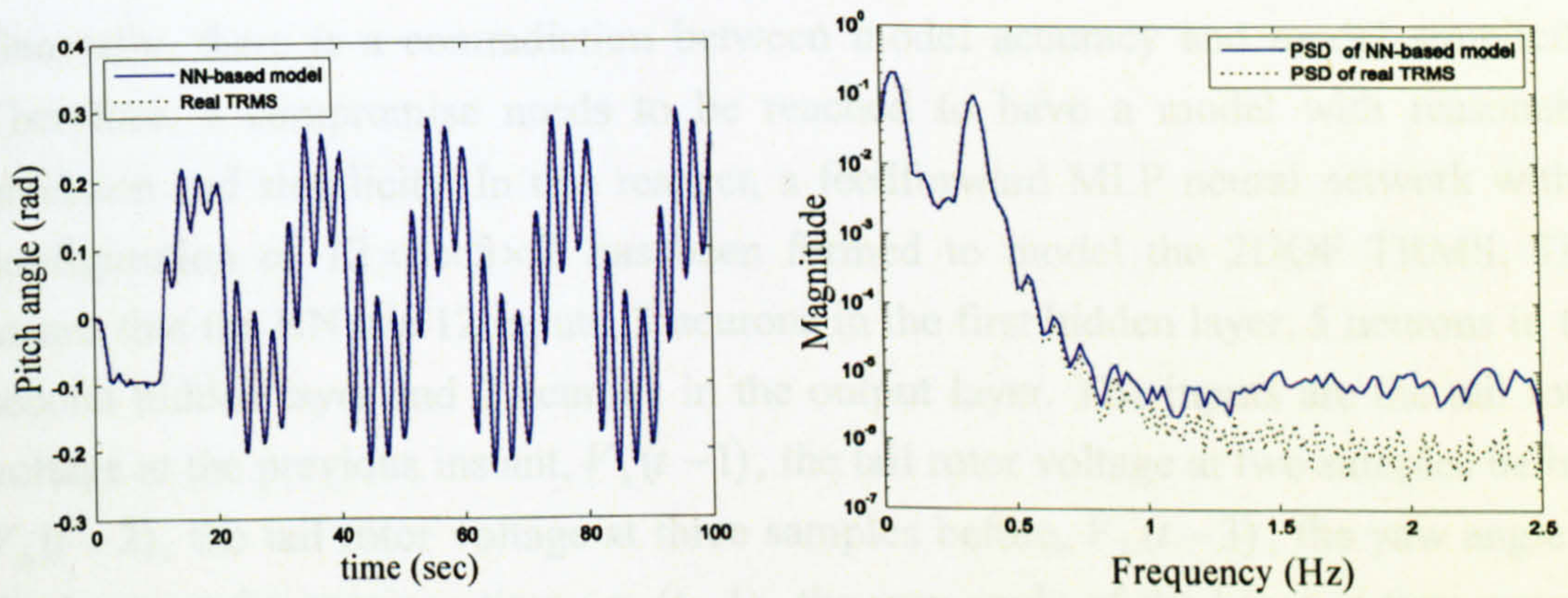


Figure 3.57 Model response with the Bayesian regularisation method

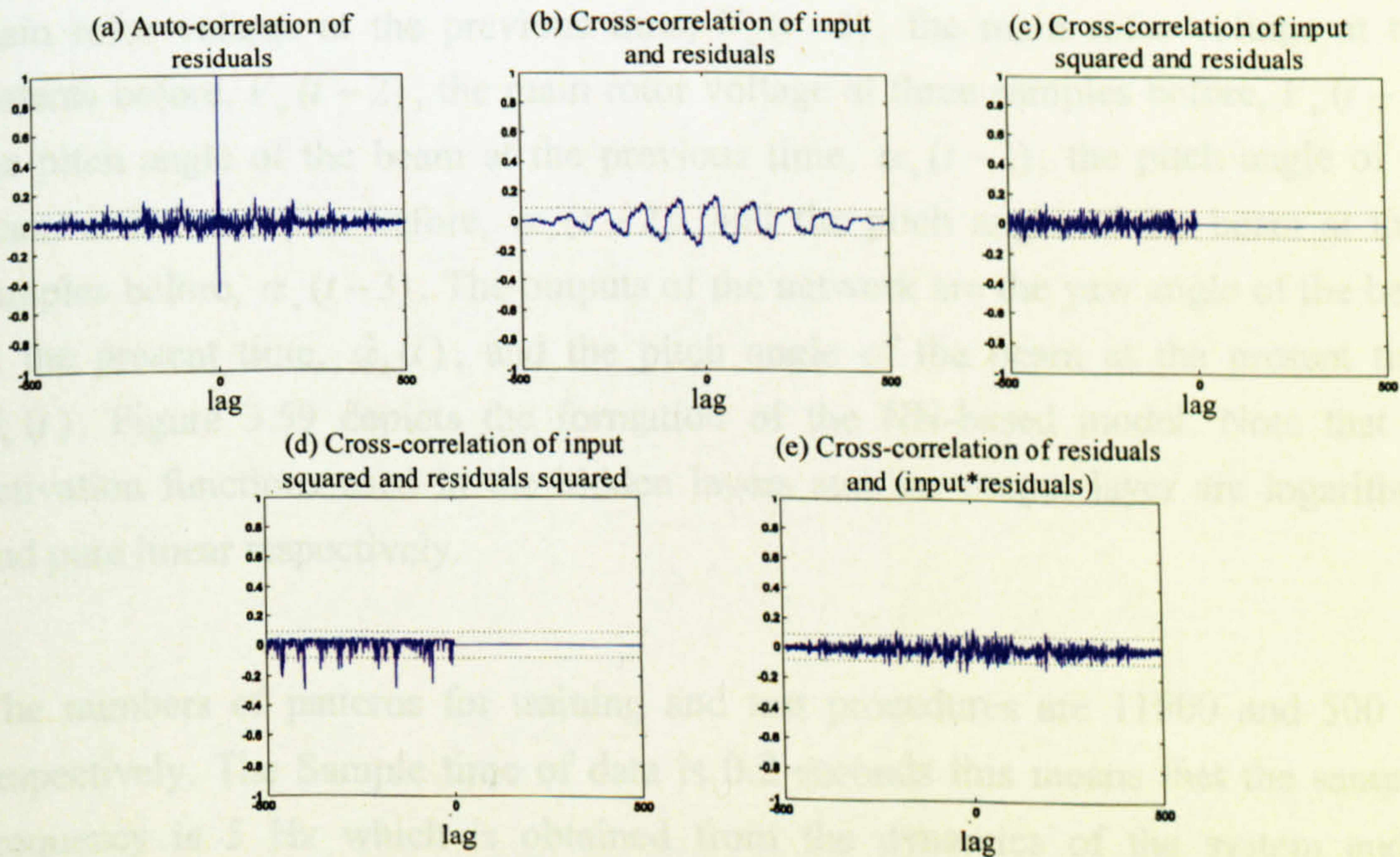


Figure 3.58 Correlation validation of the model with the BR training method

A comparative study of these training methods in terms of MSE has been carried out for both the training and test processes as shown in Table 3.2. For comparison purpose, the maximum number of iterations (epochs) for all the methods is chosen to be 500. Overall, the predictive capabilities of the model trained with the Levenberg-Marquardt and the Bayesian regularisation are quite satisfactory and can be considered the best training algorithms in this case. Also, the gradient descent and the gradient descent with momentum present the worst performances among these training methods. The BFGS quasi-Newton has performed quite well and has ranked third after the Levenberg-Marquardt and the Bayesian regularisation algorithms. The scaled conjugate and other three types of conjugate training approaches have ranked from the fourth to the seventh positions.

3.4.3 The 2DOF Model

Generally, there is a contradiction between model accuracy and model simplicity. Therefore, a compromise needs to be reached to have a model with reasonable precision and simplicity. In this respect, a feedforward MLP neural network with a configuration of $12 \times 3 \times 5 \times 2$ has been formed to model the 2DOF TRMS. This means that the NN has 12 inputs, 3 neurons in the first hidden layer, 5 neurons in the second hidden layer and 2 neurons in the output layer. The inputs are the tail rotor voltage at the previous instant, $V_h(t-1)$, the tail rotor voltage at two samples before, $V_h(t-2)$, the tail rotor voltage at three samples before, $V_h(t-3)$, the yaw angle of the beam at the previous time, $\alpha_h(t-1)$, the yaw angle of the beam at two samples before, $\alpha_h(t-2)$, the yaw angle of the beam at three samples before, $\alpha_h(t-3)$, the main rotor voltage at the previous time, $V_v(t-1)$, the main rotor voltage at two instants before, $V_v(t-2)$, the main rotor voltage at three samples before, $V_v(t-3)$, the pitch angle of the beam at the previous time, $\alpha_v(t-1)$, the pitch angle of the beam at two samples before, $\alpha_v(t-2)$, and the pitch angle of the beam at three samples before, $\alpha_v(t-3)$. The outputs of the network are the yaw angle of the beam at the present time, $\hat{\alpha}_h(t)$, and the pitch angle of the beam at the present time, $\hat{\alpha}_v(t)$. Figure 3.59 depicts the formation of the NN-based model. Note that the activation functions used in the hidden layers and the output layer are logarithmic and pure linear respectively.

The numbers of patterns for training and test procedures are 11500 and 500 sets respectively. The Sample time of data is 0.2 seconds this means that the sampling frequency is 5 Hz which is obtained from the dynamics of the system and its dominant time constant. The NN has been trained with different training methods.

Table 3.2 Performance comparison of the training algorithms

Learning algorithm	Training MSE	Test MSE
Gradient descent	2.37×10^{-2}	1.15×10^{-3}
Gradient descent with momentum	1.82×10^{-2}	2.81×10^{-3}
Gradient descent with adaptive learning	1.73×10^{-3}	6.07×10^{-4}
Gradient descent with momentum and adaptive learning	4.85×10^{-3}	3.45×10^{-4}
Resilient	4.29×10^{-4}	4.73×10^{-5}
Conjugate gradient Fletcher-Reeves	2.81×10^{-5}	9.79×10^{-6}
Conjugate gradient Polak-Ribiere	2.42×10^{-5}	9.11×10^{-6}
Conjugate gradient Powell-Beale	2.79×10^{-5}	1.01×10^{-5}
Scaled conjugate gradient	1.82×10^{-5}	9.55×10^{-6}
BFGS quasi-Newton	1.76×10^{-5}	9.41×10^{-6}
One step secant	9.73×10^{-5}	1.18×10^{-5}
Levenberg-Marquardt	1.54×10^{-5}	8.84×10^{-6}
Bayesian regularisation	1.47×10^{-5}	8.88×10^{-6}

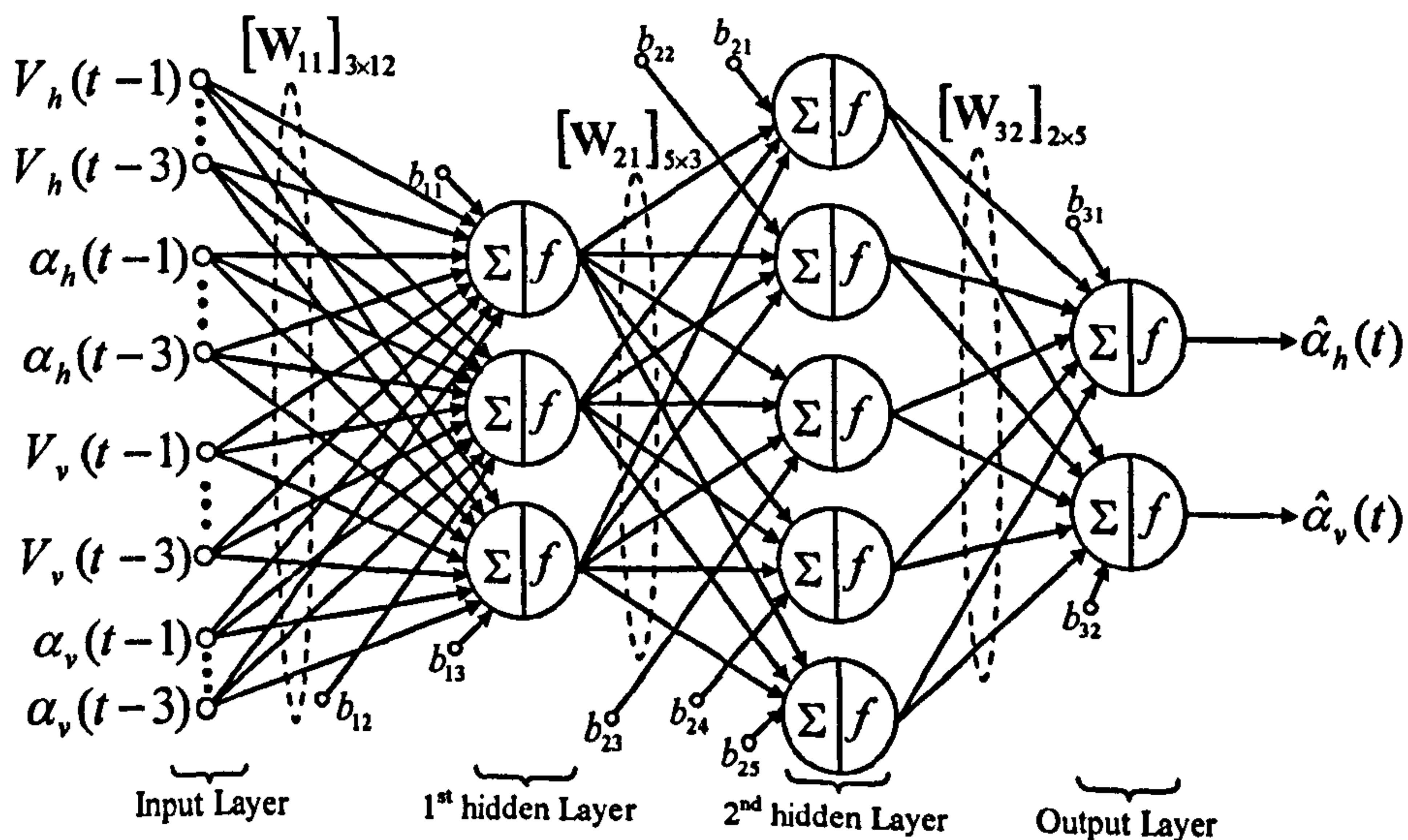


Figure 3.59 The structure of the NN-based model of the TRMS

For validation the NN-based model performance, the trained network has been tested with different types of input signals. Figures 3.60 to 3.80 show the output of the NN-based model with different training methods in the test procedure, their corresponding PSDs and their correlation tests [118], [119]. As pointed out earlier, all the tests apply equally well to MIMO systems. The correlation tests have been carried out for each combination of the input-output.

Figures 3.60 to 3.65 show that the LM learning algorithm has a satisfactory result to model the TRMS in terms of its 2DOF dynamics. Figure 3.60 depicts the yaw angle overlapping and PSD test results. The results related to the vertical channel can be seen in Figure 3.61. The correlation tests related to each input-output pair are shown in Figures 3.62 to 3.65. The satisfactory test results regarding the BFGS learning technique are depicted in Figures 3.66 to 3.68. Note that for the sake of brevity only one set of correlation tests is presented for this learning algorithm. NN-based model trained using the Bayesian regularisation learning algorithm shows an acceptable performance as can be seen from Figures 3.69 to 3.71.

Figures 3.72 to 3.74 depict the results of the conjugate gradient Powell-Beale learning algorithms. It is clear that this learning algorithm can satisfactorily model the system. Figures 3.75 to 3.77 show the overlapping, PSD and correlation tests of the network trained using the resilient back-propagation training algorithm. The results of the NN-based model trained using the scaled conjugate algorithm can be found in Figures 2.78 to 2.80. The performance is quite good with respect to all available tests.

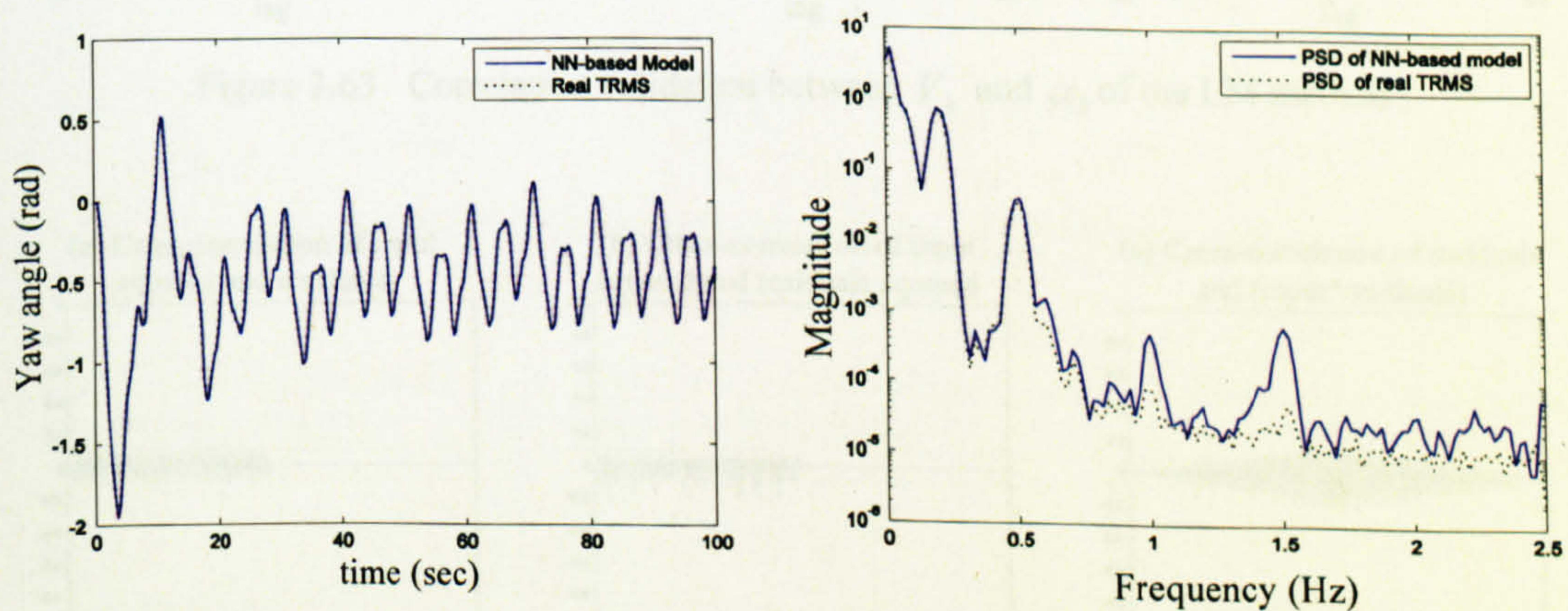


Figure 3.60 Yaw angle response with the LM learning method

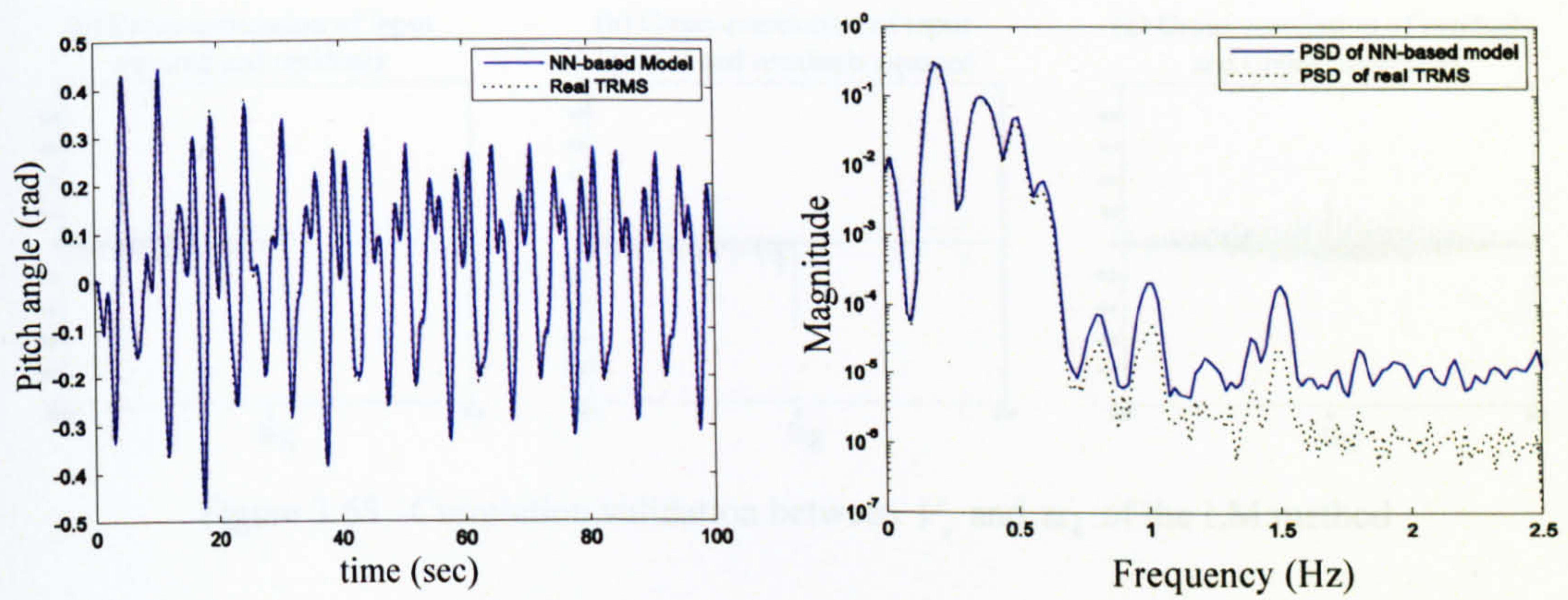


Figure 3.61 Pitch angle response with the LM learning method

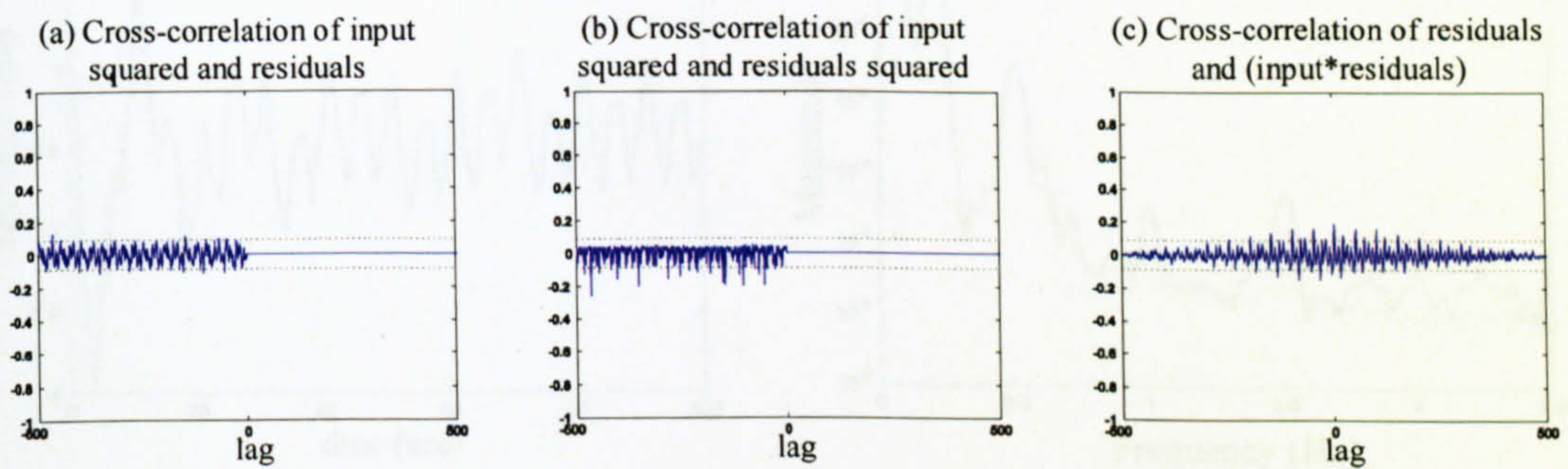


Figure 3.62 Correlation validation between V_v and α_v of the LM method

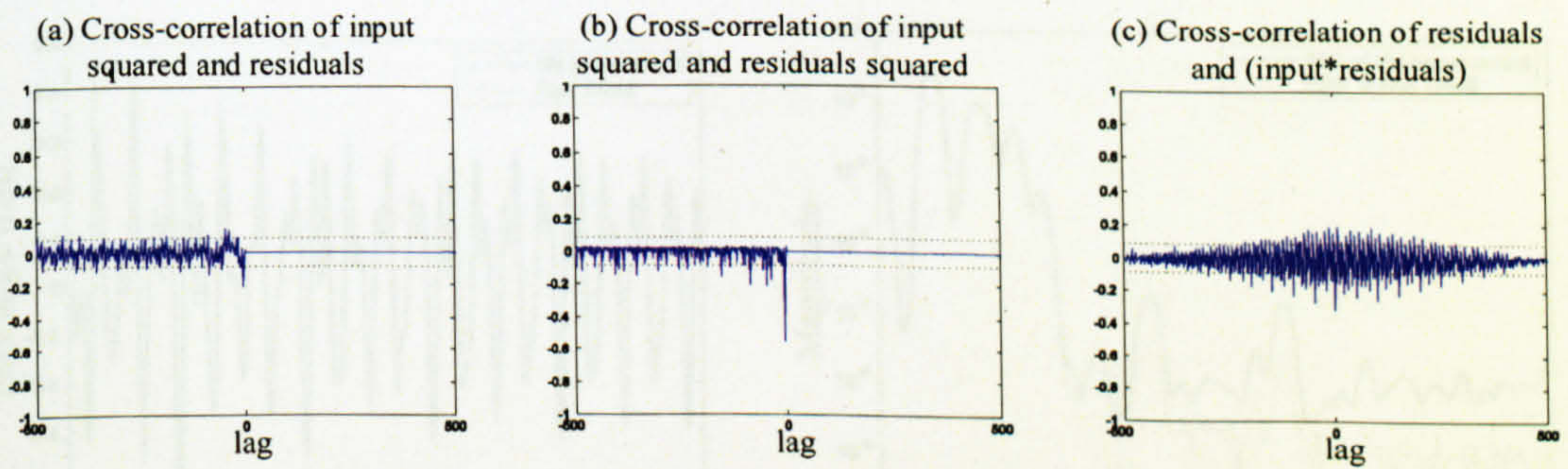


Figure 3.63 Correlation validation between V_h and α_h of the LM method

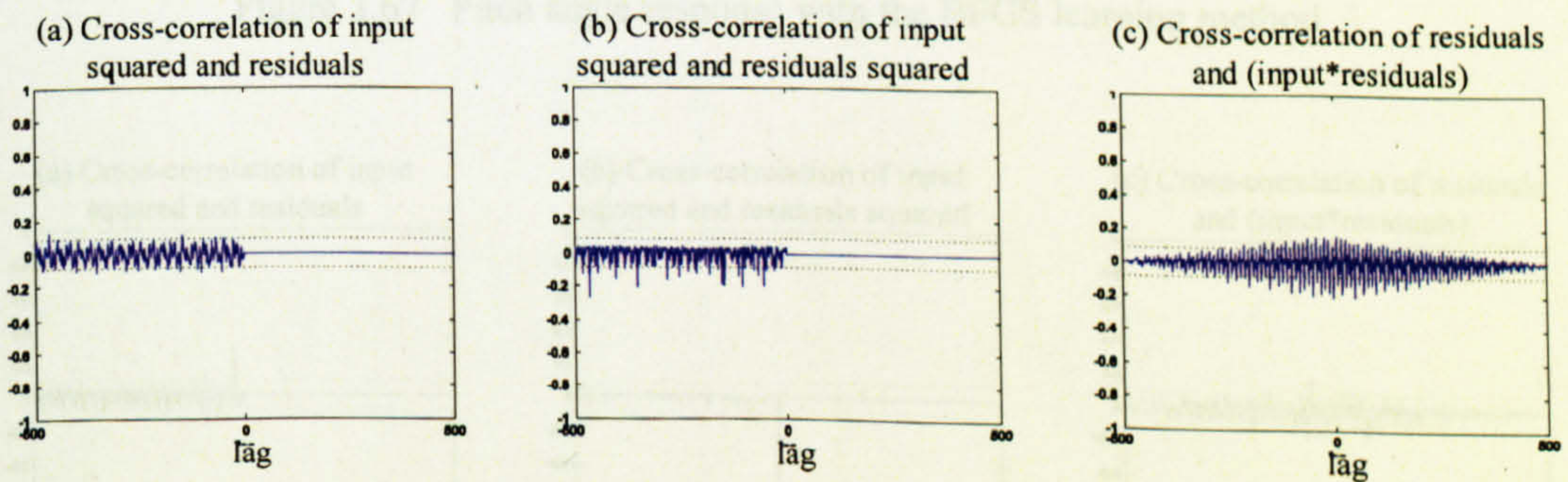


Figure 3.64 Correlation validation between V_h and α_v of the LM method

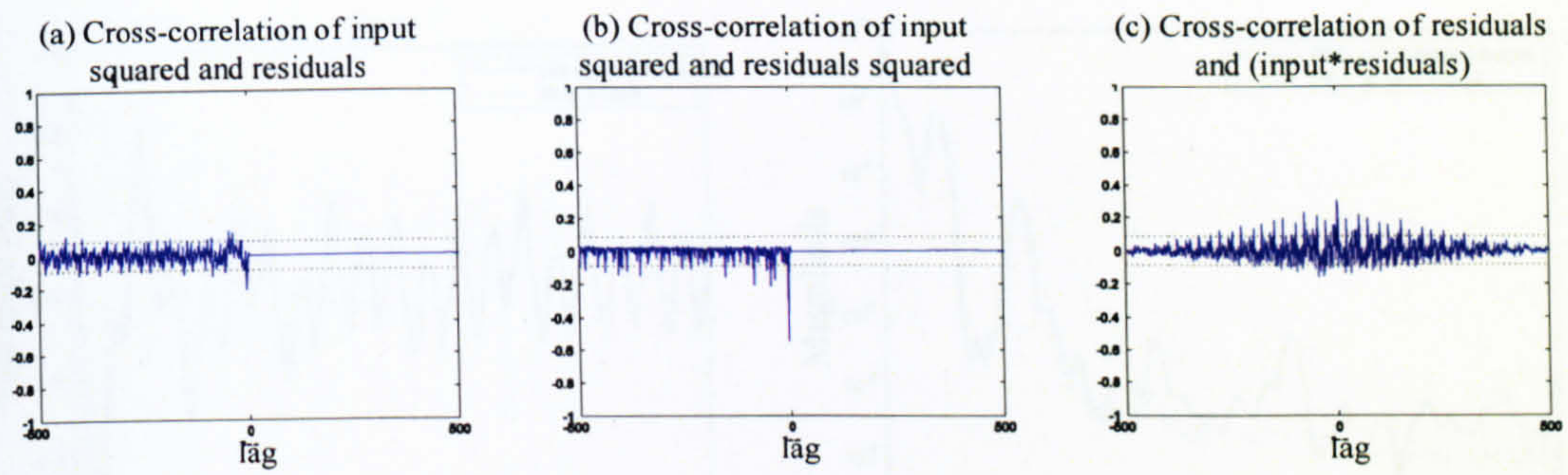


Figure 3.65 Correlation validation between V_v and α_h of the LM method

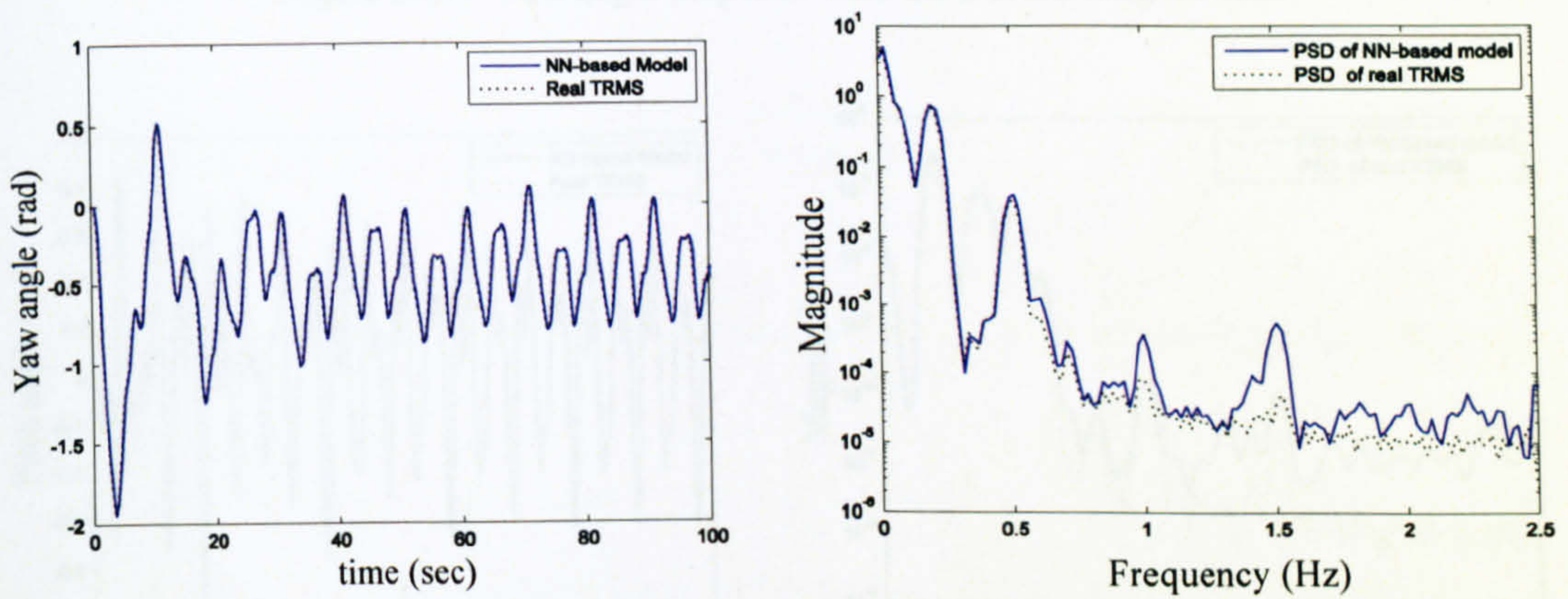


Figure 3.66 Yaw angle response with the BFGS learning method

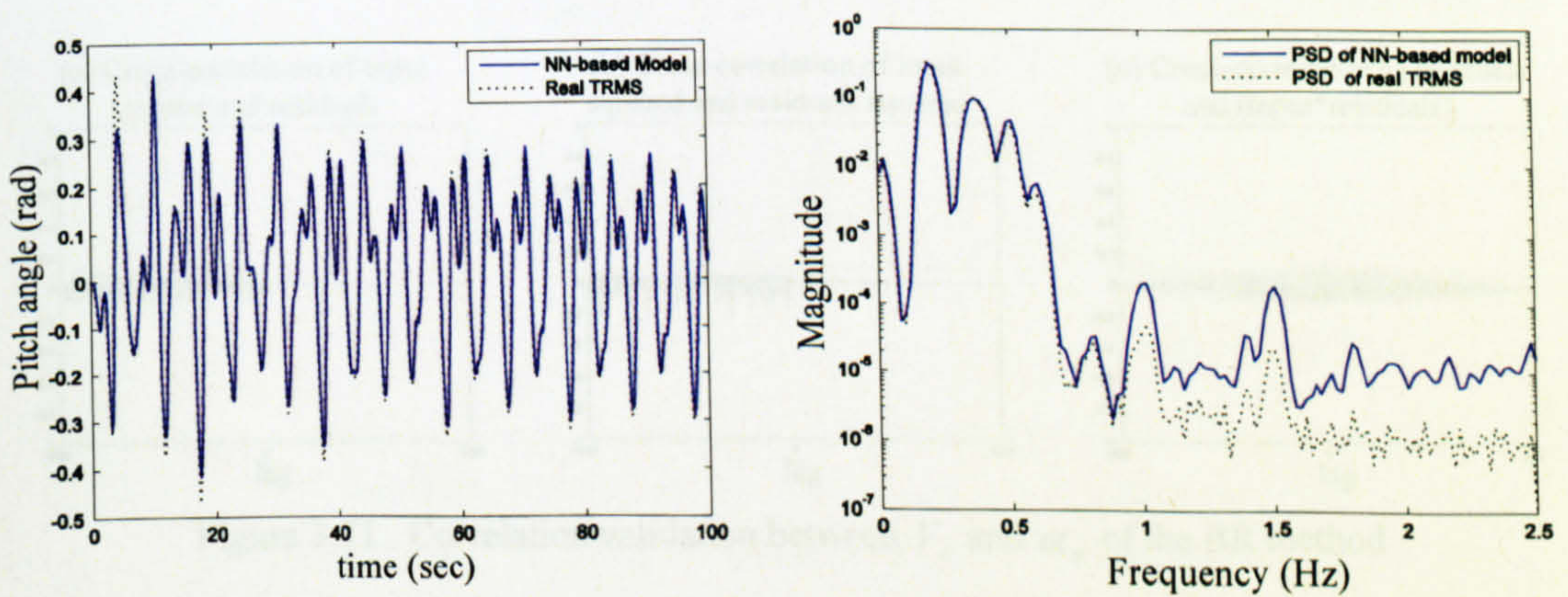


Figure 3.67 Pitch angle response with the BFGS learning method

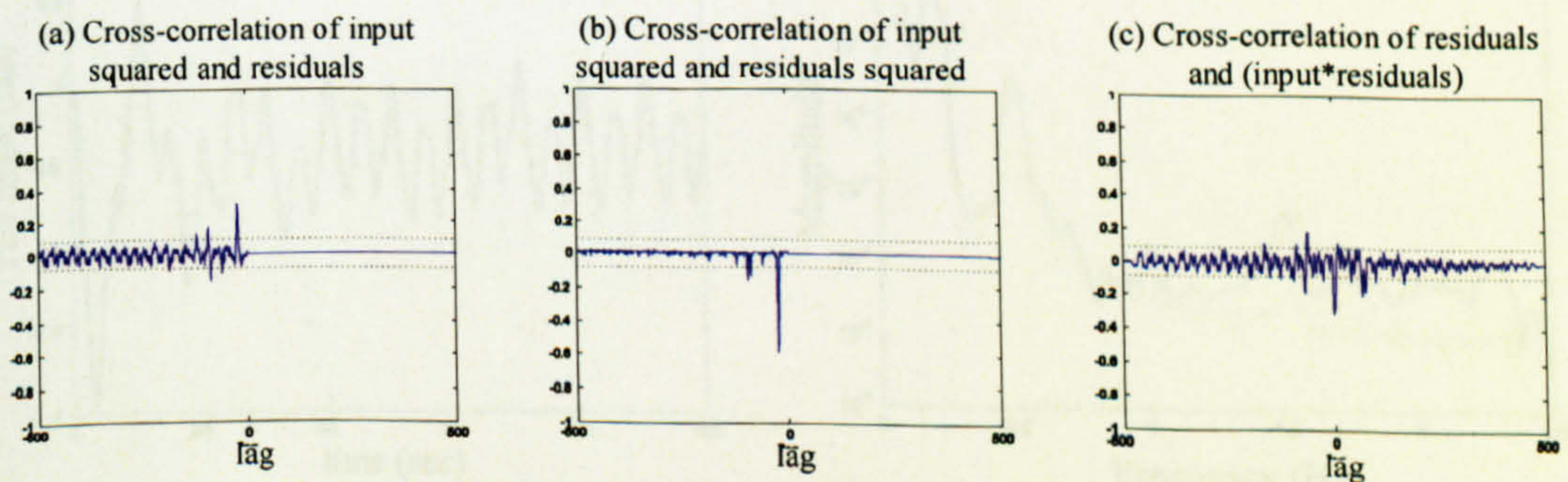


Figure 3.68 Correlation validation between V_v and α_v of the BFGS method

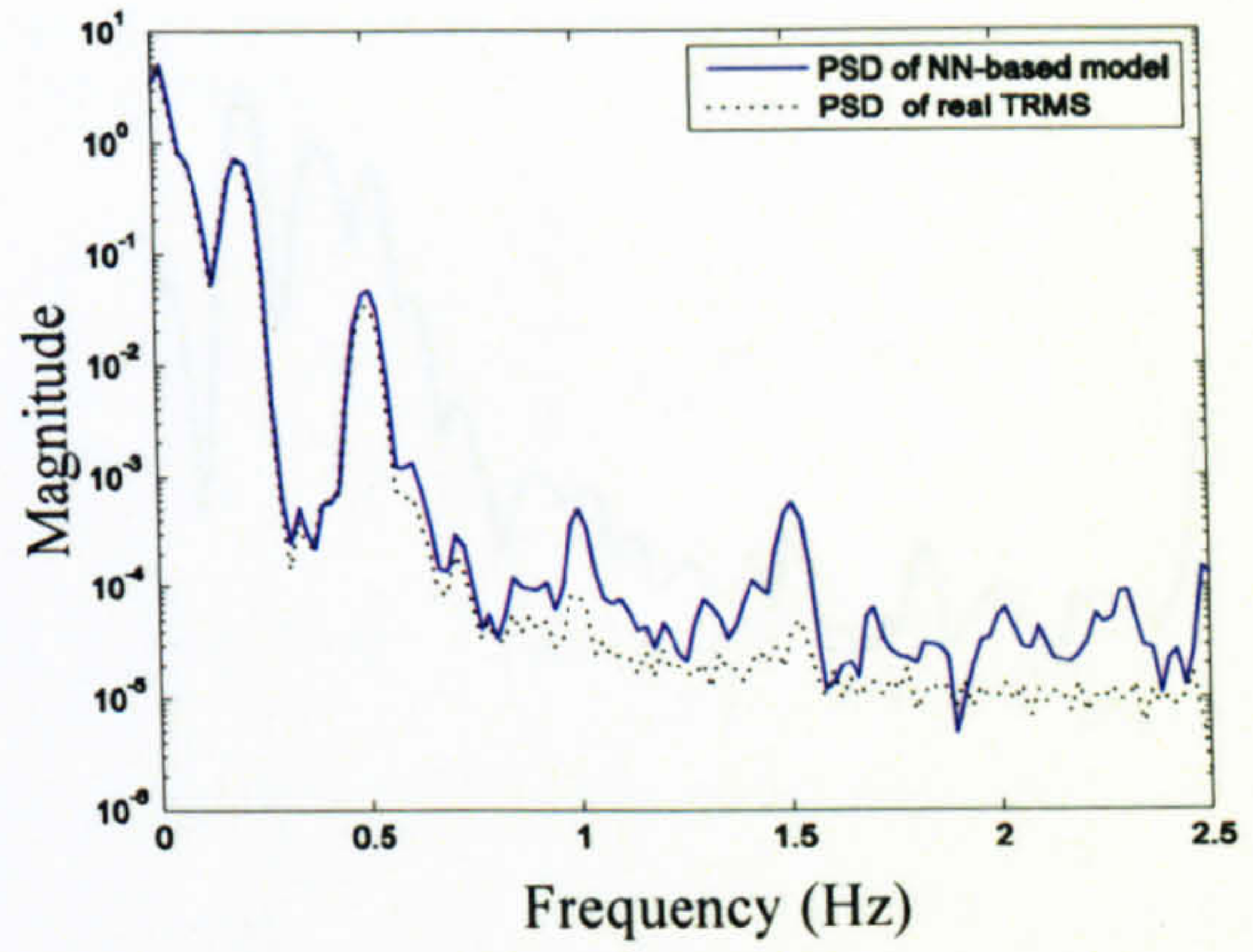
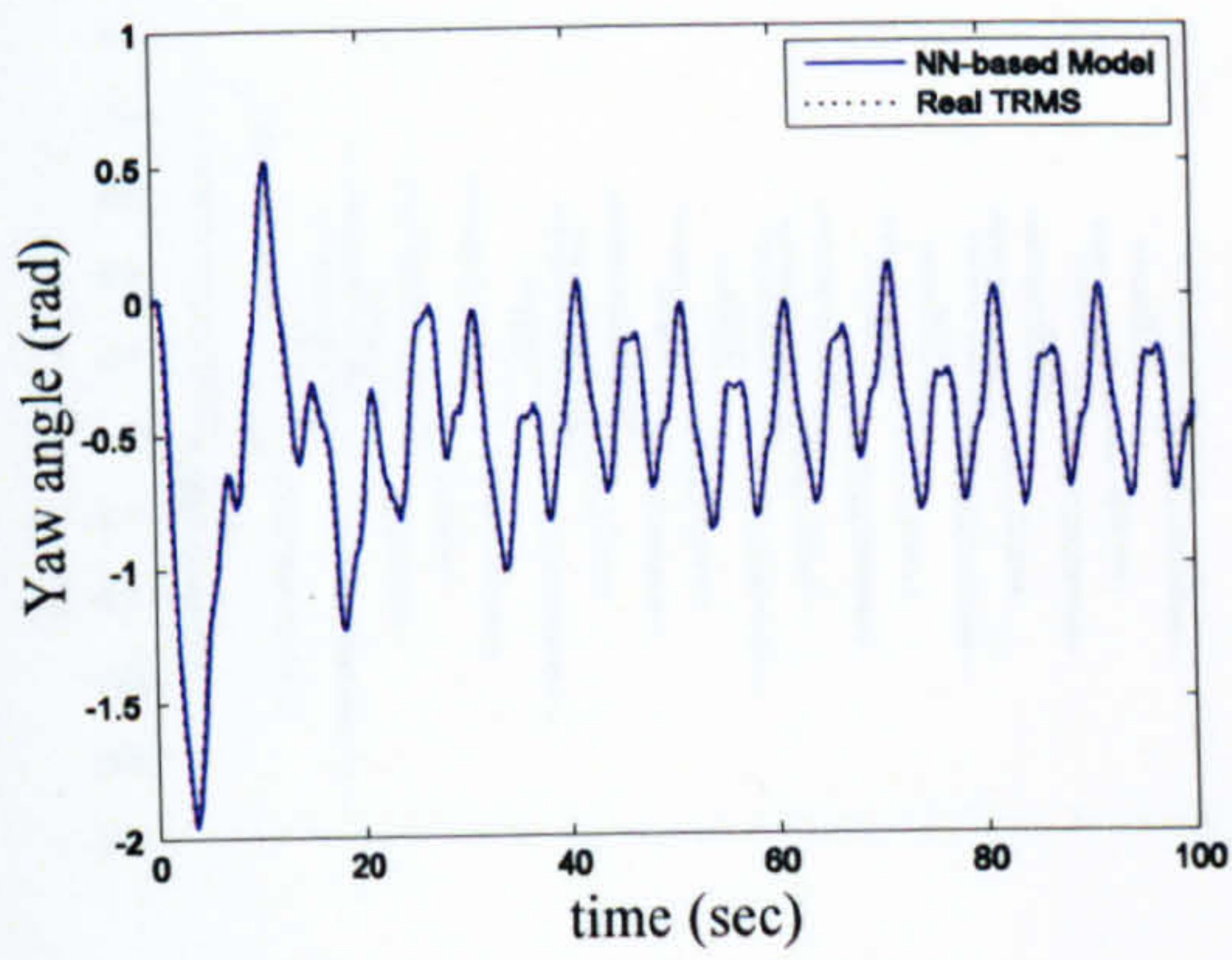


Figure 3.69 Yaw angle response with the BR learning method

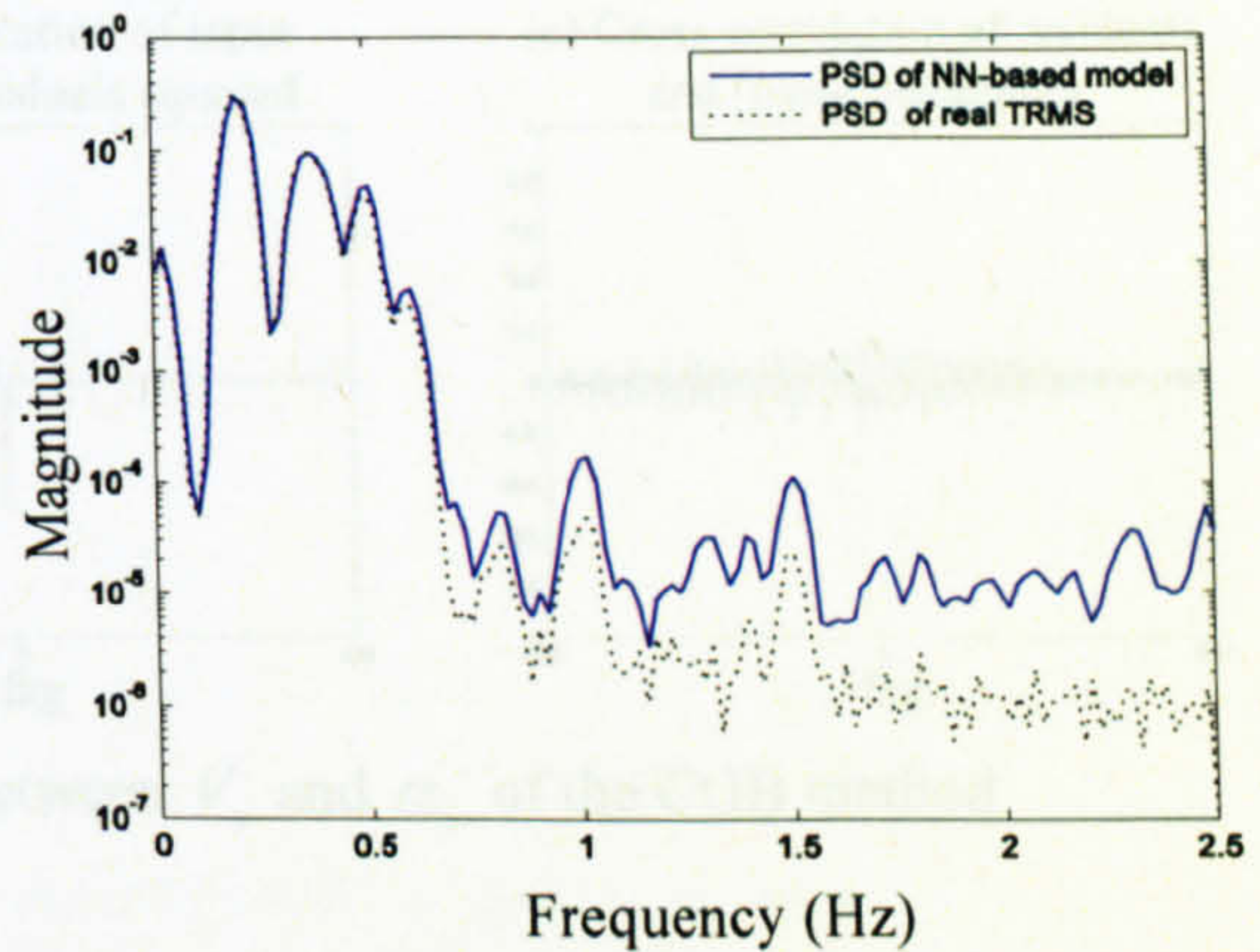
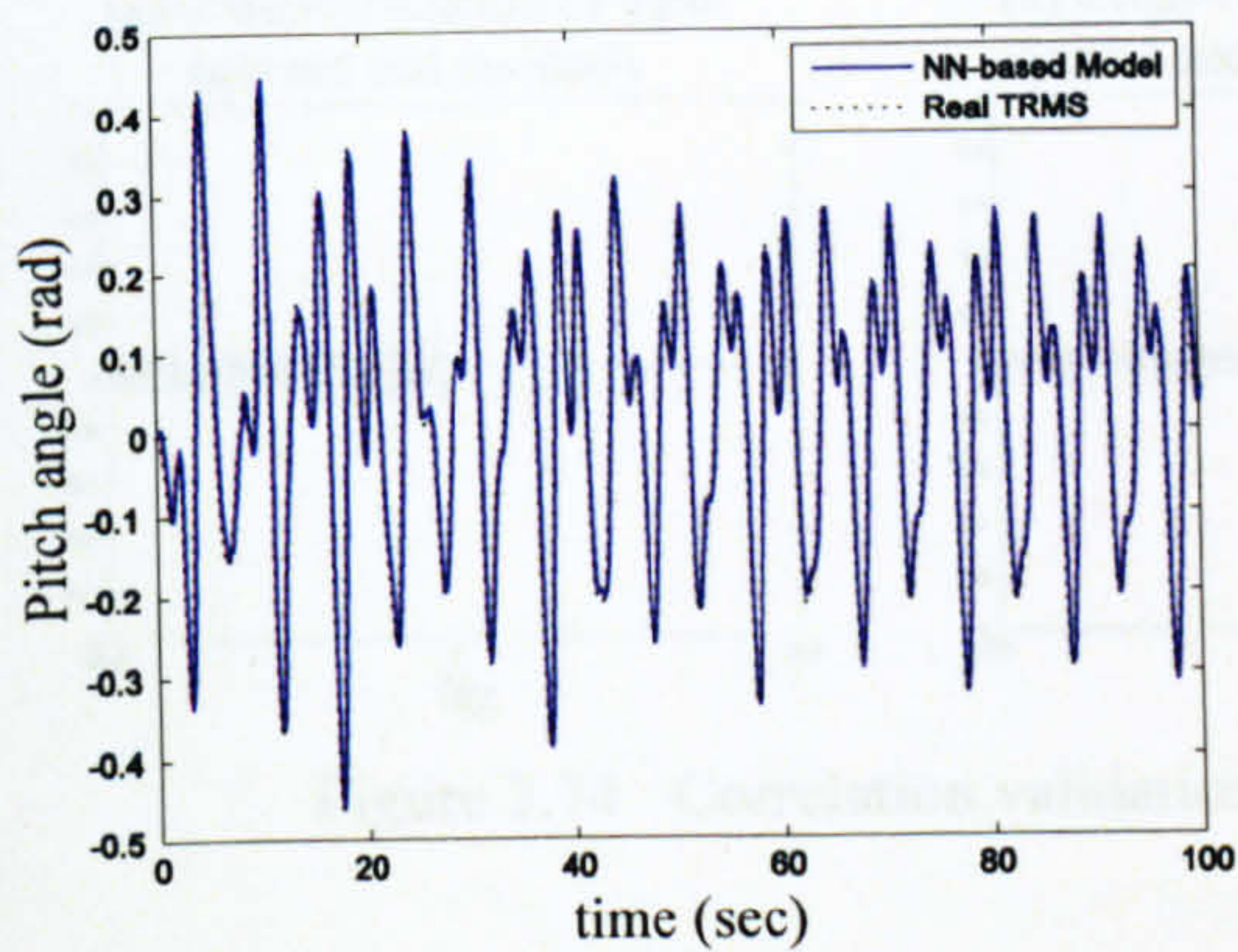


Figure 3.70 Pitch angle response with the BR learning method

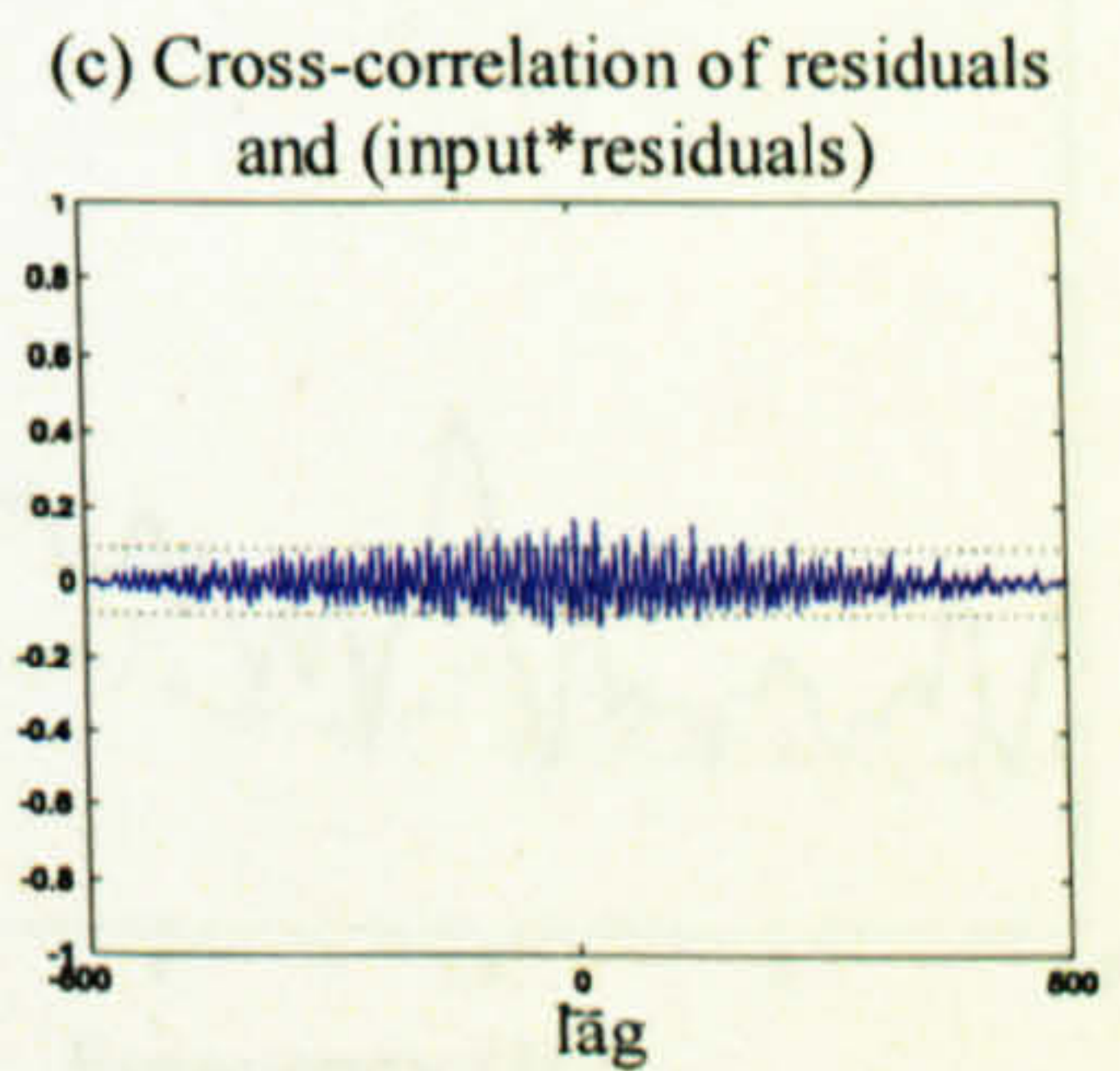
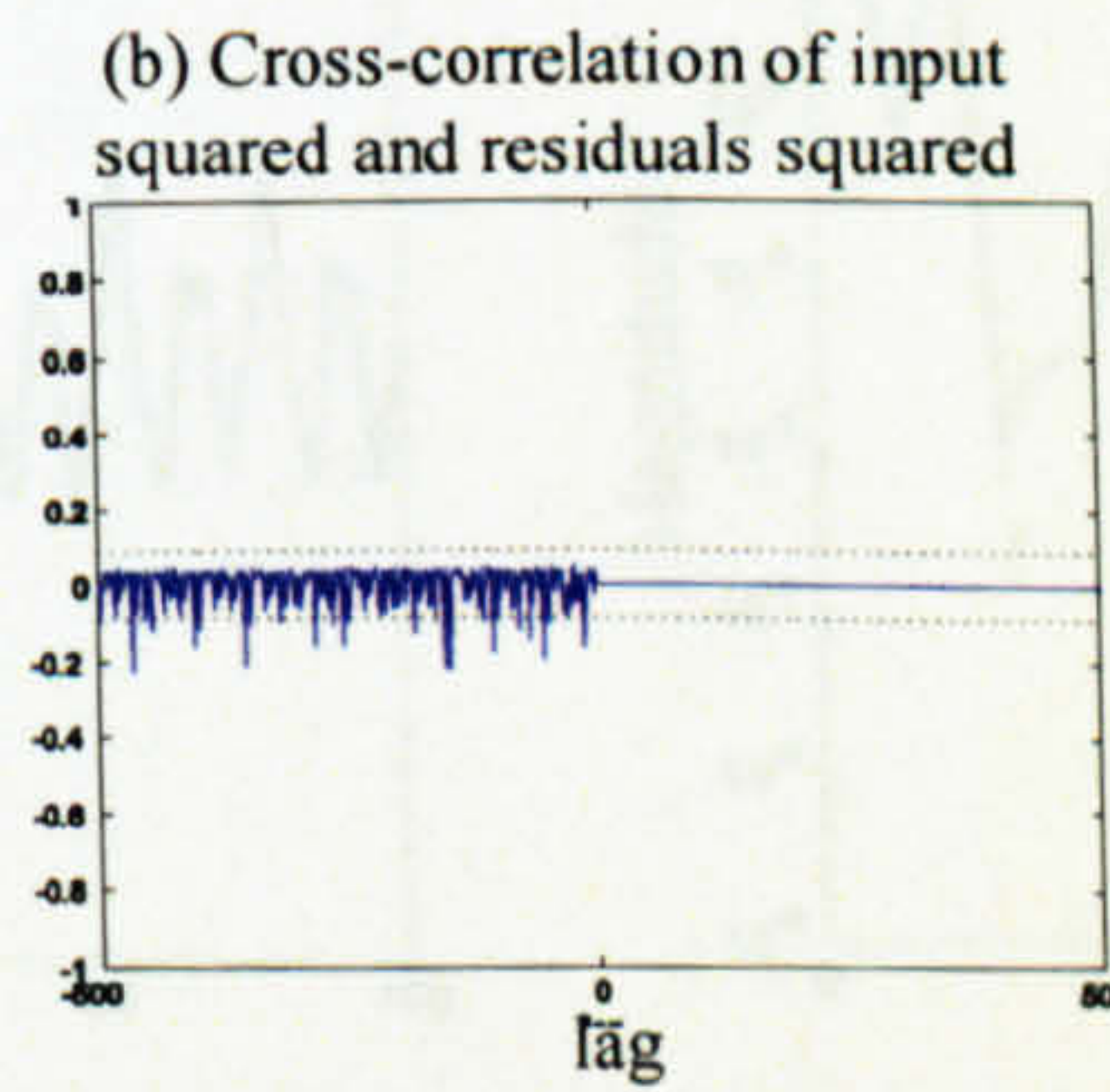
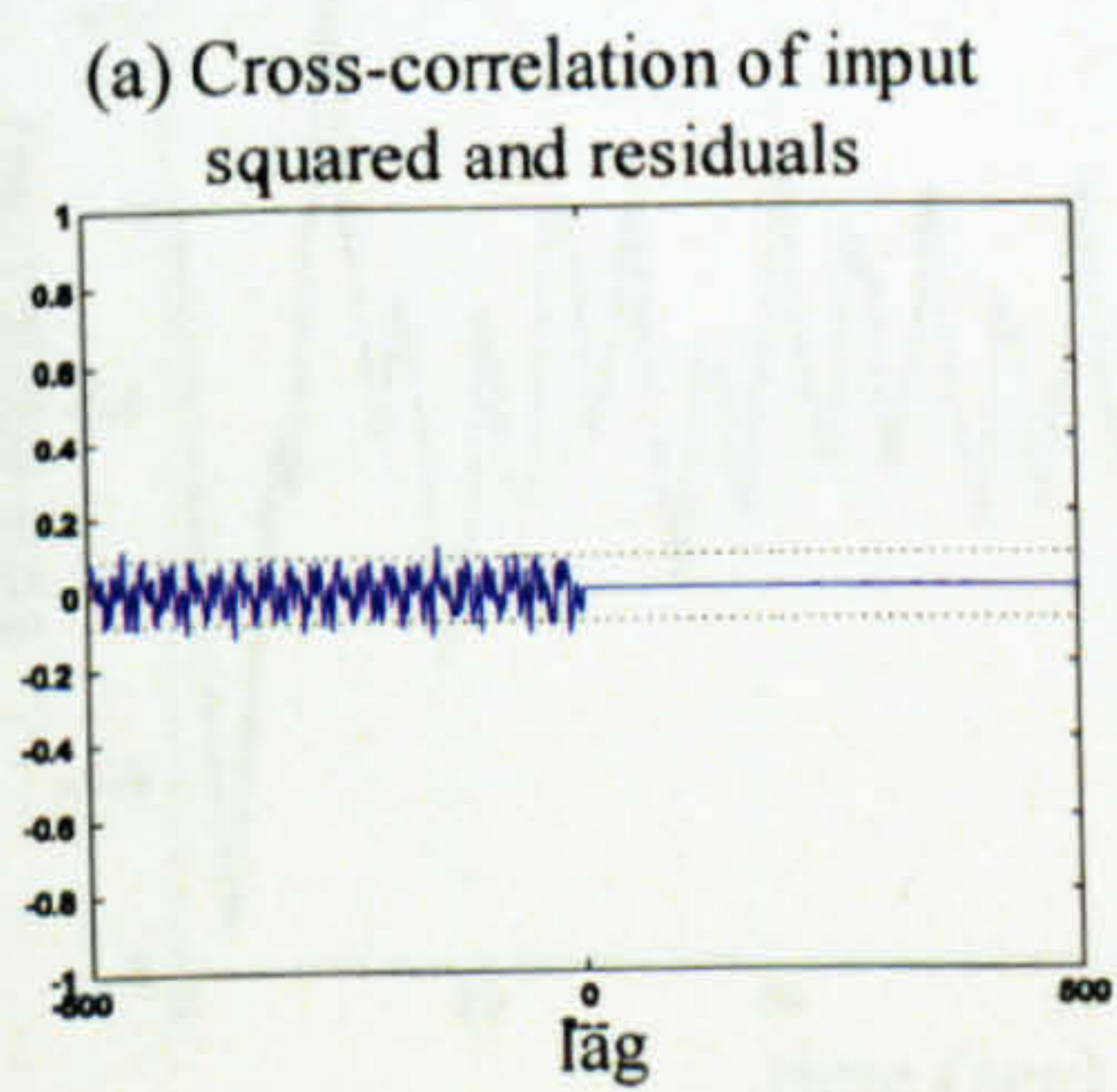


Figure 3.71 Correlation validation between V_v and α_v of the BR method

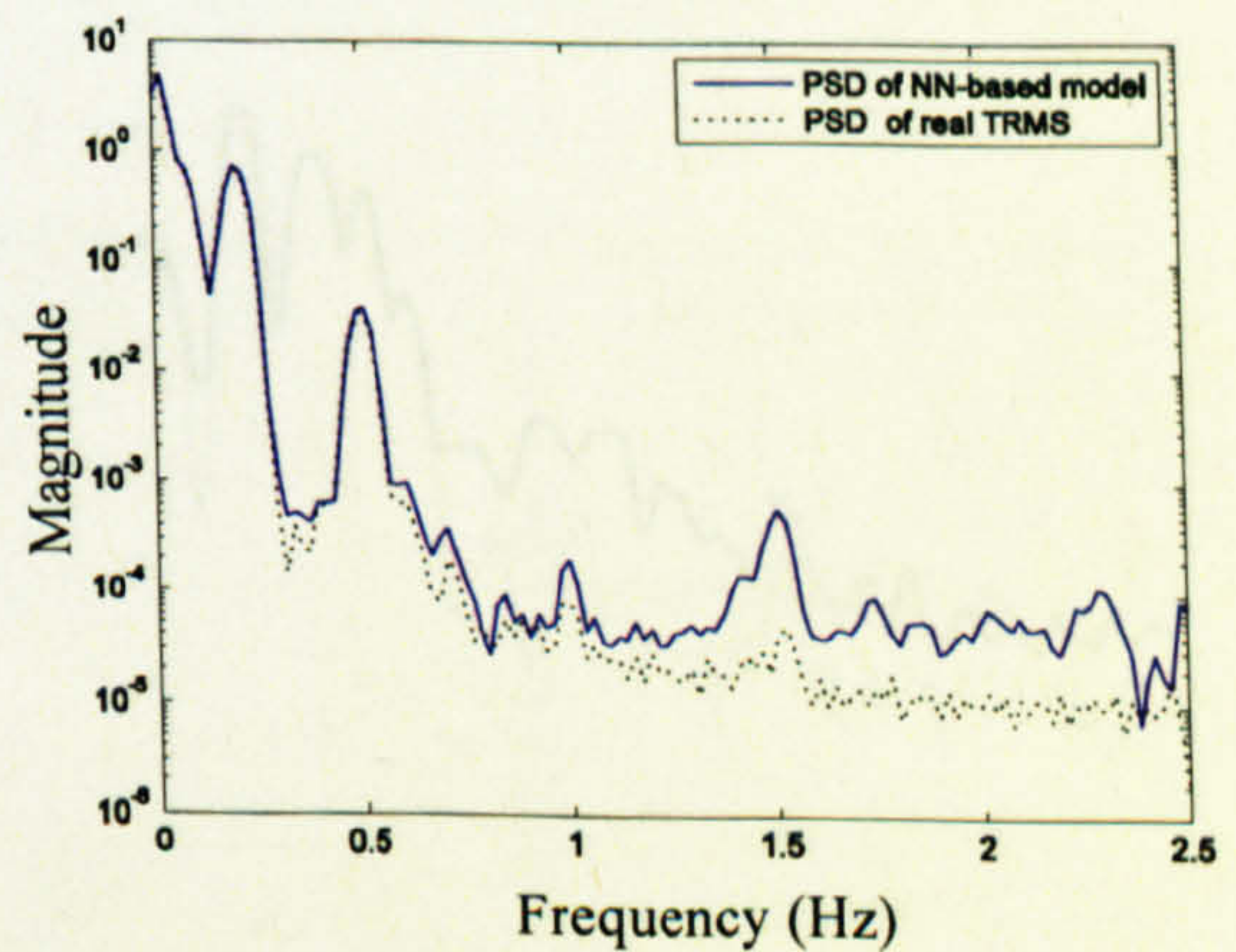
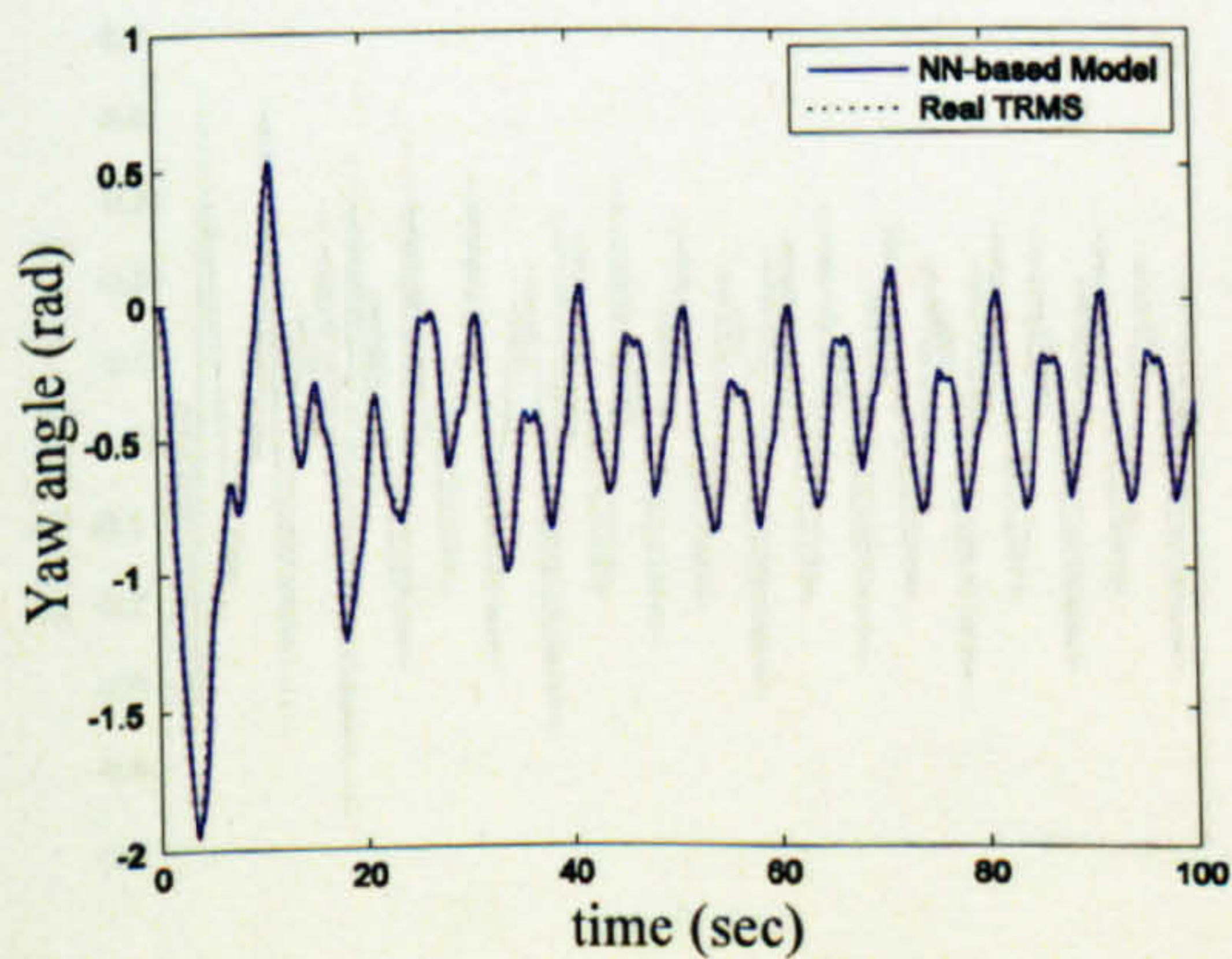


Figure 3.72 Yaw angle response with the CGB learning method

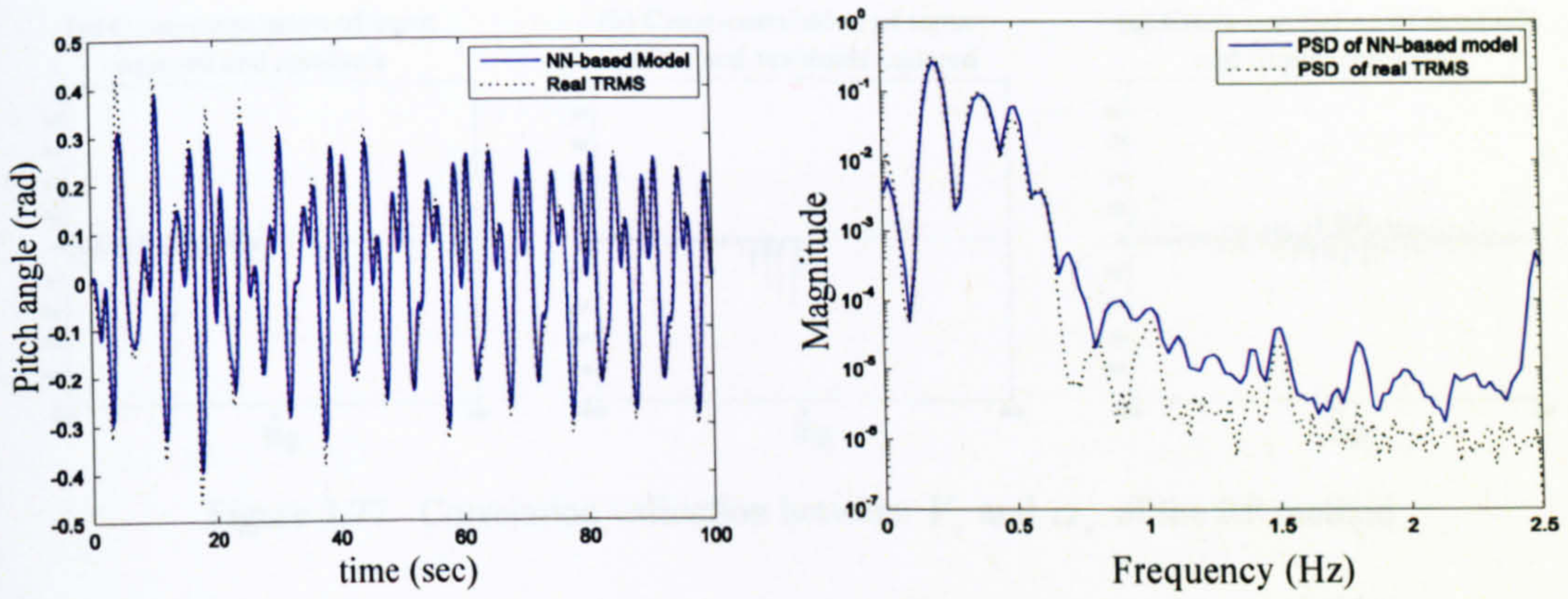


Figure 3.73 Pitch angle response with the CGB learning method

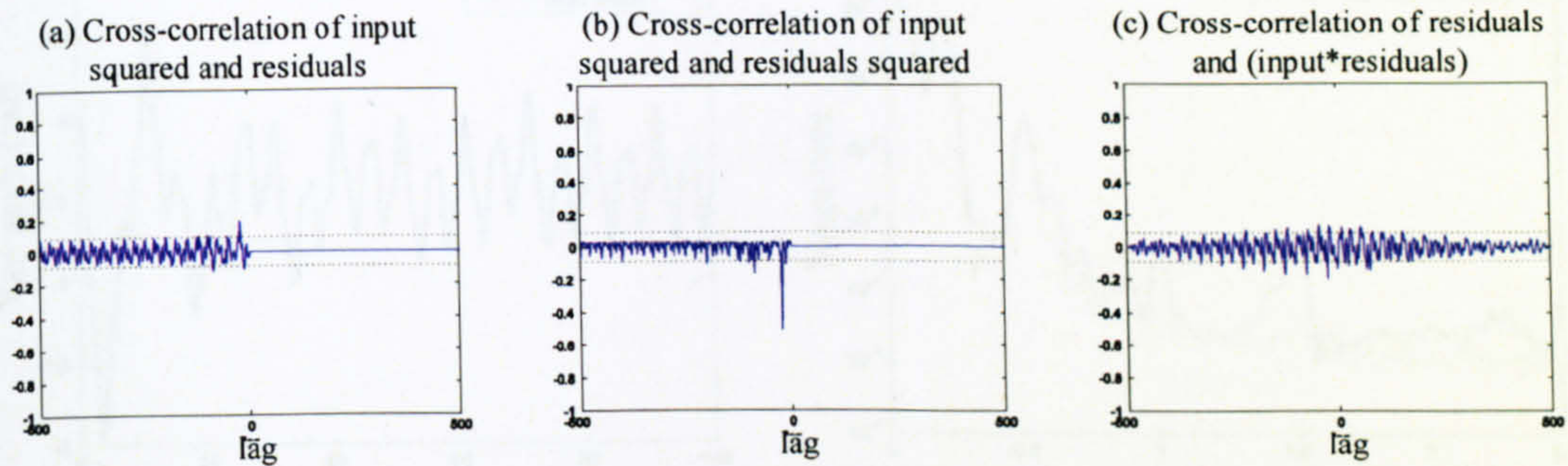


Figure 3.74 Correlation validation between V_v and α_v of the CGB method

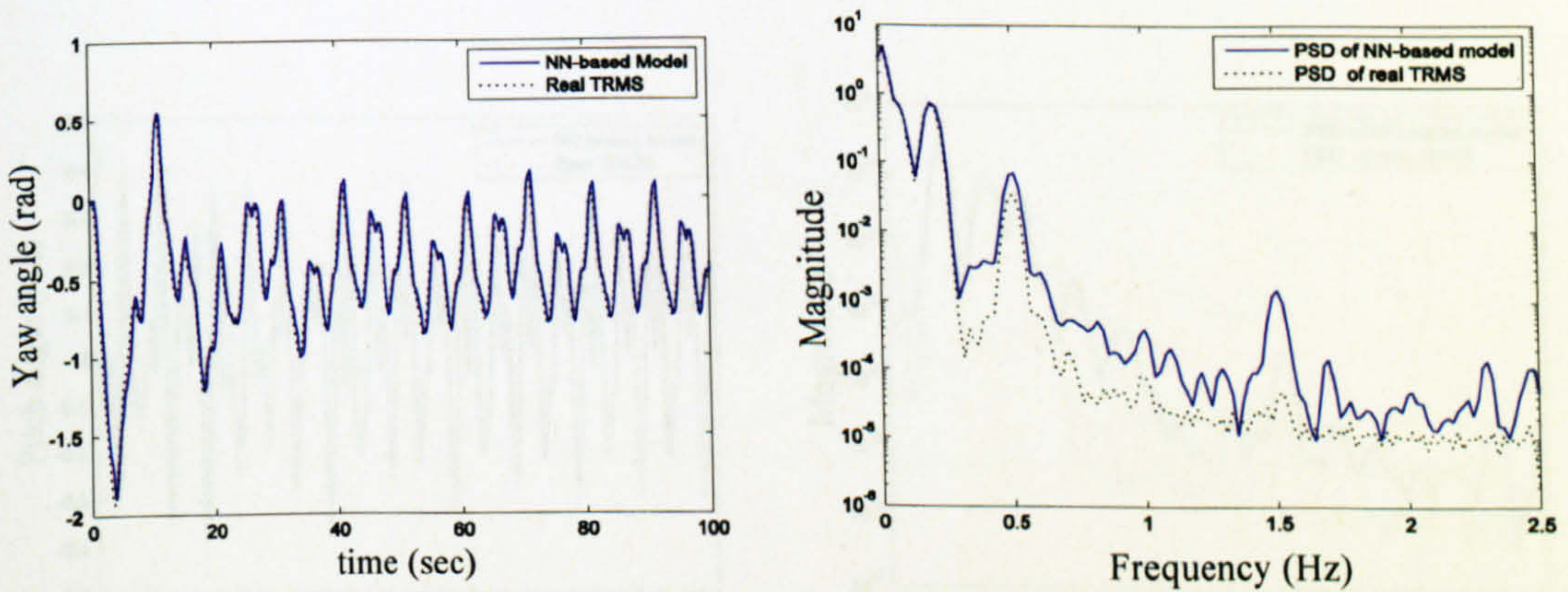


Figure 3.75 Yaw angle response with the RP learning method

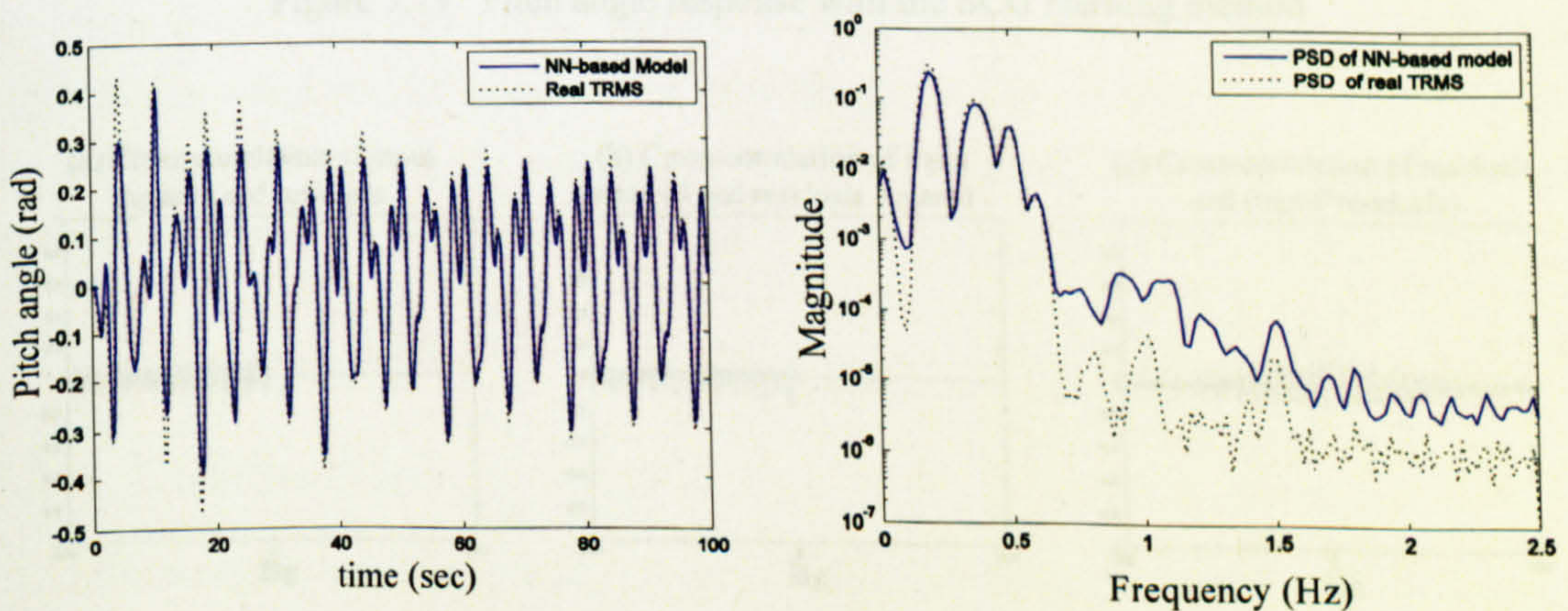


Figure 3.76 Pitch angle response with the RP learning method

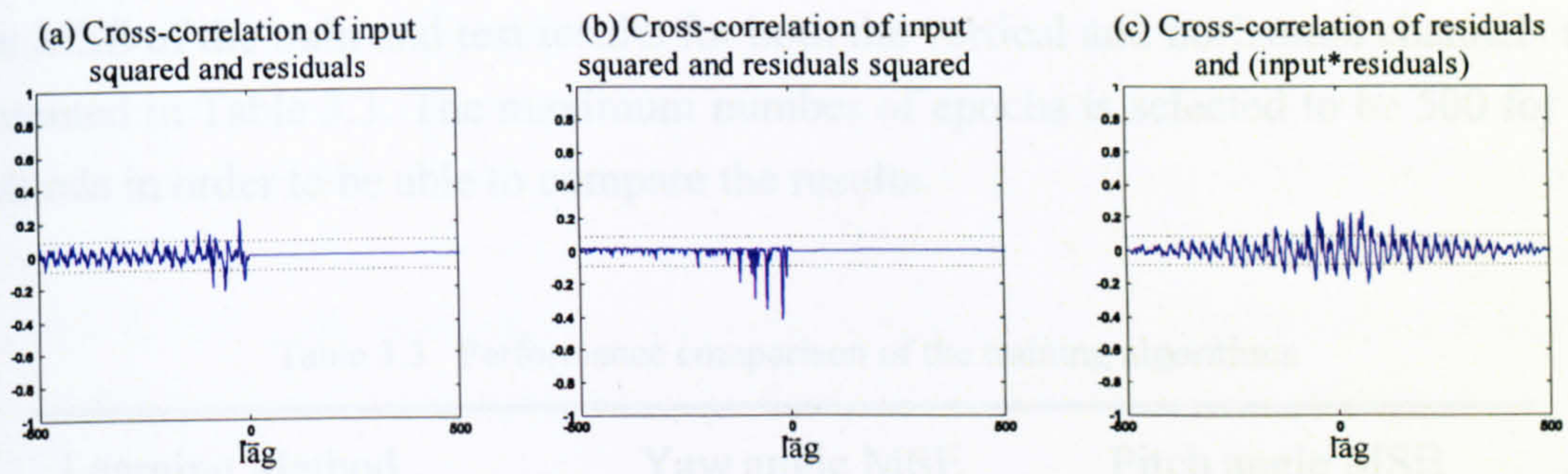


Figure 3.77 Correlation validation between V_v and α_v of the RP method

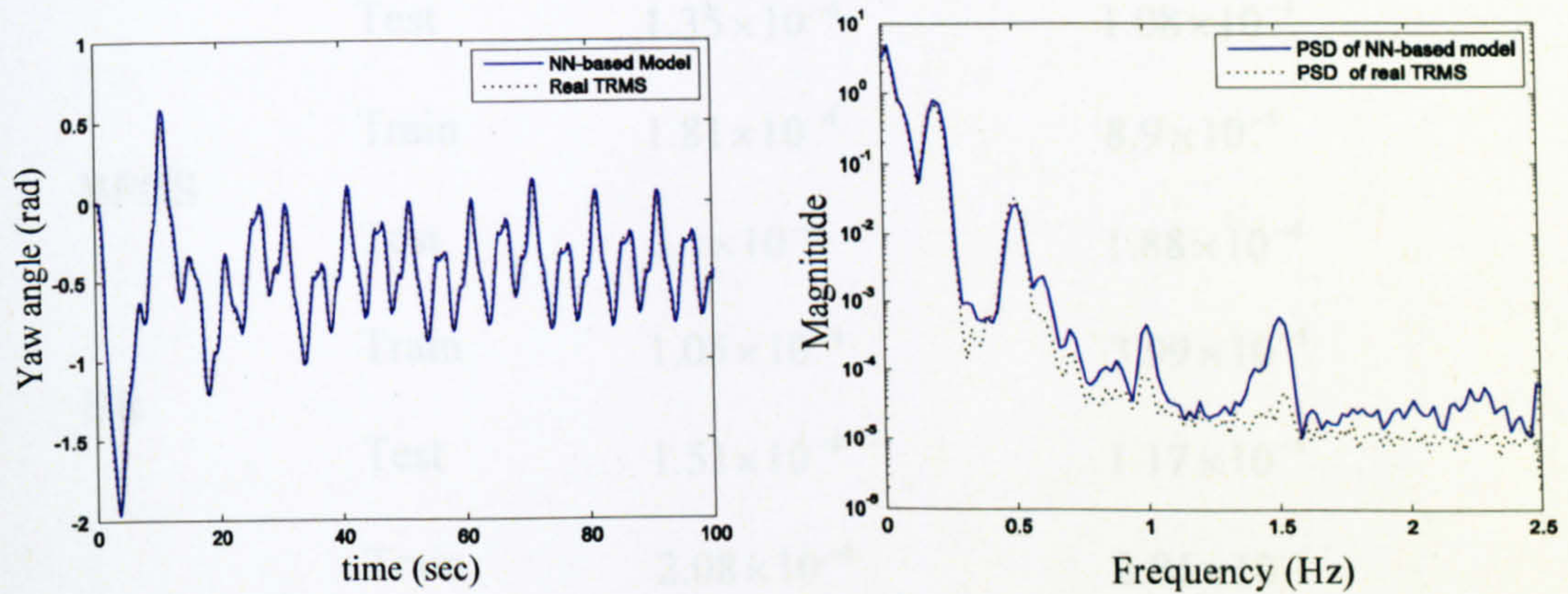


Figure 3.78 Yaw angle response with the SCG learning method

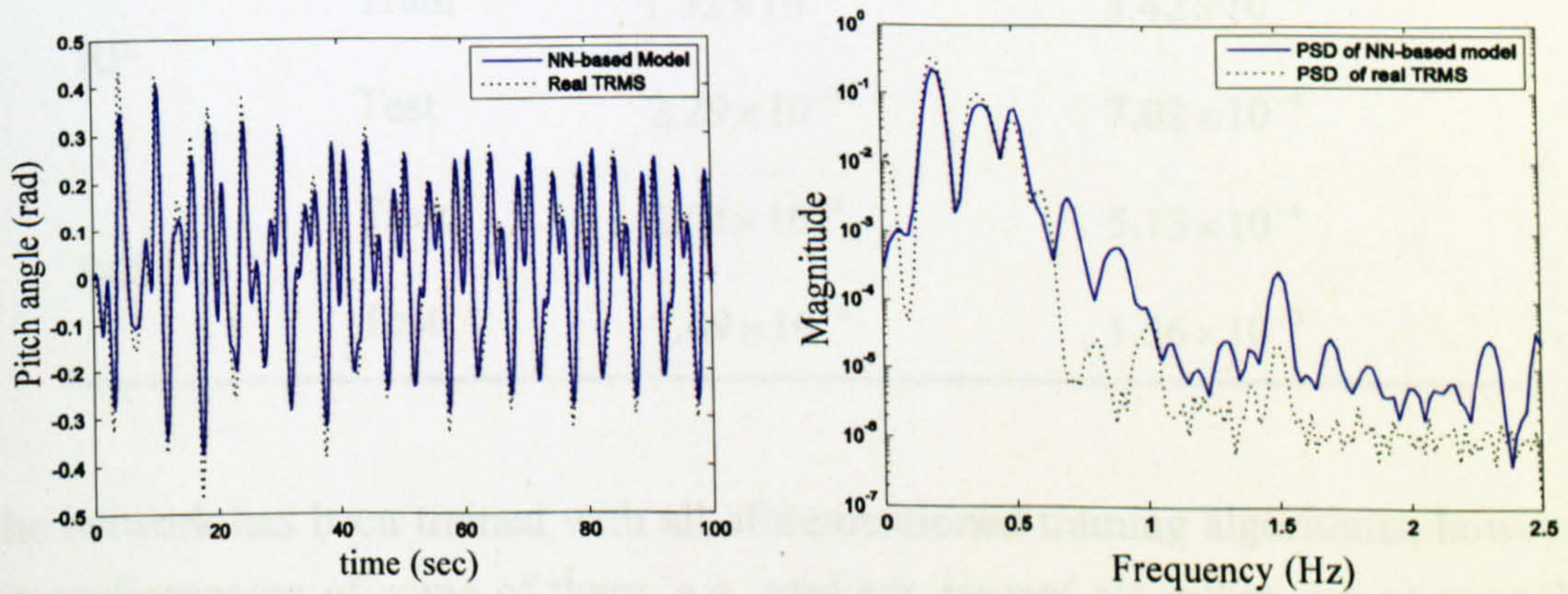


Figure 3.79 Pitch angle response with the SCG learning method

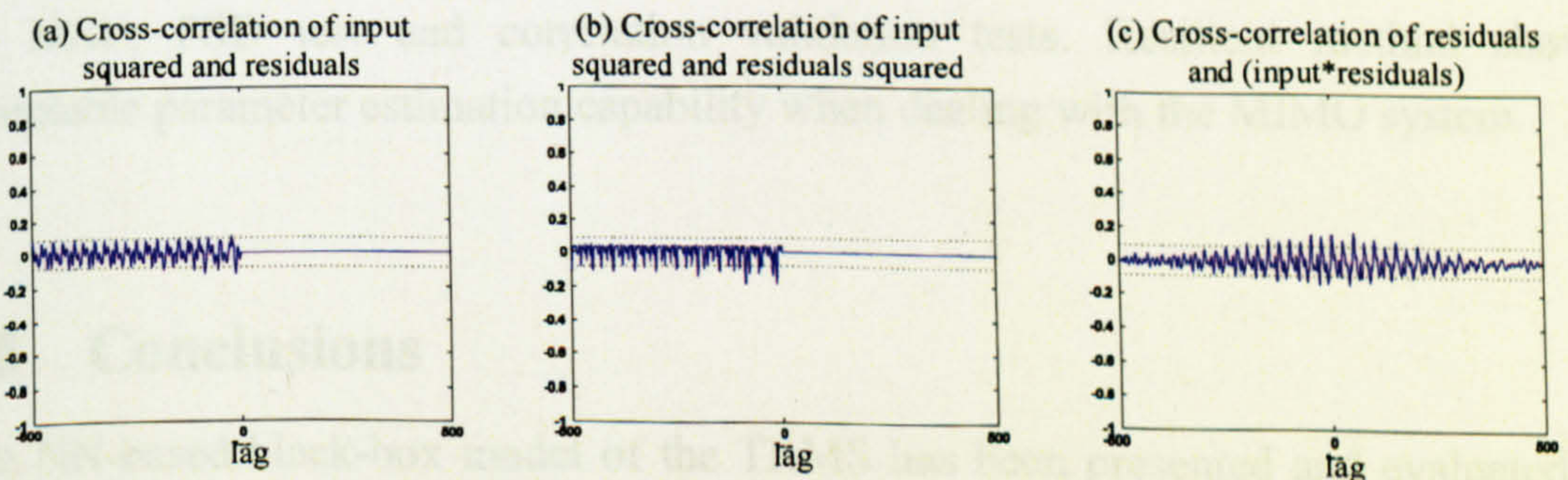


Figure 3.80 Correlation validation between V_v and α_v of the SCG method

The MSE of the train and test results for both the vertical and horizontal channels are presented in Table 3.3. The maximum number of epochs is selected to be 500 for all methods in order to be able to compare the results.

Table 3.3 Performance comparison of the training algorithms

Learning Method		Yaw angle MSE	Pitch angle MSE
LM	Train	1.21×10^{-4}	4.66×10^{-5}
	Test	1.35×10^{-4}	1.08×10^{-4}
BFGS	Train	1.81×10^{-4}	8.9×10^{-5}
	Test	3.1×10^{-4}	1.88×10^{-4}
BR	Train	1.05×10^{-4}	3.99×10^{-5}
	Test	1.51×10^{-4}	1.17×10^{-4}
CGB	Train	2.08×10^{-4}	2.81×10^{-4}
	Test	3.76×10^{-4}	7.79×10^{-4}
RP	Train	1.52×10^{-3}	3.42×10^{-4}
	Test	2.29×10^{-3}	7.02×10^{-4}
SCG	Train	2.51×10^{-4}	5.13×10^{-4}
	Test	7.69×10^{-4}	1.56×10^{-3}

The network has been trained with all aforementioned training algorithms; however, the performances of some of them, e.g. gradient descent algorithm, are so poor that their results are not presented here. Clearly, Levenberg-Marquardt and Bayesian regularisation algorithms outperform their rivals on the basis of the decisive factors, i.e. MSE, PSD test and correlation validation tests. Resilient method shows acceptable parameter estimation capability when dealing with the MIMO system.

3.5 Conclusions

The NN-based black-box model of the TRMS has been presented and evaluated in this chapter. It can be observed from the results that the error of the empirical model

using efficient learning algorithms is less than that of the analytical model. Although the analytical model is less accurate, it is generally valid for any input signal with any frequency. The empirical model is only valid in the range of the training data provided. Also, in the case of black-box modelling, the structure of the process is lost and this is one of the drawbacks of empirical modelling. The significant advantage of NN-based modelling techniques is that the model is easy and fast to develop.

CHAPTER 4

Nonlinear Grey-box Modelling

This chapter presents a grey-box modelling approach for the TRMS using GAs. The dynamic equations of the system in terms of its 2DOF dynamics, developed using the Newtonian method in Chapter 2, are used here. In order to improve the accuracy of the white-box model, the estimated parameters are retuned using a GA-based optimisation approach. A set of data is extracted from the real TRMS, and the objective has been to minimise the error between the real and the model outputs. The estimated parameters of the system are utilised as the initial populations of the GA to lead to the convergence faster. The performances of the white-box and the grey-box models are compared with respect to each other to validate the improvement of the grey-box model.

4.1 Introduction

Most of the industrial systems are nonlinear, but they are often presented by a set of linear differential equations such as state-space equations for simplicity. Since differential equations are usually derived from the physical operational mechanism of a system, they should accommodate its nonlinear description to represent an accurate model. However, a practical system may contain unknown parameters that appear in modelling and should be identified as accurately as possible. It is thus often impossible to measure or calculate every parameter to build an accurate white-box model for a practical system. If a black-box model is attempted instead, structural information of the system is lost.

The underlying dynamics of a practical engineering system and some of its physical parameters are usually known *a-priori*. However, some of the parameters cannot be precisely obtained with a white-box modelling approach, due to the system nonlinearity, complexity and constraints on the physical measurement. The system is, thus, a partially-known system and may be best modelled as a grey-box. Note that obtaining accurate differential equations that have a focused dynamic structure is

desired in many applications. Hence, a grey-box model should explicitly utilise prior knowledge the way it is used to derive white-box dynamic structure using physical laws. This forms the white part of the grey-box, and the unknown parameters can be obtained through an appropriate optimisation process.

Such an identification problem is, in general, a nonlinear optimisation problem in a multi-dimensional and, usually, multi-modal space. The optimisation task is difficult to be performed by conventional optimisation techniques, since they require a smooth search space or a differentiable error objective function. In addition, these conventional gradient-based techniques may encounter difficulties due to noisy data or only offer local optima. These numerical difficulties can, however, be overcome by meta-heuristic optimisation algorithms such as GAs that intelligently explore a poorly understood solution space in parallel by evolutionary trials without the need for differentiating the index or for linearly separable parameters [33]. Such non-deterministic methods have been shown to be able to search for optimal parameters and structures simultaneously and to be very efficient and powerful in solving high order systems and control engineering problems.

4.2 Analytical Equations of the TRMS

For the sake of simplicity, the developed nonlinear state-space equations of the system in the previous chapters are represented here.

$$\frac{di_{ah}}{dt} = -\frac{R_{ah}}{L_{ah}}i_{ah} - \frac{k_{ah}\varphi_h}{L_{ah}}\omega_h + \frac{k_1}{L_{ah}}U_h \quad (4.1)$$

$$\frac{d\omega_h}{dt} = \frac{k_{ah}\varphi_h}{J_{tr}}i_{ah} - \frac{B_{tr}}{J_{tr}}\omega_h - \frac{f_1(\omega_h)}{J_{tr}} \quad (4.2)$$

$$\frac{dS_h}{dt} = \frac{l_1 f_2(\omega_h) \cos \alpha_v - k_{vfh} \Omega_h - f_3(\alpha_h)}{D \cos^2 \alpha_v + E \sin^2 \alpha_v + F} \quad (4.3)$$

$$\frac{d\alpha_h}{dt} = S_h + \frac{k_m \omega_v \cos \alpha_v}{D \cos^2 \alpha_v + E \sin^2 \alpha_v + F} \quad (4.4)$$

$$\frac{di_{av}}{dt} = -\frac{R_{av}}{L_{av}}i_{av} - \frac{k_{av}\varphi_v}{L_{av}}\omega_v + \frac{k_2}{L_{av}}U_v \quad (4.5)$$

$$\frac{d\omega_v}{dt} = \frac{k_{av}\varphi_v}{J_{mr}}i_{av} - \frac{B_{mr}}{J_{mr}}\omega_v - \frac{f_4(\omega_v)}{J_{mr}} \quad (4.6)$$

$$\frac{dS_v}{dt} = \frac{l_m f_5(\omega_v) + k_g \Omega_h f_5(\omega_v) \cos \alpha_v - k_{vf} \Omega_v}{J_v} + \frac{g[(A-B) \cos \alpha_v - C \sin \alpha_v] - 0.5 \Omega_h^2 H \sin 2\alpha_v}{J_v} \quad (4.7)$$

$$\frac{d\alpha_v}{dt} = S_v + \frac{k_t}{J_v} \omega_h \quad (4.8)$$

where,

$$f_1(\omega_h) = \begin{cases} k_{ihp} \omega_h^2 & \text{if } \omega_h \geq 0 \\ -k_{ihn} \omega_h^2 & \text{if } \omega_h < 0 \end{cases}$$

$$f_2(\omega_h) = \begin{cases} k_{fhp} \omega_h^2 & \text{if } \omega_h \geq 0 \\ -k_{fhn} \omega_h^2 & \text{if } \omega_h < 0 \end{cases}$$

$$f_3(\alpha_h) = \begin{cases} k_{chp} \alpha_h & \text{if } \alpha_h \geq 0 \\ k_{chn} \alpha_h & \text{if } \alpha_h < 0 \end{cases}$$

$$f_4(\omega_v) = \begin{cases} k_{ivp} \omega_v^2 & \text{if } \omega_v \geq 0 \\ -k_{ivn} \omega_v^2 & \text{if } \omega_v < 0 \end{cases}$$

$$f_5(\omega_v) = \begin{cases} k_{fvp} \omega_v^2 & \text{if } \omega_v \geq 0 \\ -k_{fvn} \omega_v^2 & \text{if } \omega_v < 0 \end{cases}$$

$$\Omega_h = S_h + \frac{k_m \omega_v \cos \alpha_v}{D \cos^2 \alpha_v + E \sin^2 \alpha_v + F}$$

$$\Omega_v = S_v + \frac{k_t \omega_h}{J_v}$$

$$A = \left(\frac{m_t}{2} + m_{tr} + m_{ts} \right) l_t$$

$$B = \left(\frac{m_m}{2} + m_{mr} + m_{ms} \right) l_m$$

$$C = \left(\frac{m_b}{2} l_b + m_{cb} l_{cb} \right)$$

$$D = \left(\frac{m_m}{3} + m_{mr} + m_{ms} \right) l_m^2 + \left(\frac{m_t}{3} + m_{tr} + m_{ts} \right) l_t^2$$

$$E = \frac{m_b}{3} l_b^2 + m_{cb} l_{cb}^2$$

$$F = m_{ms} r_{ms}^2 + \frac{m_{ts}}{2} r_{ts}^2$$

$$H = A l_i + B l_m - \frac{m_b}{2} l_b^2 - m_{cb} l_{cb}^2$$

Note that it is assumed that the interface circuits between the computer and the TRMS for the two channels are linear and their gains are k_1 and k_2 for the horizontal and the vertical channels respectively. For their nonlinear relationships, see Chapter 2. Also, the viscous friction torques of the beam in the horizontal and the vertical planes have been considered, and the Coulomb and static frictions have been neglected in this study.

4.3 GA-based Parameters Identification

The grey-box approach to be employed to model the system combines the merits of both the white- and black-box modelling approaches. The approach outperforms each of the white-box and the black-box modelling approach used separately. In doing so, the nonlinear state-space equations of the system have been extracted considering all the effective forces. The measurable parameters have then been measured as precisely as possible and the unknown parameters have been estimated off-line using a powerful biologically inspired optimisation method, a GA. Note that before executing the optimisation program it is necessary to initialise a reasonable range, upper and lower limits, for each of the unknown parameters. A set of data has been extracted from the real TRMS and the objective is to minimise the weighted sum squared error between the real and the model outputs. The unknown parameters have been estimated by physical description of the system to be used as the initial values of the optimisation methods to speed up the convergence procedure. Figure 4.1 shows the block diagram of the iterative identification procedure.

From an operational perspective, a GA comprises two basic elements: a set of individuals, i.e. potential solutions (the population) and a set of biologically inspired operators active over the population. A new set of solutions is created at each generation, by the process of selecting individuals according to their level of fitness in the problem domain and breeding them together using the operators. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals that they were created from, just as in natural adaptation [121]. The flowchart of a simple GA is shown in Figure 4.2.

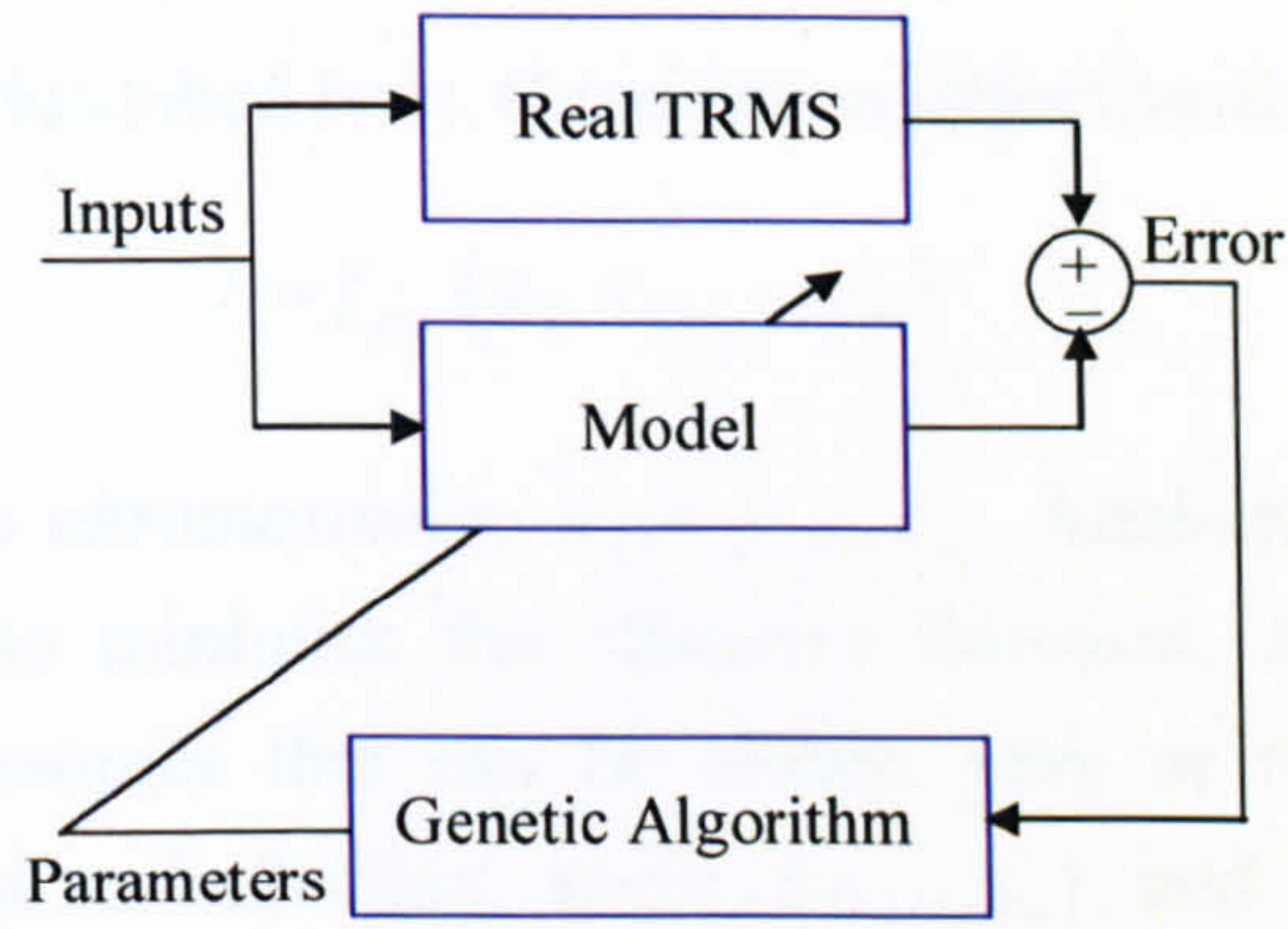


Figure 4.1 The block diagram of the iterative identification procedure

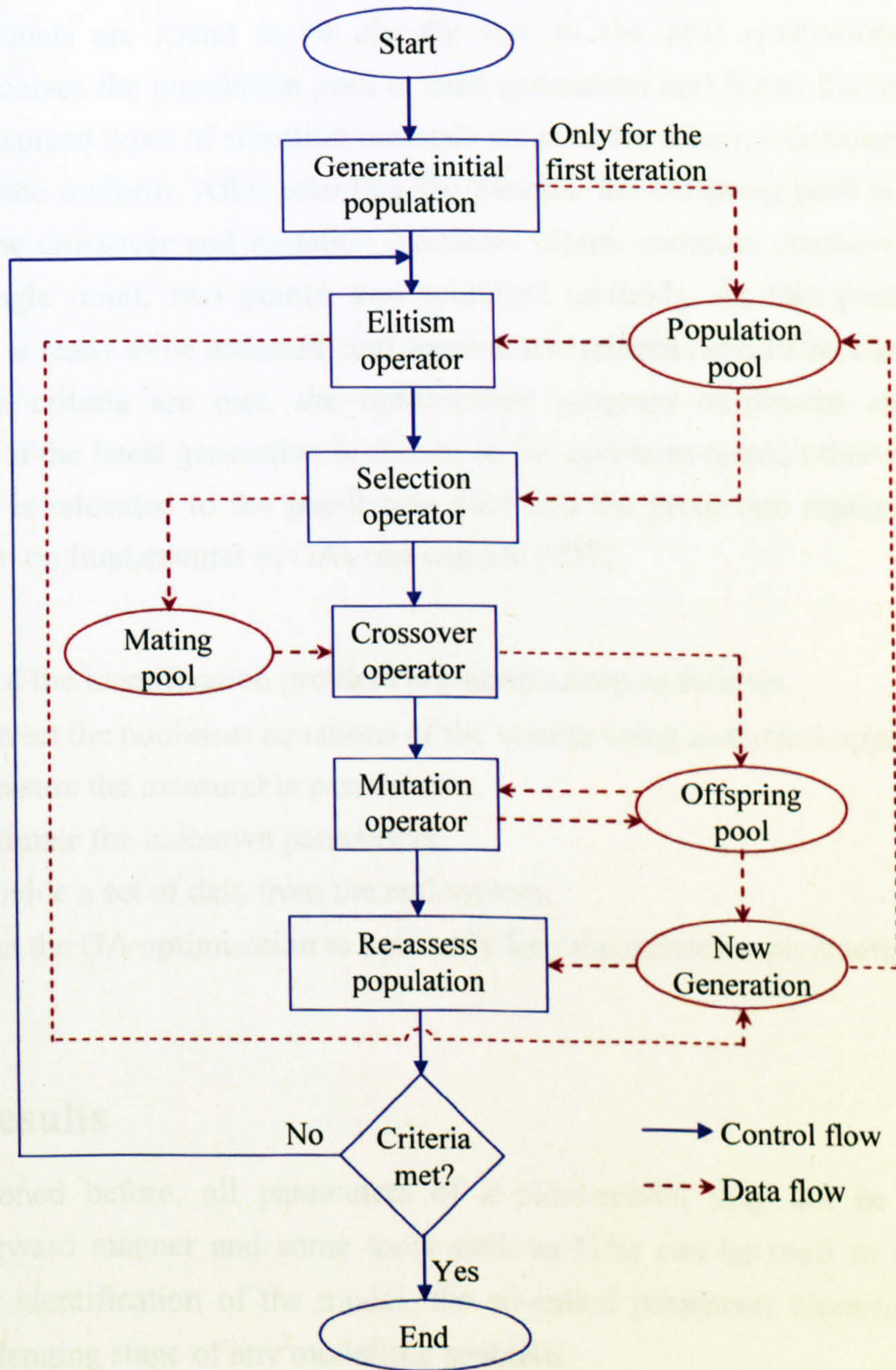


Figure 4.2 The control and data flow of a GA

The process is briefly described here. Consider an objective function as,

$$J = f_{obj} (x_1, x_2, \dots, x_n) \quad (4.9)$$

which depends on the *chromosomes*, x_1, x_2, \dots, x_n . Analogous to any optimisation problem, the aim is to minimise the objective function, J , by finding optimum values of the chromosomes that can be binary, grey or real codes. A vector of chromosomes is called an *individual*, $\mathbf{x} = [x_1, x_2, \dots, x_n]$, and a number of individuals form a *population*. At each *generation* a new set of individuals is created to have probably higher fitness values compared to the previous generation. The first stage of a GA optimisation is generating an initial population. Using the elitism operator, the best individuals are found to be directly sent to the next generation. Selection operator receives the population pool at each generation and forms the mating pool. Some widespread types of selection methods are roulette wheel, tournament, uniform and stochastic uniform. After selecting the parents, the offspring pool is created by applying the crossover and mutation operators. Some common crossover functions include single point, two points, and scattered methods. At this point, the new generation is ready to be assessed, and termination criteria need to be checked. If the termination criteria are met, the optimisation program terminates and the best individual of the latest generation is chosen as the optimum result, otherwise the new generation is relocated to the population pool and the procedure repeats. For more information on fundamental of GAs one can see [121].

The steps of the identification problem are summarised as follows,

- Extract the nonlinear equations of the system using analytical approaches,
- Measure the measurable parameters,
- Estimate the unknown parameters,
- Provide a set of data from the real system,
- Run the GA optimisation to optimally find the unknown parameters.

4.4 Results

As mentioned before, all parameters of a plant-model may not be found in a straightforward manner and some tools such as GAs can be used to facilitate the parameter identification of the model; the so-called parameter identification is the most challenging stage of any modelling problem.

In this investigation, the unknown parameters, chromosomes, are assumed to be k_{fhp} , k_{fhn} , k_{fvp} , k_{fvn} , k_{thp} , k_{thn} , k_{tvp} , k_{tvn} , k_{chp} , k_{chn} , k_{vfh} , k_{vfv} , k_t , k_m , k_g , J_{tr} , J_{mr} , B_{tr} , B_{mr} , J_v . Although some of these parameters can be calculated using physical approaches, in order to obtain a more accurate model, these parameters have also been considered to be optimally obtained. A vector containing a set of values for these 20 chromosomes forms an individual. In this work, the number of population at each generation is set to be 25 individuals, and the maximum number of generations is chosen to be 200. Note that the type of population, selection function, crossover function, and mutation function are real double vector code, roulette wheel, scattered, and uniform respectively. Number of elites, crossover probability, and mutation probability are set to be 2, 0.8, and 0.01 respectively. The MATLAB toolbox on GAs [122] is used to carry out the optimisation.

In order to validate the grey-box model, it has been tested with various input signals. The response of the developed grey-box and white-box models in terms of the 2DOF dynamics of the system are presented in Figures 4.3 to 4.24. In some cases, both white-box and grey-box models demonstrate perfect responses; therefore, only the results with observable differences are shown here. For instance, according to Figures 4.3 and 4.4, the grey-box approach outperforms white-box method; however, no significant improvement can be observed in the case of Figures 4.11 and 4.12. It is clearly demonstrated that the grey-box model has, overall, shown better performance compared to the white-box one. Note that all the figures have been zoomed in to make the improvement clearer.

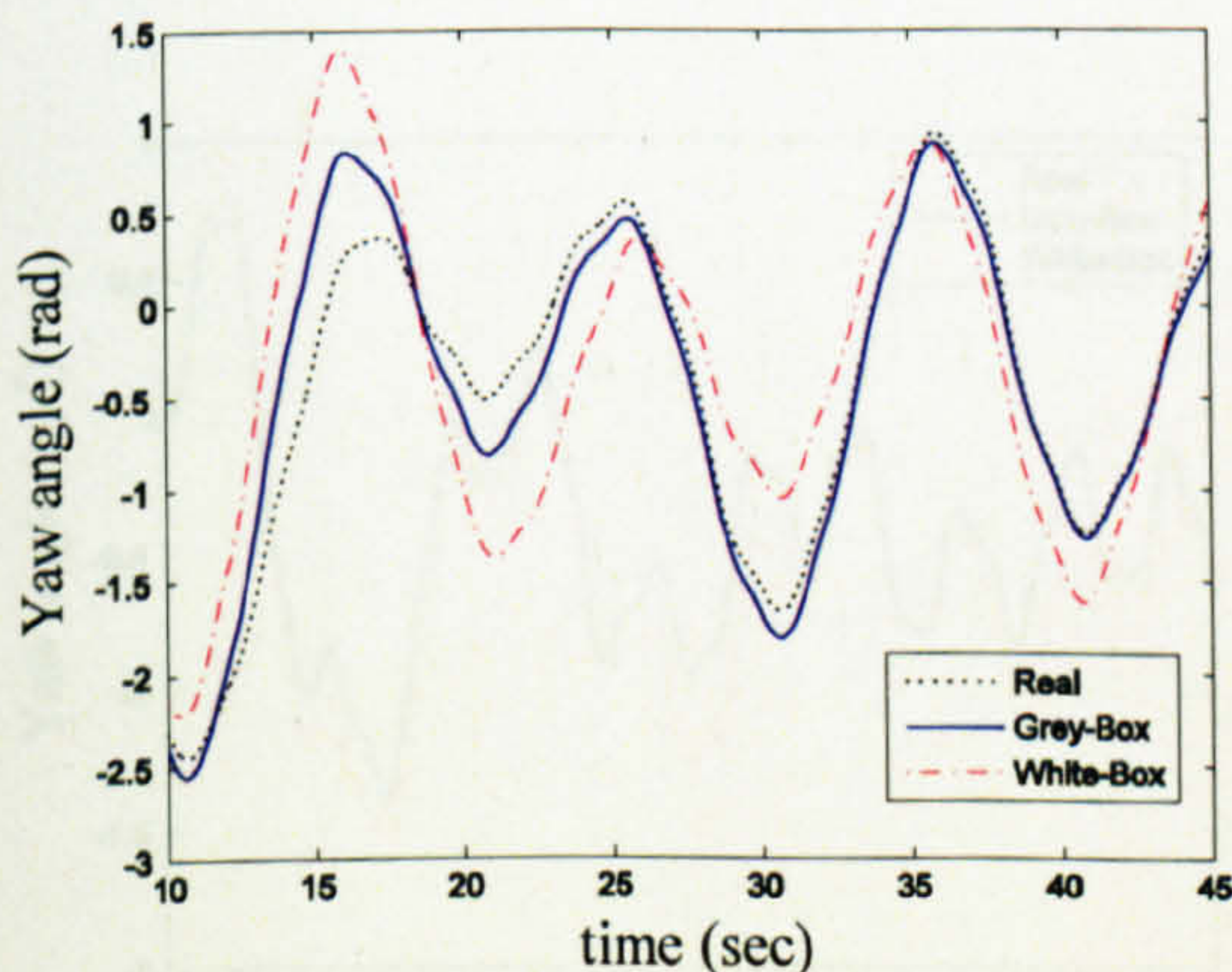


Figure 4.3 The yaw angle of the beam (case 1)

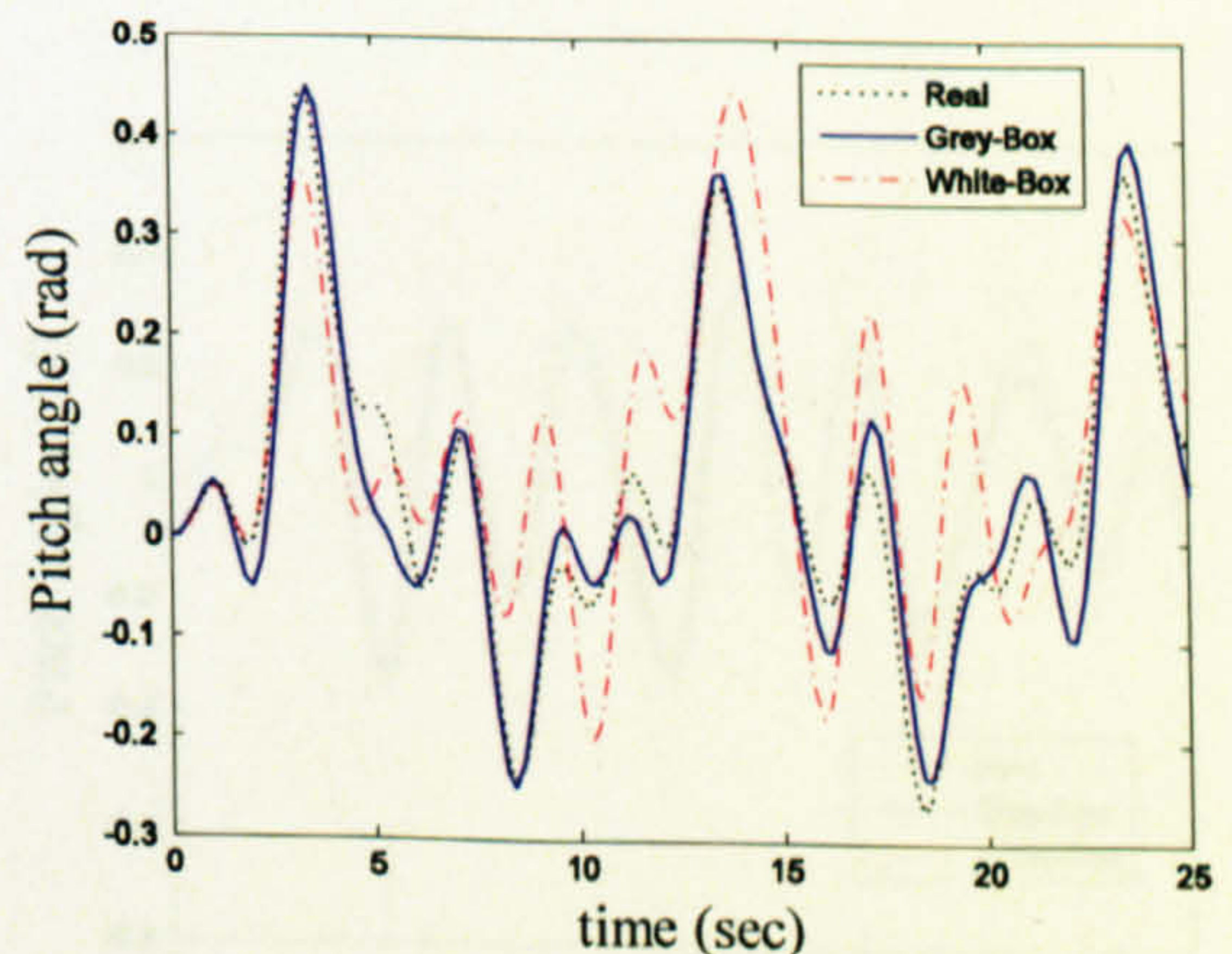


Figure 4.4 The pitch angle of the beam (case 1)

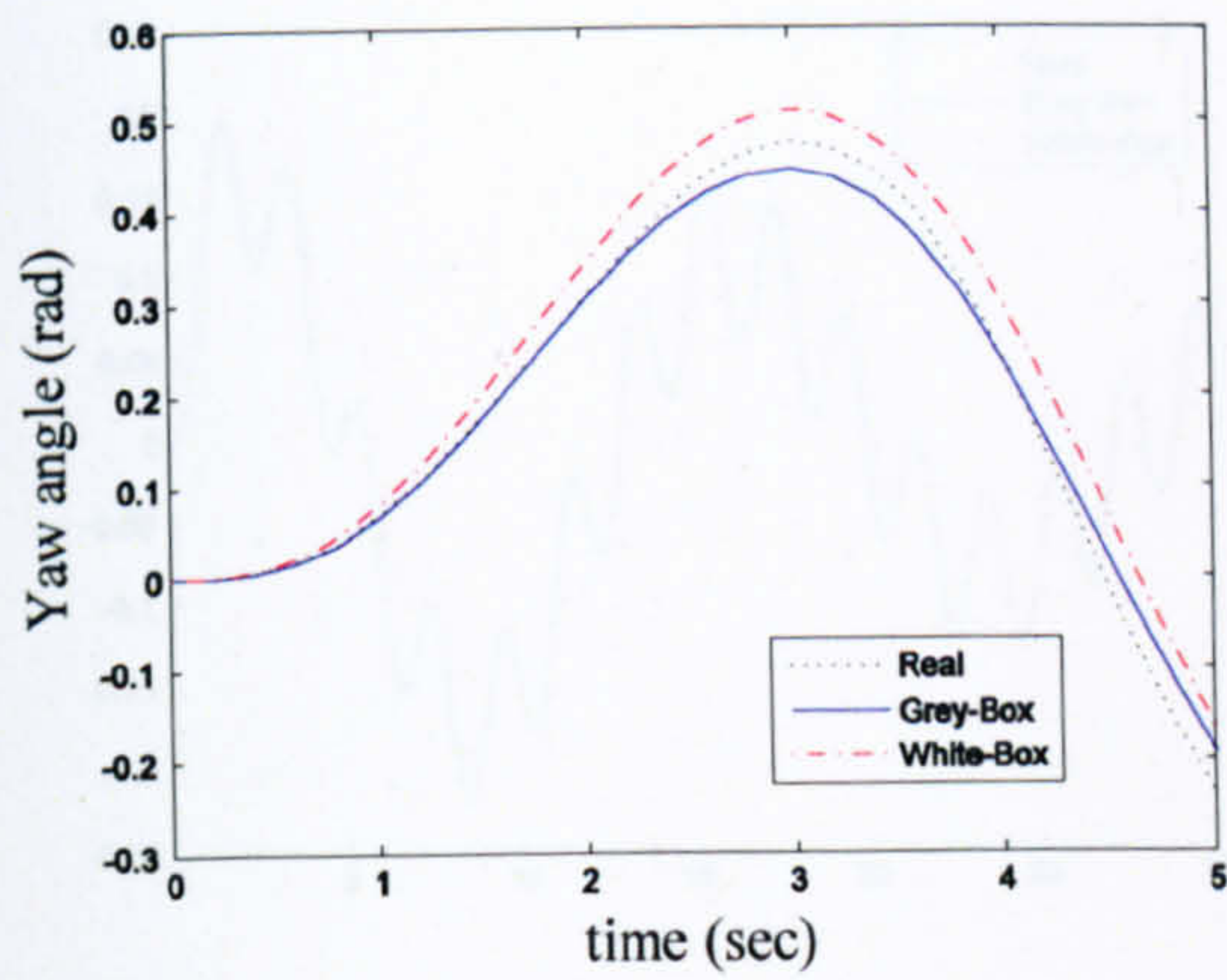


Figure 4.5 The yaw angle of the beam (case 2)

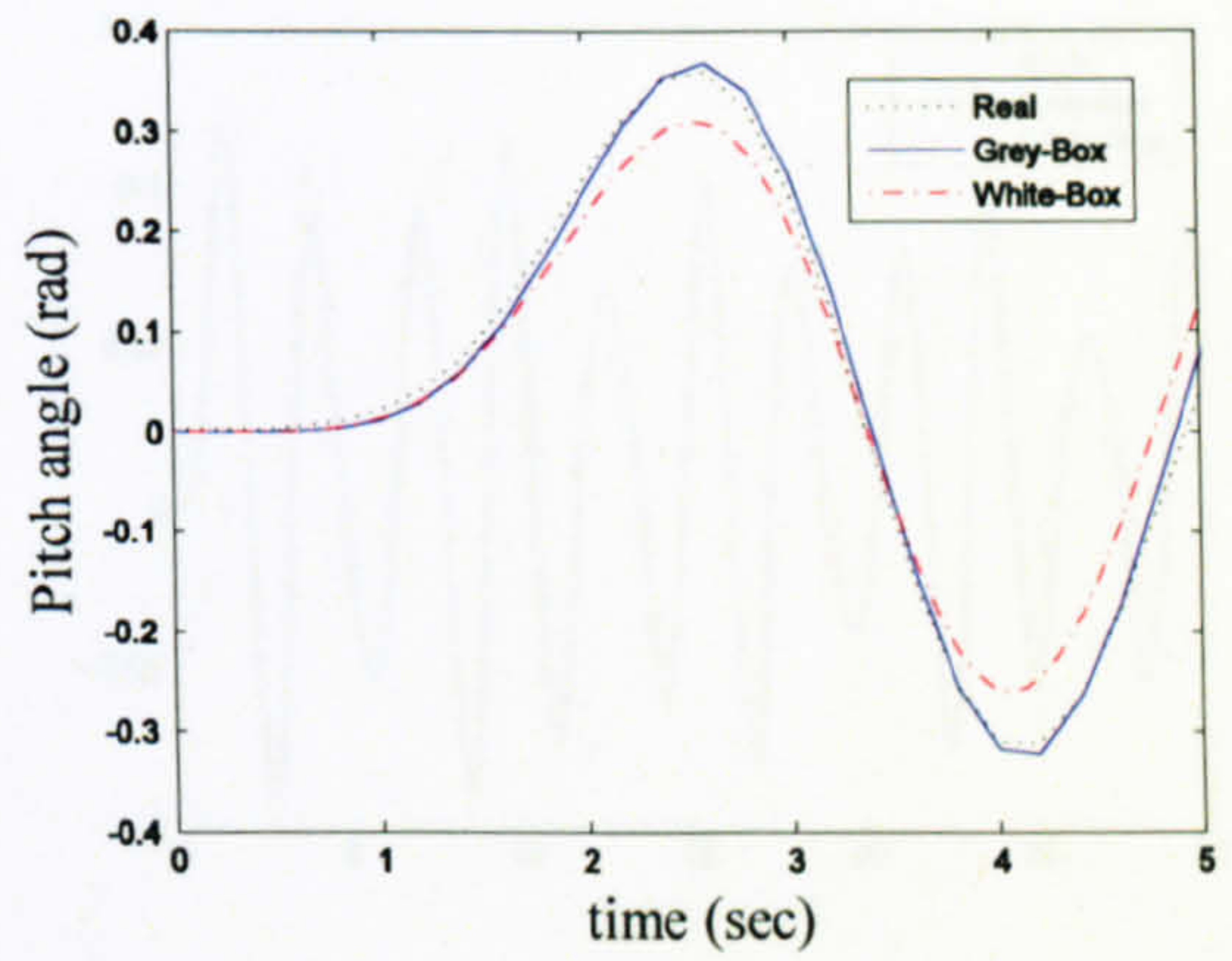


Figure 4.6 The pitch angle of the beam (case 2)

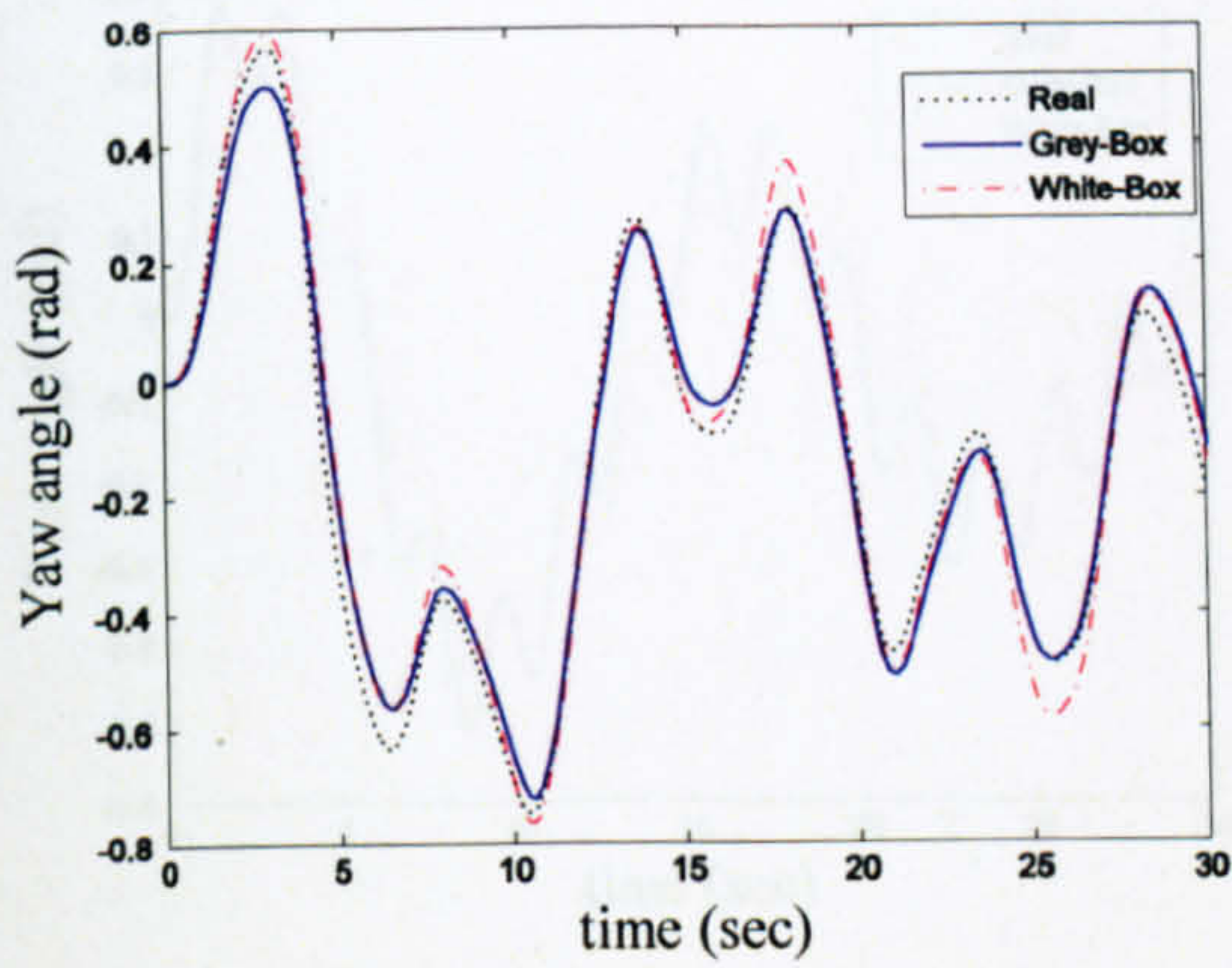


Figure 4.7 The yaw angle of the beam (case 3)

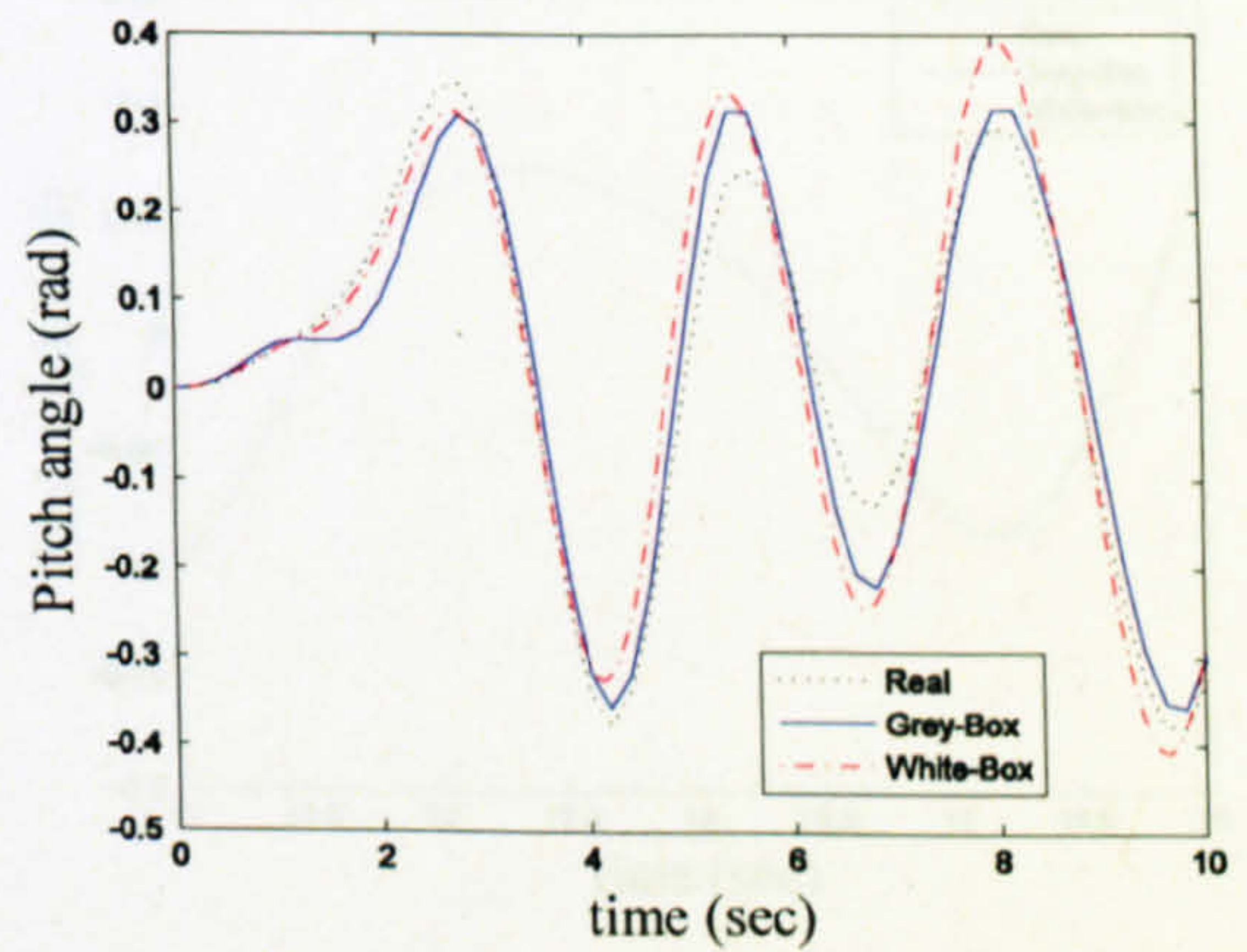


Figure 4.8 The pitch angle of the beam (case 3)

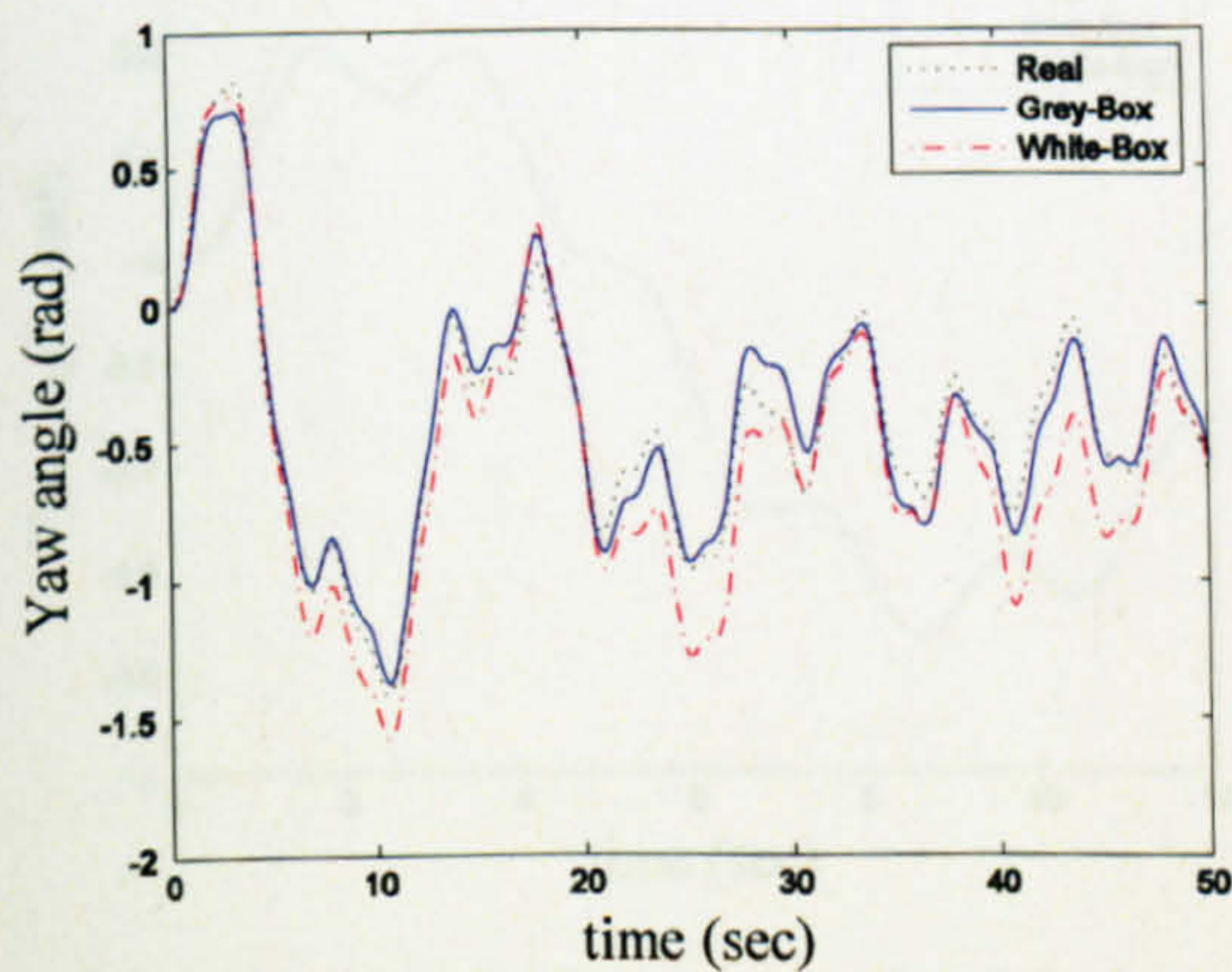


Figure 4.9 The yaw angle of the beam (case 4)

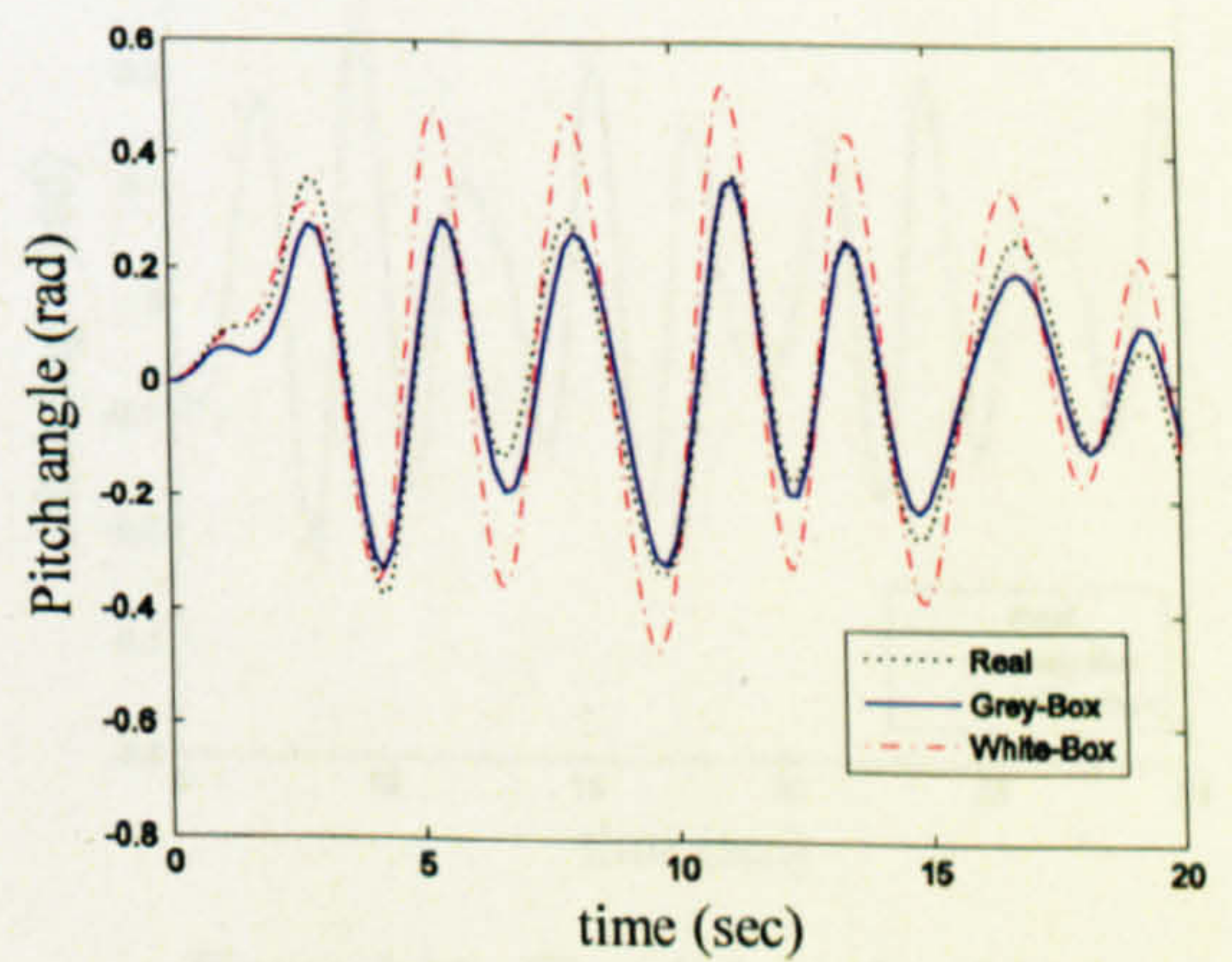


Figure 4.10 The pitch angle of the beam (case 4)

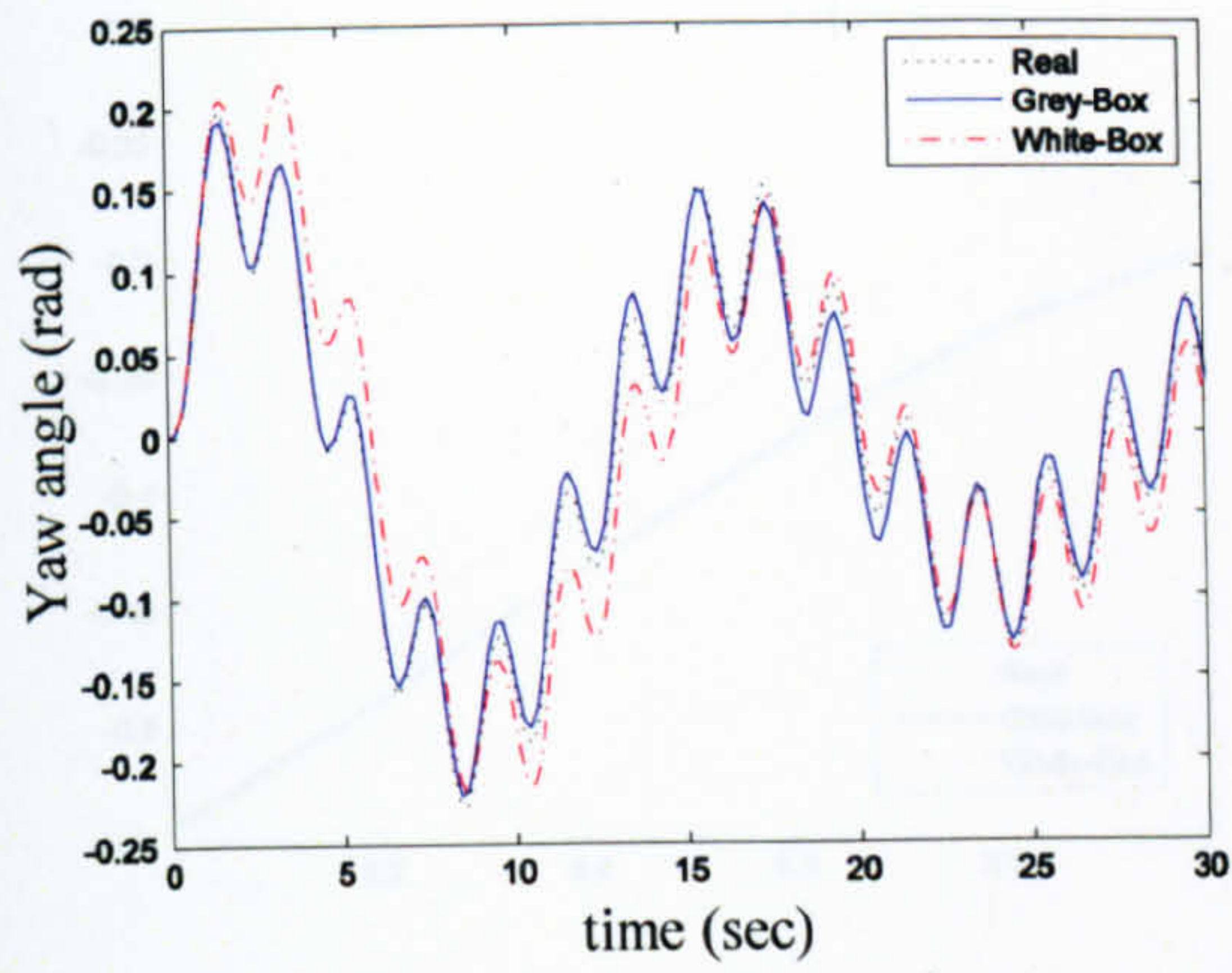


Figure 4.11 The yaw angle of the beam (case 5)

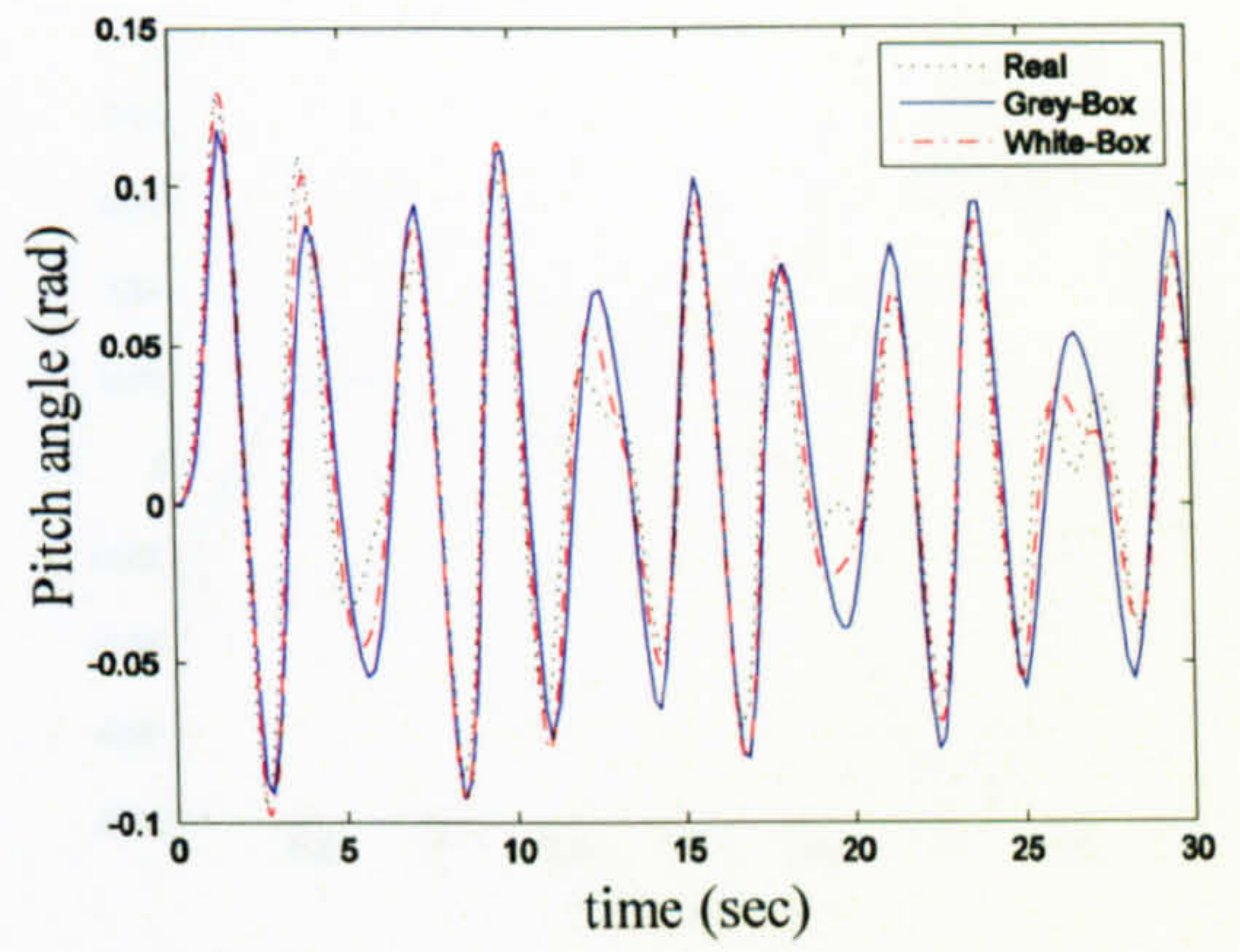


Figure 4.12 The pitch angle of the beam (case 5)

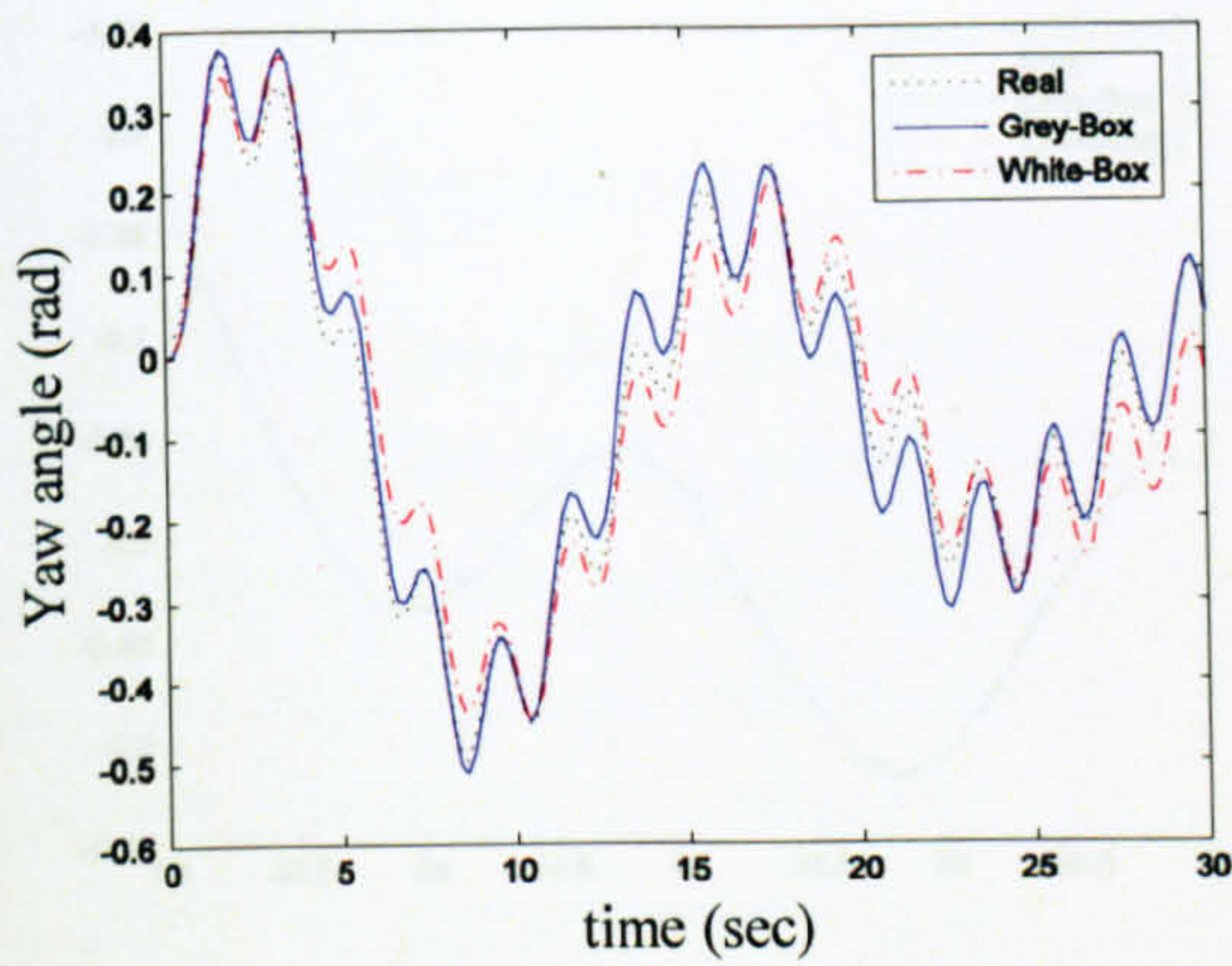


Figure 4.13 The yaw angle of the beam (case 6)

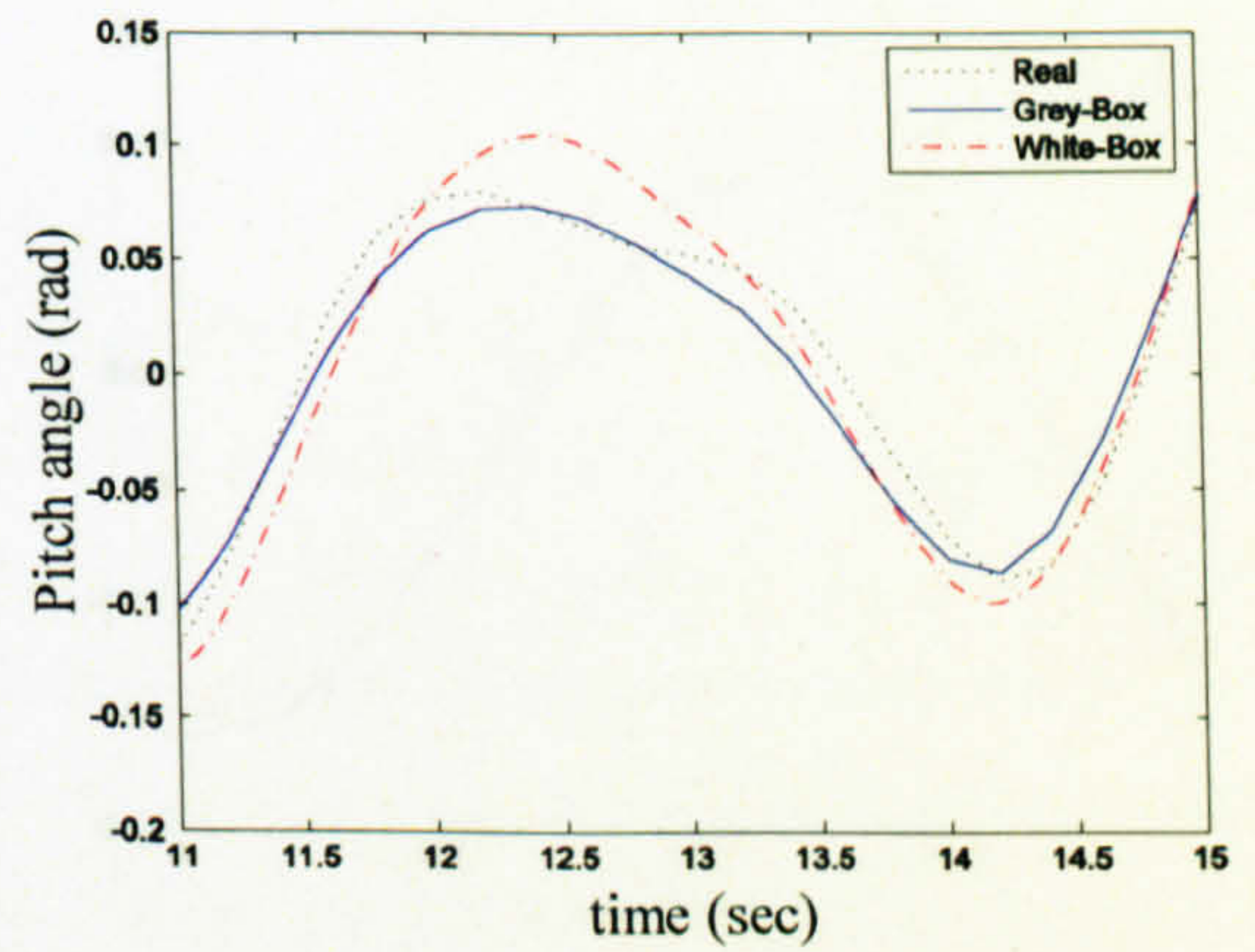


Figure 4.14 The pitch angle of the beam (case 6)

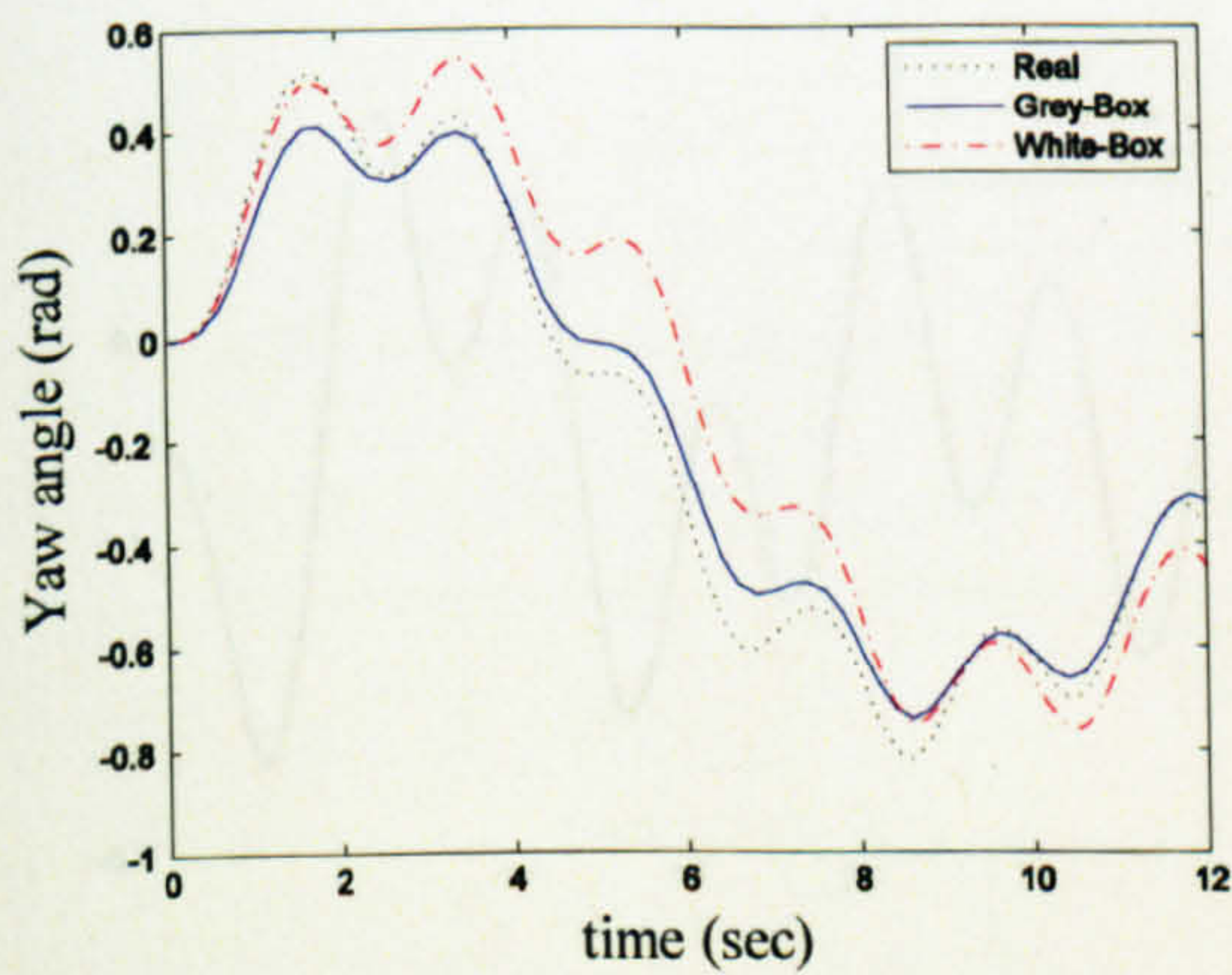


Figure 4.15 The yaw angle of the beam (case 7)

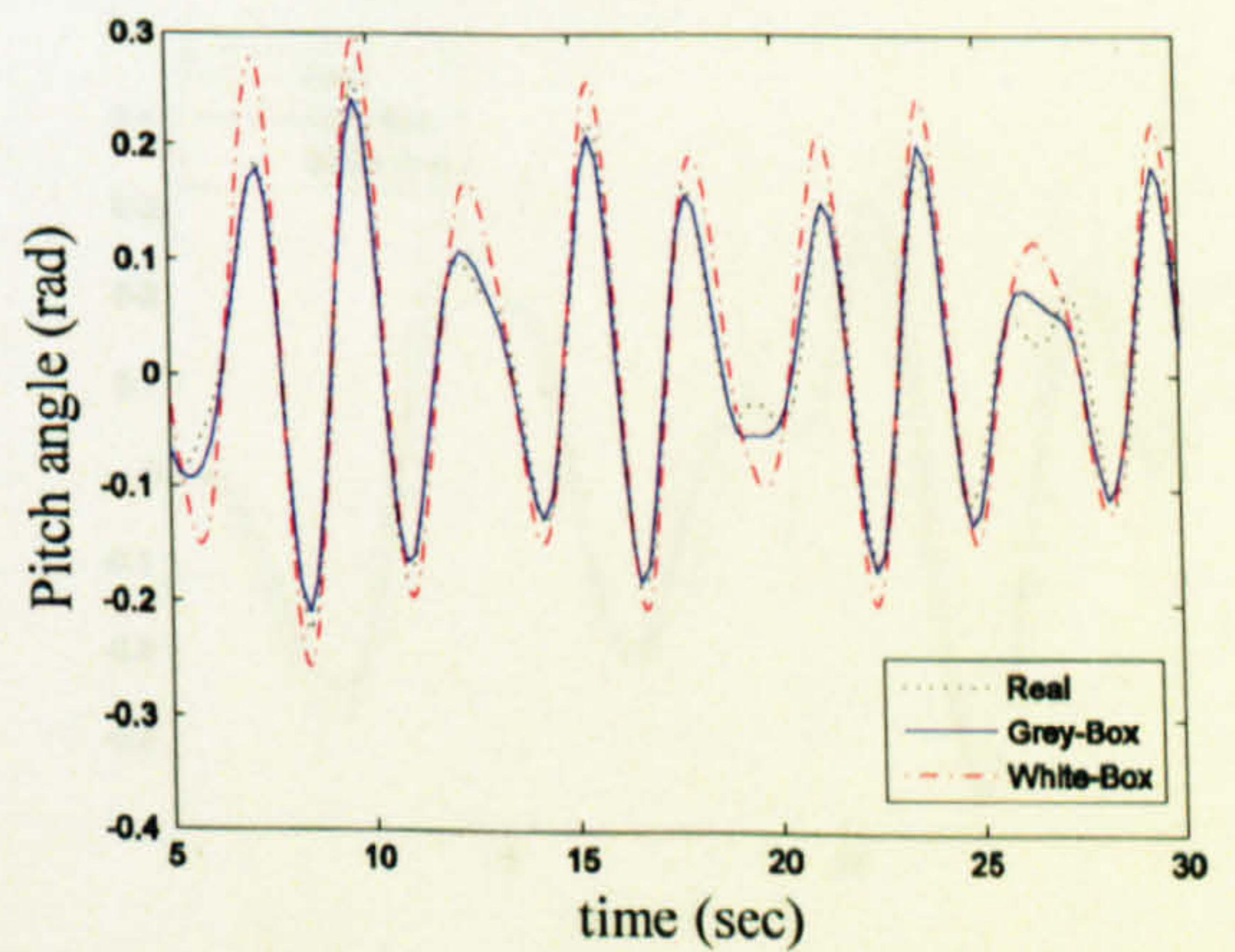


Figure 4.16 The pitch angle of the beam (case 7)

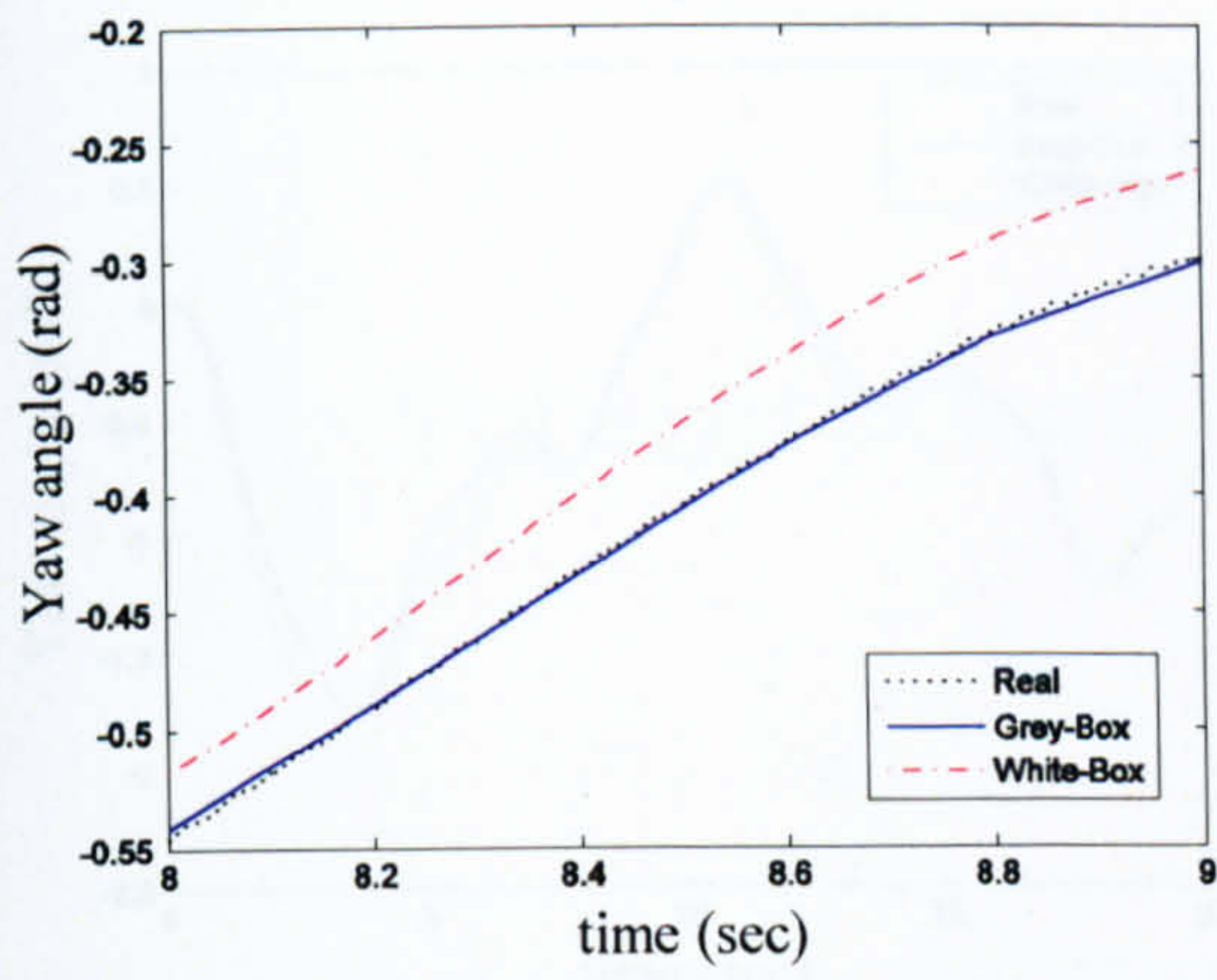


Figure 4.17 The yaw angle of the beam (case 8)

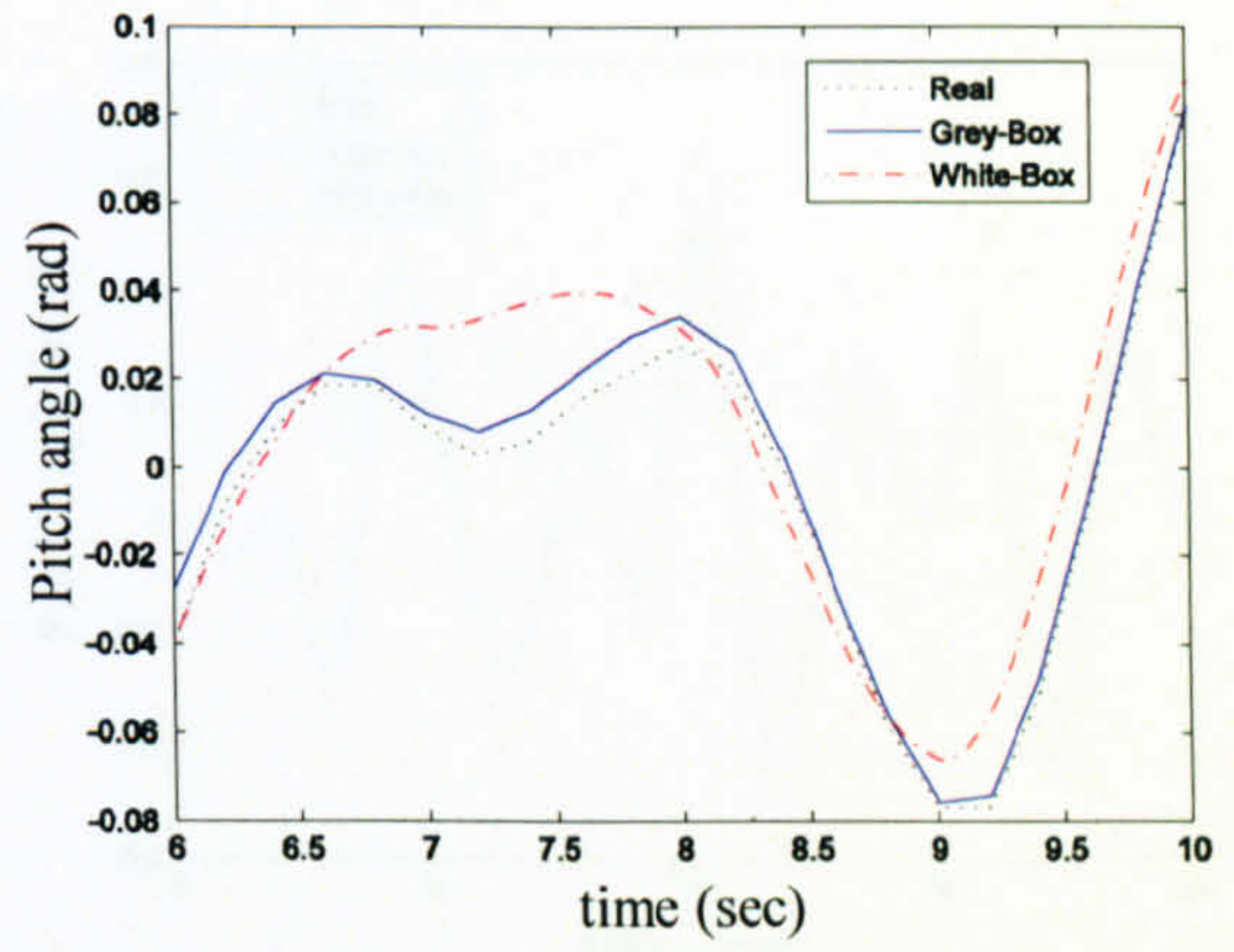


Figure 4.18 The pitch angle of the beam (case 8)

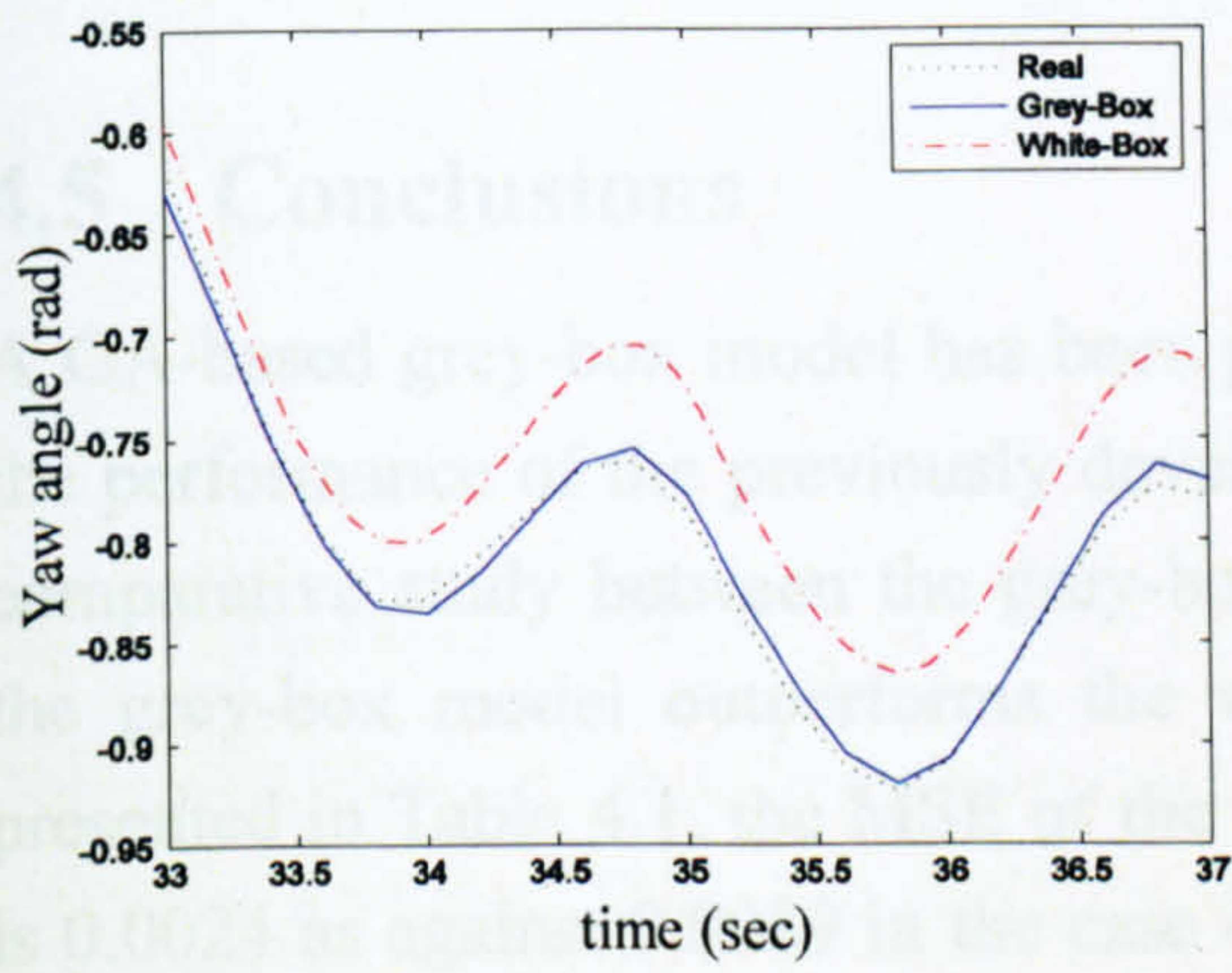


Figure 4.19 The yaw angle of the beam (case 9)

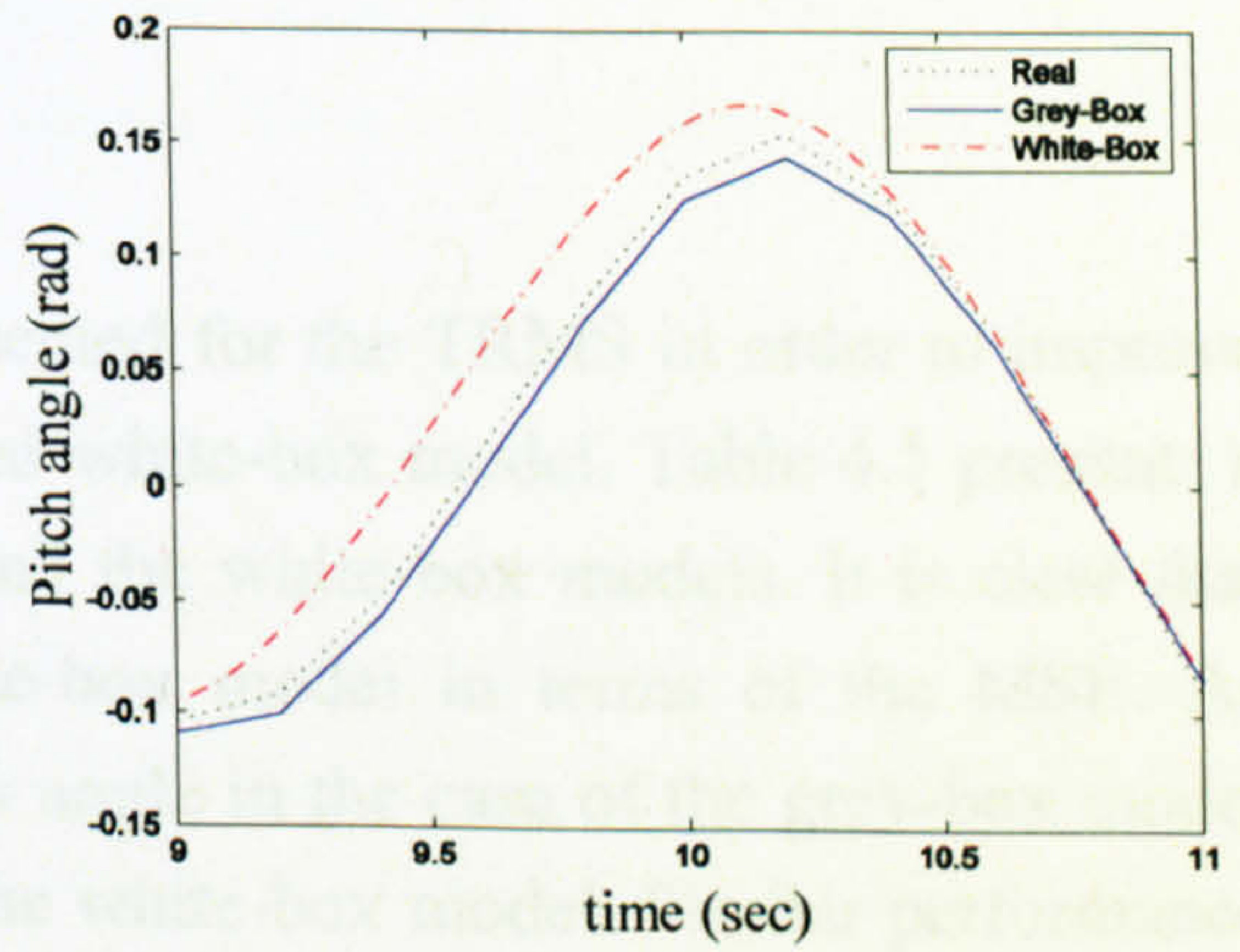


Figure 4.20 The pitch angle of the beam (case 9)

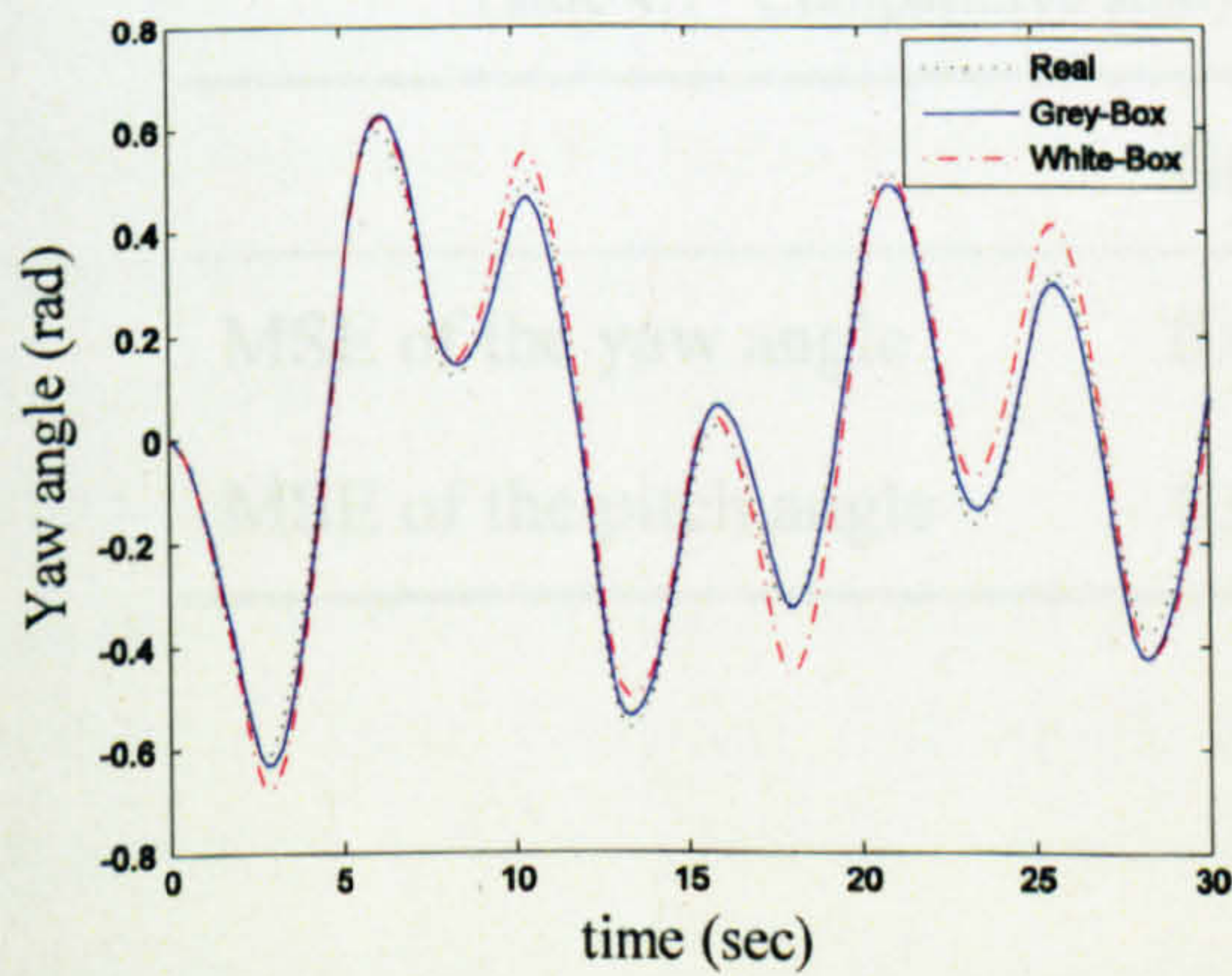


Figure 4.21 The yaw angle of the beam (case 10)

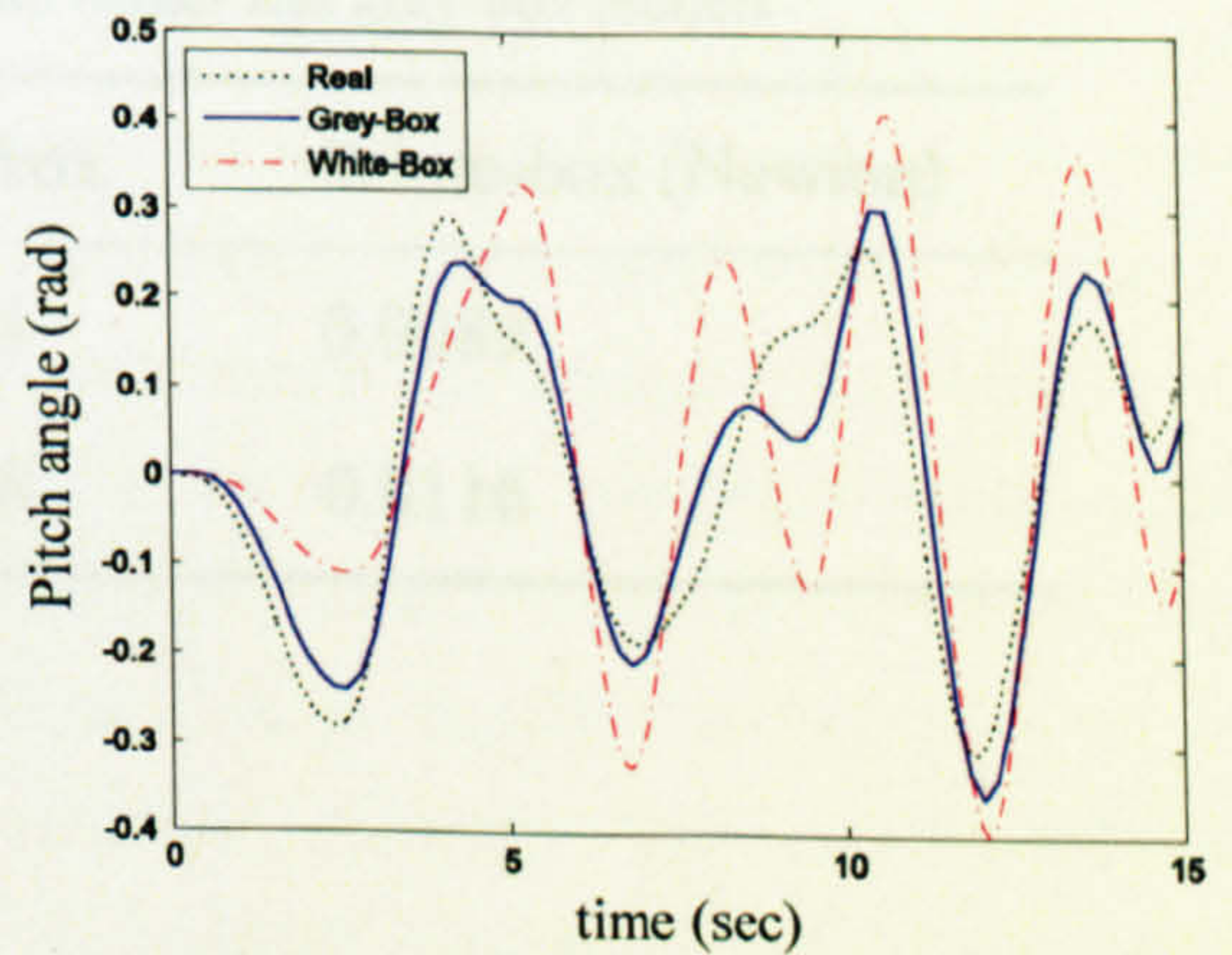


Figure 4.22 The pitch angle of the beam (case 10)

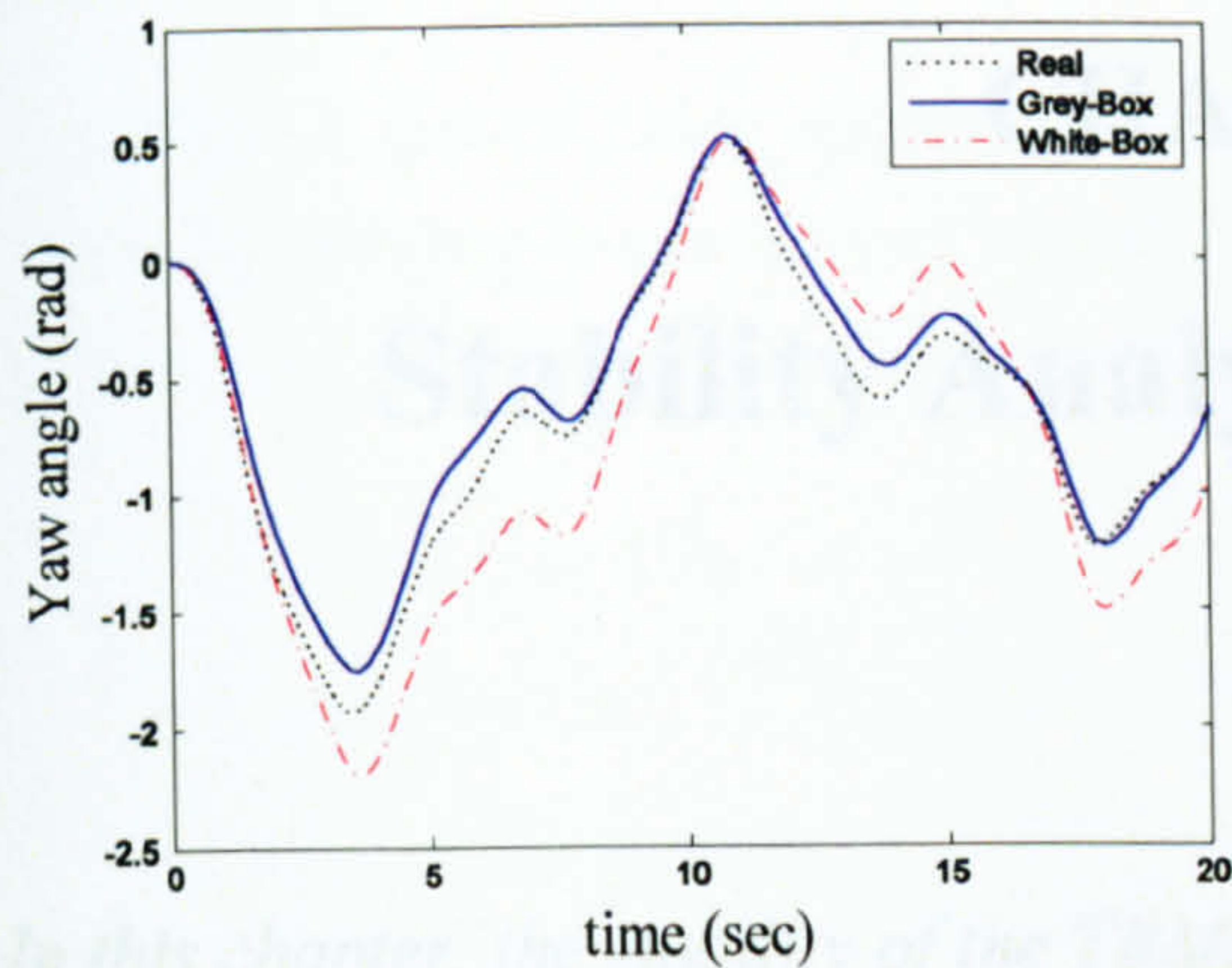


Figure 4.23 The yaw angle of the beam (case 11)

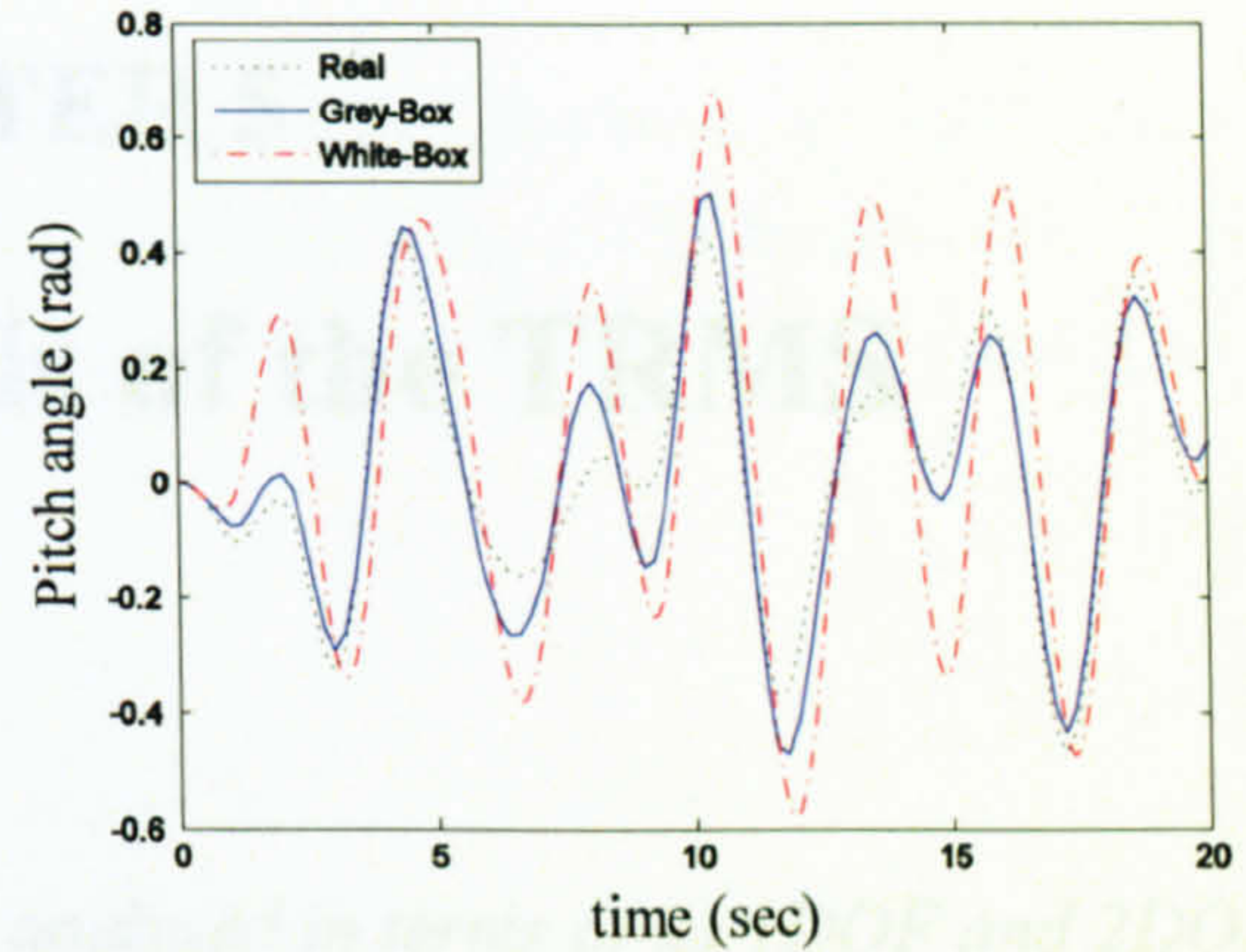


Figure 4.24 The pitch angle of the beam (case 11)

4.5 Conclusions

A GA-based grey-box model has been presented for the TRMS in order to improve the performance of the previously developed white-box model. Table 4.1 presents a comparative study between the grey-box and the white-box models. It is clear that the grey-box model outperforms the white-box model in terms of the MSE. As presented in Table 4.1, the MSE of the yaw angle in the case of the grey-box model is 0.0024 as against 0.0089 in the case of the white-box model. Similar performance is achieved in the case of the pitch angle. In this case, the MSE 0.0068 is obtained with the grey-box model as opposed to 0.0116 in the case of the white-box model.

Table 4.1 Comparative study of the white- and grey-box models

	Grey-box	White-box (Newton)
MSE of the yaw angle	0.0024	0.0089
MSE of the pitch angle	0.0068	0.0116

CHAPTER 5

Stability Analysis of the TRMS

In this chapter, the stability of the TRMS is analysed in terms of its 1DOF and 2DOF dynamics. First, the equilibrium points of the system are calculated and then using Lyapunov's linearisation method, the local stability of the system in terms of the 2DOF is proven. The global stability of the system in terms of its 1DOF dynamics is analysed using the Lyapunov and invariant set theorem. The visual method of stability, phase plane, is employed to show the stability of the system in terms of the 1DOF dynamics.

5.1 Introduction

Nonlinear systems may have some special properties, which cannot occur in the case of linear systems [123]. For instance, in an unforced linear time-invariant system of the form of $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$, the system has only one equilibrium point if the matrix \mathbf{A} is a non-singular matrix. However, in the case of nonlinear systems, it is frequently possible to encounter with multiple equilibrium points. Therefore, it is of great importance to find all the possible equilibrium points of a nonlinear system before controller design and synthesis. A nonlinear system may exhibit fixed amplitude and fixed frequency oscillation without any external excitation, called limit cycles. Note that the amplitude of a limit cycle is independent of the initial state values. Similar to linear systems, if the parameters of a nonlinear system are changed, the stability of the equilibrium point may vary. However, in the case of nonlinear systems the number of equilibrium points may also alter as a result of parameter variations. The values of these parameters at which this phenomenon happens are well-known as bifurcation values. Chaotic behaviour is another property that some nonlinear systems display. If the behaviour of a nonlinear system is extraordinarily sensitive to its initial state values, the system is considered to be a chaotic system and the phenomenon is called chaos.

5.2 Equilibrium Points

Equilibrium points of an autonomous nonlinear system of the form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ can be found by solving the nonlinear simultaneous equations $\mathbf{f}(\mathbf{x}_{eq}) = \mathbf{0}$. In the case of the TRMS, the following equations need to be solved to obtain the equilibrium points,

$$-R_{ah}i_{ah} - k_{ah}\varphi_h\omega_h = 0 \quad (5.1)$$

$$k_{ah}\varphi_hi_{ah} - B_{tr}\omega_h - f_1(\omega_h) = 0 \quad (5.2)$$

$$l_t f_2(\omega_h) \cos(\alpha'_v + \alpha_{v0}) - f_3(\alpha_h) = 0 \quad (5.3)$$

$$S_h + \frac{k_m \omega_v \cos(\alpha'_v + \alpha_{v0})}{D \cos^2(\alpha'_v + \alpha_{v0}) + E \sin^2(\alpha'_v + \alpha_{v0}) + F} = 0 \quad (5.4)$$

$$-R_{av}i_{av} - k_{av}\varphi_v\omega_v = 0 \quad (5.5)$$

$$k_{av}\varphi_vi_{av} - B_{mr}\omega_v - f_4(\omega_v) = 0 \quad (5.6)$$

$$l_m f_5(\omega_v) + g \left[(A - B) \cos(\alpha'_v + \alpha_{v0}) - C \sin(\alpha'_v + \alpha_{v0}) \right] = 0 \quad (5.7)$$

$$S_v + \frac{k_t}{J_v} \omega_h = 0 \quad (5.8)$$

where,

$$f_1(\omega_h) = \begin{cases} k_{thp}\omega_h^2 & \text{if } \omega_h \geq 0 \\ -k_{thn}\omega_h^2 & \text{if } \omega_h < 0 \end{cases}$$

$$f_2(\omega_h) = \begin{cases} k_{fhp}\omega_h^2 & \text{if } \omega_h \geq 0 \\ -k_{fhn}\omega_h^2 & \text{if } \omega_h < 0 \end{cases}$$

$$f_3(\alpha_h) = \begin{cases} k_{chp}\alpha_h & \text{if } \alpha_h \geq 0 \\ k_{chn}\alpha_h & \text{if } \alpha_h < 0 \end{cases}$$

$$f_4(\omega_v) = \begin{cases} k_{vtp}\omega_v^2 & \text{if } \omega_v \geq 0 \\ -k_{vtn}\omega_v^2 & \text{if } \omega_v < 0 \end{cases}$$

$$f_5(\omega_v) = \begin{cases} k_{fvp}\omega_v^2 & \text{if } \omega_v \geq 0 \\ -k_{fvn}\omega_v^2 & \text{if } \omega_v < 0 \end{cases}$$

It is evident that the system has multiple equilibrium points as,

$$\mathbf{x}_{eq} = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ a]^T \quad (5.9)$$

where,

$$\mathbf{x} = [i_{ah} \ \omega_h \ S_h \ \alpha_h \ i_{av} \ \omega_v \ S_v \ \alpha'_v]^T$$

$$a = \pm n\pi \quad \text{for} \quad n = 0, 1, \dots$$

However, due to a mechanical restriction on the system only one equilibrium point is feasible that is the origin.

5.3 Local Stability

According to Lyapunov's linearisation stability, if a linearised system around its equilibrium point is strictly stable, then the equilibrium is locally asymptotically stable for the nonlinear system. The linearised equations, $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$, of the TRMS around the equilibrium is as follows,

$$\frac{d}{dt} \begin{bmatrix} i_{ah} \\ \omega_h \\ S_h \\ \alpha_h \\ i_{av} \\ \omega_v \\ S_v \\ \alpha_v \end{bmatrix} = \begin{bmatrix} -\frac{R_{ah}}{L_{ah}} & -\frac{k_{ah}\varphi_h}{L_{ah}} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{k_{ah}\varphi_h}{J_{tr}} & -\frac{B_{tr}}{J_{tr}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{k_{vfh}}{k} & -\frac{k_{chp}}{k} & 0 & -\frac{k_{vfh}k_m}{k^2}\cos\alpha_{v0} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \frac{k_m}{k}\cos\alpha_{v0} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{R_{av}}{L_{av}} & -\frac{k_{av}\varphi_v}{L_{av}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{k_{av}\varphi_v}{J_{mr}} & -\frac{B_{mr}}{J_{mr}} & 0 & 0 \\ 0 & -\frac{k_{vfv}k_t}{J_v^2} & 0 & 0 & 0 & 0 & -\frac{k_{vfv}}{J_v} & \frac{gk'}{J_v} \\ 0 & \frac{k_t}{J_v} & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} i_{ah} \\ \omega_h \\ S_h \\ \alpha_h \\ i_{av} \\ \omega_v \\ S_v \\ \alpha_v \end{bmatrix} \quad (5.10)$$

where $k = D\cos^2\alpha_{v0} + E\sin^2\alpha_{v0} + F$ and $k' = (B - A)\sin\alpha_{v0} - C\cos\alpha_{v0}$. The eigenvalues of the matrix \mathbf{A} are strictly in the left-half complex plane and therefore, the equilibrium point is locally asymptotically stable for the nonlinear system.

5.4 Phase Plane Analysis

As mentioned before, the TRMS is a system of the order of 8 in the case of its 2DOF dynamics. However, when dealing with the 1DOF dynamics, the system is divided into two fourth-order subsystems. One of the powerful methods to graphically analyse the stability of second-order systems is the so-called phase plane analysis. Although the approach can be extended to third-order systems, it is restricted for higher order

systems such as the TRMS even in its 1DOF dynamics. Nevertheless, in the case of the 1DOF TRMS the second-order equations of the electric motor can be analysed separately from the other parts of the system in the case of both channels due to the cascade nature of the system. Figure 5.1 shows the block diagram of the 2DOF TRMS. Its decoupled systems which can be considered two separate systems, the 1DOF horizontal and the 1DOF vertical, are depicted in Figure 5.2.

Therefore, four systems of the order of two (i.e. the main motor, the tail motor, the vertical dynamics and the horizontal dynamics) need to be analysed. Note that based on the stability of separate parts of a system one cannot conclude the stability of the overall system; however, when the subsystems are connected to each other in cascade, some results on the overall stability can be concluded.

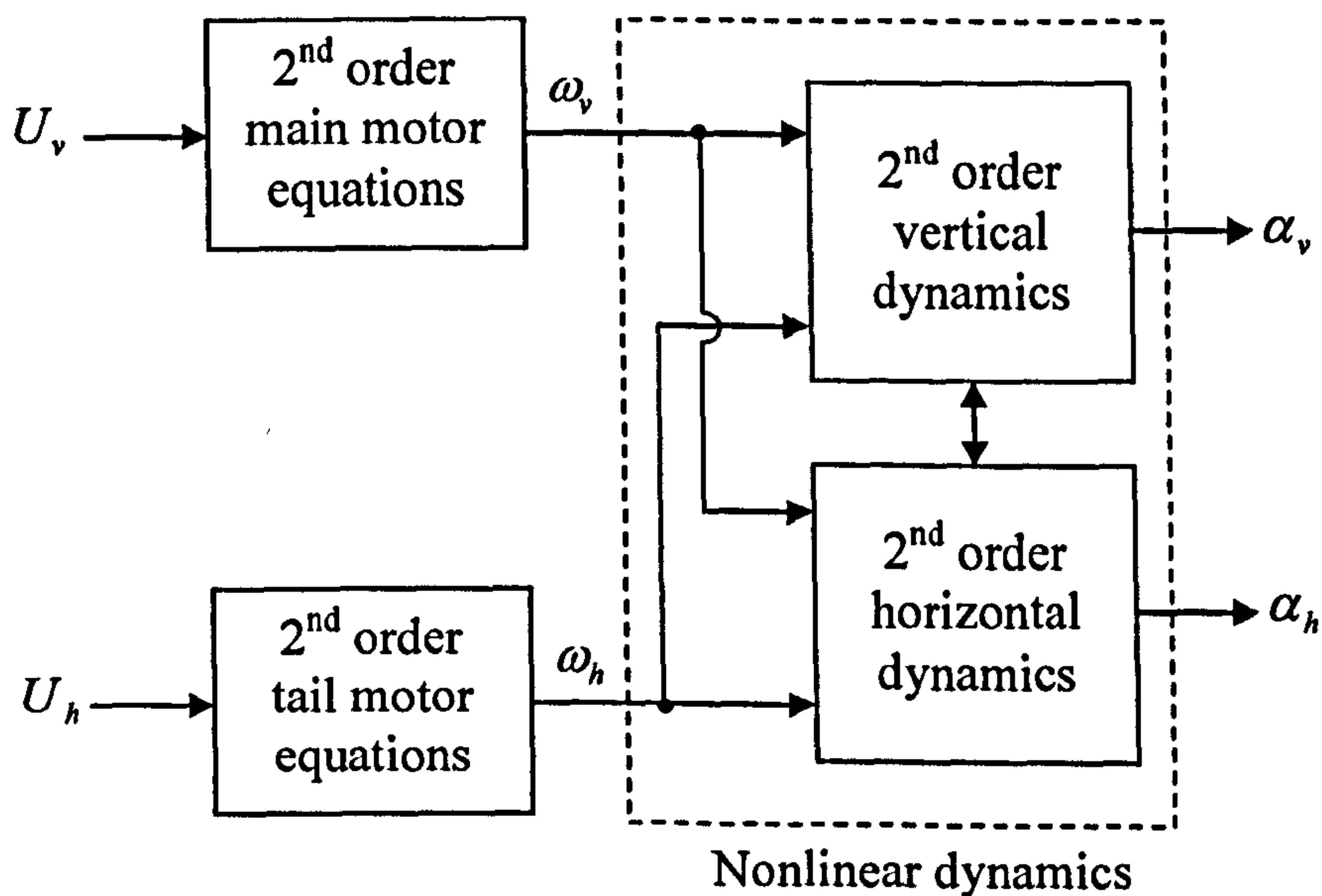


Figure 5.1 The block diagram of the 2DOF TRMS

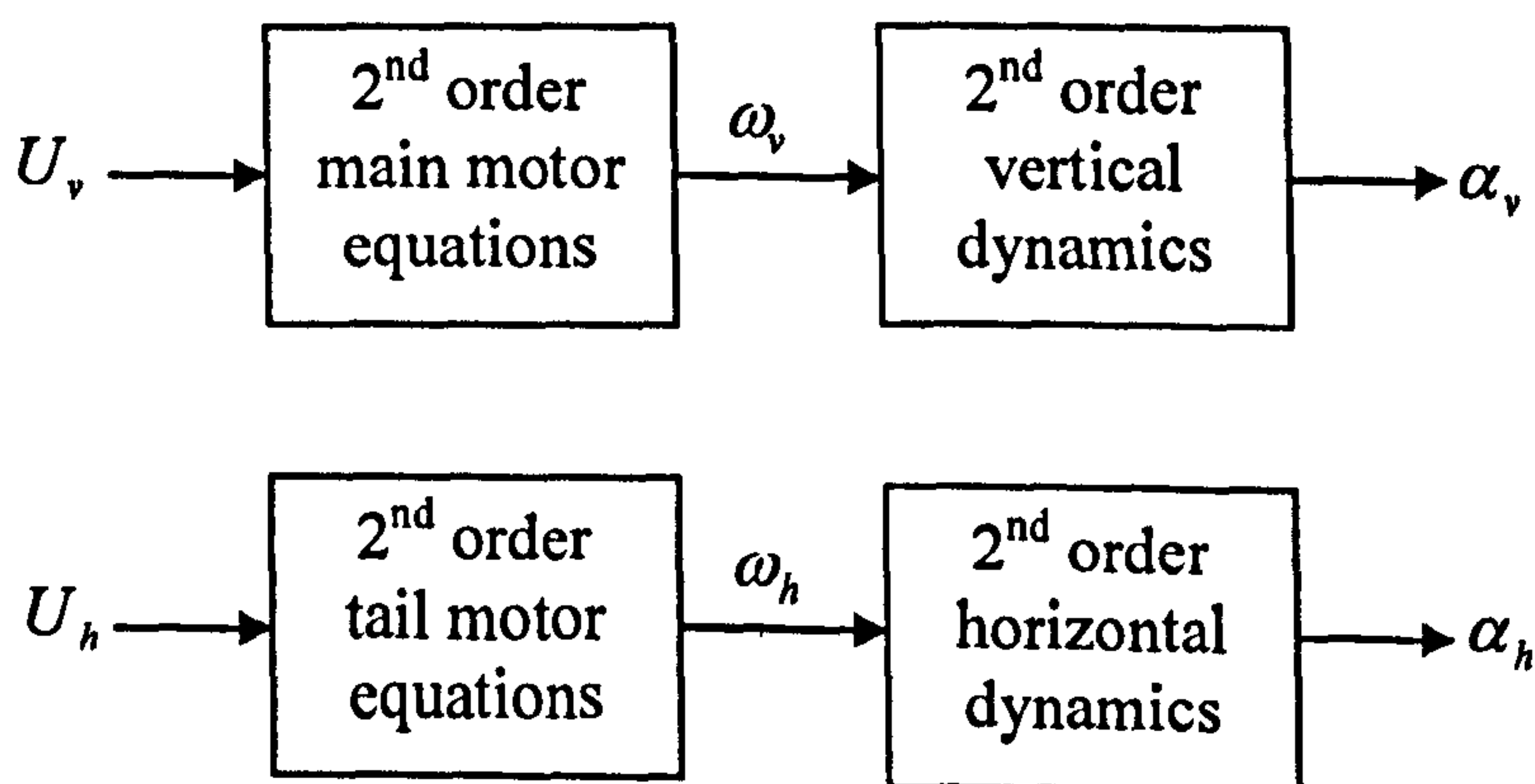


Figure 5.2 The block diagram of the decoupled TRMS

Since the equations of the main motor are independent of other parts of the TRMS, therefore its stability can be analysed using its phase portrait. Let recall the second-order unforced autonomous equations of the main motor as follows,

$$\frac{di_{av}}{dt} = -\frac{R_{av}}{L_{av}}i_{av} - \frac{k_{av}\varphi_v}{L_{av}}\omega_v \quad (5.11)$$

$$\frac{d\omega_v}{dt} = \frac{k_{av}\varphi_v}{J_{mr}}i_{av} - \frac{B_{mr}}{J_{mr}}\omega_v - \frac{f_4(\omega_v)}{J_{mr}} \quad (5.12)$$

where,

$$f_4(\omega_v) = \begin{cases} k_{tvp}\omega_v^2 & \text{if } \omega_v \geq 0 \\ -k_{tvn}\omega_v^2 & \text{if } \omega_v < 0 \end{cases}$$

Solving the above nonlinear differential equations according to various initial conditions and plotting on one graph leads to the phase portrait of the system. One can easily obtain the phase portrait of the main motor as shown in Figure 5.3. In order to obtain a more sensible graph, the phase portrait of the main motor has been redrawn for a higher value of armature inductance, L_{av} , as shown in Figure 5.4.

The same approach is repeated for the tail motor as shown in Figures 5.5 and 5.6 according to the following unforced autonomous equations,

$$\frac{di_{ah}}{dt} = -\frac{R_{ah}}{L_{ah}}i_{ah} - \frac{k_{ah}\varphi_h}{L_{ah}}\omega_h \quad (5.13)$$

$$\frac{d\omega_h}{dt} = \frac{k_{ah}\varphi_h}{J_{tr}}i_{ah} - \frac{B_{tr}}{J_{tr}}\omega_h - \frac{f_1(\omega_h)}{J_{tr}} \quad (5.14)$$

where,

$$f_1(\omega_h) = \begin{cases} k_{thp}\omega_h^2 & \text{if } \omega_h \geq 0 \\ -k_{thn}\omega_h^2 & \text{if } \omega_h < 0 \end{cases}$$

As it is evident from the figures, the equilibrium point is a stable node for both motors, as all the phase planes converge to the origin, the equilibrium point, regardless of initial conditions.

Now, consider the unexcited autonomous vertical dynamics of the system by the following equations,

$$\frac{d\Omega_v}{dt} = \frac{g[(A-B)\cos(\alpha'_v + \alpha_{v0}) - C\sin(\alpha'_v + \alpha_{v0})] - k_{vf}\Omega_v}{J_v} \quad (5.15)$$

$$\frac{d\alpha'_v}{dt} = \Omega_v \quad (5.16)$$

The phase portrait of the second-order vertical dynamics of the system is depicted in Figure 5.7. Clearly, the equilibrium point of the system is stable as all the trajectories converge to the origin. The governing equations of the horizontal dynamics can be expressed as,

$$\frac{d\Omega_h}{dt} = \frac{-k_{vfh}\Omega_h - f_3(\alpha_h)}{D + F} \quad (5.17)$$

$$\frac{d\alpha_h}{dt} = \Omega_h \quad (5.18)$$

where,

$$f_3(\alpha_h) = \begin{cases} k_{chp}\alpha_h & \text{if } \alpha_h \geq 0 \\ k_{chn}\alpha_h & \text{if } \alpha_h < 0 \end{cases}$$

The phase portrait of the horizontal dynamics can be found in Figure 5.8. In order to show the stability of the TRMS in terms of its 1DOF horizontal and vertical, two forth-order systems are presented and analysed. The equations of the 1DOF horizontal dynamics of the TRMS can be expressed as,

$$\frac{di_{ah}}{dt} = -\frac{R_{ah}}{L_{ah}}i_{ah} - \frac{k_{ah}\varphi_h}{L_{ah}}\omega_h \quad (5.19)$$

$$\frac{d\omega_h}{dt} = \frac{k_{ah}\varphi_h}{J_{tr}}i_{ah} - \frac{B_{tr}}{J_{tr}}\omega_h - \frac{f_1(\omega_h)}{J_{tr}} \quad (5.20)$$

$$\frac{d\Omega_h}{dt} = \frac{l_t f_2(\omega_h) - k_{vfh}\Omega_h - f_3(\alpha_h)}{D + F} \quad (5.21)$$

$$\frac{d\alpha_h}{dt} = \Omega_h \quad (5.22)$$

The above system is a forth-order system. As mentioned before, due to the fact that the governing equations of the tail motor is independent of the last two state variables, Ω_h and α_h , its stability can be considered using a 2D phase portrait as shown in Figure 5.5. Also, the stability of the overall 1DOF horizontal system can be represented by a 3D phase portrait. In this context, the nonlinear differential equations (5.19)-(5.22) are solved with various initial conditions. Then, only the last three state variables of the system are considered in the 3D phase portrait because the first state, i_{ah} , has direct effect on the second state variable and their stability has already been proven by another 2D phase portrait. The 3D phase portrait of the

overall horizontal dynamics is depicted in Figure 5.9 that shows the system is asymptotically stable. Similarly, the overall forth-order vertical dynamics is stable as shown its 3D phase portrait in Figure 5.10. The governing equations of the 1DOF vertical TRMS are as follows,

$$\frac{di_{av}}{dt} = -\frac{R_{av}}{L_{av}}i_{av} - \frac{k_{av}\phi_v}{L_{av}}\omega_v \quad (5.23)$$

$$\frac{d\omega_v}{dt} = \frac{k_{av}\phi_v}{J_{mr}}i_{av} - \frac{B_{mr}}{J_{mr}}\omega_v - \frac{f_4(\omega_v)}{J_{mr}} \quad (5.24)$$

$$\frac{d\Omega_v}{dt} = \frac{l_m f_5(\omega_v) + g[(A-B)\cos(\alpha'_v + \alpha_{v0}) - C\sin(\alpha'_v + \alpha_{v0})] - k_{vfv}\Omega_v}{J_v} \quad (5.25)$$

$$\frac{d\alpha'_v}{dt} = \Omega_v \quad (5.26)$$

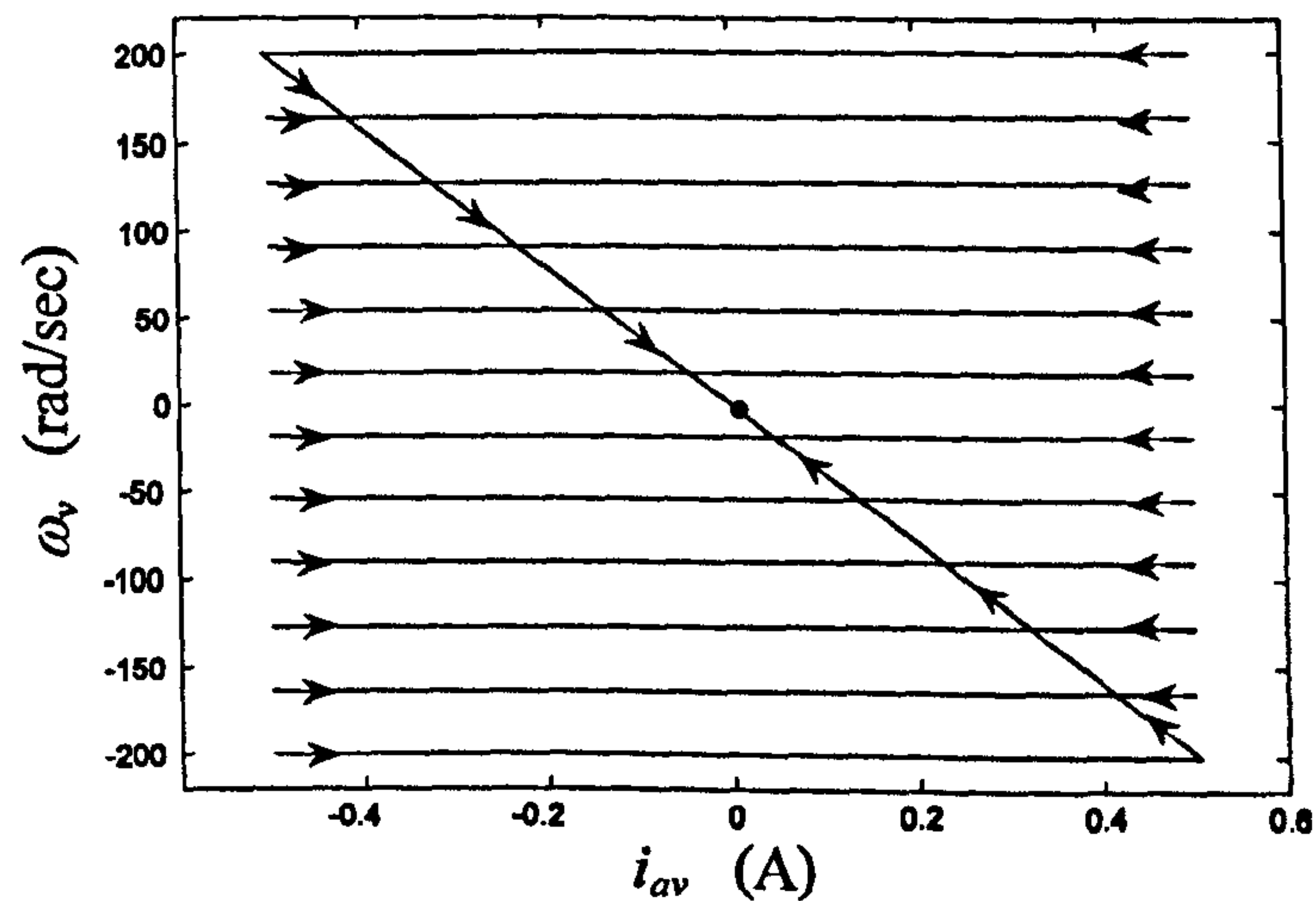


Figure 5.3 The phase portrait of the main motor

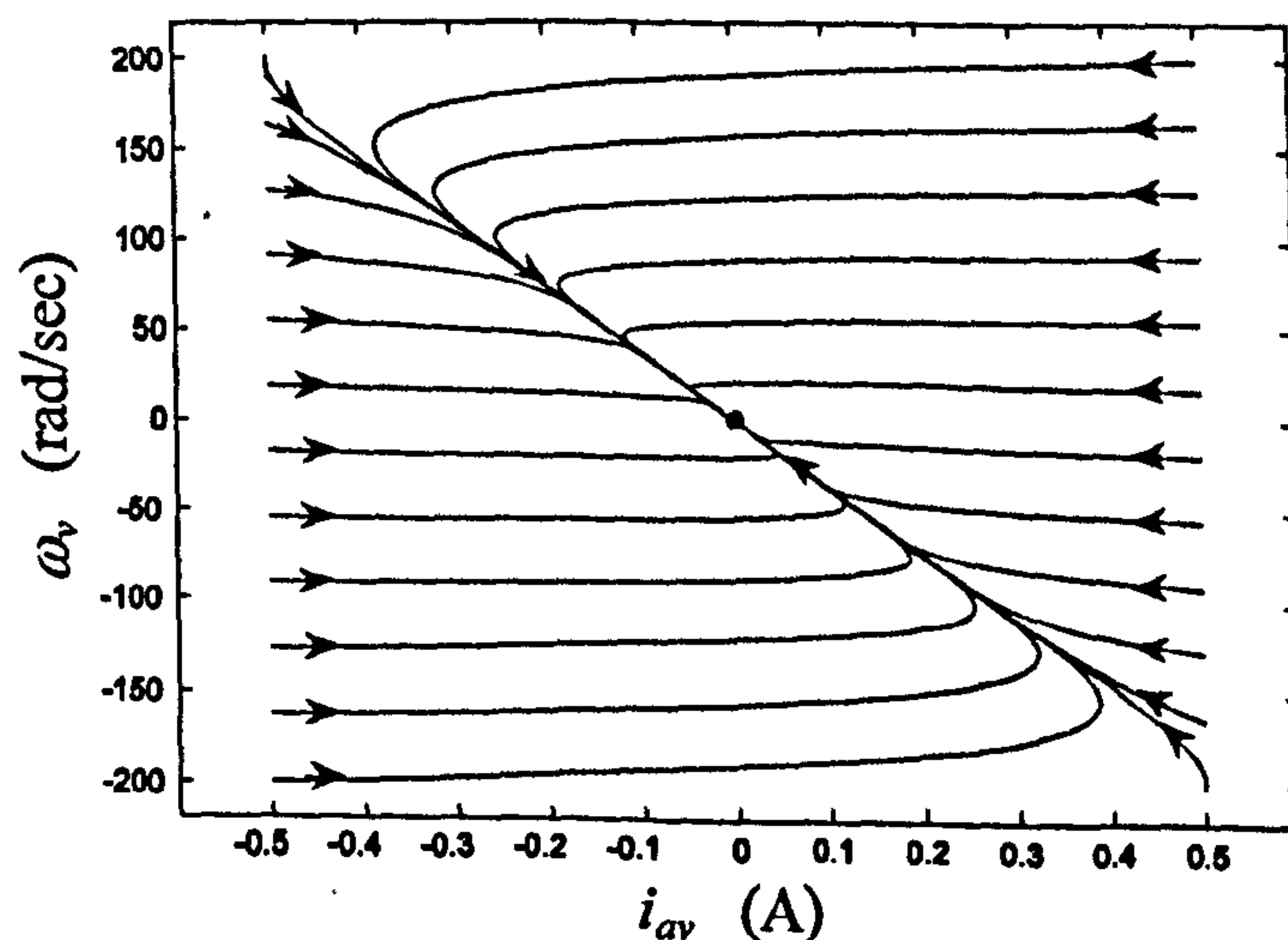


Figure 5.4 The phase portrait of the main motor with a higher inductance

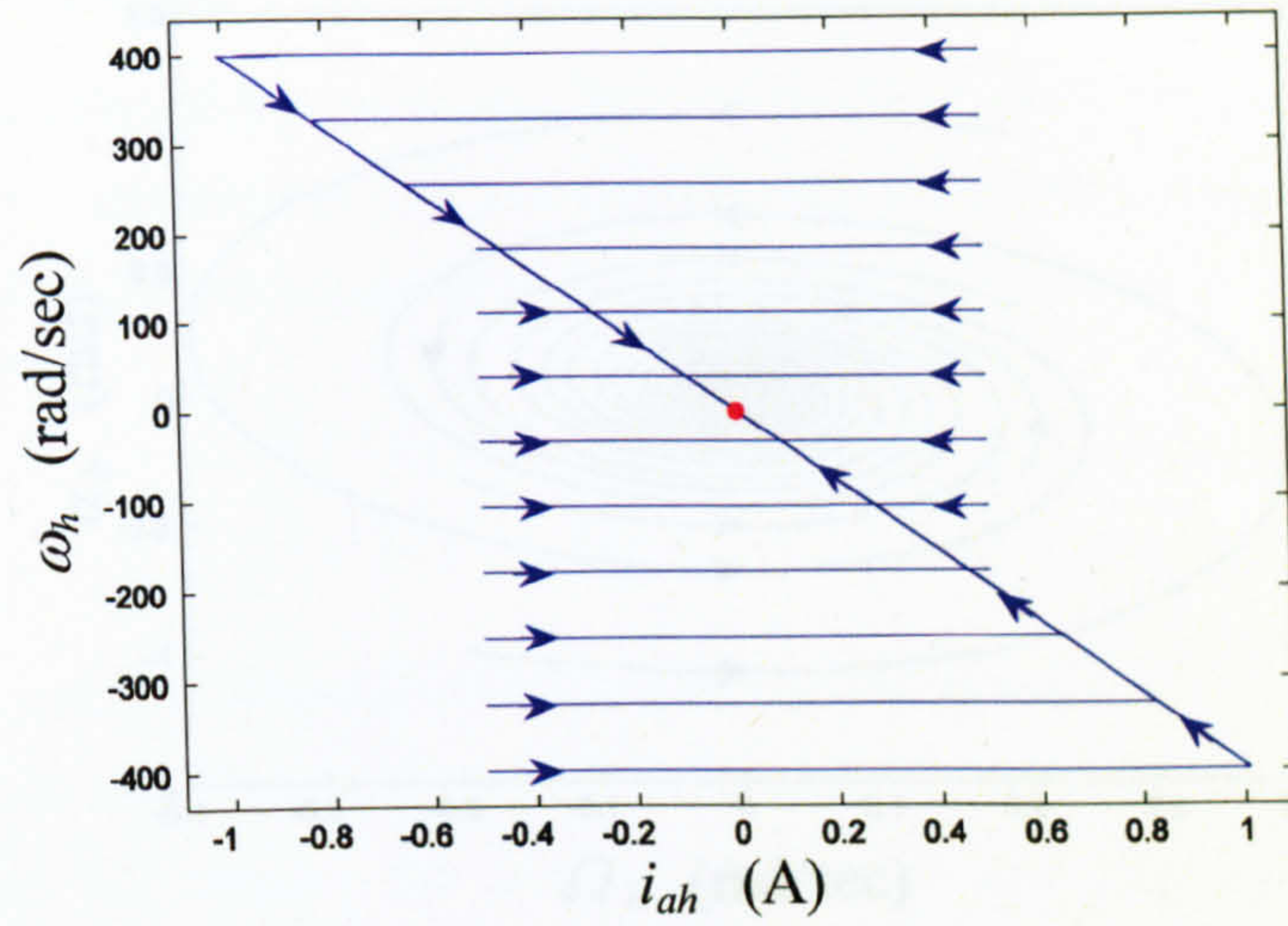


Figure 5.5 The phase portrait of the tail motor

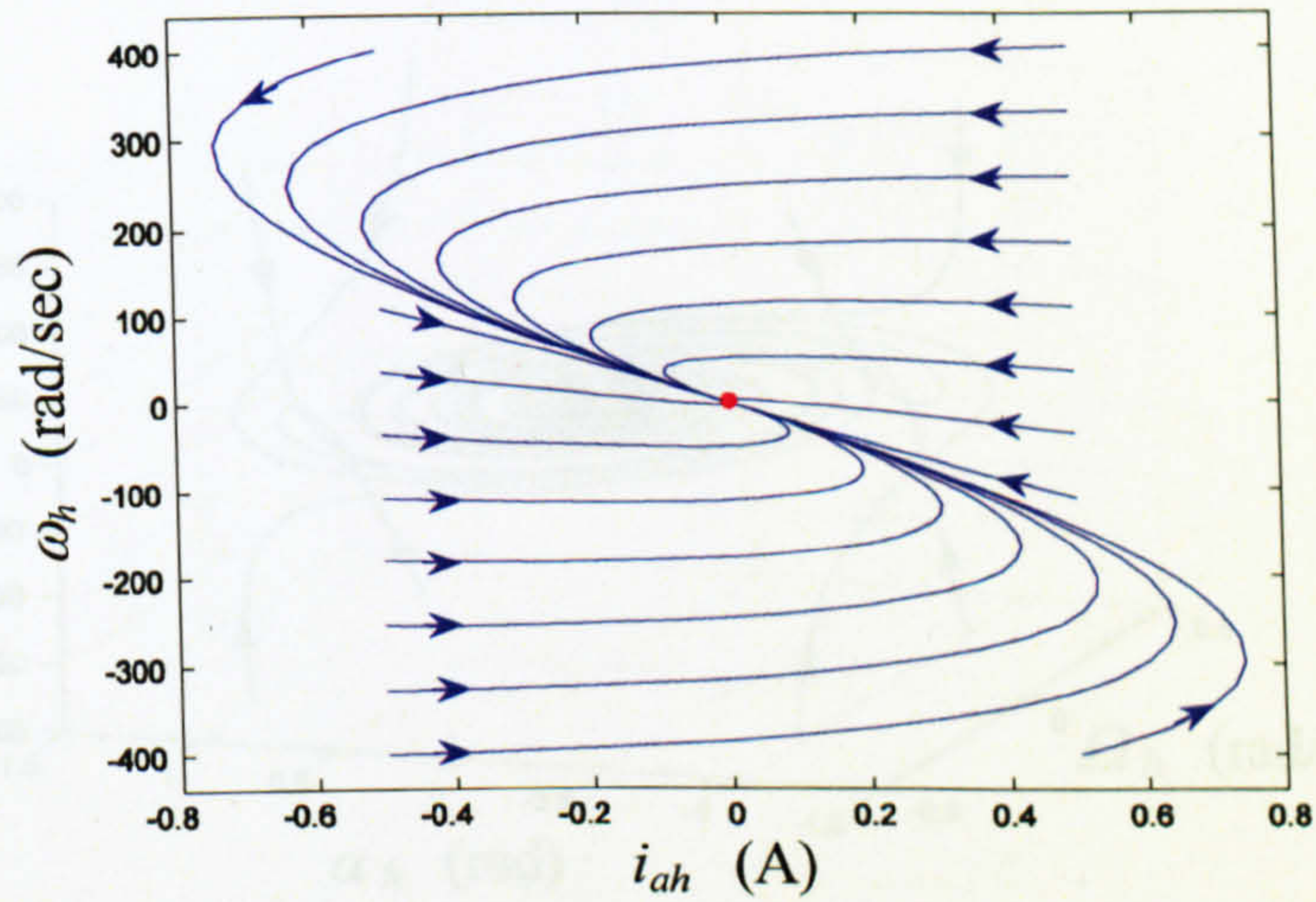


Figure 5.6 The phase portrait of the tail motor with a higher inductance

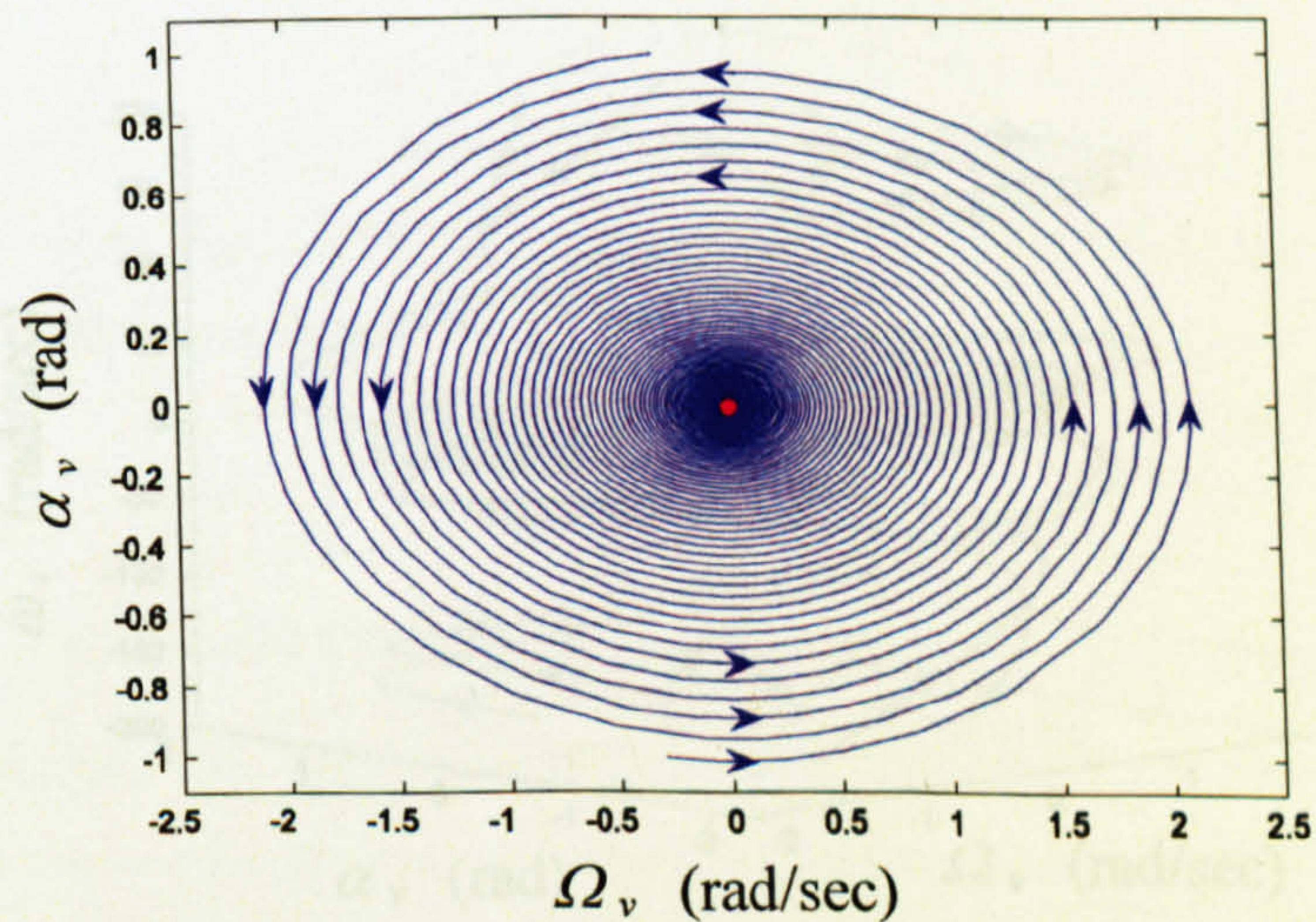


Figure 5.7 The phase portrait of the vertical dynamics

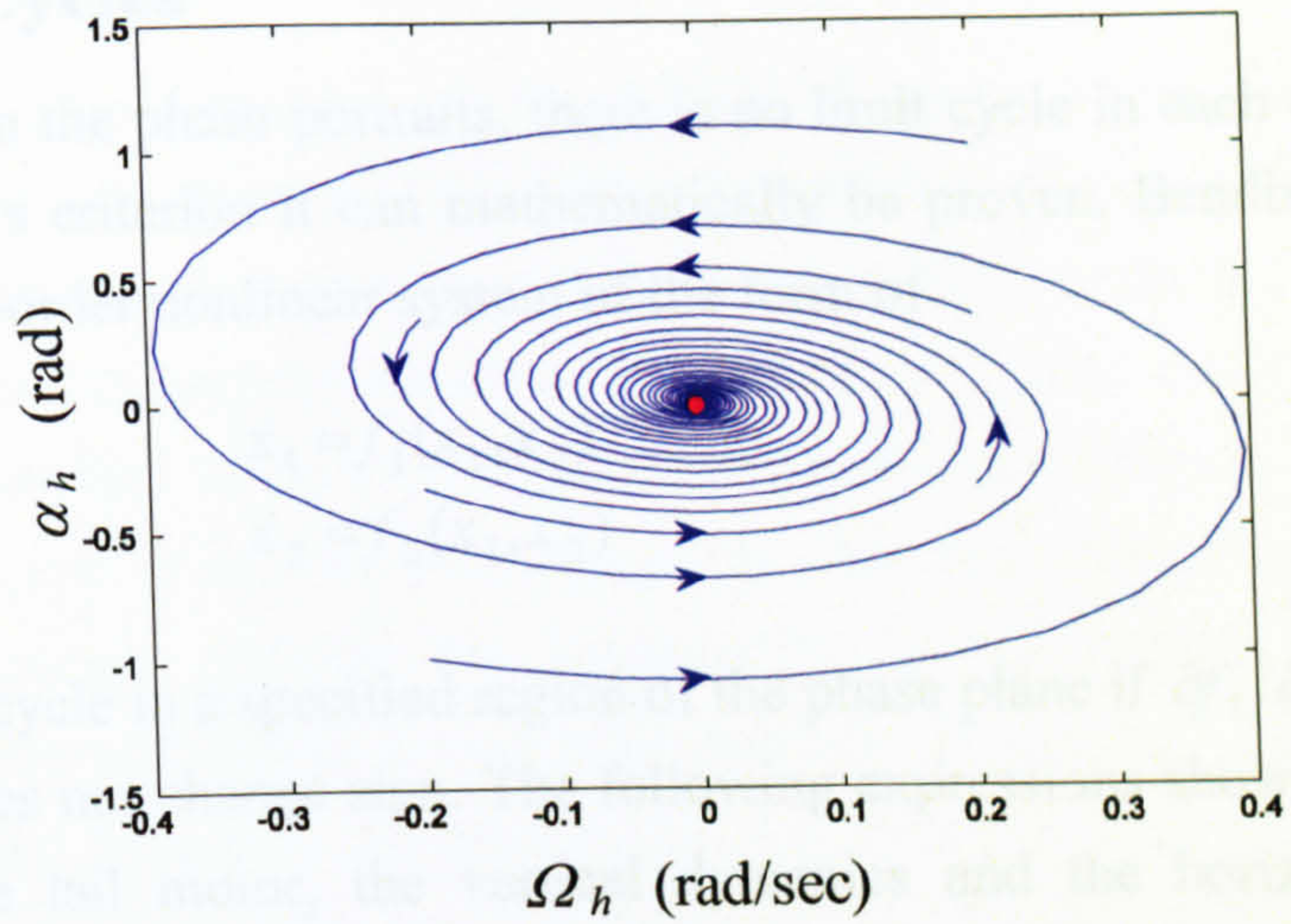


Figure 5.8 The phase portrait of the horizontal dynamics

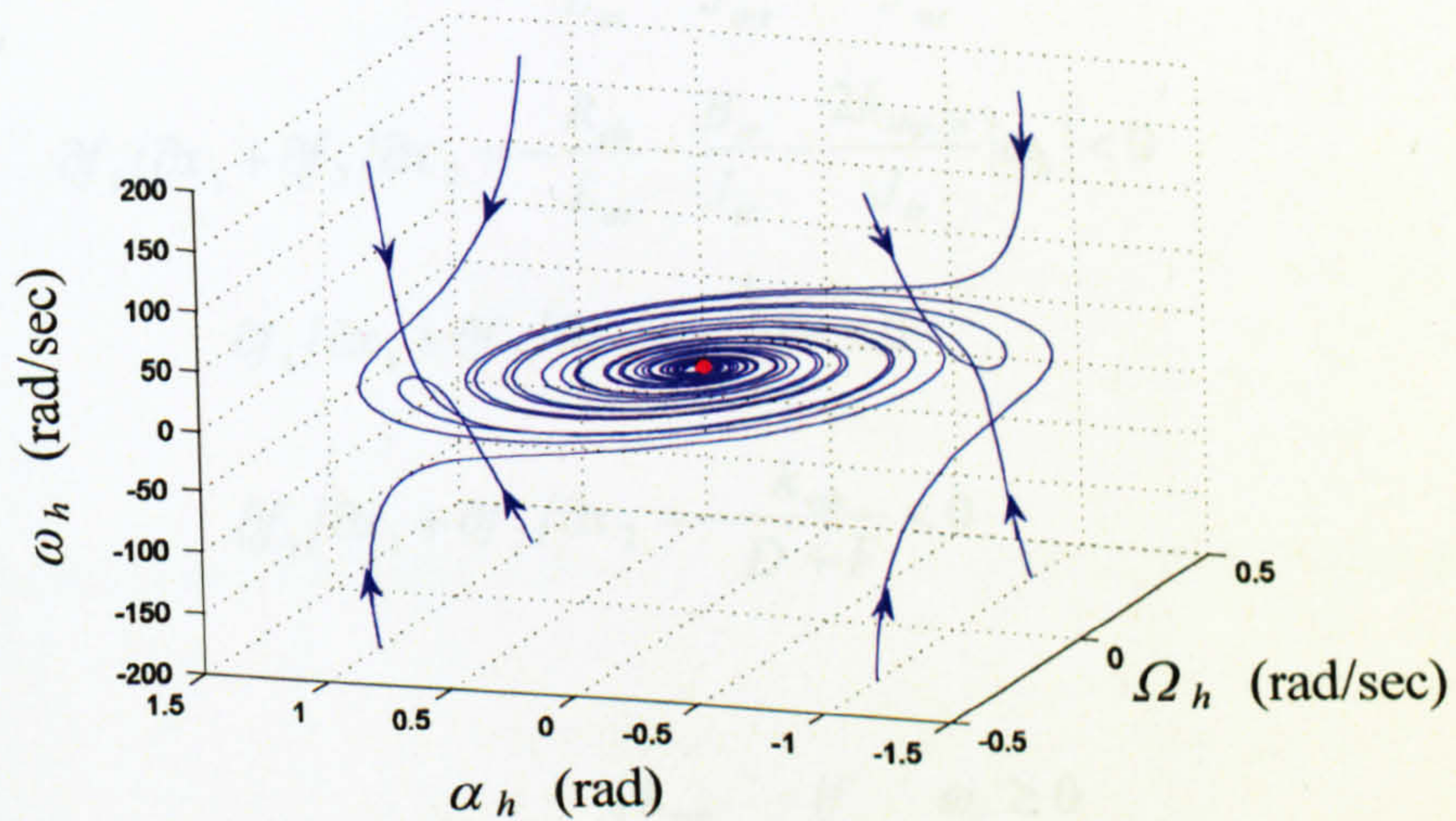


Figure 5.9 The 3D phase portrait of the horizontal dynamics

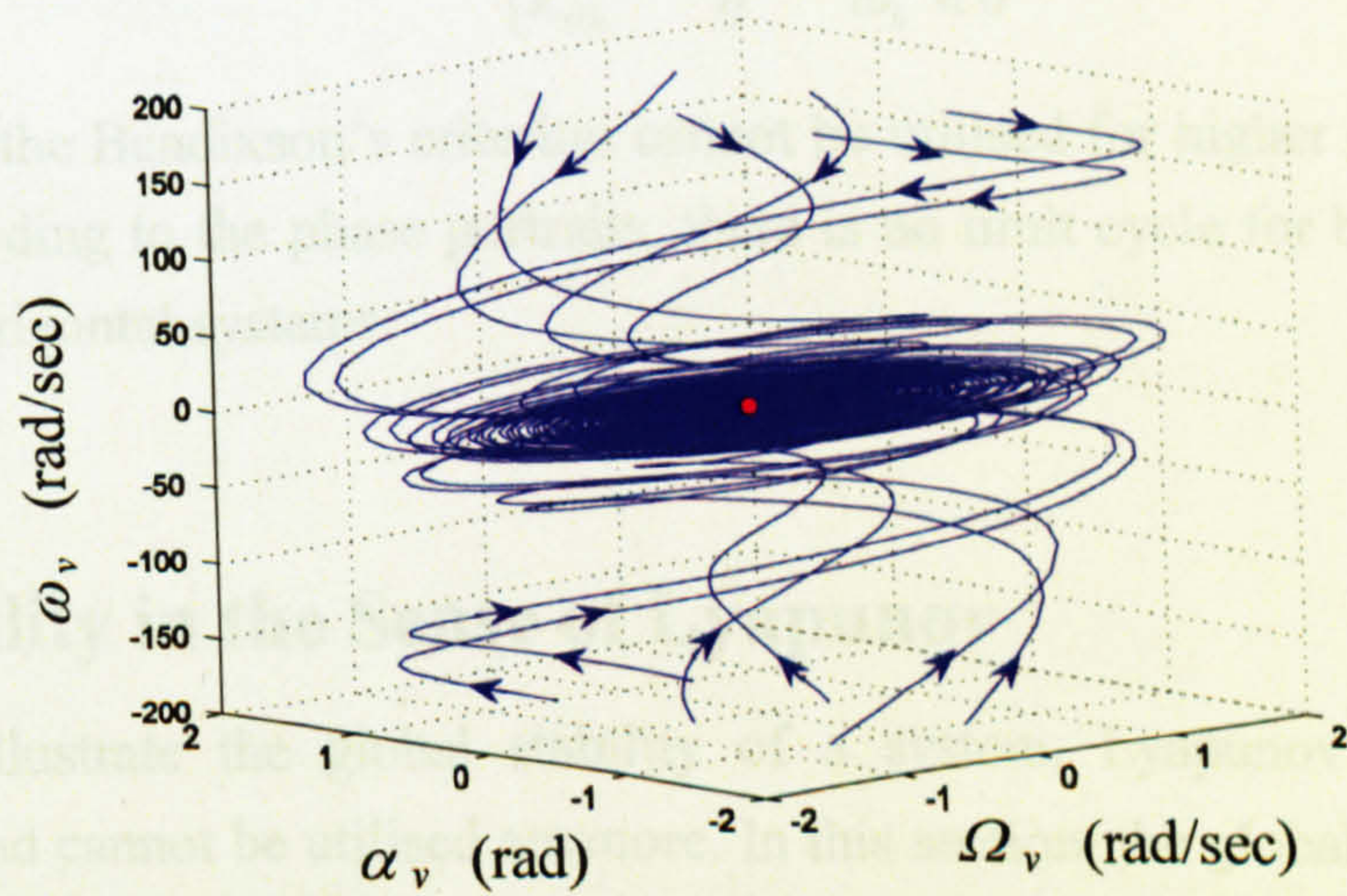


Figure 5.10 The 3D phase portrait of the vertical dynamics

5.5 Limit Cycles

As it is clear from the phase portraits, there is no limit cycle in each case. However, using Bendixson's criterion it can mathematically be proven. Bendixson's criterion says for a second-order nonlinear system of the form of

$$\begin{aligned}\dot{x}_1 &= f_1(x_1, x_2) \\ \dot{x}_2 &= f_2(x_1, x_2)\end{aligned}\tag{5.27}$$

there is no limit cycle in a specified region of the phase plane if $\partial f_1/\partial x_1 + \partial f_2/\partial x_2$ is non-zero and does not change sign. The following expressions show this test for the main motor, the tail motor, the vertical dynamics and the horizontal dynamics respectively.

$$\partial f_1/\partial x_1 + \partial f_2/\partial x_2 = -\frac{R_{av}}{L_{av}} - \frac{B_{mr}}{J_{mr}} - \frac{2k_{tvp/n}}{J_{mr}} |\omega_v| < 0\tag{5.28}$$

$$\partial f_1/\partial x_1 + \partial f_2/\partial x_2 = -\frac{R_{ah}}{L_{ah}} - \frac{B_{tr}}{J_{tr}} - \frac{2k_{thp/n}}{J_{tr}} |\omega_h| < 0\tag{5.29}$$

$$\partial f_1/\partial x_1 + \partial f_2/\partial x_2 = -\frac{k_{vfv}}{J_v} < 0\tag{5.30}$$

$$\partial f_1/\partial x_1 + \partial f_2/\partial x_2 = -\frac{k_{vfh}}{D+F} < 0\tag{5.31}$$

where

$$k_{tvp/n} = \begin{cases} k_{tvp} & \text{if } \omega_v \geq 0 \\ k_{tvn} & \text{if } \omega_v < 0 \end{cases}$$

$$k_{thp/n} = \begin{cases} k_{thp} & \text{if } \omega_h \geq 0 \\ k_{thn} & \text{if } \omega_h < 0 \end{cases}$$

Unfortunately, the Bendixson's criterion cannot be utilised for higher order systems; however, according to the phase portraits, there is no limit cycle for both the 1DOF vertical and horizontal systems.

5.6 Stability in the Sense of Lyapunov

In order to illustrate the global stability of a system, Lyapunov's linearisation stability method cannot be utilised anymore. In this section, the global stability of the TRMS is considered.

5.6.1 The 1DOF Horizontal

Consider the dynamic equations of the TRMS in terms of its 1DOF horizontal, equations (5.19)-(5.22). Since the tail motor dynamics is decoupled from the rest of the dynamics, if one can guarantee the exponential stability of the tail motor, the first two equations, then the asymptotic stability of the total system can be guaranteed by the exponential stability of the second part, horizontal dynamics and if the horizontal dynamics is Lipschitz in its state variables [125]. Generally, the phase portrait shown in Figure 5.5 cannot provide the opportunity to distinguish between global asymptotic stability and global exponential stability. Therefore, one needs to find a radially unbounded Lyapunov function $V(\mathbf{x})$ (i.e. $V(\mathbf{x}) \rightarrow \infty$ as $\|\mathbf{x}\| \rightarrow \infty$) to satisfy $\dot{V}(\mathbf{x}) \leq -\alpha V(\mathbf{x})$ for a strictly positive value of α . To this effect, the Lyapunov function of the form of $V(\mathbf{x}) = \mathbf{x}^T \mathbf{P} \mathbf{x}$ is chosen where,

$$\mathbf{x} = [i_{ah} \ \omega_h]^T$$

and

$$\mathbf{P} = \begin{bmatrix} \frac{k_{ch}\phi_h}{J_{lr}} & 0 \\ 0 & \frac{k_{ch}\phi_h}{L_{ah}} \end{bmatrix}.$$

The tail motor is exponentially stable for any positive value of α which satisfies,

$$\alpha < \min\left(\frac{2R_{ah}}{L_{ah}}, \frac{2B_{lr}}{J_{lr}}\right).$$

Since the tail motor system is globally exponentially stable, the state variables i_{ah} and ω_h in equations (5.21) and (5.22) can be substituted by \bar{i}_{ah} and $\bar{\omega}_h$. The equations can be rewritten as,

$$\frac{d}{dt} \begin{bmatrix} \alpha_h \\ \Omega_h \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k_{chp/n}}{D+F} & -\frac{k_{vfn}}{D+F} \end{bmatrix} \begin{bmatrix} \alpha_h \\ \Omega_h \end{bmatrix} + \begin{bmatrix} 0 \\ l_1 f_2(\bar{\omega}_h) \end{bmatrix} \quad (5.32)$$

where

$$k_{chp/n} = \begin{cases} k_{chp} & \text{if } \alpha_h \geq 0 \\ k_{chn} & \text{if } \alpha_h < 0 \end{cases}$$

which is a globally exponentially stable system as it is a linear system in the standard controllable form with all negative coefficients in the last row of related matrix \mathbf{A} .

The horizontal dynamics is globally Lipschitz in α_h and Ω_h , and also Lipschitz in ω_h from the practical point of view because ω_h has upper and lower limits. Therefore, the forth-order 1DOF horizontal system is globally exponentially stable.

5.6.2 The 1DOF Vertical

Similar to the tail motor, the main motor is globally exponentially stable. Therefore, it is enough to show the global exponential stability of the system dynamics presented by equations (5.25) and (5.26) where ω_v is substituted by $\bar{\omega}_v$. The equations can be rewritten as follows,

$$\frac{d\Omega_v}{dt} = -a_1\Omega_v - a_2 \sin \alpha'_v + a_3 |\bar{\omega}_v| \bar{\omega}_v \quad (5.33)$$

$$\frac{d\alpha'_v}{dt} = \Omega_v \quad (5.34)$$

where, a_1 , a_2 and a_3 are positive constants. The candidate Lyapunov function can be obtained from the augmentation of the kinetic and potential energies as follow,

$$V = \frac{1}{2}\Omega_v^2 - a_2 \cos \alpha'_v + a_2 \quad (5.35)$$

Unfortunately, the differentiation of the Lyapunov function is not negative definite and, in fact, it is negative semi-definite,

$$\dot{V} = -a_1\Omega_v^2 \leq 0 \quad (5.36)$$

Fortunately, in such a situation it is still possible to provide the asymptotic stability of the system with the aid of LaSalle's invariance principle [124]. Consider an autonomous dynamic system as $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$. Assume a compact (closed and bounded) positively invariant set Ω and a Lyapunov function V on Ω exist for the mentioned dynamic system. Define the set

$$N = \{\mathbf{x} \in R^n \mid \dot{V}(\mathbf{x}) = 0\} \quad (5.37)$$

This means N is the set of all points within Ω where $\dot{V}(\mathbf{x}) = 0$. Now let N^* be the largest positively invariant set in N . Then every solution of the dynamic system on Ω approaches N^* as $t \rightarrow +\infty$. This theorem is well-known as LaSalle's invariance principle [124].

With regard to the dynamic system represented by equations (5.33) and (5.34), the

sets Ω and N can be defined as,

$$\Omega = \{(\Omega_v, \alpha'_v) \in R^2\} \quad (5.38)$$

$$N = \{(0, \alpha'_v) | \alpha'_v \in R\} \quad (5.39)$$

and consequently, the largest positively invariant set in N is found as,

$$N^* = \{(0, k\pi) | k \in Z\} \quad (5.40)$$

However, due to the mechanical restriction, the set Ω and N are modified as,

$$\Omega = \left\{ (\Omega_v, \alpha'_v) \in R^2 \mid -\frac{\pi}{2} - \alpha_{v0} < \alpha'_v < \frac{\pi}{2} - \alpha_{v0} \right\} \quad (5.41)$$

$$N = \left\{ (0, \alpha'_v) \mid -\frac{\pi}{2} - \alpha_{v0} < \alpha'_v < \frac{\pi}{2} - \alpha_{v0} \right\} \quad (5.42)$$

and the plausible invariant set is,

$$N^* = (0, 0) \quad (5.43)$$

Therefore, every solution originating from the Ω tends to N^* as $t \rightarrow +\infty$ and subsequently, the system is uniformly (almost) globally asymptotically stable. The Lyapunov function and the sets N (blue line) and N^* (red dot) have been shown in Figure 5.11.

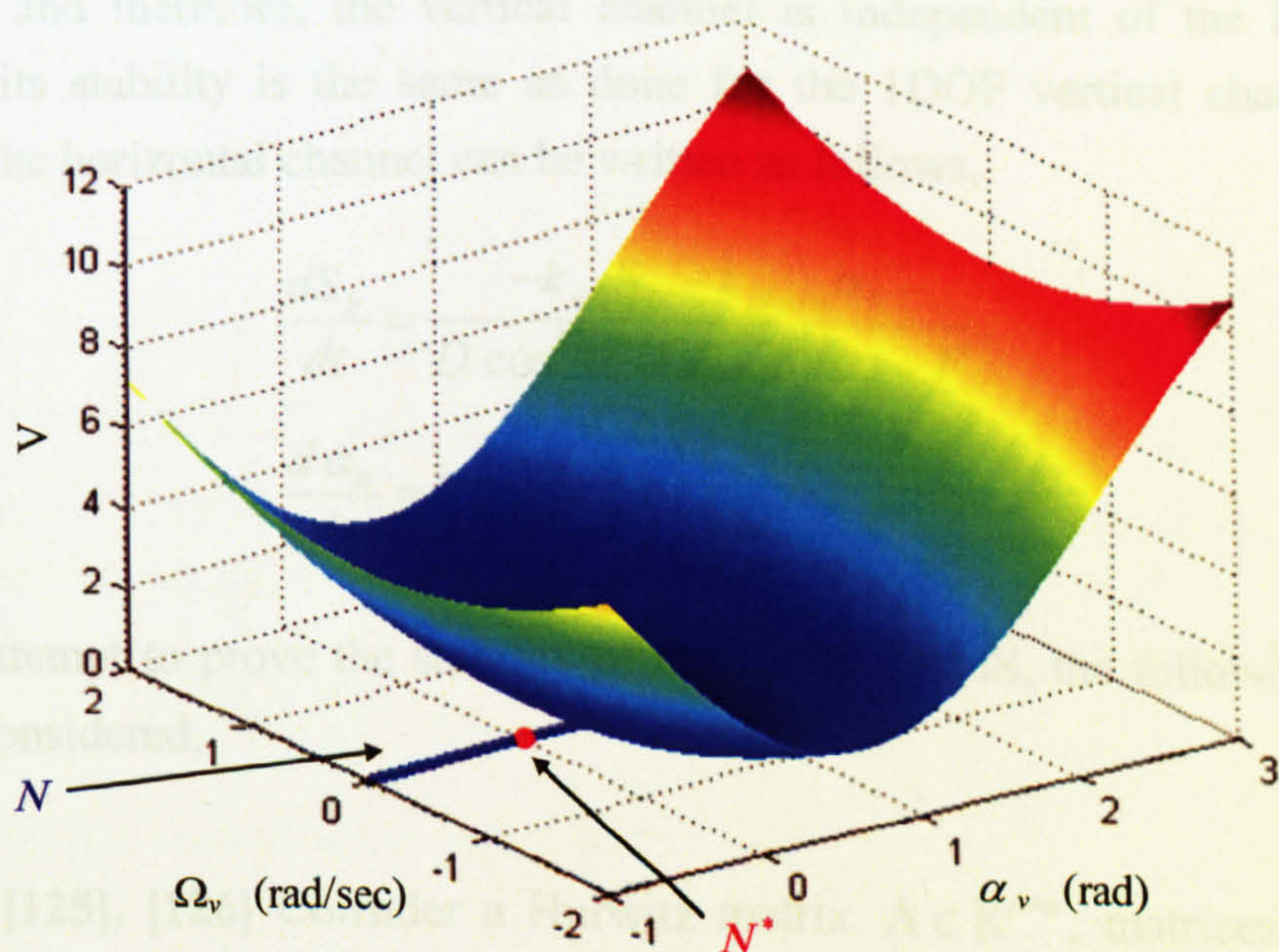


Figure 5.11 The Lyapunov function and the sets N and N^*

Unfortunately we could not show the exponential stability of the vertical dynamics and only the asymptotic stability has been guaranteed. Therefore, the stability of the vertical dynamics is guaranteed when the effect of the main motor is neglected.

5.6.3 The 2DOF TRMS

In order to show the stability of the 2DOF TRMS, consider the global exponential stability of the tail and main motors as shown before. The forth-order equations of the rest of the dynamics can be expressed as,

$$\frac{dS_h}{dt} = \frac{l_t f_2(\bar{\omega}_h) \cos \alpha_v - k_{vfh} \Omega_h - f_3(\alpha_h)}{D \cos^2 \alpha_v + E \sin^2 \alpha_v + F} \quad (5.44)$$

$$\frac{d\alpha_h}{dt} = S_h + \frac{k_m \bar{\omega}_v \cos \alpha_v}{D \cos^2 \alpha_v + E \sin^2 \alpha_v + F} \quad (5.45)$$

$$\begin{aligned} \frac{dS_v}{dt} = & \frac{f_5(\bar{\omega}_v)(l_m + k_g \Omega_h \cos \alpha_v) - k_{vfv} \Omega_v}{J_v} \\ & + \frac{g[(A - B) \cos \alpha_v - C \sin \alpha_v] - 0.5 \Omega_h^2 H \sin 2\alpha_v}{J_v} \end{aligned} \quad (5.46)$$

$$\frac{d\alpha_v}{dt} = S_v + \frac{k_t}{J_v} \bar{\omega}_h \quad (5.47)$$

where, $\Omega_h = \dot{\alpha}_h$ and $\Omega_v = \dot{\alpha}_v$. In order to prove the stability of the TRMS in terms of its 2DOF dynamics, the centrifugal and gyroscopic torques are neglected (i.e. $H = k_g = 0$) and therefore, the vertical channel is independent of the horizontal channel and its stability is the same as done for the 1DOF vertical channel. The equations of the horizontal channel can be written as follows,

$$\frac{dS_h}{dt} = \frac{-k_{vfh} S_h - k_{chp/n} \alpha_h}{D \cos^2 \alpha_v + E \sin^2 \alpha_v + F} \quad (5.48)$$

$$\frac{d\alpha_h}{dt} = S_h \quad (5.49)$$

Before any attempt to prove the stability of the 2DOF TRMS, the following lemma needs to be considered,

Lemma 5.1 [125], [126] Consider a Hurwitz matrix $A \in \mathbb{R}^{n \times n}$, matrices $B \in \mathbb{R}^{n \times 1}$, $C \in \mathbb{R}^{1 \times n}$ and some number $\gamma > 0$. Denote $A_\gamma^- := A - \gamma BC$ and $A_\gamma^+ := A + \gamma BC$. The following conditions are equivalent:

(i) There exists $\mathbf{P} = \mathbf{P}^T > 0$ such that

$$\mathbf{P}\mathbf{A}_\gamma^- + (\mathbf{A}_\gamma^-)^T \mathbf{P} < 0, \quad \mathbf{P}\mathbf{A}_\gamma^+ + (\mathbf{A}_\gamma^+)^T \mathbf{P} < 0 \quad (5.50)$$

(ii) The matrix \mathbf{A} is Hurwitz and $|\mathbf{C}(i\omega\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}| < \frac{1}{\gamma}$ for all $\omega \in \mathbb{R}$

Let rewrite equations (5.48) and (5.49) as follows,

$$\dot{q}_1 = q_2 \quad (5.51)$$

$$\dot{q}_2 = \frac{1}{f(t)}(-a_1 q_1 - a_2 q_2) \quad (5.52)$$

where $\frac{1}{D+F} = \alpha \leq \frac{1}{f(t)} \leq \beta = \frac{1}{D+F+(E-D)\sin^2\alpha_\gamma^{\max}}$. Note that α_γ^{\max} is assumed to be 0.7 rad.

Let $\delta = \frac{\alpha+\beta}{2}$ and $\gamma = \frac{\beta-\alpha}{2}$ and define the nominal matrix of \mathbf{A} and matrices \mathbf{B} and \mathbf{C} as,

$$\mathbf{A}_{\text{nom}} = \begin{bmatrix} 0 & 1 \\ -\delta a_1 & -\delta a_2 \end{bmatrix} \quad (5.53)$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (5.54)$$

$$\mathbf{C} = [a_1 \quad a_2] \quad (5.55)$$

Now \mathbf{A}_γ^- and \mathbf{A}_γ^+ can easily be found as

$$\mathbf{A}_\gamma^- := \mathbf{A}_{\text{nom}} - \gamma\mathbf{B}\mathbf{C} = \begin{bmatrix} 0 & 1 \\ -\beta a_1 & -\beta a_2 \end{bmatrix} \quad (5.56)$$

$$\mathbf{A}_\gamma^+ := \mathbf{A}_{\text{nom}} + \gamma\mathbf{B}\mathbf{C} = \begin{bmatrix} 0 & 1 \\ -\alpha a_1 & -\alpha a_2 \end{bmatrix} \quad (5.57)$$

According to lemma 5.1 if $|\mathbf{C}(i\omega\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}| < \frac{1}{\gamma}$ for all $\omega \in \mathbb{R}$ then there exists a positive definite matrix \mathbf{P} such that $\mathbf{P}\mathbf{A}_\gamma^- + (\mathbf{A}_\gamma^-)^T \mathbf{P} < 0$ and $\mathbf{P}\mathbf{A}_\gamma^+ + (\mathbf{A}_\gamma^+)^T \mathbf{P} < 0$ satisfy. Then the Lyapunov function is $V = \frac{1}{2}[q_1 \quad q_2]\mathbf{P}[q_1 \quad q_2]^T$. To this end, the following condition needs to be held,

$$\omega^4 + [\alpha\beta a_2^2 - (\beta + \alpha)a_1]\omega^2 + \alpha\beta a_1^2 > 0 \quad (5.58)$$

It will be satisfied if its discriminant is negative, viz,

$$\Delta = [\alpha\beta a_2^2 - (\beta + \alpha)a_1]^2 - 4\alpha\beta a_1^2 < 0 \quad (5.59)$$

The above condition holds as long as the pitch angle remains in the interval of [-0.7,

0.7] rad. Therefore, the TRMS in terms of its 2DOF dynamics is asymptotically stable as long as the pitch angle stays inside the aforementioned interval.

Here, we are trying to use another approach to prove the stability of the TRMS using generalised Krasovskii theorem.

Theorem 5.1 (Generalised Krasovskii Theorem) [123] Consider an autonomous system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ with the equilibrium point of interest being the origin, and let $\mathbf{A}(\mathbf{x})$ denote the Jacobian matrix of the system. Then a sufficient condition for the origin to be asymptotically stable is that there exists a symmetric positive definite matrix \mathbf{P} , such that $\forall \mathbf{x} \neq \mathbf{0}$, the matrix

$$\mathbf{F}(\mathbf{x}) = \mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} \quad (5.60)$$

is negative definite in some neighbourhood of the origin. The function $V(\mathbf{x}) = \mathbf{f}^T \mathbf{P} \mathbf{f}$ is then a Lyapunov function for the system.

According to equations (5.51) and (5.52), the Jacobian matrix can be defined as,

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -\frac{a_1}{f(t)} & -\frac{a_2}{f(t)} \end{bmatrix} \quad (5.61)$$

where again $\alpha \leq \frac{1}{f(t)} \leq \beta$. Assume that the symmetric positive definite matrix \mathbf{P} is defined as,

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} \quad (5.62)$$

Therefore, the matrix $\mathbf{F}(\mathbf{x})$ can be formed while the entries of \mathbf{P} are unknown and should be found in such a way the matrices $-\mathbf{F}(\mathbf{x})$ and \mathbf{P} become positive definite. To this end, the principal minors of $-\mathbf{F}(\mathbf{x})$ and \mathbf{P} must be positive. The following matrix satisfies the conditions,

$$\mathbf{P} = \begin{bmatrix} 3.3 & 1 \\ 1 & 20 \end{bmatrix}$$

5.7 Conclusions

The TRMS is a nonlinear system with multiple equilibrium points; however, due to

the mechanical restrictors only one equilibrium point is feasible, which is the origin. On the basis of Lyapunov's linearisation method, the equilibrium point is locally asymptotically stable. The dynamics of the TRMS in terms of 1DOF horizontal and 1DOF vertical are globally asymptotically stable using Lyapunov and invariant set theorems. The TRMS in terms of its 2DOF dynamics is asymptotically stable as long as the pitch angle lies in the interval of $[-0.7, 0.7]$ rad.

CHAPTER 6

Nonlinear Model Predictive Control and Stability

This chapter investigates the development of a full state-feedback NMPC approach for the TRMS. The method is well-known as multistep Newton-type control strategy; however, the formulation here differs from the original one and has some advantages over it. A state-dependent model is used to find the linear models at each instant. The stability of the nonlinear MPC control system is provided using terminal equality constraints.

6.1 Introduction

Nowadays, there is no doubt that MPC is one of the popularly used multivariable control algorithms in industrial processes and indeed in other areas [9]. Note that MPC displays its main strength when applied to problems with some characteristics as follows,

- Large number of manipulated and controlled variables
- Constraints imposed on both the manipulated and controlled variables
- Changing control objectives and equipment failure
- Presence of time delays

The concept of all the controllers belonging to the MPC family is characterised by the following steps,

1. A plant-model predicts the future outputs for a pre-determined horizon, called the prediction horizon N_p , at each instant k . These predicted outputs $\hat{y}(k+i|k)$, $i=1,2,\dots,N_p$, depend on the past inputs and outputs, and also the future control signals $u(k+i|k)$ for $i=0,1,\dots,N_p-1$ ($u(k+i|k)=u(k+N_c|k)$ for $i>N_c$, where N_c is the control horizon).

2. According to specific criteria a set of future control signals is optimally calculated to have the process output as close as possible to the reference trajectory (which is either the setpoint itself or an approximation of it). The augmented criterion, objective function, usually takes the form of a quadratic function of the errors between the predicted output signal and the predicted reference trajectory. The control effort is included in the objective function in most cases.

3. Based on the receding horizon concept, the first part of the control signal, $u(k | k)$, is sent to the process, whilst the remaining parts of the control signal are rejected [9].

The various MPC algorithms may differ from the perspective of the model used to represent the process, the noise, the constraints and the cost function to be minimised. The block diagram of a generic MPC is depicted in Figure 6.1.

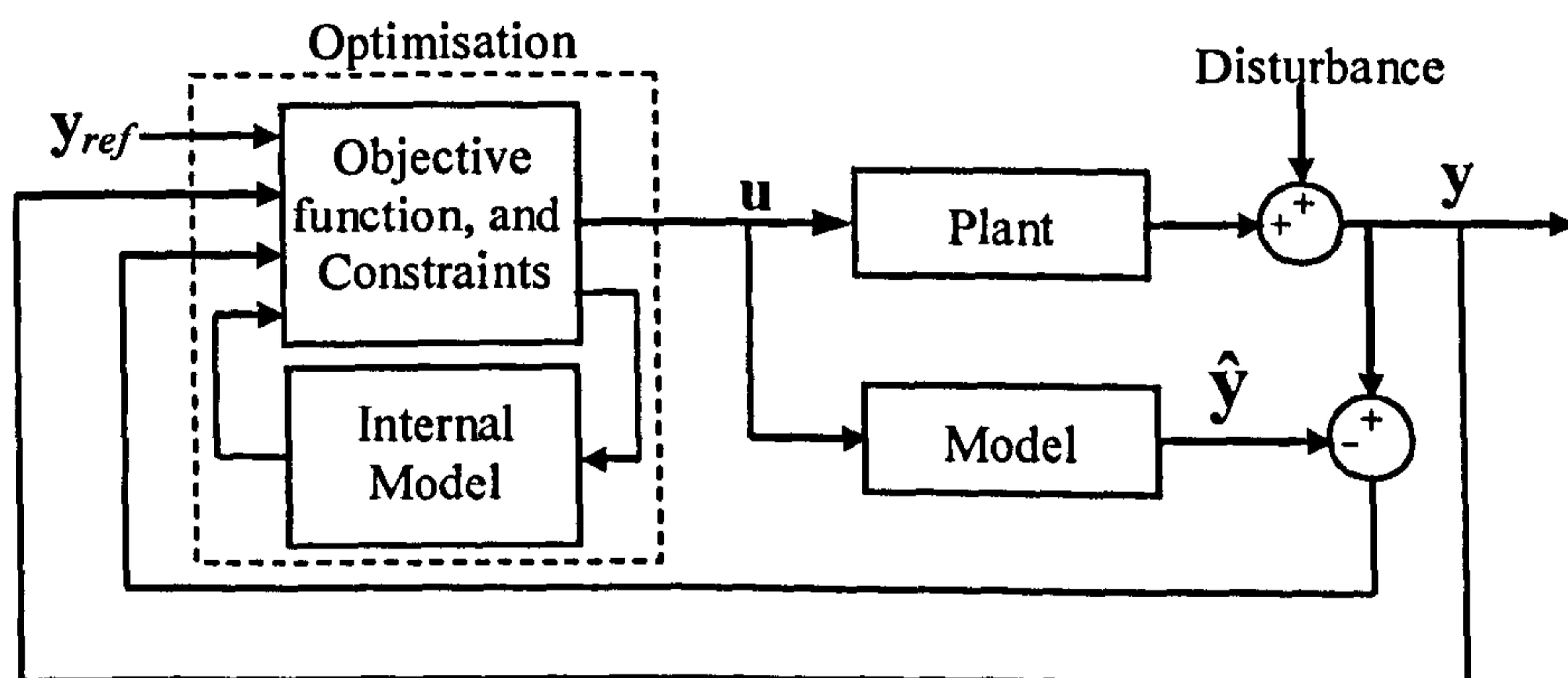


Figure 6.1 The block diagram of a generic MPC

When the internal plant-model and the constraints are linear, one is dealing with a linear MPC system and the mature linear MPC algorithms can be utilised. However, as stated previously, in the case of strict nonlinear systems, a linear internal model causes considerable performance degradation due to the fact that the performance of a model predictive controller severely depends on the model accuracy. Therefore, the necessity of nonlinear MPC approaches is obvious from a practical point of view.

In this chapter, a stable multistep Newton-type MPC is used to control a constrained nonlinear MIMO system, the TRMS. The proposed approach is envisaged to be used for any other nonlinear system when an accurate nonlinear plant-model is available.

As mentioned before, the idea has been originated from the work of Li and Biegler [57]. The present research differs from the original one in terms of the way of formulation. In the original one, the objective function variables are the difference between the current and the nominal input trajectories; however, here the variables are input changes during the control horizon. Also, in this study, a state-dependent model is used to create a set of linear models at each instant based on the previous operating point; however, Li *et al.* [57] have used a Jacobian-based linearisation.

6.2 Newton-type MPC Method

A conventional linear time-varying MPC is an MPC method that uses a nonlinear model to update the linear model only at each sample time, k . This method has no satisfactory performance for a severe nonlinear system, since the method uses only a single linear model at each iteration and then updates it in the new step according to the current operating point. Specially, if the prediction horizon is assumed to be large, the error between the linearised model and the nonlinear model gradually increases as approaches to the end of the prediction horizon. In this research, the linear model is updated during the prediction horizon as well. The main problem for updating the linear model during the prediction horizon is that the operating points during the prediction horizon are unknown and the linearisation can be carried out only on the basis of known operating points. In this situation, one approach is using all the control effort from the previous iteration of the optimisation to linearise the nonlinear model during the prediction horizon. Instead of linearisation based on Taylor-expansion, a state-dependent state-space model is used to form a linear model by substituting the previous operating point into the state-dependent matrices. The state-dependent linearisation method is believed to perform significantly better compared to the Taylor-expansion-based linearisation, from the viewpoint of its application in multistep Newton-type MPC.

6.2.1 Methodology

Consider a nonlinear system with n_u inputs, n_y outputs and n_x states described as discrete state-space equations as follows,

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) \\ \mathbf{y}(k) = \mathbf{h}(\mathbf{x}(k)) \end{cases} \quad (6.1)$$

where $\mathbf{x}(k)$ is the state vector, $\mathbf{u}(k)$ is the input vector, and $\mathbf{y}(k)$ is the output

vector, all at instant k . It can adaptively be linearised at each *real*¹ sample time, k , as the following discrete state-space equations,

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}(k) \mathbf{x}(k) + \mathbf{B}(k) \mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{C}(k) \mathbf{x}(k) \end{cases} \quad (6.2)$$

or can be presented by a set of state-dependent state-space equations as,

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}(\mathbf{x}(k)) \mathbf{x}(k) + \mathbf{B}(\mathbf{x}(k)) \mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{C}(\mathbf{x}(k)) \mathbf{x}(k) \end{cases} \quad (6.3)$$

Note that the state variables and inputs related to the previous instant are used as initial conditions to linearise the nonlinear system at each time. Now one needs to linearise the nonlinear system N_p times at each sampling instant according to the N_p operating points from the previous instant of the optimisation results.

$$\begin{cases} \hat{\mathbf{x}}(k+i+1|k) = \mathbf{A}(\mathbf{x}(k+i|k)) \hat{\mathbf{x}}(k+i|k) + \mathbf{B}(\mathbf{x}(k+i|k)) \hat{\mathbf{u}}(k+i|k) \\ \hat{\mathbf{y}}(k+i|k) = \mathbf{C}(\mathbf{x}(k+i|k)) \hat{\mathbf{x}}(k+i|k) \end{cases} \quad i = 0, 1, \dots, N_p - 1 \quad (6.4)$$

It is noted that for the simplicity of representation of the equations, the state-dependent matrix $\mathbf{A}(\mathbf{x}(k+i|k))$ is shown as $\mathbf{A}(k+i|k)$ and similarly for the other state-dependent matrices.

To find the linear models, one can easily use the known values of $\mathbf{x}(k+i|k-1)$ instead of unknown $\mathbf{x}(k+i|k)$ where, $i = 0, 1, \dots, N_p - 1$. In order to solve the optimisation problem of an MPC, one needs to obtain the relationship between the internal model outputs during the prediction horizon interval, $1 \leq i \leq N_p$, and the internal model inputs during the control horizon interval, $0 \leq i \leq N_c - 1$, where N_p and N_c are the prediction and control horizons respectively, see Figure 6.2. If this relationship is linear and the constraints are also linear then the optimisation problem can be a linear quadratic problem.

¹ In model predictive control, two sample instants are considered and should be clarified to prevent any misunderstanding. One is real sample time, and the other is internal sample time. In term $\mathbf{u}(k+i|k)$, k is the real sample time and $k+i$ is the internal sample time.

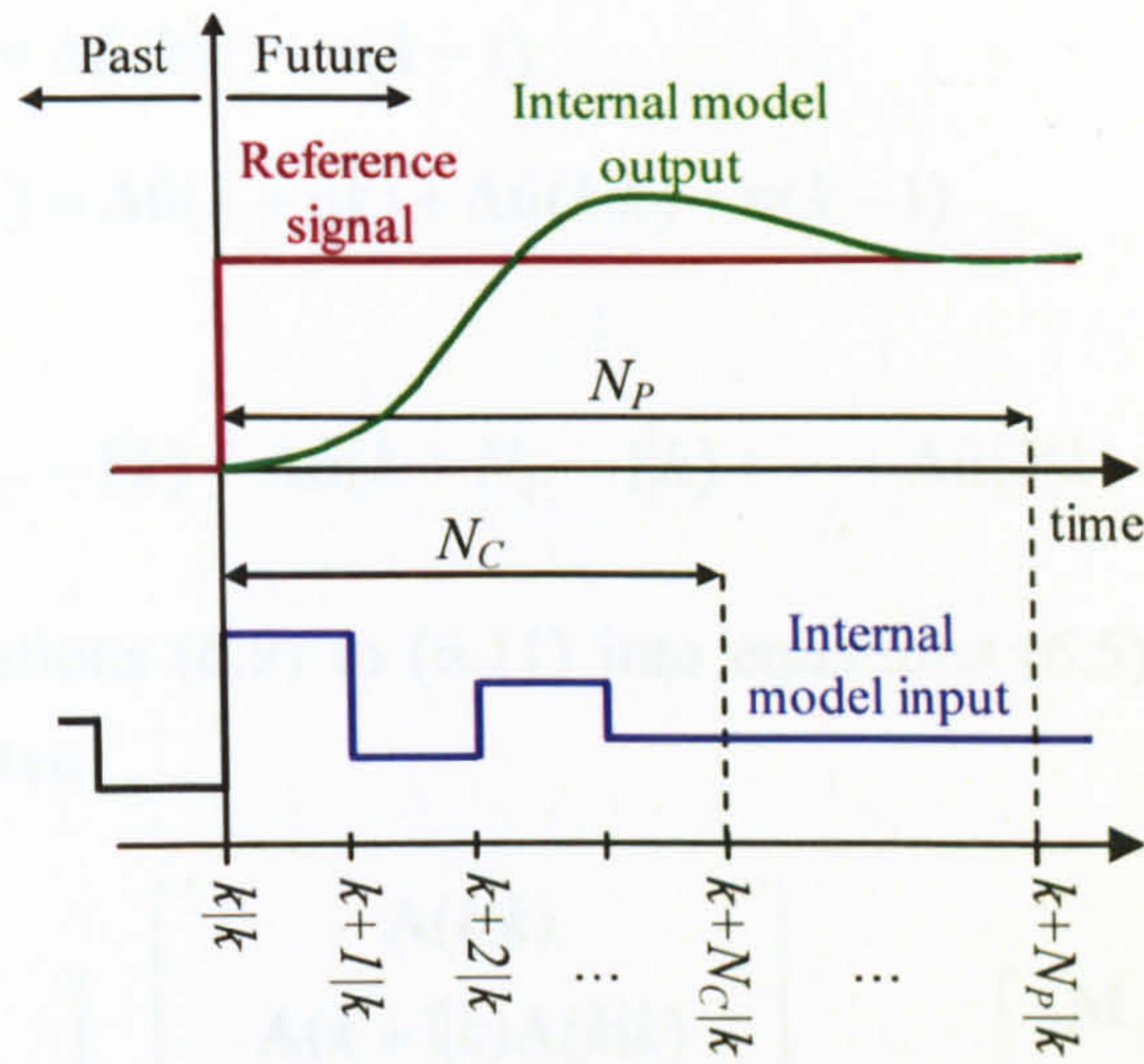


Figure 6.2 The MPC approach of a SISO plant at time k

Assume that all the state variables of the system are available (i.e. they can be measured or observed). The model state variables in the prediction horizon interval with respect to the current state variables and the future inputs can be expressed as,

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{A}(k|k)\mathbf{x}(k) + \mathbf{B}(k|k)\hat{\mathbf{u}}(k|k) \quad (6.5)$$

$$\begin{aligned} \hat{\mathbf{x}}(k+2|k) &= \mathbf{A}(k+1|k)\hat{\mathbf{x}}(k+1|k) + \mathbf{B}(k+1|k)\hat{\mathbf{u}}(k+1|k) \\ &= \mathbf{A}(k+1|k)\mathbf{A}(k|k)\mathbf{x}(k) + \mathbf{A}(k+1|k)\mathbf{B}(k|k)\hat{\mathbf{u}}(k|k) + \mathbf{B}(k+1|k)\hat{\mathbf{u}}(k+1|k) \end{aligned} \quad (6.6)$$

$$\begin{aligned} \hat{\mathbf{x}}(k+3|k) &= \mathbf{A}(k+2|k)\hat{\mathbf{x}}(k+2|k) + \mathbf{B}(k+2|k)\hat{\mathbf{u}}(k+2|k) \\ &= \mathbf{A}(k+2|k)\mathbf{A}(k+1|k)\mathbf{A}(k|k)\mathbf{x}(k) + \mathbf{A}(k+2|k)\mathbf{A}(k+1|k)\mathbf{B}(k|k)\hat{\mathbf{u}}(k|k) \\ &\quad + \mathbf{A}(k+2|k)\mathbf{B}(k+1|k)\hat{\mathbf{u}}(k+1|k) + \mathbf{B}(k+2|k)\hat{\mathbf{u}}(k+2|k) \end{aligned} \quad (6.7)$$

⋮

$$\begin{aligned} \hat{\mathbf{x}}(k+N_p|k) &= \mathbf{A}(k+N_p-1|k)\hat{\mathbf{x}}(k+N_p-1|k) + \mathbf{B}(k+N_p-1|k)\hat{\mathbf{u}}(k+N_p-1|k) \\ &= \mathbf{A}(k+N_p-1|k)\cdots\mathbf{A}(k+1|k)\mathbf{A}(k|k)\mathbf{x}(k) \\ &\quad + \mathbf{A}(k+N_p-1|k)\cdots\mathbf{A}(k+1|k)\mathbf{B}(k|k)\hat{\mathbf{u}}(k|k) \\ &\quad + \mathbf{A}(k+N_p-1|k)\cdots\mathbf{A}(k+2|k)\mathbf{B}(k+1|k)\hat{\mathbf{u}}(k+1|k) \\ &\quad \vdots \\ &\quad + \mathbf{A}(k+N_p-1|k)\mathbf{B}(k+N_p-2|k)\hat{\mathbf{u}}(k+N_p-2|k) \\ &\quad + \mathbf{B}(k+N_p-1|k)\hat{\mathbf{u}}(k+N_p-1|k) \end{aligned} \quad (6.8)$$

It is common to use the change of the input, $\Delta\hat{\mathbf{u}}(k+i|k)$, instead of the input itself, $\hat{\mathbf{u}}(k+i|k)$, where $\Delta\hat{\mathbf{u}}(k+i|k) = \hat{\mathbf{u}}(k+i|k) - \hat{\mathbf{u}}(k+i-1|k)$ [127]. The inputs only change during the control horizon interval and remain constant after that, i.e. $\hat{\mathbf{u}}(k+i|k) = \hat{\mathbf{u}}(k+N_c-1|k)$ or $\Delta\hat{\mathbf{u}}(k+i|k) = 0$ for $N_c \leq i \leq N_p-1$. The relationship between the inputs and the changes of the inputs are as follows,

$$\hat{\mathbf{u}}(k|k) = \Delta \hat{\mathbf{u}}(k|k) + \mathbf{u}(k-1) \quad (6.9)$$

$$\hat{\mathbf{u}}(k+1|k) = \Delta \hat{\mathbf{u}}(k+1|k) + \Delta \hat{\mathbf{u}}(k|k) + \mathbf{u}(k-1) \quad (6.10)$$

⋮

$$\hat{\mathbf{u}}(k+N_C-1|k) = \Delta \hat{\mathbf{u}}(k+N_C-1|k) + \cdots + \Delta \hat{\mathbf{u}}(k|k) + \mathbf{u}(k-1) \quad (6.11)$$

By substituting equations (6.9) to (6.11) into equations (6.5) to (6.8) the following equation can be written,

$$\begin{bmatrix} \hat{\mathbf{x}}(k+1|k) \\ \hat{\mathbf{x}}(k+2|k) \\ \vdots \\ \hat{\mathbf{x}}(k+N_C|k) \\ \hat{\mathbf{x}}(k+N_C+1|k) \\ \vdots \\ \hat{\mathbf{x}}(k+N_P|k) \end{bmatrix}_{n_x N_P \times 1} = \begin{bmatrix} \mathbf{A}(k|k) \\ \mathbf{A}(k+1|k)\mathbf{A}(k|k) \\ \vdots \\ \prod_{i=1}^{N_C} \mathbf{A}(k+N_C-i|k) \\ \prod_{i=1}^{N_C+1} \mathbf{A}(k+N_C+1-i|k) \\ \vdots \\ \prod_{i=1}^{N_P} \mathbf{A}(k+N_P-i|k) \end{bmatrix}_{n_x N_P \times n_x} \mathbf{x}(k) + \begin{bmatrix} \mathbf{M}_{1,1}(k) \\ \mathbf{M}_{2,1}(k) \\ \vdots \\ \mathbf{M}_{N_C,1}(k) \\ \mathbf{M}_{N_C+1,1}(k) \\ \vdots \\ \mathbf{M}_{N_P,1}(k) \end{bmatrix}_{n_x N_P \times n_u} \mathbf{u}(k-1) + \begin{bmatrix} \mathbf{M}_{1,1}(k) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{M}_{2,1}(k) & \mathbf{M}_{2,2}(k) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{M}_{N_C,1}(k) & \mathbf{M}_{N_C,2}(k) & \cdots & \mathbf{M}_{N_C,N_C}(k) \\ \mathbf{M}_{N_C+1,1}(k) & \mathbf{M}_{N_C+1,2}(k) & \cdots & \mathbf{M}_{N_C+1,N_C}(k) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{M}_{N_P,1}(k) & \mathbf{M}_{N_P,2}(k) & \cdots & \mathbf{M}_{N_P,N_C}(k) \end{bmatrix}_{n_x N_P \times n_u N_C} \begin{bmatrix} \Delta \hat{\mathbf{u}}(k|k) \\ \Delta \hat{\mathbf{u}}(k+1|k) \\ \vdots \\ \Delta \hat{\mathbf{u}}(k+N_C-1|k) \end{bmatrix}_{n_u N_C \times 1} \quad (6.12)$$

where,

$$\mathbf{M}_{1,1}(k) = \mathbf{B}(k|k)$$

$$\mathbf{M}_{2,1}(k) = \mathbf{A}(k+1|k)\mathbf{B}(k|k) + \mathbf{B}(k+1|k)$$

$$\mathbf{M}_{2,2}(k) = \mathbf{B}(k+1|k)$$

⋮

$$\mathbf{M}_{N_C,1}(k) = \sum_{j=0}^{N_C-2} \left(\prod_{i=1}^{N_C-1-j} \mathbf{A}(k+N_C-i|k) \right) \mathbf{B}(k+j|k) + \mathbf{B}(k+N_C-1|k)$$

$$\mathbf{M}_{N_C,2}(k) = \sum_{j=1}^{N_C-2} \left(\prod_{i=1}^{N_C-1-j} \mathbf{A}(k+N_C-i|k) \right) \mathbf{B}(k+j|k) + \mathbf{B}(k+N_C-1|k)$$

⋮

$$\mathbf{M}_{N_C, N_C}(k) = \mathbf{B}(k + N_C - 1|k)$$

$$\mathbf{M}_{N_C+1, 1}(k) = \sum_{j=0}^{N_C-1} \left(\prod_{i=1}^{N_C-j} \mathbf{A}(k + N_C + 1 - i|k) \right) \mathbf{B}(k + j|k) + \mathbf{B}(k + N_C|k)$$

$$\mathbf{M}_{N_C+1, 2}(k) = \sum_{j=1}^{N_C-1} \left(\prod_{i=1}^{N_C-j} \mathbf{A}(k + N_C + 1 - i|k) \right) \mathbf{B}(k + j|k) + \mathbf{B}(k + N_C|k)$$

⋮

$$\mathbf{M}_{N_C+1, N_C}(k) = \mathbf{A}(k + N_C|k) \mathbf{B}(k + N_C - 1|k) + \mathbf{B}(k + N_C|k)$$

⋮

$$\mathbf{M}_{N_P, 1}(k) = \sum_{j=0}^{N_P-2} \left(\prod_{i=1}^{N_P-1-j} \mathbf{A}(k + N_P - i|k) \right) \mathbf{B}(k + j|k) + \mathbf{B}(k + N_P - 1|k)$$

$$\mathbf{M}_{N_P, 2}(k) = \sum_{j=1}^{N_P-2} \left(\prod_{i=1}^{N_P-1-j} \mathbf{A}(k + N_P - i|k) \right) \mathbf{B}(k + j|k) + \mathbf{B}(k + N_P - 1|k)$$

⋮

$$\mathbf{M}_{N_P, N_C}(k) = \sum_{j=N_C-1}^{N_P-2} \left(\prod_{i=1}^{N_P-1-j} \mathbf{A}(k + N_P - i|k) \right) \mathbf{B}(k + j|k) + \mathbf{B}(k + N_P - 1|k)$$

Note that in equation (6.12), the first two terms are related to the past variables that are known and the last term is associated with the future signals that should be optimally calculated using an optimisation technique. The output predictions can be obtained as,

$$\underbrace{\begin{bmatrix} \hat{\mathbf{y}}(k+1|k) \\ \hat{\mathbf{y}}(k+2|k) \\ \vdots \\ \hat{\mathbf{y}}(k+N_P|k) \end{bmatrix}}_{n_y N_P \times 1} = \underbrace{\begin{bmatrix} \mathbf{C}(k+1|k) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}(k+2|k) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{C}(k+N_P|k) \end{bmatrix}}_{n_y N_P \times n_x N_P} \underbrace{\begin{bmatrix} \hat{\mathbf{x}}(k+1|k) \\ \hat{\mathbf{x}}(k+2|k) \\ \vdots \\ \hat{\mathbf{x}}(k+N_P|k) \end{bmatrix}}_{n_x N_P \times 1} + \underbrace{\begin{bmatrix} \hat{\mathbf{d}}(k+1|k) \\ \hat{\mathbf{d}}(k+2|k) \\ \vdots \\ \hat{\mathbf{d}}(k+N_P|k) \end{bmatrix}}_{n_y N_P \times 1} \quad (6.13)$$

where $\hat{\mathbf{d}}(k+i|k)$ is the disturbance that can be considered either a constant value for all i or can be estimated as the difference between the real and estimated outputs. Substituting (6.12) into (6.13) leads to the following equation,

$$\mathbf{Y}(k) = \mathbf{M}_C(k) \mathbf{M}_A(k) \mathbf{x}(k) + \mathbf{M}_C(k) \mathbf{M}_B(k) \mathbf{u}(k-1) + \mathbf{M}_C(k) \mathbf{M}_U(k) \Delta \mathbf{U}(k) + \mathbf{M}_d(k) \quad (6.14)$$

where,

$$\mathbf{Y}(k) = \begin{bmatrix} \hat{\mathbf{y}}(k+1|k) \\ \hat{\mathbf{y}}(k+2|k) \\ \vdots \\ \hat{\mathbf{y}}(k+N_P|k) \end{bmatrix}_{n_y N_P \times 1}$$

$$\mathbf{M}_C(k) = \begin{bmatrix} \mathbf{C}(k+1|k) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}(k+2|k) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{C}(k+N_P|k) \end{bmatrix}_{n_y N_P \times n_x N_P}$$

$$\mathbf{M}_A(k) = \begin{bmatrix} \mathbf{A}(k|k) \\ \mathbf{A}(k+1|k)\mathbf{A}(k|k) \\ \vdots \\ \prod_{i=1}^{N_P} \mathbf{A}(k+N_P-i|k) \end{bmatrix}_{n_x N_P \times n_x}$$

$$\mathbf{M}_B(k) = \begin{bmatrix} \mathbf{M}_{1,1}(k) \\ \mathbf{M}_{2,1}(k) \\ \vdots \\ \mathbf{M}_{N_P,1}(k) \end{bmatrix}_{n_x N_P \times n_u}$$

$$\mathbf{M}_U(k) = \begin{bmatrix} \mathbf{M}_{1,1}(k) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{M}_{2,1}(k) & \mathbf{M}_{2,2}(k) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{M}_{N_P,1}(k) & \mathbf{M}_{N_P,2}(k) & \cdots & \mathbf{M}_{N_P,N_C}(k) \end{bmatrix}_{n_x N_P \times n_u N_C}$$

$$\Delta \mathbf{U}(k) = \begin{bmatrix} \Delta \hat{\mathbf{u}}(k|k) \\ \Delta \hat{\mathbf{u}}(k+1|k) \\ \vdots \\ \Delta \hat{\mathbf{u}}(k+N_C-1|k) \end{bmatrix}_{n_u N_C \times 1}$$

$$\mathbf{M}_d(k) = \begin{bmatrix} \hat{\mathbf{d}}(k+1|k) \\ \hat{\mathbf{d}}(k+2|k) \\ \vdots \\ \hat{\mathbf{d}}(k+N_P|k) \end{bmatrix}_{n_y N_P \times 1}$$

6.2.2 Objective Function

Assume that the following objective function should be minimised according to the constraints (6.16) to (6.18),

$$J(k) = \sum_{i=1}^{N_p} [\mathbf{r}(k+i) - \hat{\mathbf{y}}(k+i|k)]^T \boldsymbol{\delta}(i) [\mathbf{r}(k+i) - \hat{\mathbf{y}}(k+i|k)] + \sum_{i=1}^{N_c} [\Delta \hat{\mathbf{u}}(k+i-1|k)]^T \boldsymbol{\lambda}(i) [\Delta \hat{\mathbf{u}}(k+i-1|k)] \quad (6.15)$$

$$\mathbf{y}_{\min} \leq \hat{\mathbf{y}}(k+i|k) \leq \mathbf{y}_{\max} \quad i = 1, 2, \dots, N_p \quad (6.16)$$

$$\mathbf{u}_{\min} \leq \hat{\mathbf{u}}(k+i-1|k) \leq \mathbf{u}_{\max} \quad i = 1, 2, \dots, N_c \quad (6.17)$$

$$\Delta \mathbf{u}_{\min} \leq \Delta \hat{\mathbf{u}}(k+i-1|k) \leq \Delta \mathbf{u}_{\max} \quad i = 1, 2, \dots, N_c \quad (6.18)$$

where,

\mathbf{r} : Reference trajectory with dimension $(n_y \times 1)$

$\boldsymbol{\delta}$: Weighting matrix of tracking error with dimension $(n_y \times n_y)$

$\boldsymbol{\lambda}$: Weighting matrix of control effort with dimension $(n_u \times n_u)$

The indices min and max highlight the lower and upper bounds respectively.

The objective function can be rewritten as,

$$J(k) = [\mathbf{M}_r(k) - \mathbf{Y}(k)]^T \mathbf{Q} [\mathbf{M}_r(k) - \mathbf{Y}(k)] + \Delta \mathbf{U}^T(k) \mathbf{R} \Delta \mathbf{U}(k) \quad (6.19)$$

where,

$$\mathbf{M}_r = \begin{bmatrix} \mathbf{r}(k+1) \\ \mathbf{r}(k+2) \\ \vdots \\ \mathbf{r}(k+N_p) \end{bmatrix}_{n_y N_p \times 1}$$

$$\mathbf{Q} = \begin{bmatrix} \boldsymbol{\delta}(1) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\delta}(2) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{\delta}(N_p) \end{bmatrix}_{n_y N_p \times n_y N_p}$$

$$\mathbf{R} = \begin{bmatrix} \boldsymbol{\lambda}(1) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\lambda}(2) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{\lambda}(N_c) \end{bmatrix}_{n_u N_c \times n_u N_c}$$

By substituting equation (6.14) into (6.19) the following linear quadratic function is obtained,

$$J(k) = \frac{1}{2} \Delta \mathbf{U}^T(k) \mathbf{H}(k) \Delta \mathbf{U}(k) + \Delta \mathbf{U}^T(k) \mathbf{G}(k) + c(k) \quad (6.20)$$

where,

$$\mathbf{H}(k) = 2(\mathbf{M}_U^T(k) \mathbf{M}_C^T(k) \mathbf{Q} \mathbf{M}_C(k) \mathbf{M}_U(k) + \mathbf{R})$$

$$\mathbf{G}(k) = -2\mathbf{M}_U^T(k) \mathbf{M}_C^T(k) \mathbf{Q} \mathbf{E}(k)$$

$$c(k) = \mathbf{E}^T(k) \mathbf{Q} \mathbf{E}(k)$$

$$\mathbf{E}(k) = \mathbf{M}_r(k) - \mathbf{M}_C(k) \mathbf{M}_A(k) \mathbf{x}(k) - \mathbf{M}_C(k) \mathbf{M}_B(k) \mathbf{u}(k-1) - \mathbf{M}_d(k)$$

6.2.3 Constraints

All the constraints should be transferred to the standard form of $\Lambda(k) \Delta \mathbf{U}(k) \leq \mathbf{b}(k)$.

Equation (6.18) can be expressed as,

$$\begin{aligned} \Delta \hat{\mathbf{u}}(k+i-1|k) &\leq \Delta \mathbf{u}_{\max} & \text{and} \\ -\Delta \hat{\mathbf{u}}(k+i-1|k) &\leq -\Delta \mathbf{u}_{\min} & i = 1, 2, \dots, N_C \end{aligned} \quad (6.21)$$

The vector-matrix form of these inequality equations can be written as,

$$\underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ -\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \cdots & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \cdots & -\mathbf{I} \end{bmatrix}}_{2n_u N_C \times n_u N_C} \underbrace{\begin{bmatrix} \Delta \hat{\mathbf{u}}(k|k) \\ \Delta \hat{\mathbf{u}}(k+1|k) \\ \vdots \\ \Delta \hat{\mathbf{u}}(k+N_C-1|k) \end{bmatrix}}_{n_u N_C \times 1} \leq \underbrace{\begin{bmatrix} \Delta \mathbf{u}_{\max} \\ -\Delta \mathbf{u}_{\min} \\ \Delta \mathbf{u}_{\max} \\ -\Delta \mathbf{u}_{\min} \\ \vdots \\ \vdots \\ \Delta \mathbf{u}_{\max} \\ -\Delta \mathbf{u}_{\min} \end{bmatrix}}_{2n_u N_C \times 1} \quad (6.22)$$

where \mathbf{I} is a proper size ($n_u \times n_u$) identity matrix. Similarly, equation (6.17) can be rearranged as,

$$\begin{aligned} \hat{\mathbf{u}}(k+i-1|k) &\leq \mathbf{u}_{\max} & \text{and} \\ -\hat{\mathbf{u}}(k+i-1|k) &\leq -\mathbf{u}_{\min} & i = 1, 2, \dots, N_C \end{aligned} \quad (6.23)$$

Also, it is clear that,

$$\hat{\mathbf{u}}(k+i-1|k) = \mathbf{u}(k-1) + \sum_{j=0}^{i-1} \Delta \hat{\mathbf{u}}(k+j|k) \quad (6.24)$$

Using (6.23) and (6.24) one can write the inputs inequality equations as,

$$\underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ -\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{I} & \mathbf{I} & \cdots & \mathbf{0} \\ -\mathbf{I} & -\mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & & \vdots \\ \mathbf{I} & \mathbf{I} & \cdots & \mathbf{I} \\ -\mathbf{I} & -\mathbf{I} & \cdots & -\mathbf{I} \end{bmatrix}}_{2n_u N_C \times n_u N_C} \underbrace{\begin{bmatrix} \Delta \hat{\mathbf{u}}(k|k) \\ \Delta \hat{\mathbf{u}}(k+1|k) \\ \vdots \\ \Delta \hat{\mathbf{u}}(k+N_C-1|k) \end{bmatrix}}_{n_u N_C \times 1} \leq \underbrace{\begin{bmatrix} \mathbf{u}_{\max} - \mathbf{u}(k-1) \\ -\mathbf{u}_{\min} + \mathbf{u}(k-1) \\ \mathbf{u}_{\max} - \mathbf{u}(k-1) \\ -\mathbf{u}_{\min} + \mathbf{u}(k-1) \\ \vdots \\ \vdots \\ \mathbf{u}_{\max} - \mathbf{u}(k-1) \\ -\mathbf{u}_{\min} + \mathbf{u}(k-1) \end{bmatrix}}_{2n_u N_C \times 1} \quad (6.25)$$

The output constraints can be expressed as follows,

$$\begin{aligned} \hat{\mathbf{y}}(k+i|k) &\leq \mathbf{y}_{\max} & \text{and} \\ -\hat{\mathbf{y}}(k+i|k) &\leq -\mathbf{y}_{\min} & i = 1, 2, \dots, N_p \end{aligned} \quad (6.26)$$

Using equation (6.14) these inequality constraints can be written in the standard form as,

$$\underbrace{\begin{bmatrix} \mathbf{M}_C(k)\mathbf{M}_U(k) \\ -\mathbf{M}_C(k)\mathbf{M}_U(k) \end{bmatrix}}_{2n_y N_p \times n_u N_C} \underbrace{\begin{bmatrix} \Delta \hat{\mathbf{u}}(k|k) \\ \Delta \hat{\mathbf{u}}(k+1|k) \\ \vdots \\ \Delta \hat{\mathbf{u}}(k+N_C-1|k) \end{bmatrix}}_{n_u N_C \times 1} \leq \underbrace{\begin{bmatrix} \mathbf{Y}_{\max} - \mathbf{M}_C(k)\mathbf{M}_A(k)\mathbf{x}(k) - \mathbf{M}_C(k)\mathbf{M}_B(k)\mathbf{u}(k-1) - \mathbf{M}_d(k) \\ -\mathbf{Y}_{\min} + \mathbf{M}_C(k)\mathbf{M}_A(k)\mathbf{x}(k) + \mathbf{M}_C(k)\mathbf{M}_B(k)\mathbf{u}(k-1) + \mathbf{M}_d(k) \end{bmatrix}}_{2n_y N_p \times 1} \quad (6.27)$$

where $\mathbf{Y}_{\max} = [\mathbf{I} \ \mathbf{I} \ \cdots \ \mathbf{I}]^T \mathbf{y}_{\max}$ and $\mathbf{Y}_{\min} = [\mathbf{I} \ \mathbf{I} \ \cdots \ \mathbf{I}]^T \mathbf{y}_{\min}$. Consequently, all these three sets of inequality constraints, (6.22), (6.25) and (6.27), can easily be used to construct the equation $\Lambda(k)\Delta\mathbf{U}(k) \leq \mathbf{b}(k)$.

6.3 Stability

The terminal equality constraints method is a way to provide stability [62]. The aim is to add a set of terminal state constraints to force the state variables to take particular equilibrium values at the end of the prediction horizon. The stability can be proved using the Lyapunov approach even in a general case [62].

As the proposed nonlinear MPC is based on a discrete model, a continuous model as

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}_c(\mathbf{x}(t), \mathbf{u}(t)) \quad (6.28)$$

can approximately be transferred into a discrete counterpart using,

$$\mathbf{x}(k+1) = \mathbf{x}(k) + T_s \mathbf{f}_c(\mathbf{x}(k), \mathbf{u}(k)) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) \quad (6.29)$$

where T_s is the sampling time, and \mathbf{f}_c and \mathbf{f} are the right hand sides of the continuous and the discrete nonlinear state-space equations respectively. Note that in an equilibrium point $(\mathbf{x}_{eq}, \mathbf{u}_{eq})$ the following expressions can be written in the case of a continuous model and a discrete model respectively,

$$\mathbf{f}_c(\mathbf{x}_{eq}, \mathbf{u}_{eq}) = \mathbf{0} \quad (6.30)$$

$$\mathbf{f}(\mathbf{x}_{eq}, \mathbf{u}_{eq}) = \mathbf{x}_{eq} \quad (6.31)$$

As pointed out earlier, the following terminal equality constraints are added to the inequality constrained objective function to guarantee the stability,

$$\hat{\mathbf{x}}(k + N_p | k) = \mathbf{x}_{eq}(k) \quad (6.32)$$

where \mathbf{x}_{eq} is an equilibrium point that satisfies the reference signals as well. The equality state constraints, (6.32), can be transferred to the equality control signals using (6.12) as,

$$\mathbf{A}_{eq}(k) \Delta \mathbf{U}(k) = \mathbf{B}_{eq}(k) \quad (6.33)$$

where,

$$\mathbf{A}_{eq}(k) = [\mathbf{M}_{N_p,1}(k) \quad \mathbf{M}_{N_p,2}(k) \quad \cdots \quad \mathbf{M}_{N_p,N_c}(k)] \quad (6.34)$$

$$\mathbf{B}_{eq}(k) = \mathbf{x}_{eq}(k) - \prod_{i=1}^{N_p} \mathbf{A}(k + N_p - i | k) \mathbf{x}(k) + \mathbf{M}_{N_p,1}(k) \mathbf{u}(k-1) \quad (6.35)$$

6.4 The TRMS

The proposed multistep Newton-type MPC based on state-dependent is implemented on the TRMS, shown in Figure 6.3. The control objective is to control the yaw and the pitch angles as accurately as possible.

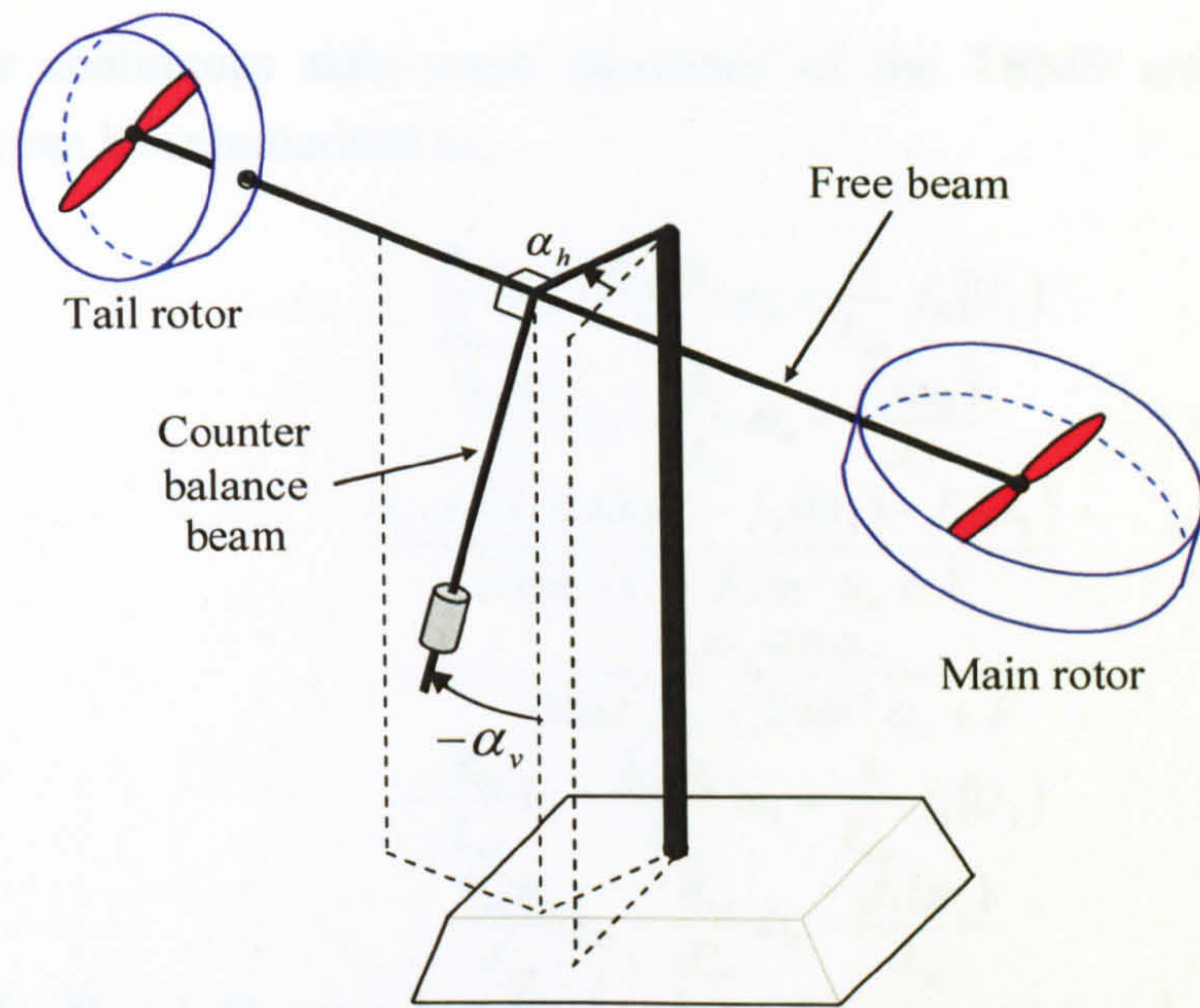


Figure 6.3 The TRMS

Let recall the state variables, inputs and outputs of the TRMS from Chapter 2 as follows:

$$\mathbf{x}(k) = [i_{ah}(k) \quad \omega_h(k) \quad S_h(k) \quad \alpha_h(k) \quad i_{av}(k) \quad \omega_v(k) \quad S_v(k) \quad \alpha_v(k)]^T \quad (6.36)$$

$$\mathbf{u}(k) = [U_h(k) \quad U_v(k)]^T \quad (6.37)$$

$$\mathbf{y}(k) = [\alpha_h(k) \quad \alpha_v(k)]^T \quad (6.38)$$

where,

i_{ah} : Armature current of the tail motor (A)

ω_h : Rotational velocity of the tail rotor (rad/s)

S_h : Angular velocity of the TRMS beam in the horizontal plane without the effect of the main rotor (rad/s)

α_h : Horizontal position (yaw angle) of the TRMS beam (rad)

i_{av} : Armature current of the main motor (A)

ω_v : Rotational velocity of the main rotor (rad/s)

S_v : Angular velocity of the TRMS beam in the vertical plane without the effect of the tail rotor (rad/s)

α_v : Vertical position (pitch angle) of the TRMS beam (rad)

U_h : Input voltage signal of the tail motor (V)

U_v : Input voltage signal of the main motor (V).

The nonlinear continuous state-space equations of the TRMS are developed in Chapter 2 and can be summarised as,

$$\frac{d}{dt} \begin{bmatrix} i_{ah} \\ \omega_h \\ S_h \\ \alpha_h \\ i_{av} \\ \omega_v \\ S_v \\ \alpha_v \end{bmatrix} = \begin{bmatrix} -\frac{R_{ah}}{L_{ah}} i_{ah} - \frac{k_{ah}\varphi_h}{L_{ah}} \omega_h + \frac{1}{L_{ah}} f_6(U_h) \\ \frac{k_{ah}\varphi_h}{J_{tr}} i_{ah} - \frac{B_{tr}}{J_{tr}} \omega_h - \frac{f_1(\omega_h)}{J_{tr}} \\ \frac{l_t f_2(\omega_h) \cos \alpha_v - f_7(\Omega_h) - f_3(\alpha_h)}{D \cos^2 \alpha_v + E \sin^2 \alpha_v + F} \\ S_h + \frac{k_m \omega_v \cos \alpha_v}{D \cos^2 \alpha_v + E \sin^2 \alpha_v + F} \\ -\frac{R_{av}}{L_{av}} i_{av} - \frac{k_{av}\varphi_v}{L_{av}} \omega_v + \frac{1}{L_{av}} f_8(U_v) \\ \frac{k_{av}\varphi_v}{J_{mr}} i_{av} - \frac{B_{mr}}{J_{mr}} \omega_v - \frac{f_4(\omega_v)}{J_{mr}} \\ \frac{f_5(\omega_v)(l_m + k_g \Omega_h \cos \alpha_v) - f_9(\Omega_v) + g[(A-B) \cos \alpha_v - C \sin \alpha_v] - 0.5 \Omega_h^2 H \sin 2\alpha_v}{J_v} \\ S_v + \frac{k_t}{J_v} \omega_h \end{bmatrix} \quad (6.39)$$

where R_{ah} , L_{ah} , $k_{ah}\varphi_h$, J_{tr} , B_{tr} , l_t , D , E , F , k_m , R_{av} , L_{av} , $k_{av}\varphi_v$, J_{mr} , B_{mr} , l_m , k_g , g , A , B , C , H , J_v , and k_t are positive constant values, and Ω_h and Ω_v are defined as,

$$\Omega_h = S_h + \frac{k_m \omega_v \cos \alpha_v}{D \cos^2 \alpha_v + E \sin^2 \alpha_v + F} \quad (6.40)$$

$$\Omega_v = S_v + \frac{k_t \omega_h}{J_v} \quad (6.41)$$

also f_1 , to f_9 are nonlinear functions as presented in Chapter 2. Since $L_{ah} \ll R_{ah}$ and $L_{av} \ll R_{av}$, without loss of accuracy, the order of the system can be reduced to six, $\mathbf{x}(k) = [\omega_h(k) \ S_h(k) \ \alpha_h(k) \ \omega_v(k) \ S_v(k) \ \alpha_v(k)]^T$, as,

$$\frac{d}{dt} \begin{bmatrix} \omega_h \\ S_h \\ \alpha_h \\ \omega_v \\ S_v \\ \alpha_v \end{bmatrix} = \begin{bmatrix} -\frac{(k_{ah}\varphi_h)^2}{J_{tr} R_{ah}} \omega_h - \frac{B_{tr}}{J_{tr}} \omega_h - \frac{f_1(\omega_h)}{J_{tr}} + \frac{k_{ah}\varphi_h}{J_{tr} R_{ah}} f_6(U_h) \\ \frac{l_t f_2(\omega_h) \cos \alpha_v - f_7(\Omega_h) - f_3(\alpha_h)}{D \cos^2 \alpha_v + E \sin^2 \alpha_v + F} \\ S_h + \frac{k_m \omega_v \cos \alpha_v}{D \cos^2 \alpha_v + E \sin^2 \alpha_v + F} \\ -\frac{(k_{av}\varphi_v)^2}{J_{mr} R_{av}} \omega_v - \frac{B_{mr}}{J_{mr}} \omega_v - \frac{f_4(\omega_v)}{J_{mr}} + \frac{k_{av}\varphi_v}{J_{mr} R_{av}} f_8(U_v) \\ \frac{f_5(\omega_v)(l_m + k_g \Omega_h \cos \alpha_v) - f_9(\Omega_v) + g[(A-B) \cos \alpha_v - C \sin \alpha_v] - 0.5 \Omega_h^2 H \sin 2\alpha_v}{J_v} \\ S_v + \frac{k_t}{J_v} \omega_h \end{bmatrix} \quad (6.42)$$

Although this model order reduction has no significant effect on the accuracy of the model, it has a considerable effect on the computational burden which reduces the load of the processor and increases the speed of the optimisation problem. The above nonlinear state-space equation can approximately be expressed as the following state-dependent state-space equation,

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x})\mathbf{x} + \mathbf{B}\mathbf{u} \quad (6.43)$$

where,

$$\mathbf{A}(\mathbf{x}) = \begin{bmatrix} \frac{-(k_{ah}\varphi_h)^2}{J_{tr}R_{ah}} - \frac{B_{tr}}{J_{tr}} - \frac{k_{thp/n}}{J_{tr}}|\omega_h| & 0 & 0 & 0 & 0 & 0 \\ \frac{l_t k_{fhp/n}|\omega_h| \cos\alpha_v}{f_{10}(\alpha_v)} & \frac{-k_{vfh}}{f_{10}(\alpha_v)} & \frac{-k_{chp/n}}{f_{10}(\alpha_v)} & \frac{-k_{vfh}k_m \cos\alpha_v}{(f_{10}(\alpha_v))^2} & 0 & 0 \\ 0 & 1 & 0 & \frac{k_m \cos\alpha_v}{f_{10}(\alpha_v)} & 0 & 0 \\ 0 & 0 & 0 & \frac{-(k_{av}\varphi_v)^2}{J_{mr}R_{av}} - \frac{B_{mr}}{J_{mr}} - \frac{k_{vfp/n}}{J_{mr}}|\omega_v| & 0 & 0 \\ \frac{-k_{vfh}k_t}{J_v^2} & 0 & 0 & \frac{k_{fvp/n}|\omega_v|(l_m + k_g\Omega_h \cos\alpha_v)}{J_v} & \frac{-k_{vfv}}{J_v} & f_{11}(\alpha_v) \\ \frac{k_t}{J_v} & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} \frac{k_1 k_{ah} \varphi_h}{J_{tr} R_{ah}} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{k_2 k_{av} \varphi_v}{J_{mr} R_{av}} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$f_{10}(\alpha_v) = D \cos^2 \alpha_v + E \sin^2 \alpha_v + F$$

$$f_{11}(\alpha_v) = \begin{cases} \frac{g[(A-B)\cos\alpha_v - C\sin\alpha_v]}{\alpha_v} & \text{if } \alpha_v \neq 0 \\ 0 & \text{if } \alpha_v = 0 \end{cases}$$

In the case of the TRMS, the steady state values of the state variables, $\mathbf{x}_{eq}(k)$, are defined as,

$$\mathbf{x}_{eq}(k) = [\omega_h^{eq} \quad S_h^{eq} \quad \alpha_h^{eq} \quad \omega_v^{eq} \quad S_v^{eq} \quad \alpha_v^{eq}]^T \quad (6.44)$$

where each element is found according to the current reference signals, $\alpha_{href}(k)$ and $\alpha_{vref}(k)$, and equations (6.30) and (6.42),

$$\alpha_h^{eq} = \alpha_{href}(k) \quad (6.45)$$

$$\alpha_v^{eq} = \alpha_{vref}(k) \quad (6.46)$$

$$\omega_h^{eq} = \begin{cases} \sqrt{\frac{f_3(\alpha_h^{eq})}{k_{fhp} l_t \cos \alpha_v^{eq}}} & \text{for } \alpha_h^{eq} \geq 0 \\ -\sqrt{\frac{f_3(\alpha_h^{eq})}{k_{fhn} l_t \cos \alpha_v^{eq}}} & \text{for } \alpha_h^{eq} < 0 \end{cases} \quad (6.47)$$

$$S_h^{eq} = -\frac{k_m \omega_v^{eq} \cos \alpha_v^{eq}}{D \cos^2 \alpha_v^{eq} + E \sin^2 \alpha_v^{eq} + F} \quad (6.48)$$

$$\omega_v^{eq} = \begin{cases} \sqrt{\frac{F_v^{eq}}{k_{fvp}}} & \text{for } F_v^{eq} \geq 0 \\ -\sqrt{\frac{F_v^{eq}}{k_{fvm}}} & \text{for } F_v^{eq} < 0 \end{cases} \quad (6.49)$$

where,

$$F_v^{eq} = -\frac{g[(A-B) \cos \alpha_v^{eq} - C \sin \alpha_v^{eq}]}{l_m}$$

$$S_v^{eq} = -\frac{k_t \omega_h^{eq}}{J_v} \quad (6.50)$$

6.5 Results

Based on the nonlinear model, N_p linear models are developed to model the nonlinear system during the prediction horizon at each instant, k . The objective function and constraints are formed based on these linear models to have a constrained linear quadratic optimisation problem. A set of terminal equality state constraints is formed to force the state to an equilibrium point at the end of the prediction horizon in order to guarantee the stability of the closed-loop system. At each iteration, the first set of the optimum input vector, $\Delta \hat{u}(k|k)$, is added to the control signal at previous time, $u(k-1)$, and the result is sent to the plant and also the linear model at $k|k$ instant. The others optimum values are kept for the next sample time, $k+1$, calculation.

The block diagram of the Newton-type MPC is shown in Figure 6.4. Note that the nonlinear model and linearisation operator have not been shown in Figure 6.4. Figure 6.5 shows the flowchart of the proposed method. Although it is assumed that all the state variables are measurable in this chapter, a state observer can be used in the case of output-feedback MPC methods. The source codes of the corresponding programme can be found in Appendix 5.

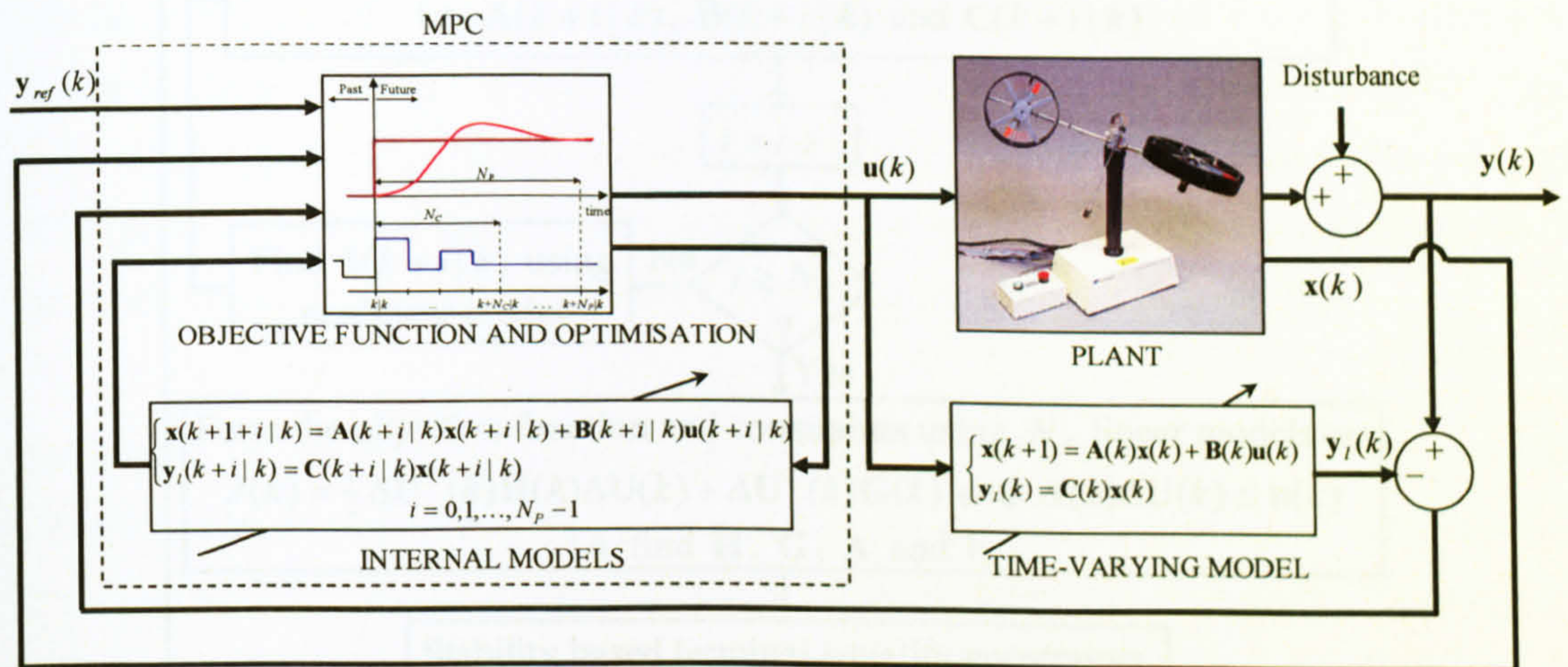


Figure 6.4 Block diagram of the proposed Newton-type MPC approach

As mentioned before, the plant has 2 inputs, 2 outputs and 6 states. The inputs are the DC voltages of the tail and main rotors, U_h and U_v , and outputs are the yaw, α_h , and pitch, α_v , angles of the beam. The state variables are the rotational velocity of the tail rotor, ω_h , the rotational velocity of the main rotor, ω_v , the angular velocity of the TRMS beam in the horizontal plane without the effect of the main rotor, S_h , the angular velocity of the TRMS beam in the vertical plane without the effect of the tail rotor, S_v , the horizontal position of the TRMS beam, α_h , and the vertical position of the TRMS beam, α_v . The parameters of the applied multistep Newton-type MPC are listed in Table 6.1. Note that the sampling time of the model predictive controller is set to be 0.2 seconds and the optimisation approach is chosen to be an active set method available in MATLAB Toolbox for linear quadratic problems. The computational time for a period of 100 seconds simulation with parameters listed in Table 6.1 is 5.6 seconds, this means the proposed method can be implemented in real time.

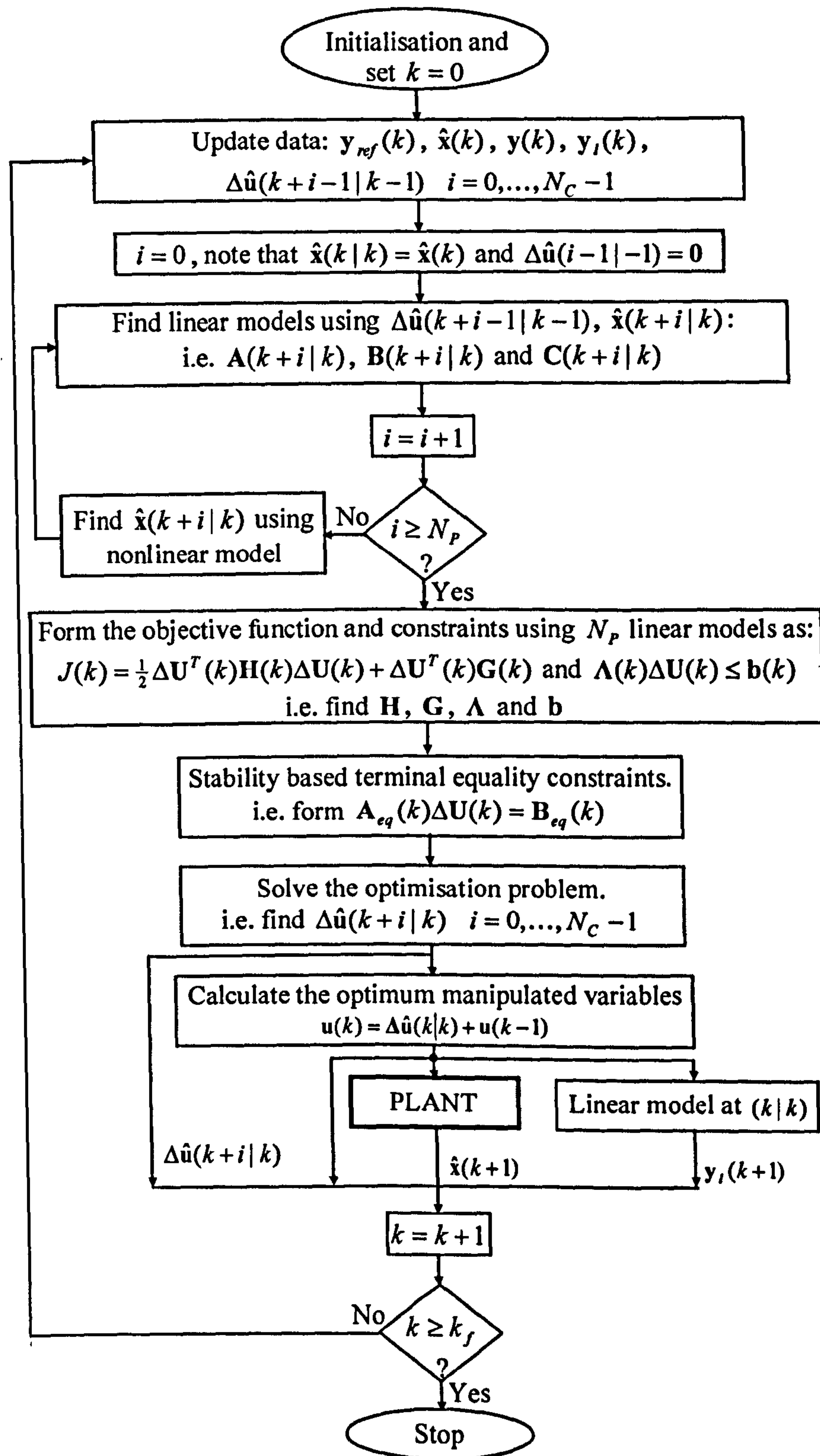


Figure 6.5 Flowchart of the proposed Newton-type MPC approach

The nonlinear MPC developed for the TRMS has been assessed with a variety of reference signals and the results obtained demonstrate that the controller has a high performance and reliability in various operating regions. The controller has been proven to be reliable under disturbances and various reference signals used. Figures

6.6 and 6.7 show the results of the horizontal (yaw) and the vertical (pitch) angles of the beam with square references. It is clear from Figures 6.6 and 6.7 that the two channels have significant effect on each other. For instance, at the 25th second, the yaw angle reference signal has been changed from 0.6 to -0.6 rad and subsequently, the yaw angle has followed the reference signal. The pitch angle, due to the mentioned coupling, has been affected by the yaw angle; however, the controller has regained the control and forced the pitch angle to follow the reference trajectory. The overshoot at the 50th second of the yaw angle response has been caused by the step change of the pitch angle reference signal at that instant. This reference signal is one of the challenging trajectories in the case of the TRMS and it has been selected deliberately to validate the performance of the proposed control method. The control signals of case 1 for the two channels are depicted in Figures 6.8 and 6.9.

The results of the horizontal and the vertical angles of the beam with sine wave references are shown in Figures 6.10 and 6.11 that show the ability of the proposed control system in following a continuously variable reference trajectory. Figures 6.12 and 6.13 shows the manipulated signals of the tail and main motors related to the second case respectively. Figures 6.14 to 6.25 illustrate the responses of the controller according to some other reference signals and with initial positions of the beam. In order to substantiate the efficacy of the proposed method, the control system is tested with sawtooth reference signals as shown in Figures 6.26 and 6.27. The input signals to the TRMS related to the sawtooth references can be seen in Figures 6.28 and 6.29.

Figure 6.30 shows two successive results of the internal model output of the horizontal channel. This figure illustrates that at the end of each prediction horizon the controlled variables exactly take the reference values and also this operating point is an equilibrium point due to the terminal equality constraints. Therefore, as discussed earlier, the closed-loop system is stable as long as the optimisation problem is feasible. As usual in stability proofs, assume that the model is perfect and there is no disturbance. As the final points at each iteration is an equilibrium point, the finite horizon control can be considered to be an infinite horizon control and the stability of the system can be verified. Note that in Figure 6.30 the two successive results have been shown without any overlap to be clearer; however, for example, instants $k+2|k$ and $k+2|k+1$ can be drawn at the same time. The optimum values of the objective function at each iteration has been depicted in Figure 6.31. This figure shows the values around the 75th second in the case related to Figures 6.6 and 6.7 where the yaw angle reference has been changed.

Table 6.1 The parameters of the proposed MPC

Parameter	Value	Parameter	Value
N_P	20	ΔU_h^{\max}	2.0 V
N_C	15	ΔU_v^{\min}	-2.0 V
U_h^{\min}	-2.5 V	ΔU_v^{\max}	2.0 V
U_h^{\max}	2.5 V	α_h^{\min}	-4.2 rad
U_v^{\min}	-2.5 V	α_h^{\max}	1.5 rad
U_v^{\max}	2.5 V	α_v^{\min}	-0.45 rad
ΔU_h^{\min}	-2.0 V	α_v^{\max}	1.65 rad
δ	$\begin{bmatrix} 0.1 & 0 \\ 0 & 1 \end{bmatrix}$	λ	$\begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix}$

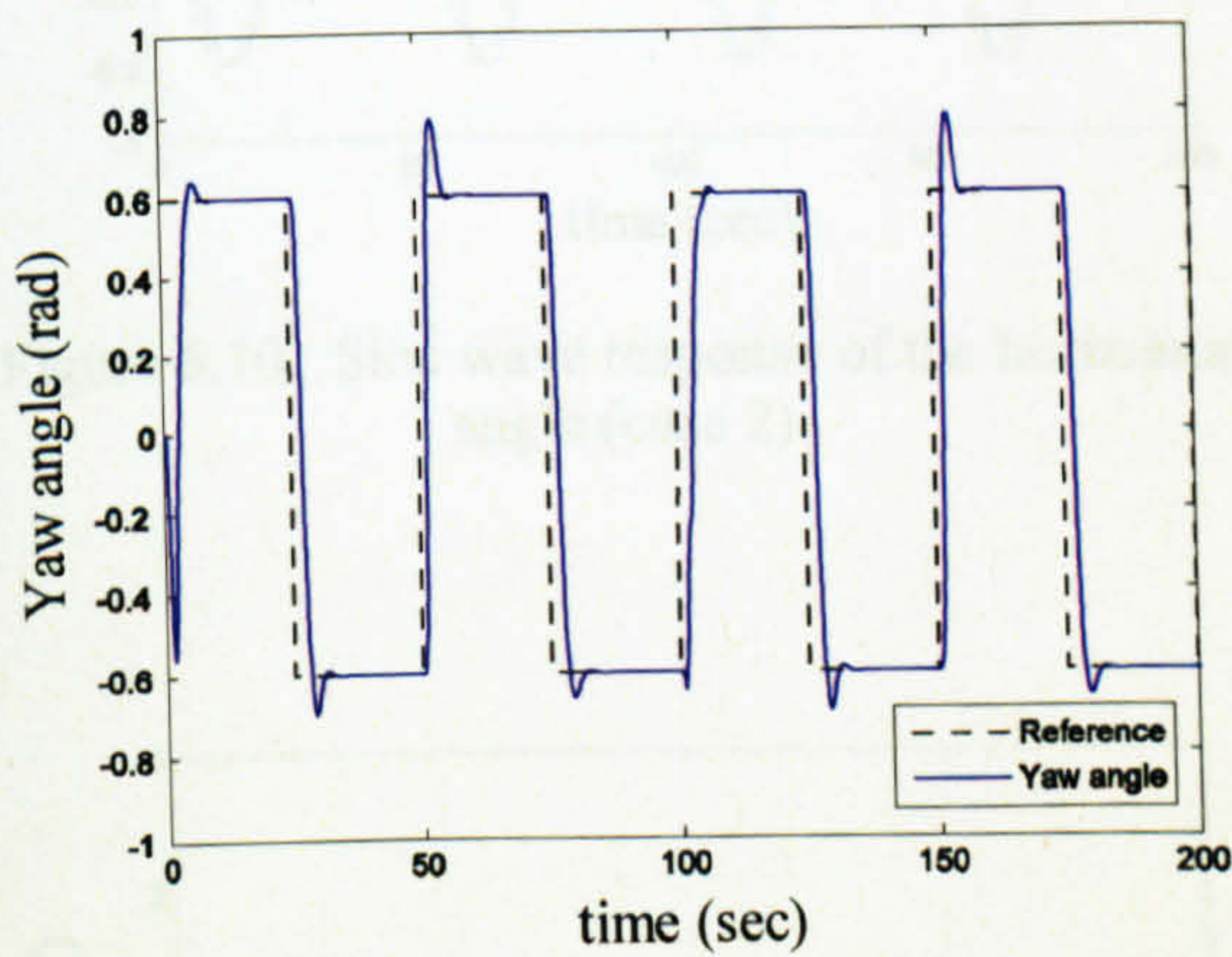


Figure 6.6 Square wave response of the horizontal angle (case 1)

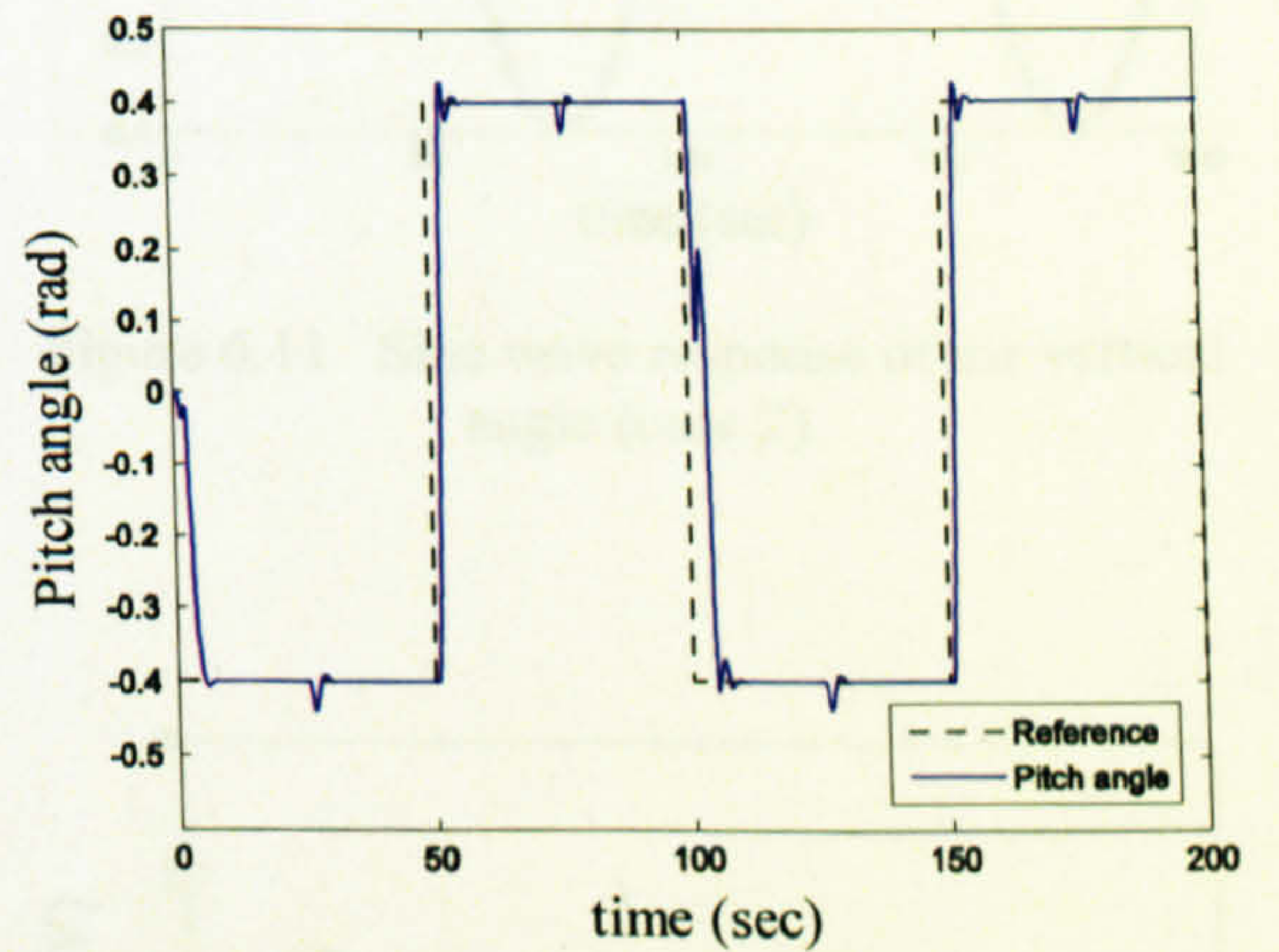


Figure 6.7 Square wave response of the vertical angle (case 1)

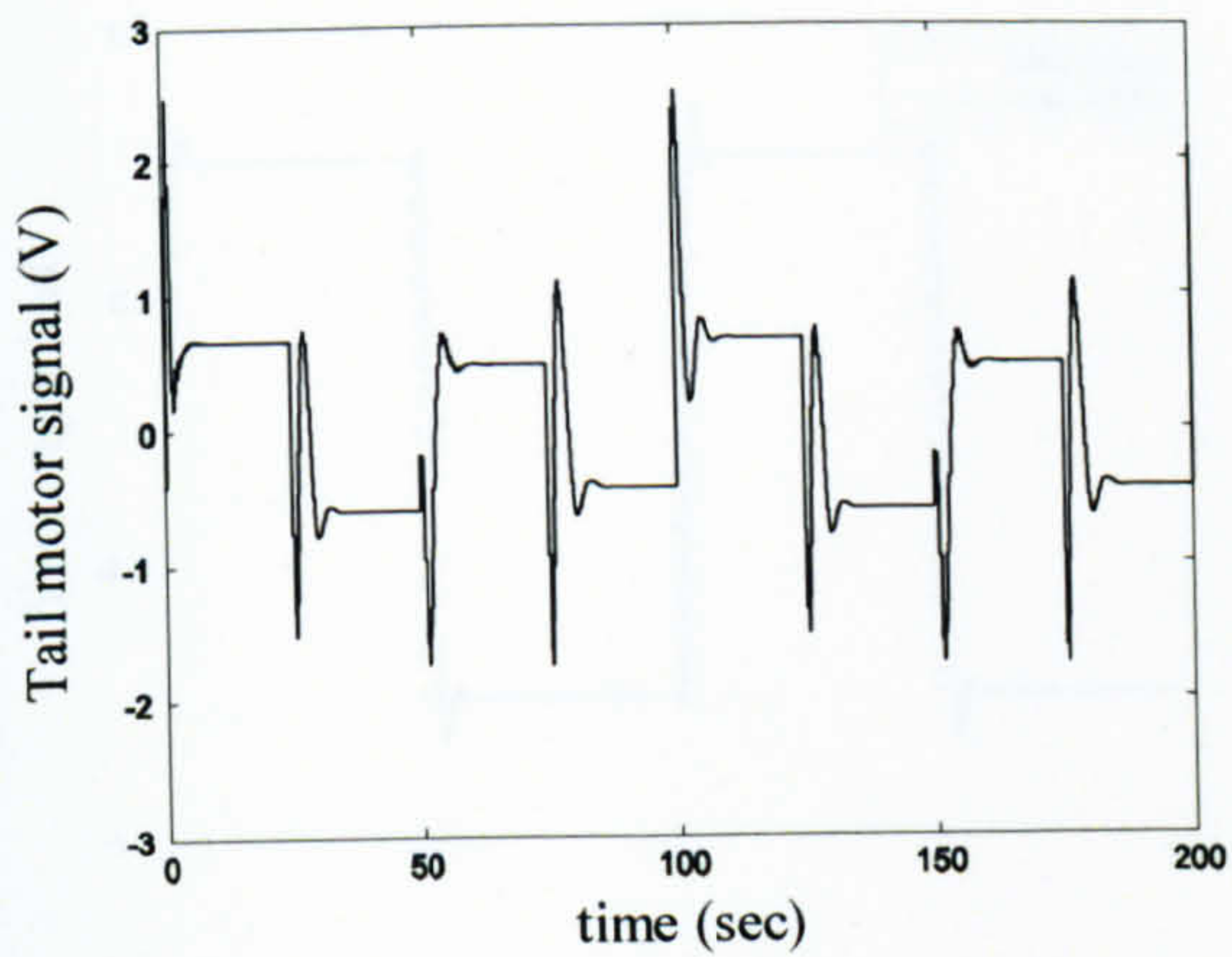


Figure 6.8 Tail motor manipulated signal, U_h , (case 1)

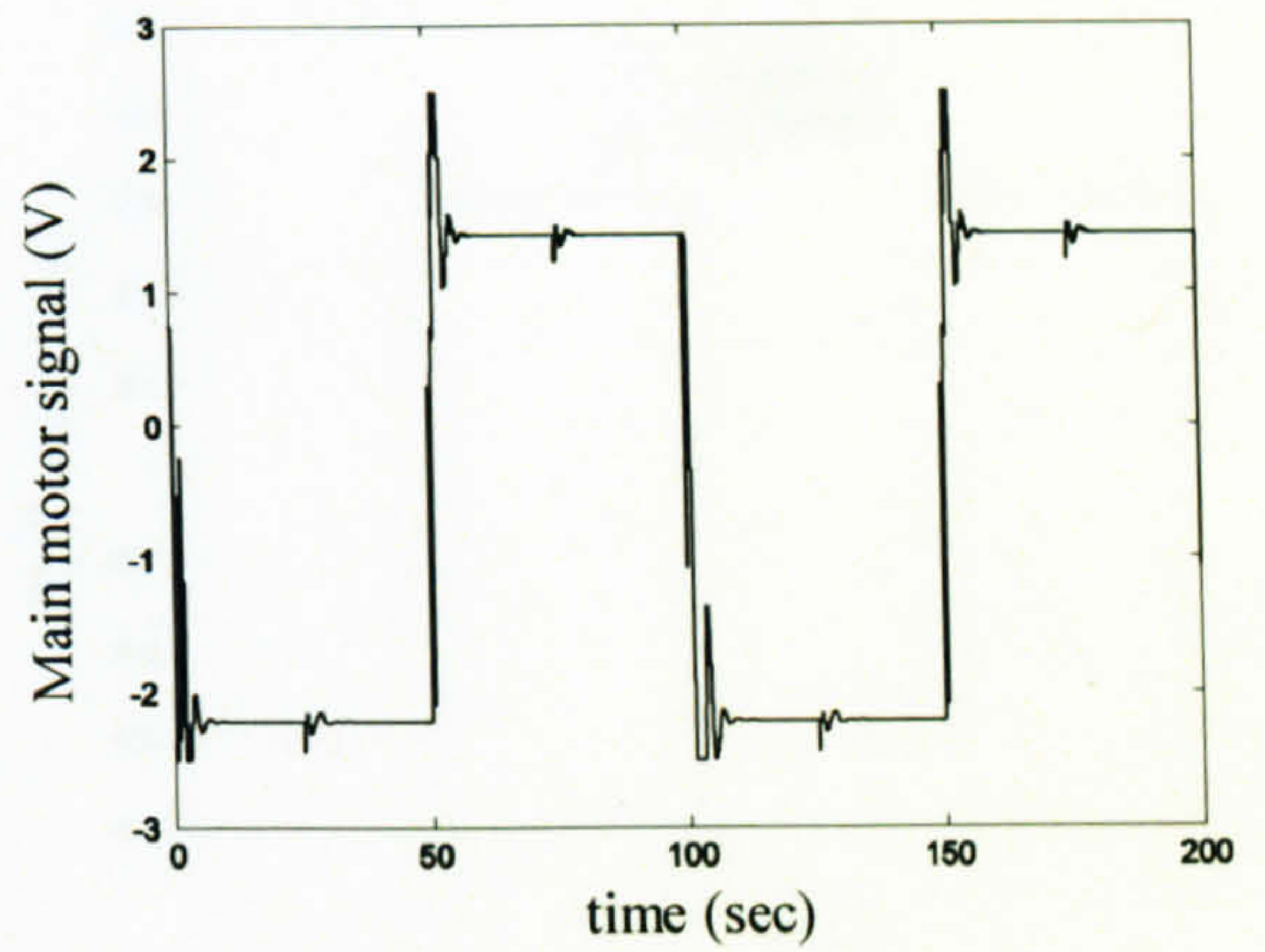


Figure 6.9 Main motor manipulated signal, U_v , (case 1)

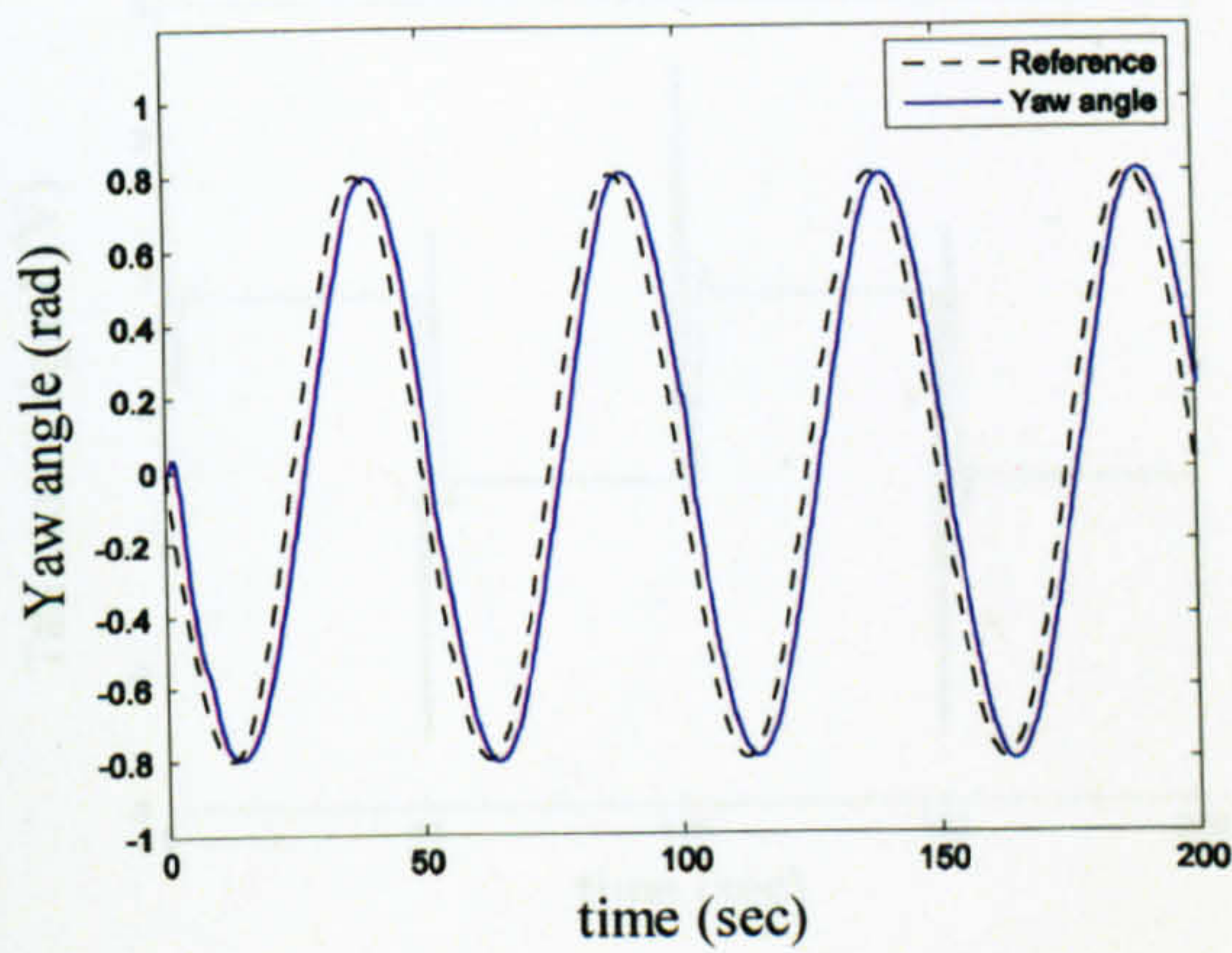


Figure 6.10 Sine wave response of the horizontal angle (case 2)

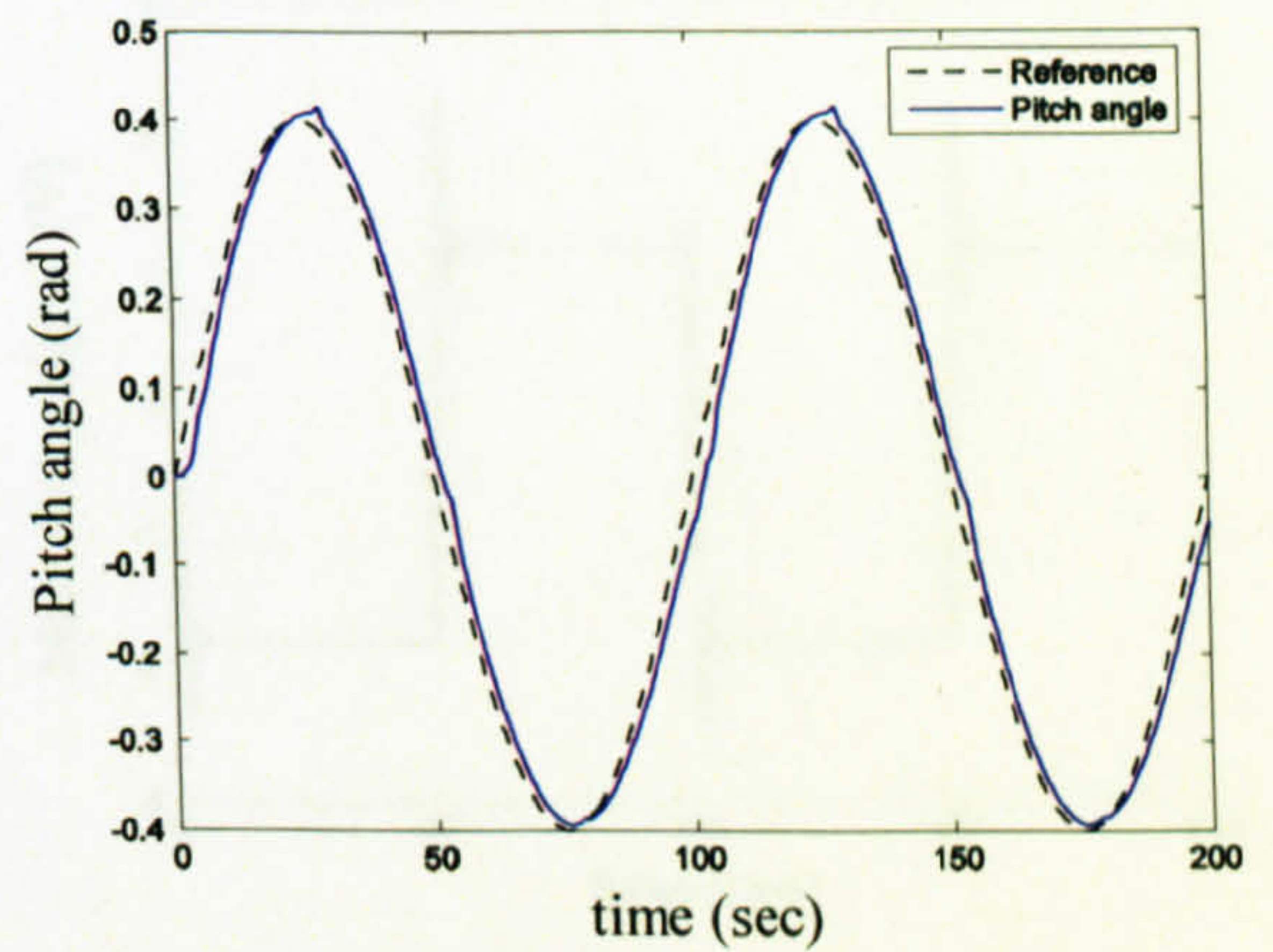


Figure 6.11 Sine wave response of the vertical angle (case 2)

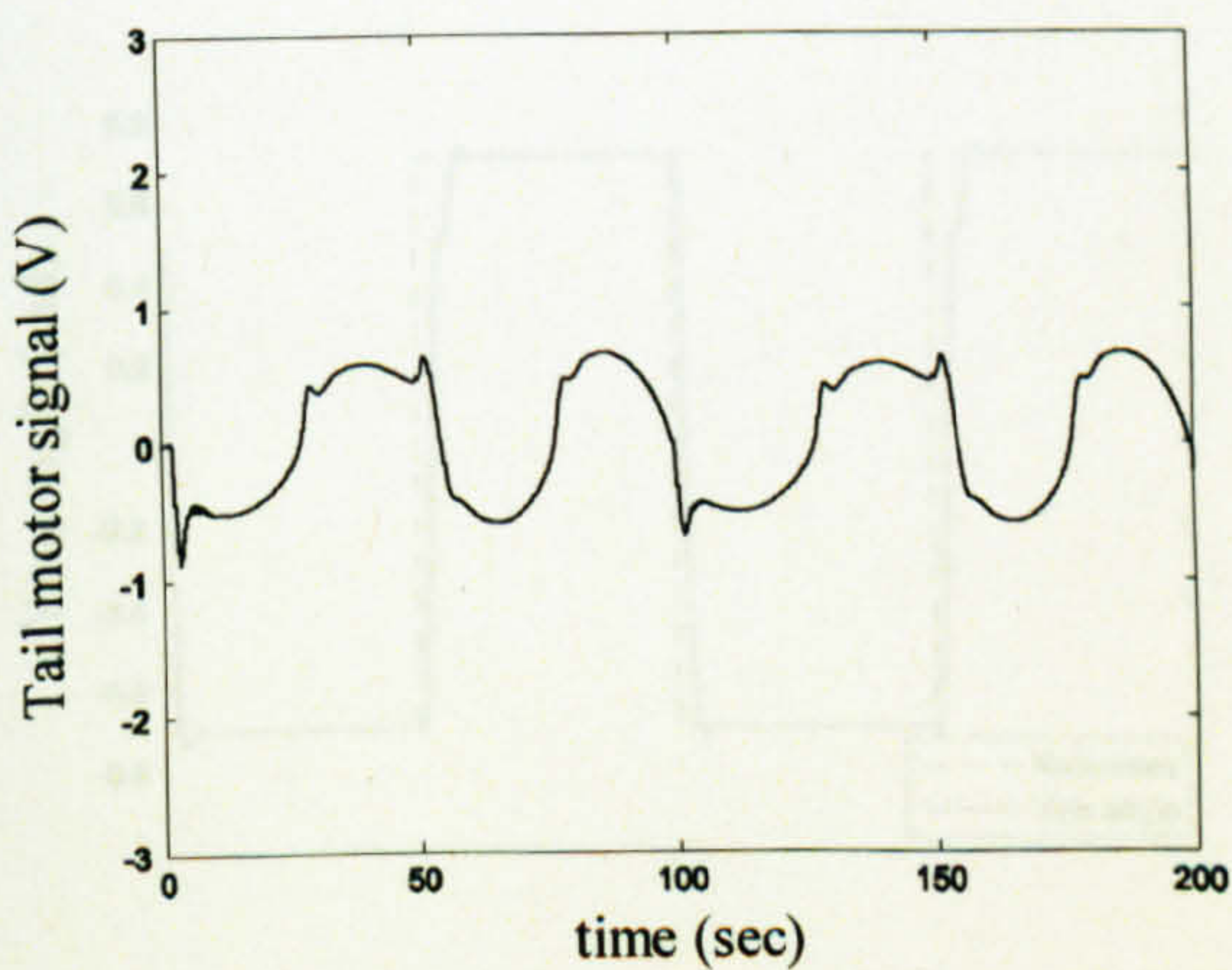


Figure 6.12 Tail motor manipulated signal, U_h , (case 2)

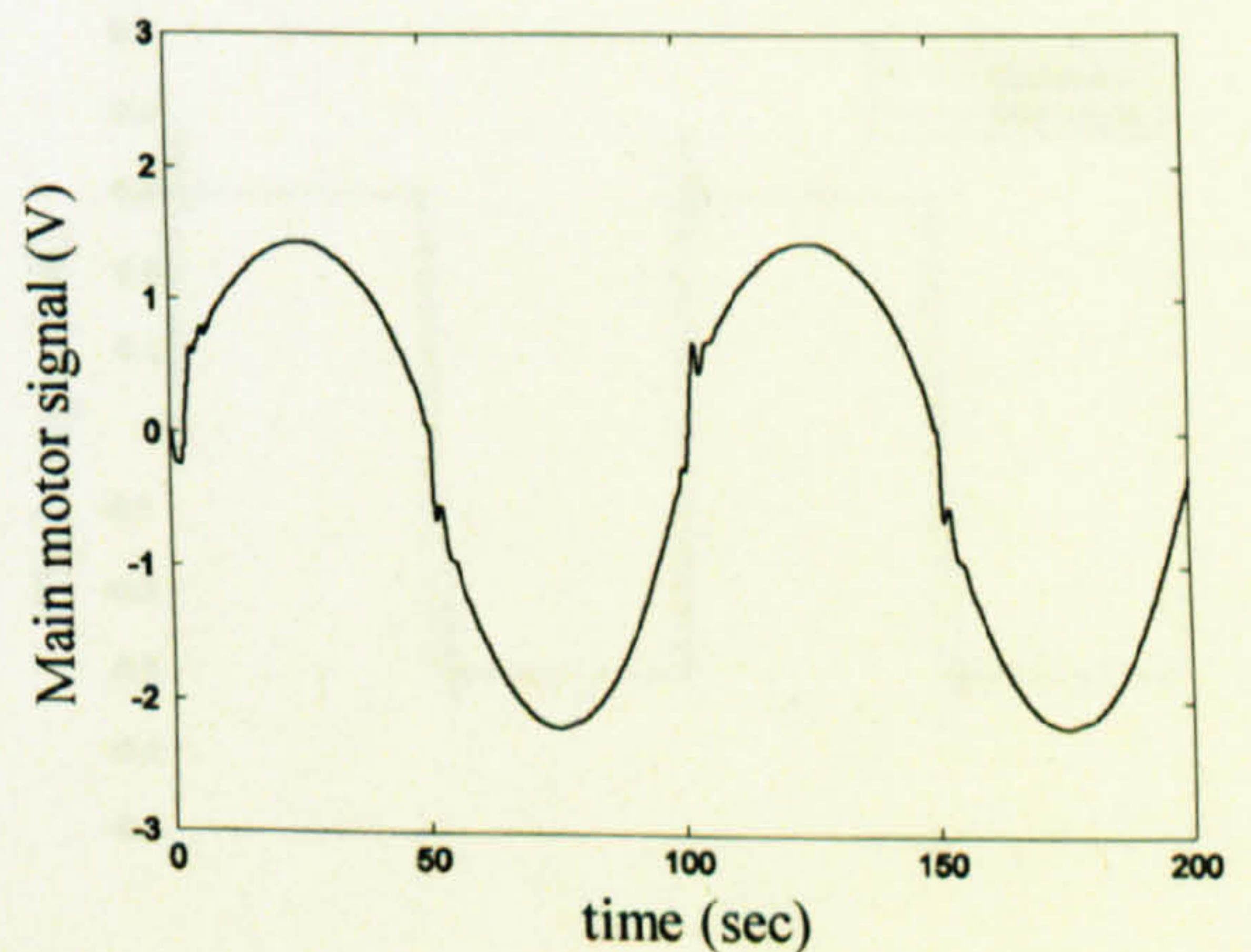


Figure 6.13 Main motor manipulated signal, U_v , (case 2)

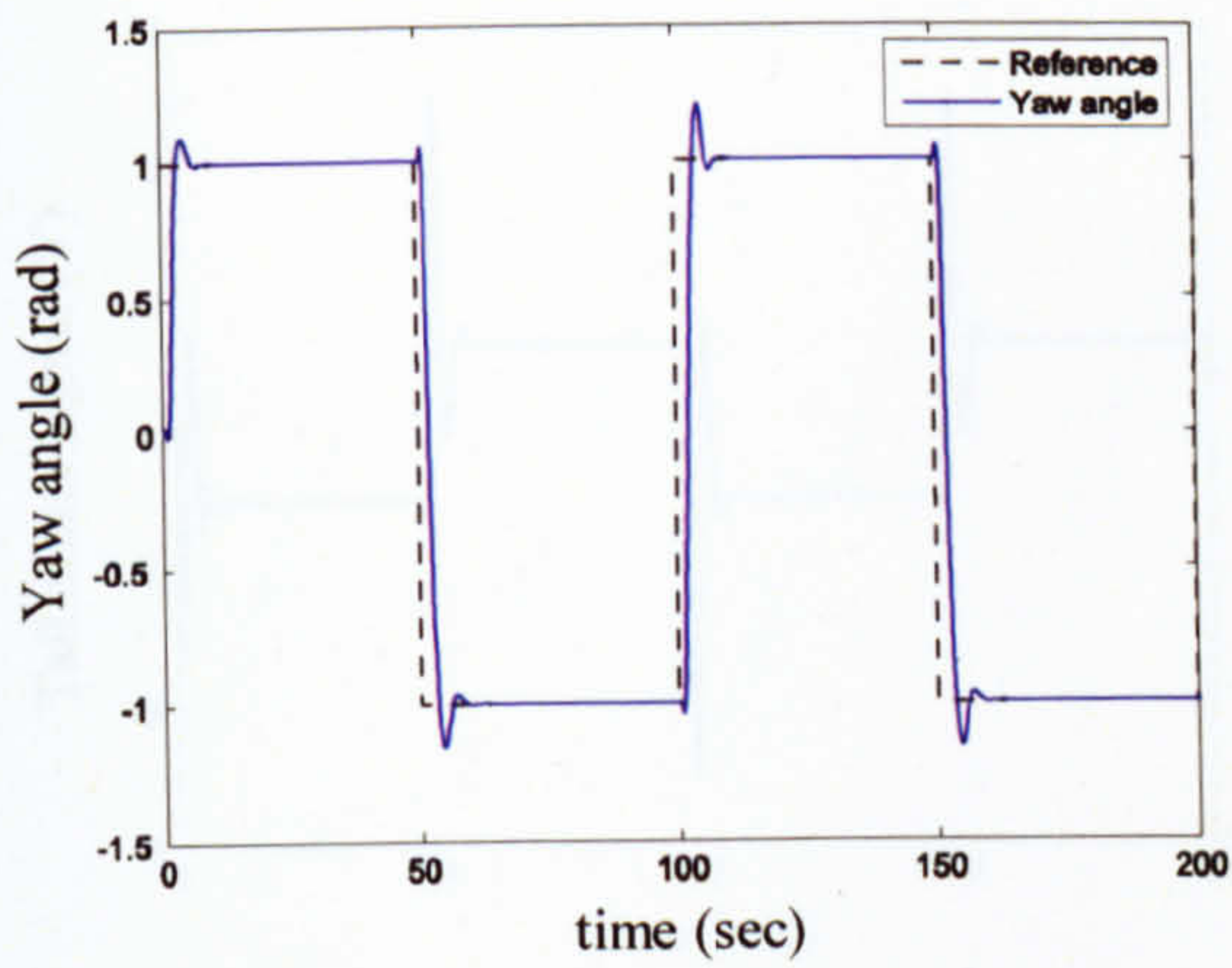


Figure 6.14 Square wave response of the horizontal angle (case 3)

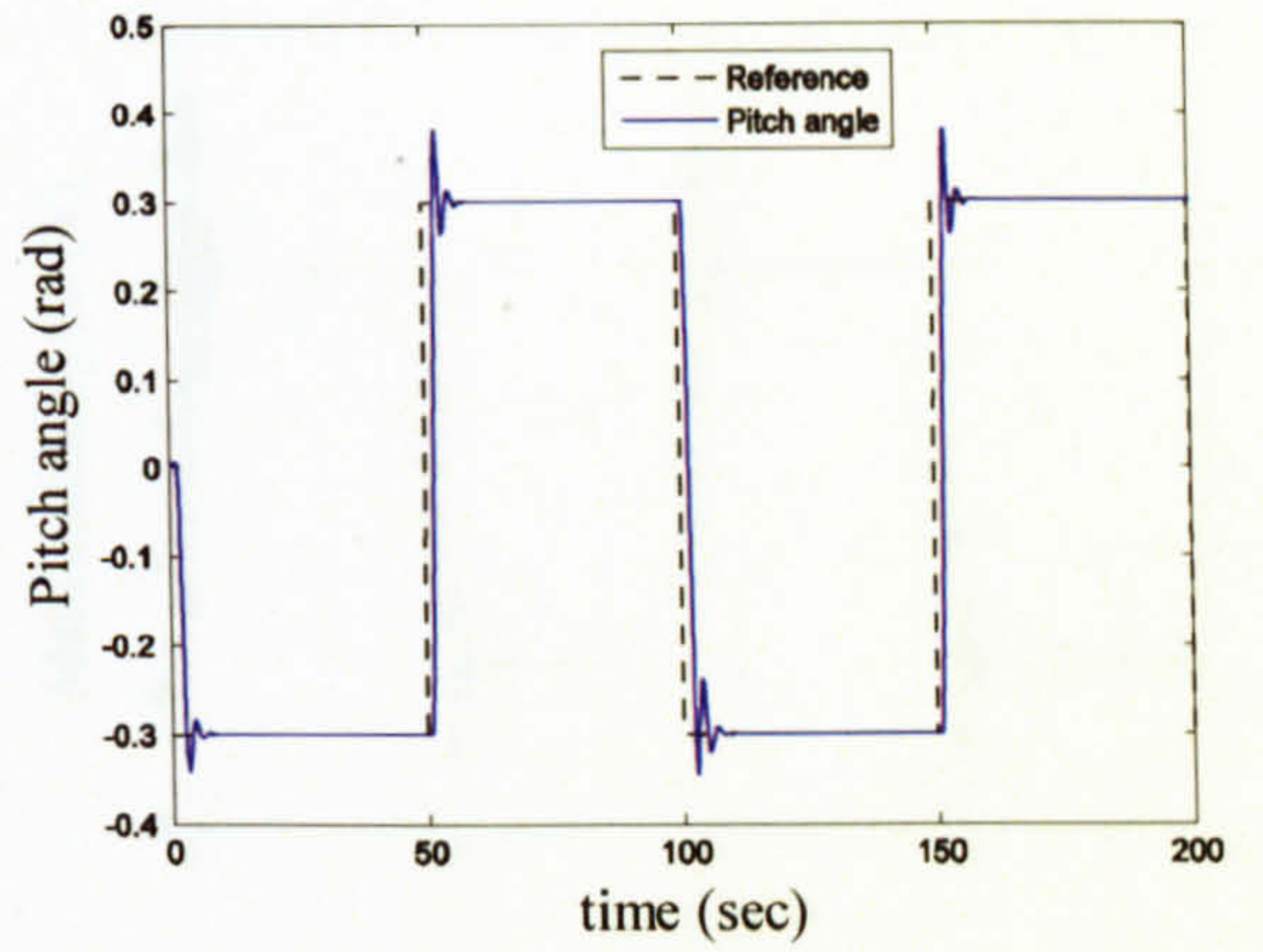


Figure 6.15 Square wave response of the vertical angle (case 3)

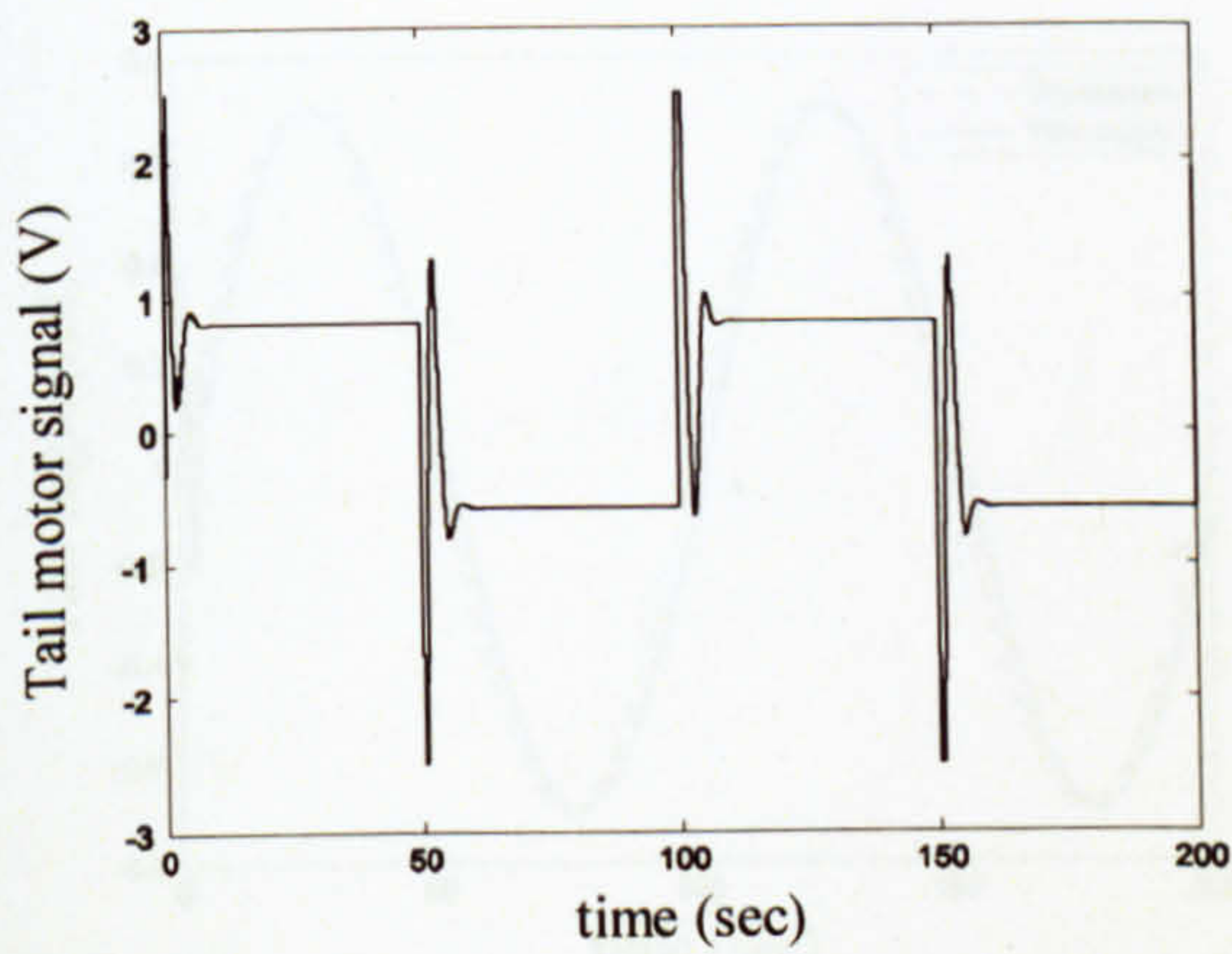


Figure 6.16 Tail motor manipulated signal, U_h , (case 3)

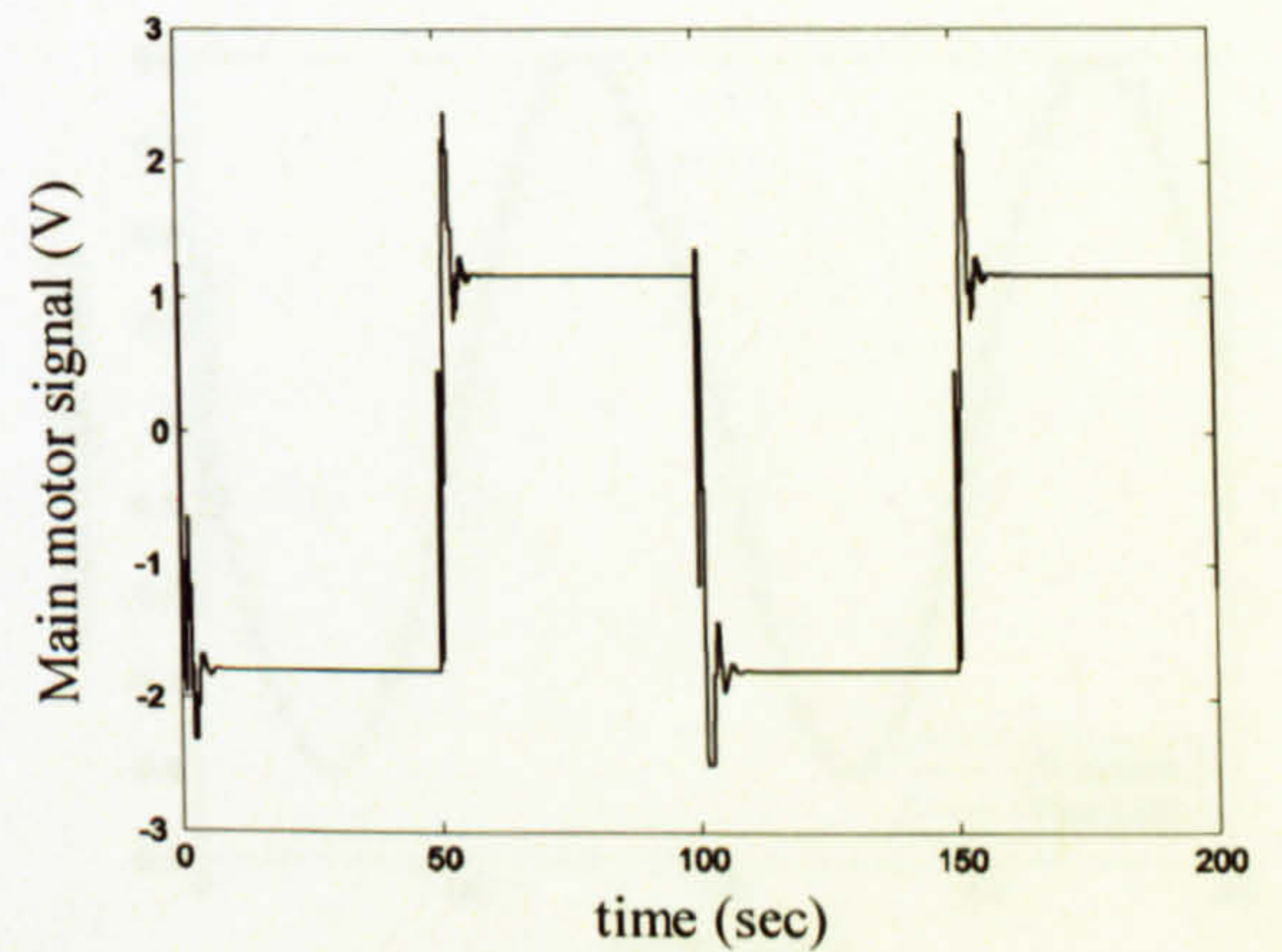


Figure 6.17 Main motor manipulated signal, U_v , (case 3)

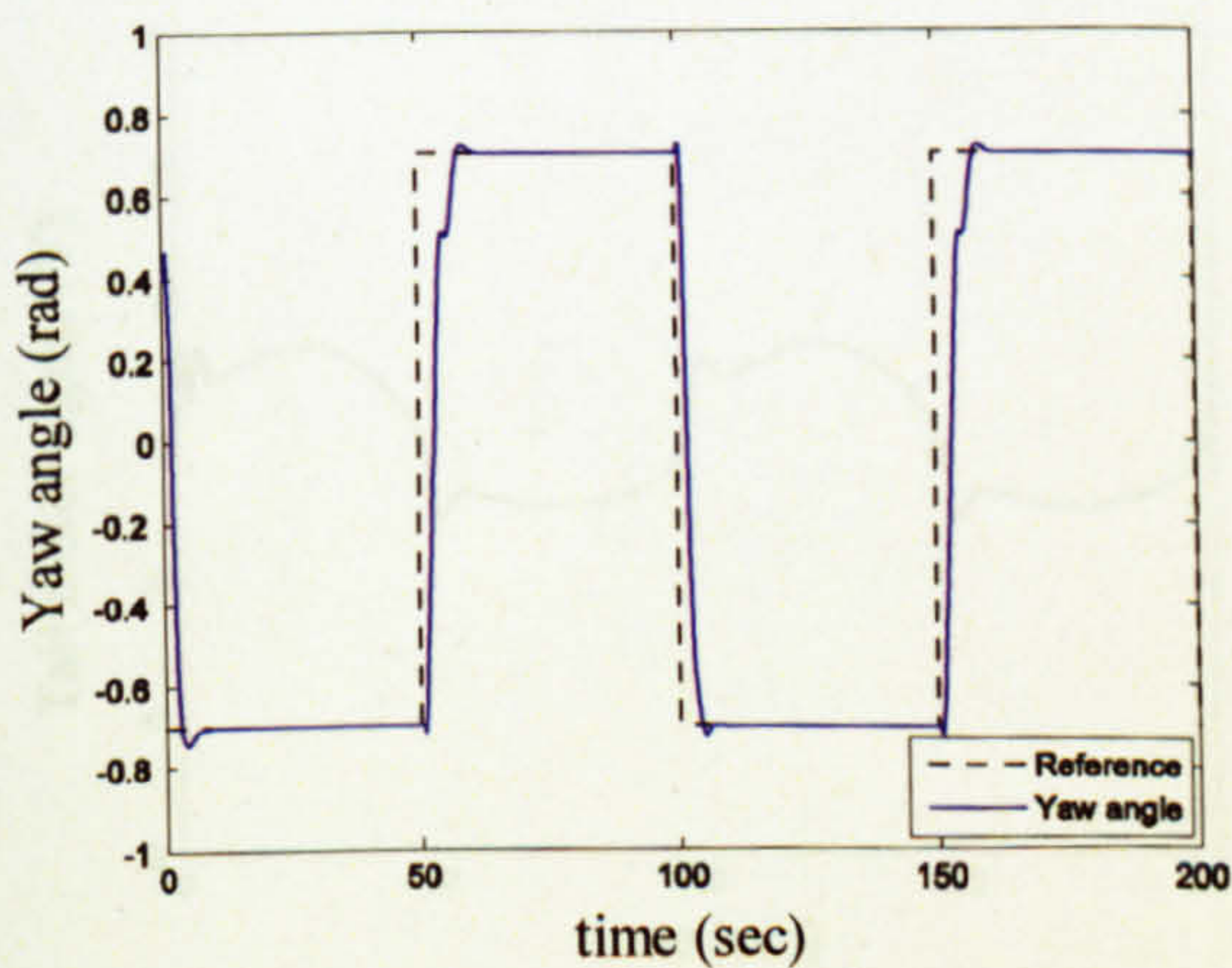


Figure 6.18 Square wave response of the horizontal angle (case 4)

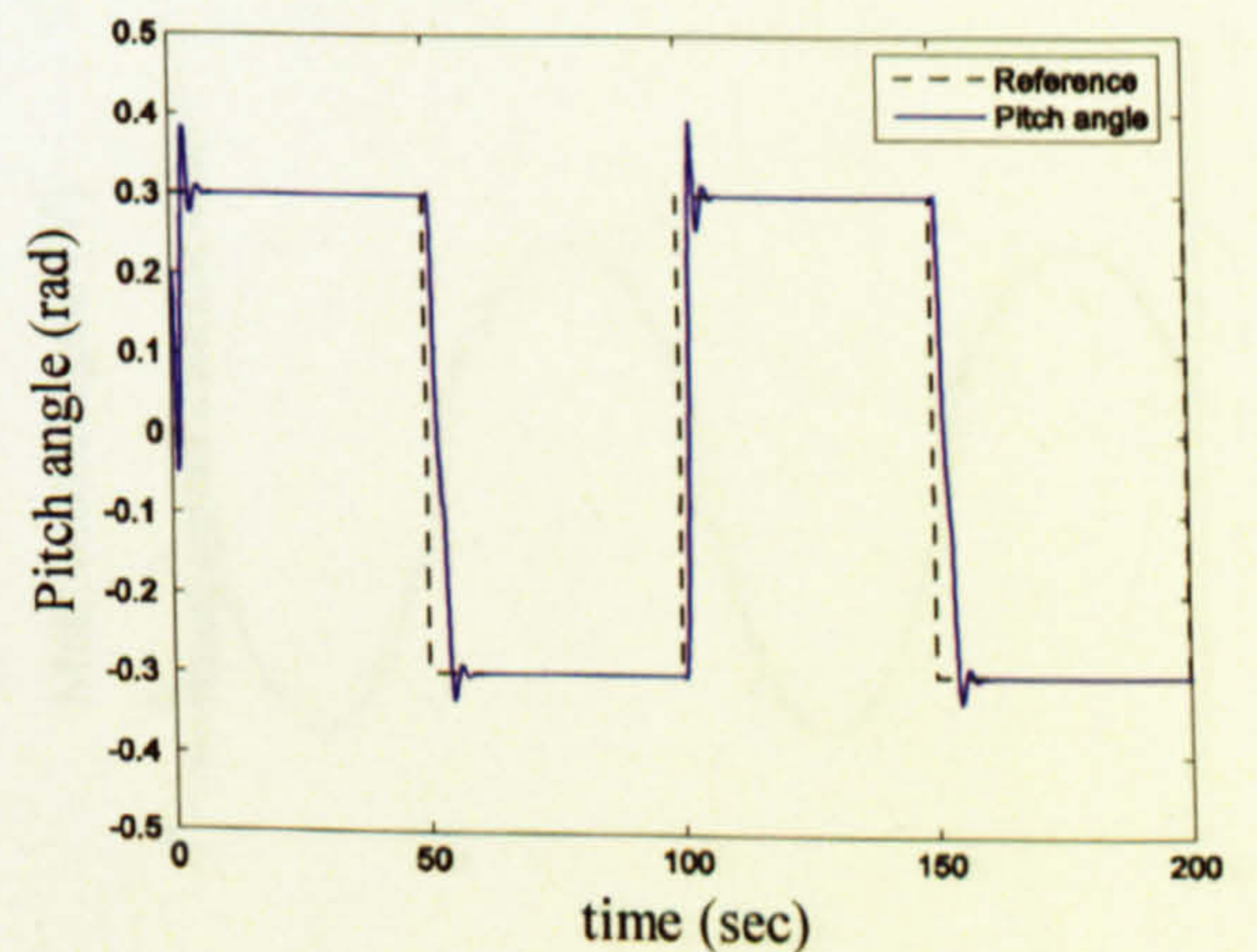


Figure 6.19 Square wave response of the vertical angle (case 4)

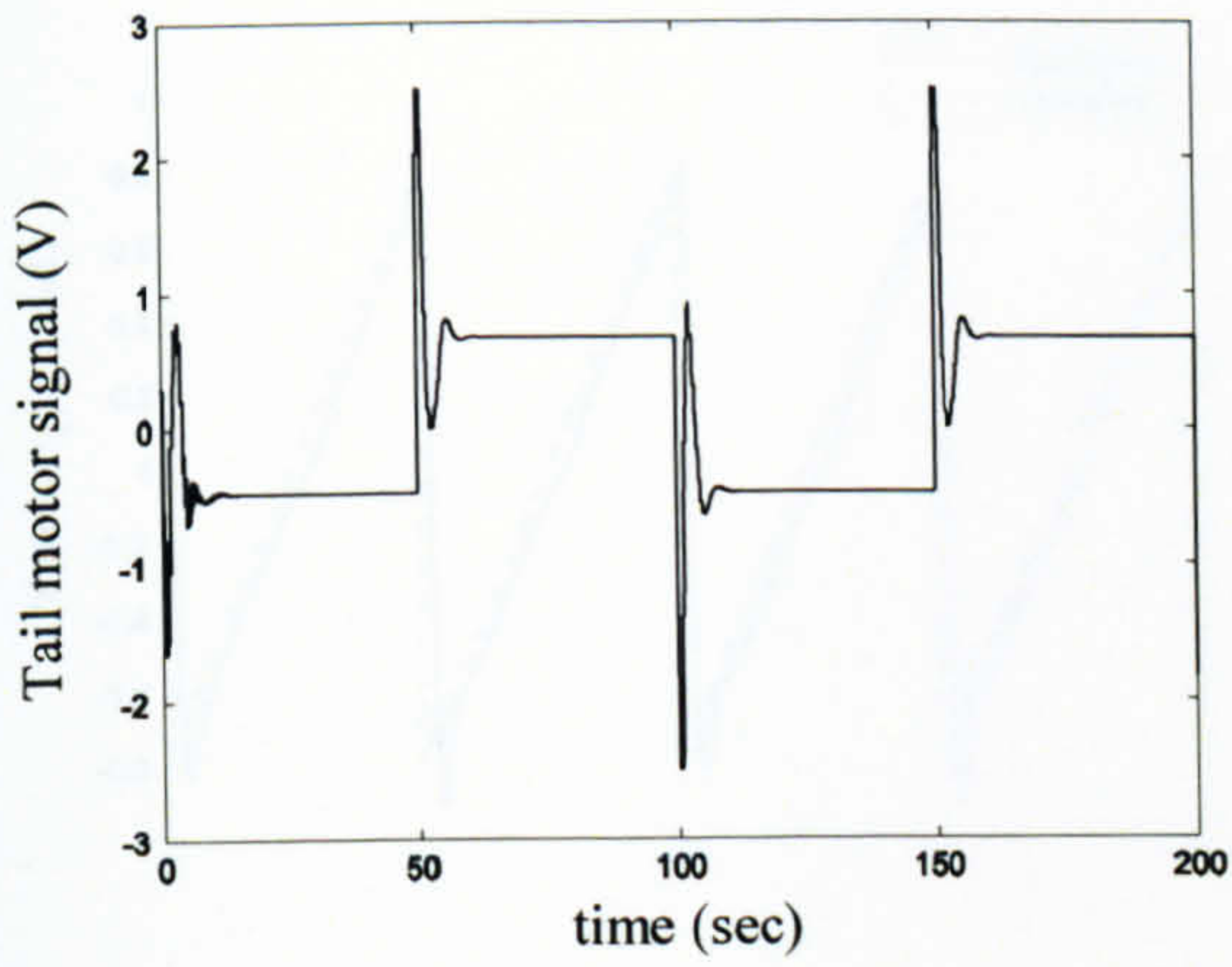


Figure 6.20 Tail motor manipulated signal, U_h , (case 4)

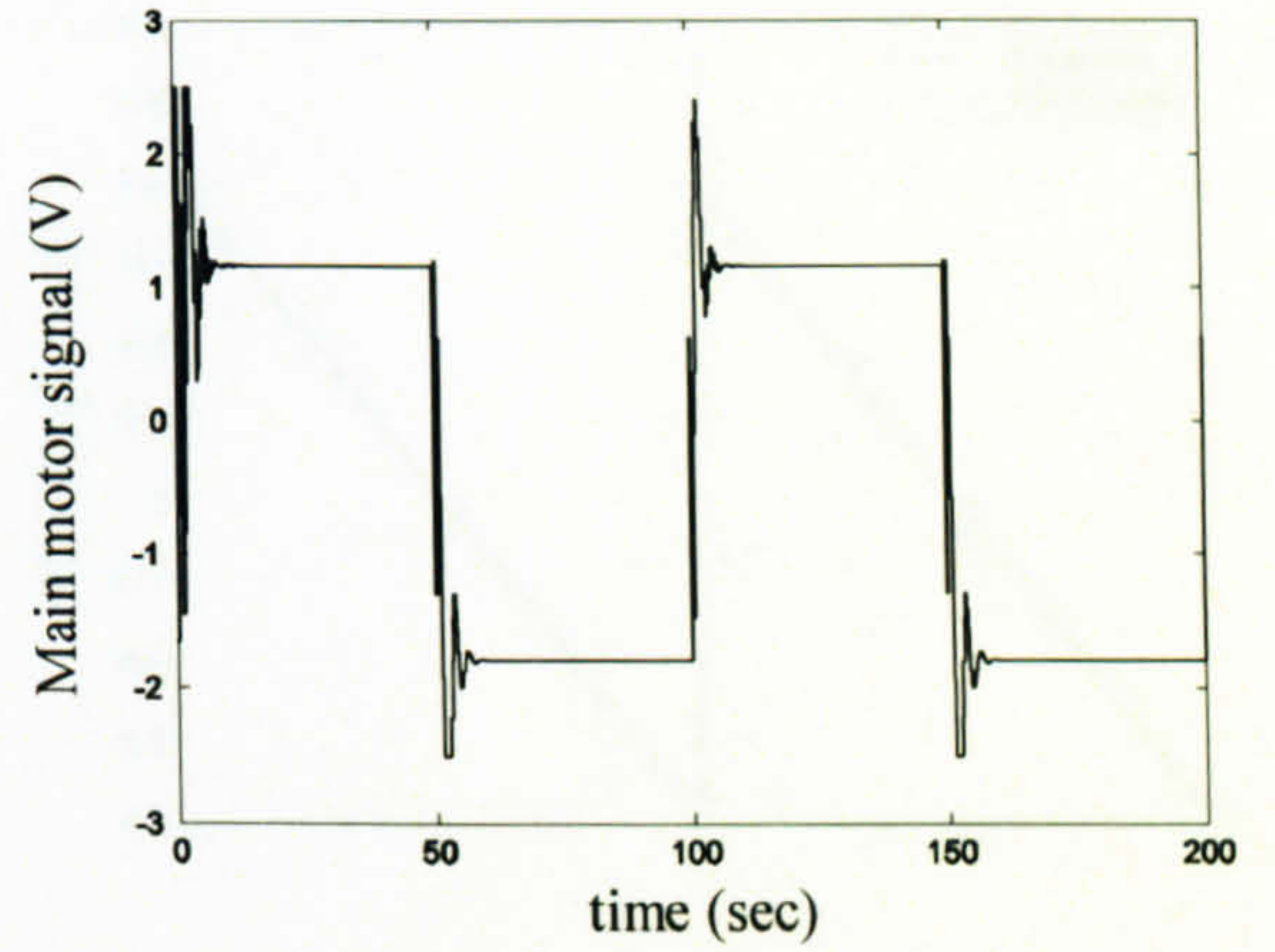


Figure 6.21 Main motor manipulated signal, U_v , (case 4)

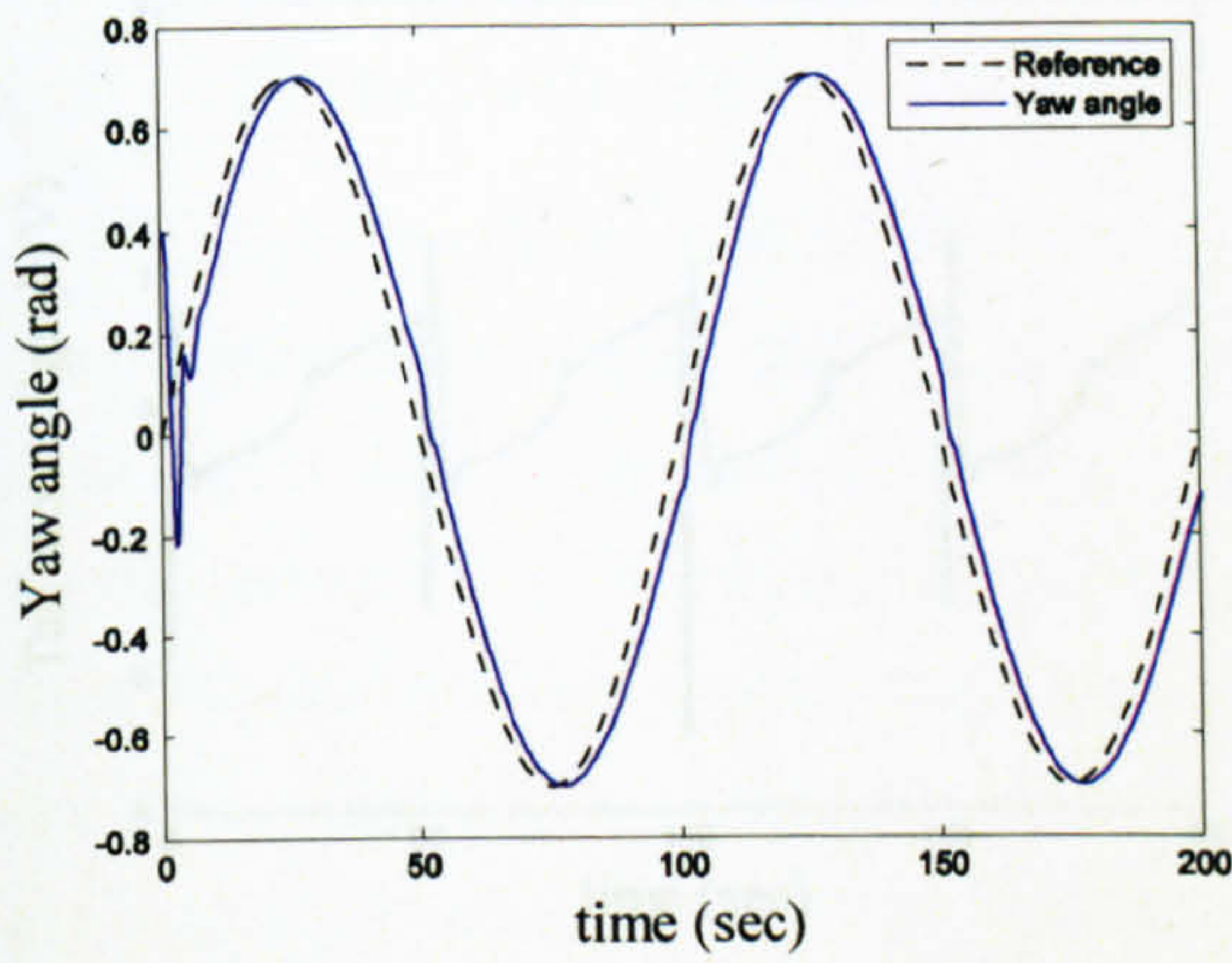


Figure 6.22 Sine wave response of the horizontal angle (case 5)

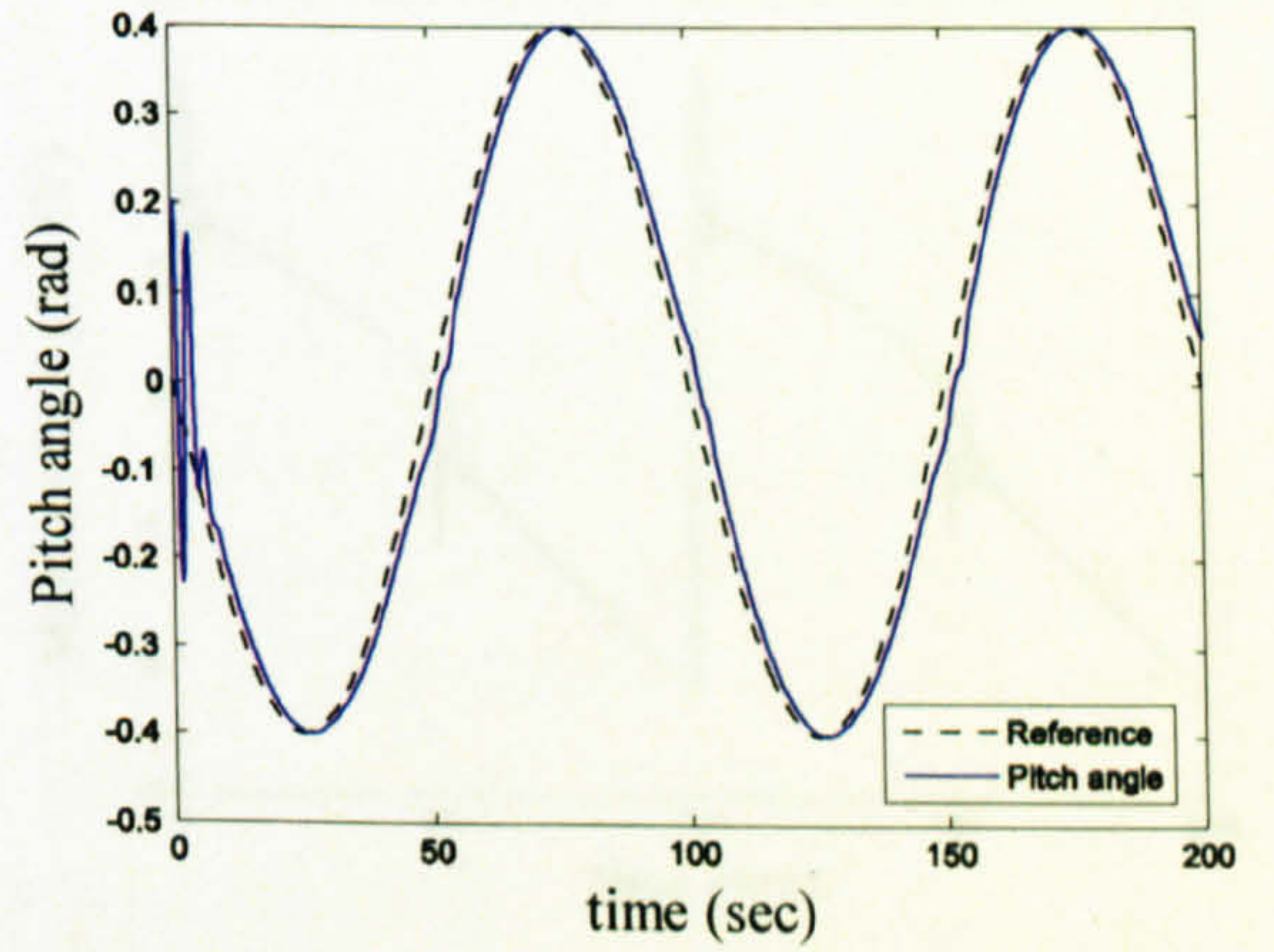


Figure 6.23 Sine wave response of the vertical angle (case 5)

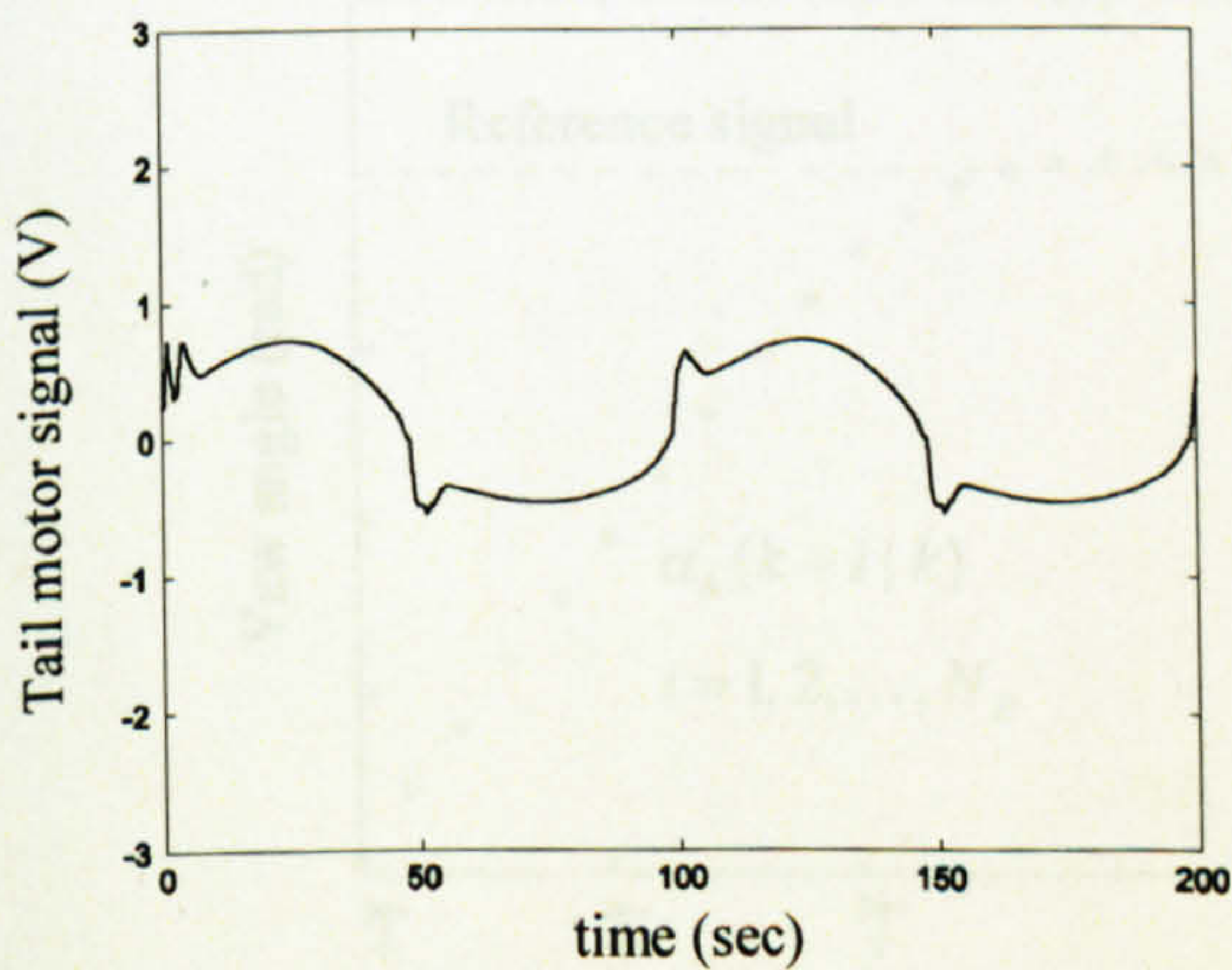


Figure 6.24 Tail motor manipulated signal, U_h , (case 5)

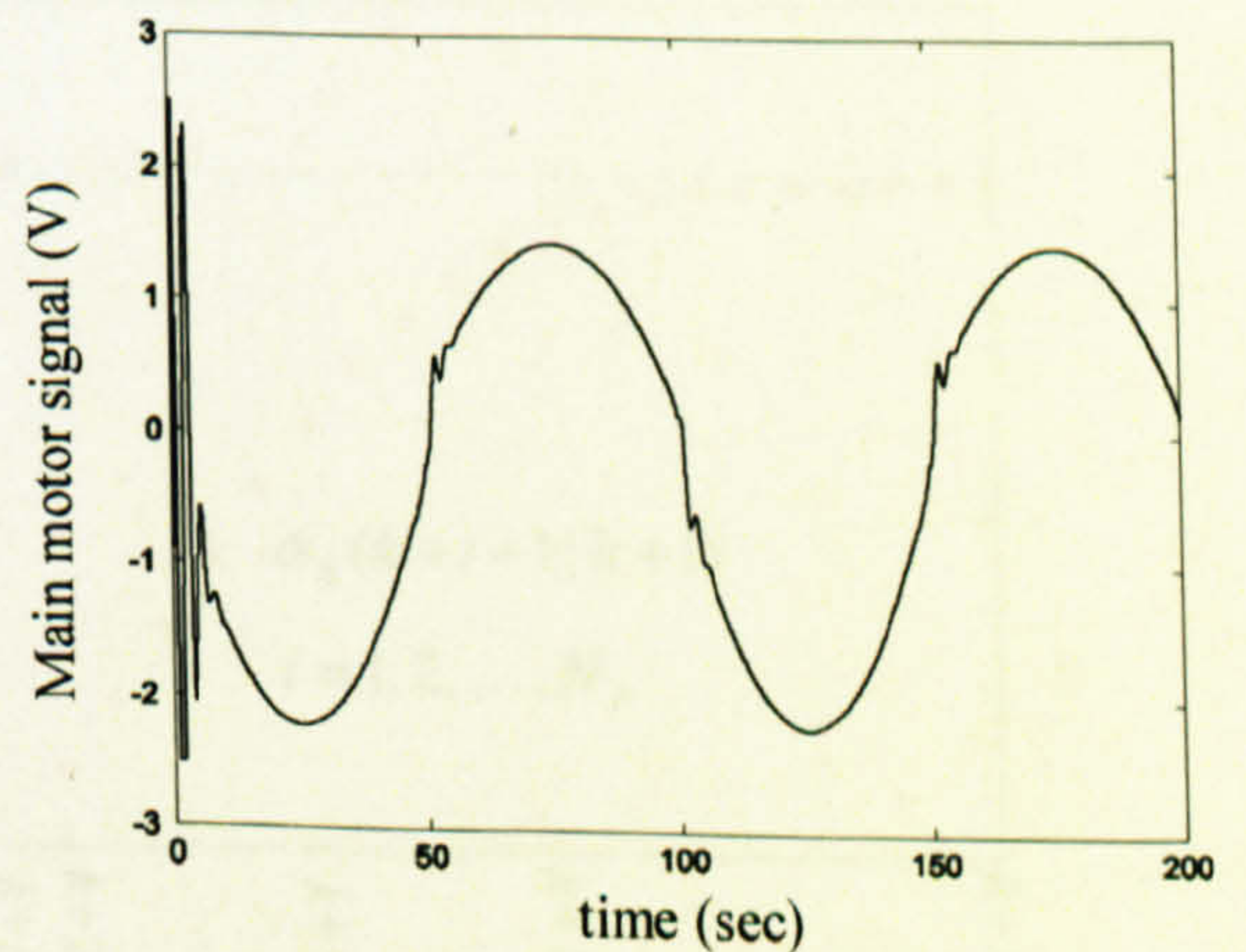


Figure 6.25 Main motor manipulated signal, U_v , (case 5)

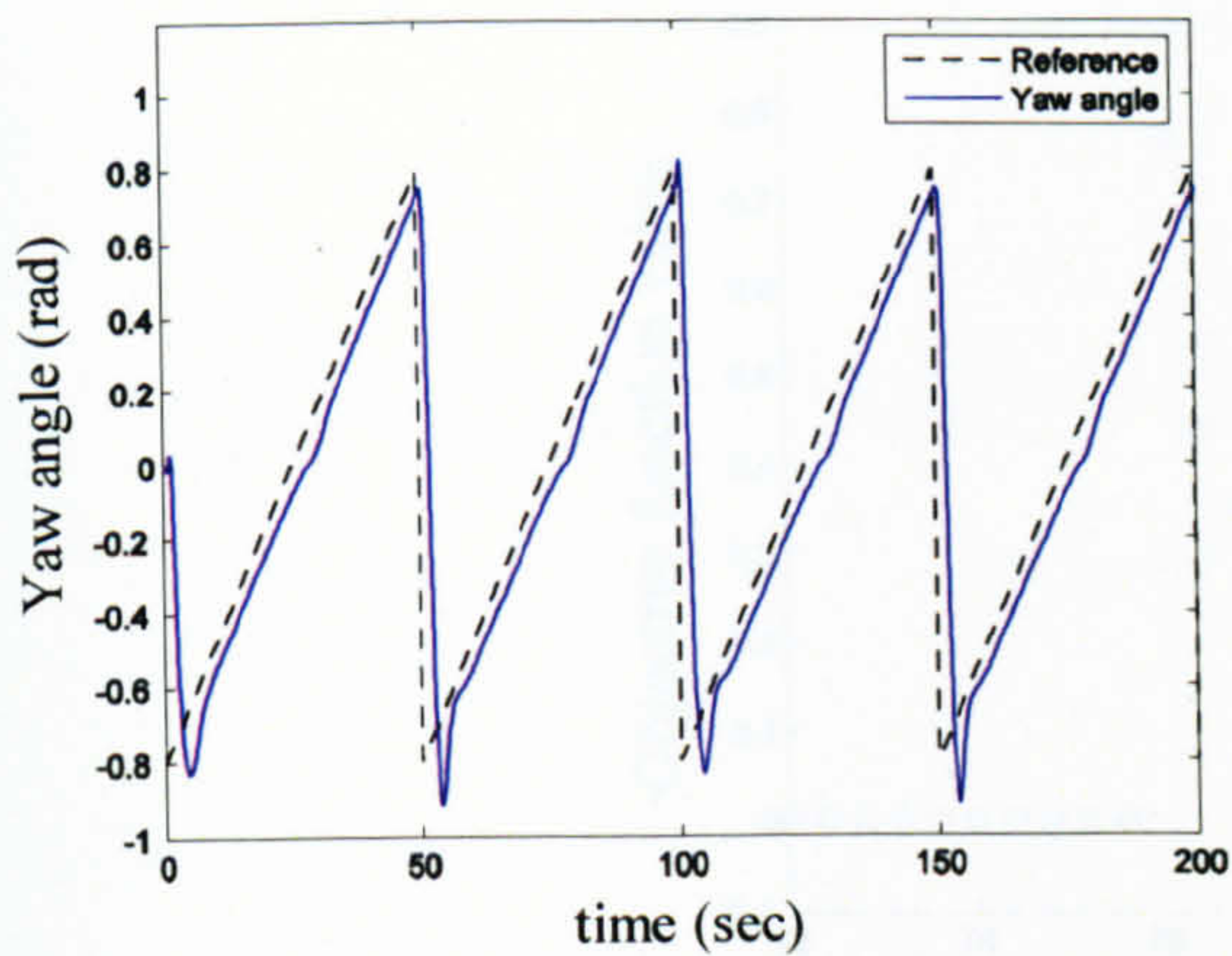


Figure 6.26 Sawtooth wave response of the horizontal angle (case 6)

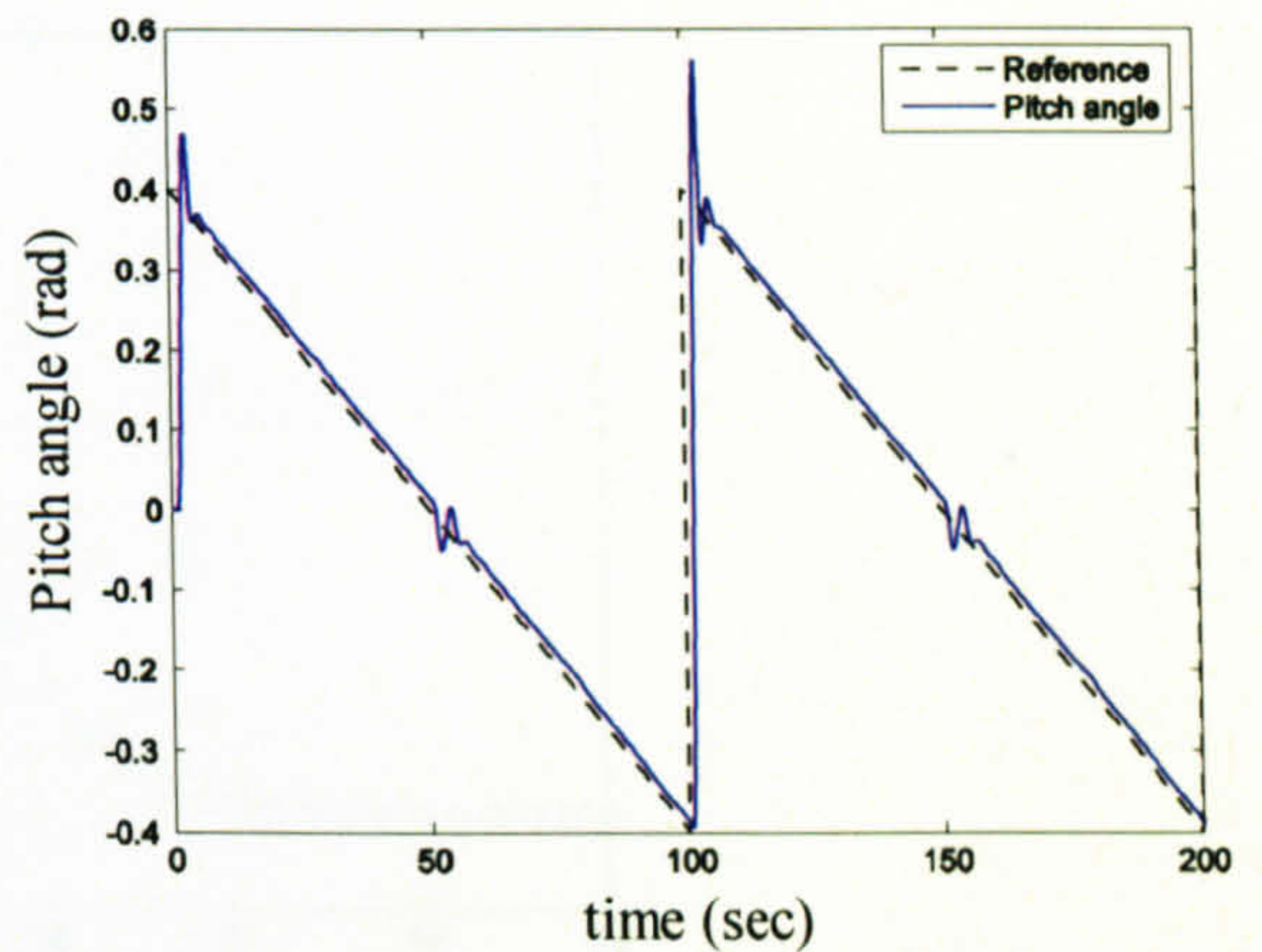


Figure 6.27 Sawtooth wave response of the vertical angle (case 6)

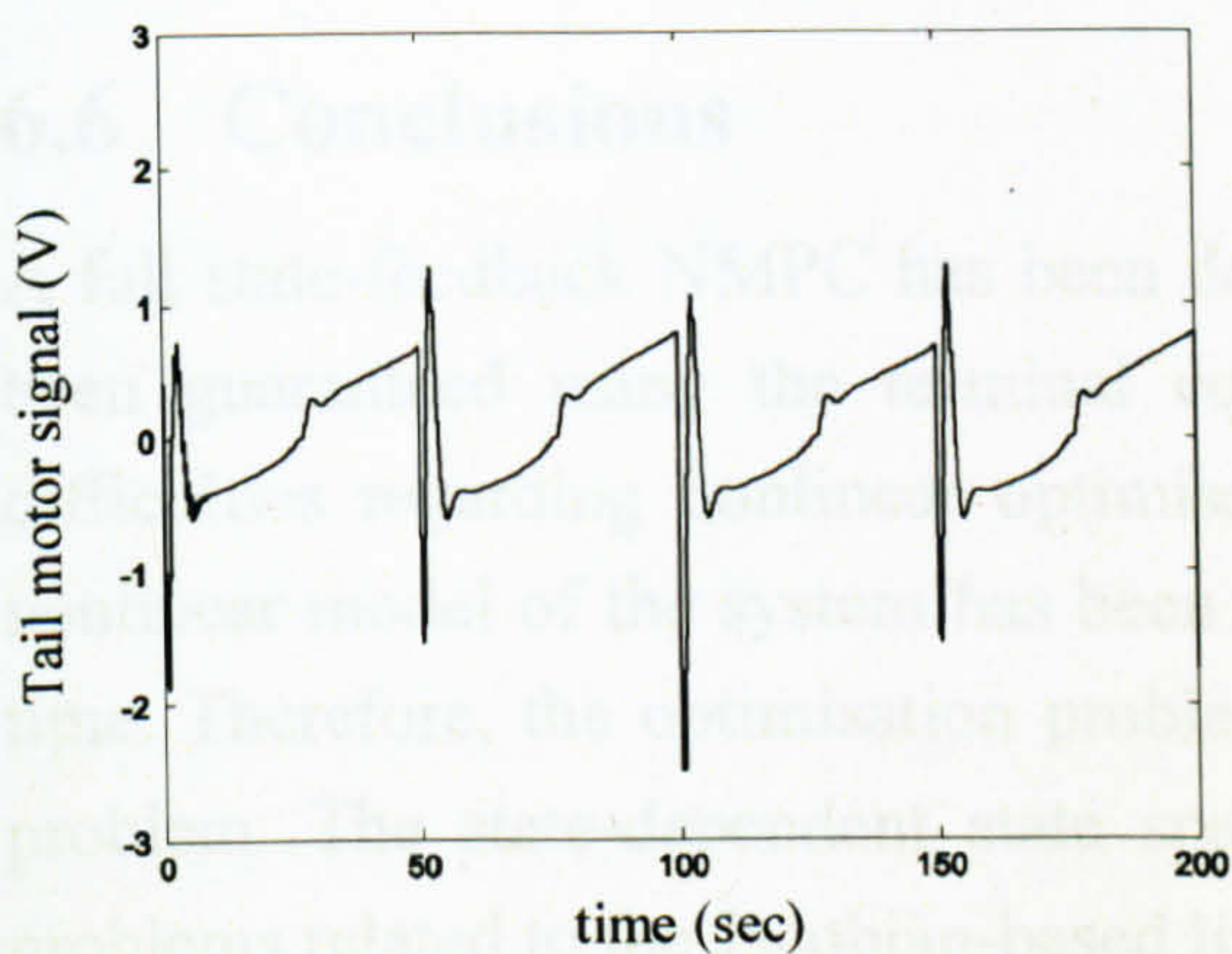


Figure 6.28 Tail motor manipulated signal, U_h , (case 6)

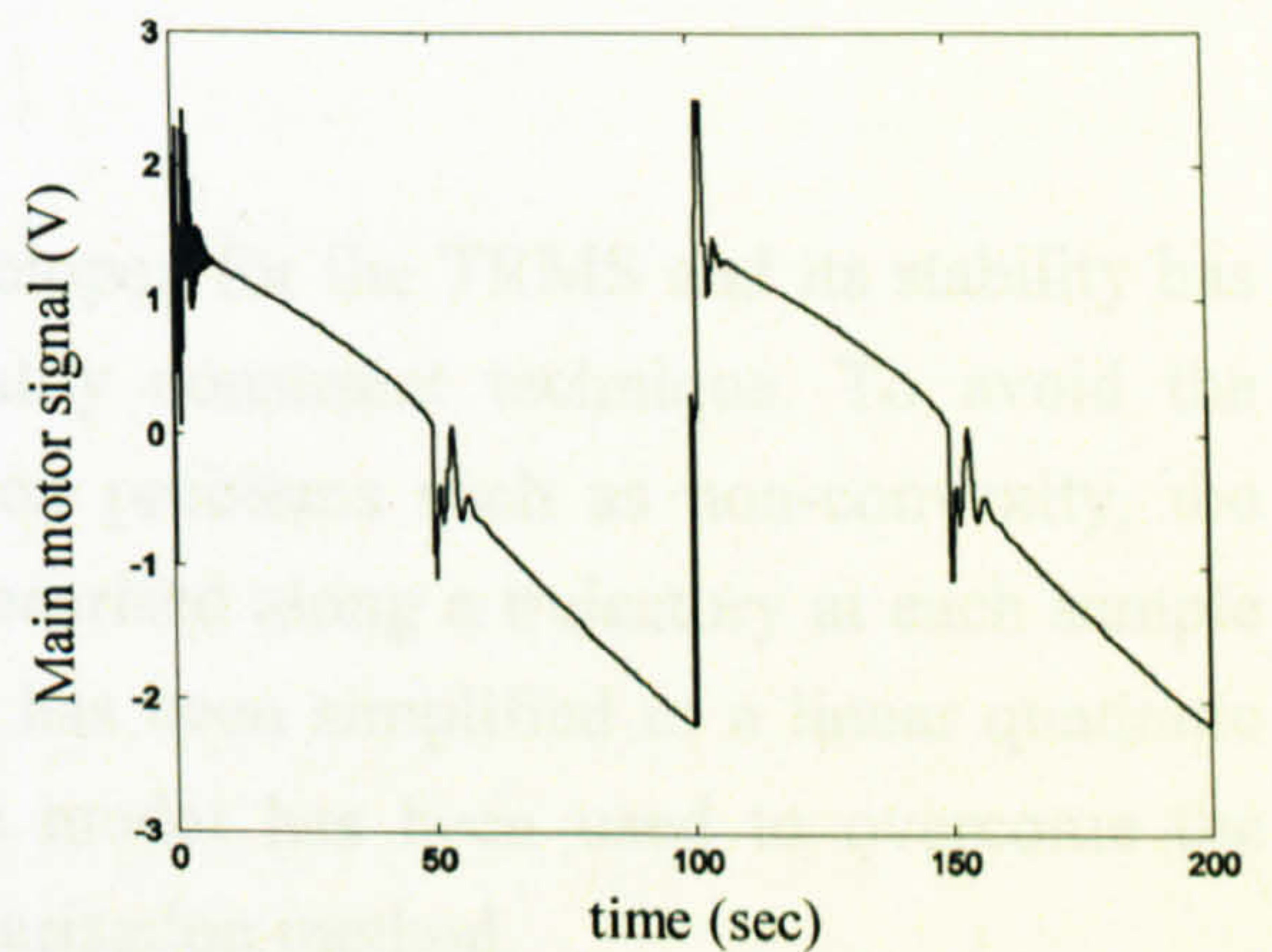


Figure 6.29 Main motor manipulated signal, U_v , (case 6)

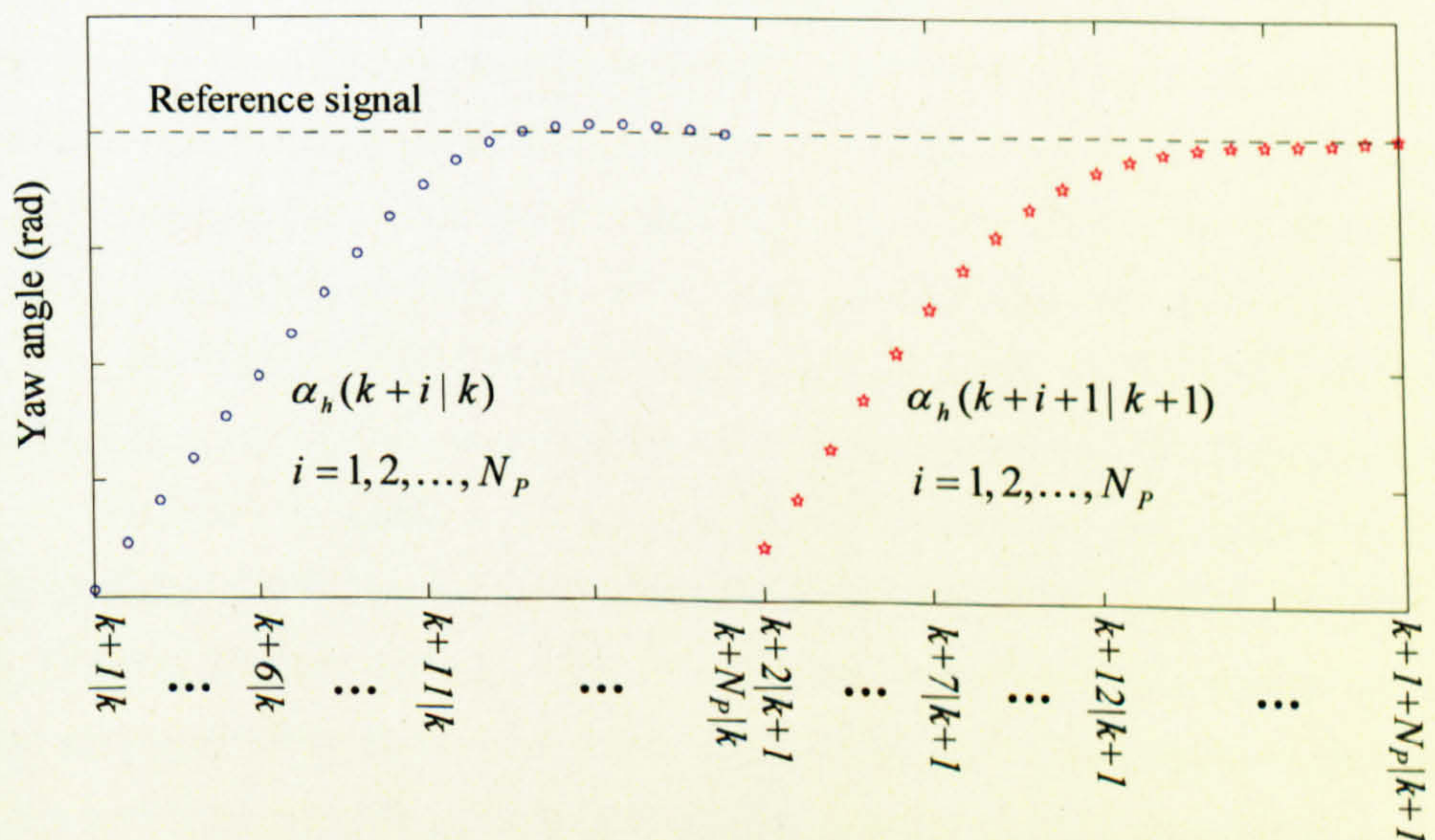


Figure 6.30 Two successive results of internal model output

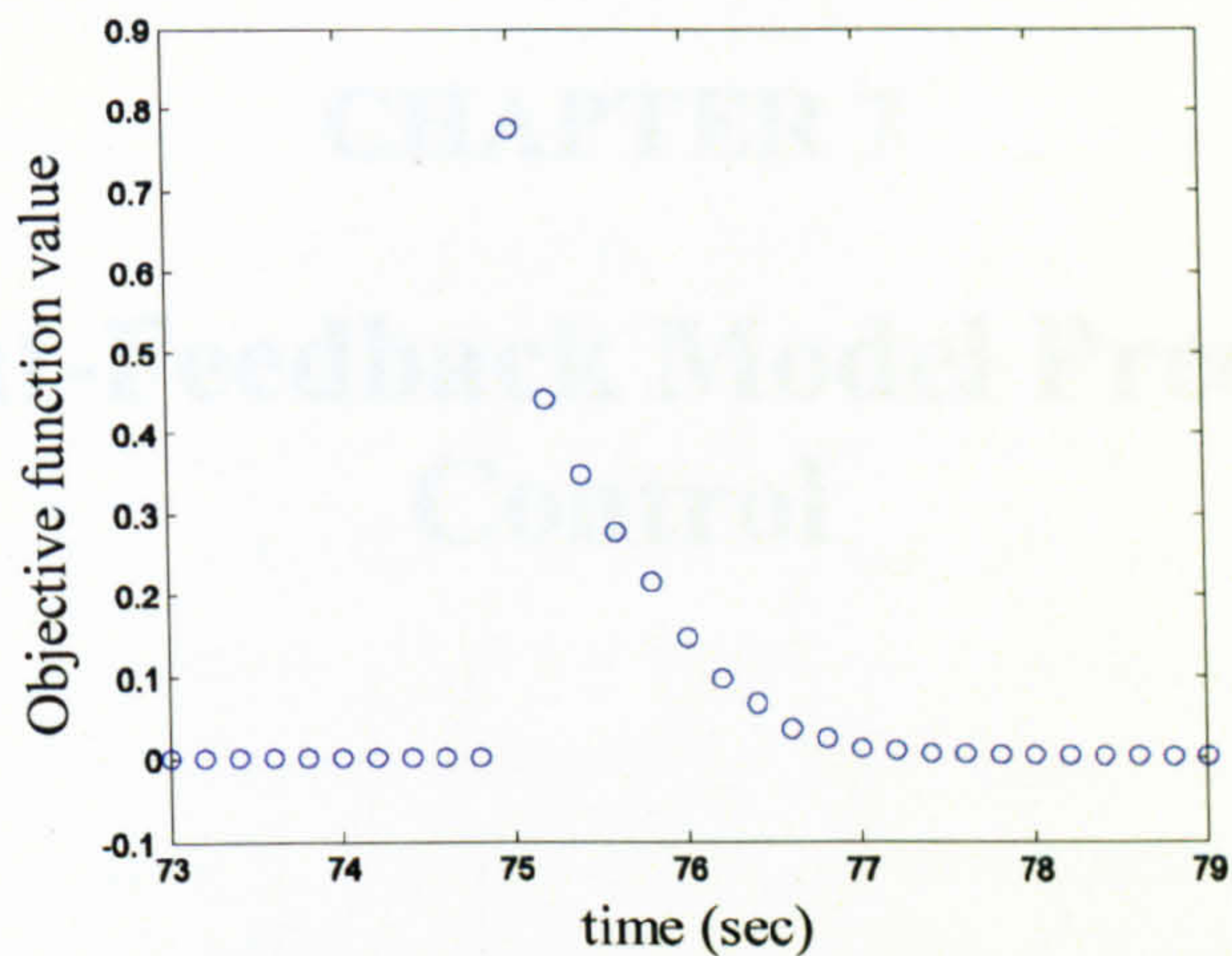


Figure 6.31 The objective function values related to the case of Figures 6.6 and 6.7

6.6 Conclusions

A full state-feedback NMPC has been developed for the TRMS and its stability has been guaranteed using the terminal equality constraint technique. To avoid the difficulties regarding nonlinear optimisation problems such as non-convexity, the nonlinear model of the system has been linearised along a trajectory at each sample time. Therefore, the optimisation problem has been simplified to a linear quadratic problem. The state-dependent state space model has been used to overcome the problems related to the Jacobian-based linearisation method.

CHAPTER 7

Output-Feedback Model Predictive Control

An output-feedback MPC approach for constrained nonlinear systems is presented in this chapter. The state variables are observed using a UKF that offers some advantages over an EKF. An EKF based on a state-dependent model is utilised in the experimental part of the study which shows a satisfactory performance. The Newton-type MPC proposed in the previous chapter is consolidated with this state observer to construct an output-feedback MPC for nonlinear systems.

7.1 Introduction

Generally, most advanced control techniques, such as MPC, need the state vector of the plant-model to be available. However, due to cost efficiency or other difficulties, full state variables are not measurable in practical systems. In such situations, state observers can be used to estimate or retrieve the state vector. Hence, before proceeding to a discussion on output-feedback MPC, it will be necessary to survey briefly on state observers. The elegant approach of state estimation, the Kalman filter [128], has widely been used for linear systems, over almost half a century. In the case of nonlinear systems its counterpart, EKF, has been introduced but, for highly nonlinear systems, it shows a poor performance. Julier *et al.* [129] have proposed a generalised version of the EKF known as UKF. The UKF shows more accurate state estimation in comparison with the EKF and also it can be implemented easily compared to the EKF [129]-[131]. However, in this study, an EKF is introduced to find the linearisation of the plant-model using a state-dependent state-space model. An output-feedback nonlinear MPC can be designed by consolidating a state-feedback nonlinear MPC and a state observer [74]-[77]; however, its stability cannot be held without further action, even if both the state-feedback controller and the observer are separately stable. Imslund *et al.* [74] have investigated the stability, robustness and performance of output-feedback nonlinear MPC methods.

7.2 Observer-based MPC Strategy

From a practical point of view, the state variables are inaccessible for measurements in the case of the TRMS. Therefore, a state observer needs to be used to reconstruct the state variables at each iteration of the control algorithm. To observe the state variables of the TRMS both the UKF and the state-dependent state-space EKF are employed. In the case of the UKF approach, a deterministic sampling technique, unscented transform, is used to select a minimal set of sample points, called sigma points, around the mean. The covariance of the estimate can be recovered using the propagation of these sigma points through the nonlinear function. Note that in the case of the UKF the calculation of Jacobian matrix is removed. In this part of research, an augmented type of UKF techniques is considered to estimate the states of the TRMS.

7.2.1 Methodology

In this section, the Newton-type MPC formulation is represented in a shortened form. For the time being, assume all state variables of the system are available. In the following section a UKF is utilised to observe all the state variables. The model state variables in the prediction horizon interval with respect to the current state variables and the future inputs can be expressed as,

$$\hat{\mathbf{x}}(k+i+1|k) = \mathbf{A}(k+i|k)\hat{\mathbf{x}}(k+i|k) + \mathbf{B}(k+i|k)\hat{\mathbf{u}}(k+i|k) \quad i = 0, \dots, N_p - 1 \quad (7.1)$$

that can be rewritten as,

$$\hat{\mathbf{x}}(k+i+1|k) = \left[\prod_{j=0}^i \mathbf{A}(k+i-j|k) \right] \mathbf{x}(k) + \sum_{n=0}^i \left\{ \left[\prod_{j=0}^{i-n-1} \mathbf{A}(k+i-j|k) \right] \mathbf{B}(k+n|k) \hat{\mathbf{u}}(k+n|k) \right\} \quad i = 0, \dots, N_p - 1 \quad (7.2)$$

where $\mathbf{x}(k) = \hat{\mathbf{x}}(k|k)$. The relationship between the inputs and the changes of the inputs is,

$$\hat{\mathbf{u}}(k+j|k) = \mathbf{u}(k-1) + \sum_{i=0}^j \Delta \hat{\mathbf{u}}(k+i|k) \quad j = 0, 1, \dots, N_c - 1 \quad (7.3)$$

By substituting equations (7.3) into equations (7.2) the following equations can be written,

$$\begin{aligned} \hat{\mathbf{x}}(k+i+1|k) = & \left[\prod_{j=0}^i \mathbf{A}(k+i-j|k) \right] \mathbf{x}(k) + \sum_{n=0}^i \left\{ \left[\prod_{j=0}^{i-n-1} \mathbf{A}(k+i-j|k) \right] \mathbf{B}(k+n|k) \right\} \Delta \hat{\mathbf{u}}(k-1) \\ & + \sum_{m=0}^{\min(i, N_C-1)} \left(\sum_{n=m}^i \left\{ \left[\prod_{j=0}^{i-n-1} \mathbf{A}(k+i-j|k) \right] \mathbf{B}(k+n|k) \right\} \Delta \hat{\mathbf{u}}(k+m|k) \right) \quad i = 0, \dots, N_p - 1 \end{aligned} \quad (7.4)$$

Note that in equation (7.4) the first two terms are related to the past variables that are known and the last term is associated with the future signals that should be optimally calculated using an optimisation technique. The output predictions can be obtained as,

$$\hat{\mathbf{y}}(k+i|k) = \mathbf{C}(k+i|k) \hat{\mathbf{x}}(k+i|k) + \hat{\mathbf{d}}(k+i|k) \quad i = 1, \dots, N_p \quad (7.5)$$

where $\hat{\mathbf{d}}(k+i|k)$ is the disturbance. Substitution (7.4) into (7.5) leads to the following equation,

$$\mathbf{Y}(k) = \mathbf{M}_C(k) \mathbf{M}_A(k) \mathbf{x}(k) + \mathbf{M}_C(k) \mathbf{M}_B(k) \mathbf{u}(k-1) + \mathbf{M}_C(k) \mathbf{M}_U(k) \Delta \mathbf{U}(k) + \mathbf{M}_d(k) \quad (7.6)$$

that has been obtained in the previous chapter.

7.2.2 The UKF

As pointed out earlier, in the case of highly nonlinear systems the EKF illustrates unacceptable performance of observation mostly due to its use of linearisation [130]. The so-called UKF provides a more precise result using unscented transformation. The flowchart of the unscented Kalman state observer is depicted in Figure 7.1.

7.2.3 The State-Dependent EKF

The EKF uses Jacobian-based linearisation approach to update the linear model at each stage which is the main cause of its poor performance for highly nonlinear systems. The so-called state-dependent EKF employs a state-dependent state-space model to find the linear models by substituting the most recent estimated or measured state variables into the state-dependent model. Assume a nonlinear state-space plant-model,

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) + \mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{h}(\mathbf{x}(k)) + \mathbf{v}(k) \end{aligned} \quad (7.7)$$

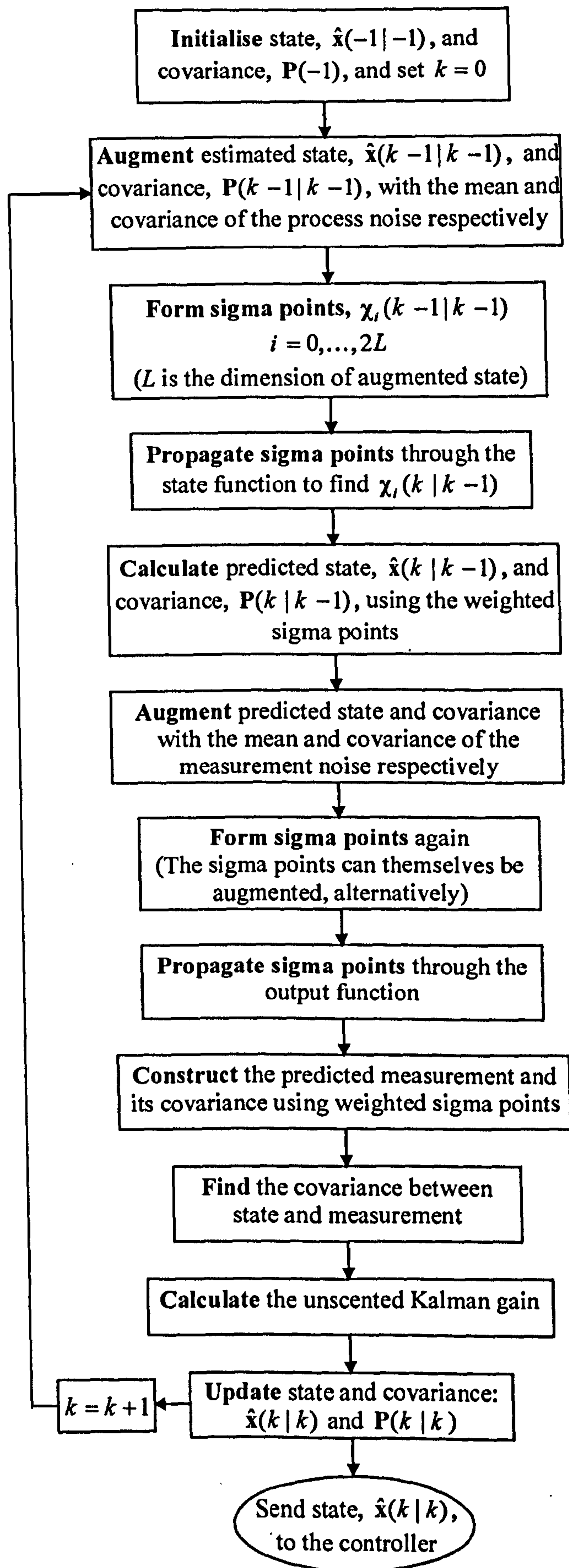


Figure 7.1 Flowchart of the unscented Kalman observer procedure

where $w(k)$ and $v(k)$ are white noise with covariance matrices $Q(k)$ and $R(k)$ respectively. The nonlinear state-space equation can be approximated by the following state-dependent state-space equation,

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}(\mathbf{x}(k))\mathbf{x}(k) + \mathbf{B}(\mathbf{x}(k))\mathbf{u}(k) + \mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{C}(\mathbf{x}(k))\mathbf{x}(k) + \mathbf{v}(k) \end{aligned} \quad (7.8)$$

Optimum state variables estimation can be obtained using the following equations iteratively,

Prediction stage

$$\hat{\mathbf{x}}(k|k-1) = \mathbf{f}(\hat{\mathbf{x}}(k-1|k-1), \mathbf{u}(k)) \quad (7.9)$$

$$\mathbf{M}(k) = \mathbf{A}(\hat{\mathbf{x}}(k|k-1))\mathbf{P}(k-1)\mathbf{A}^T(\hat{\mathbf{x}}(k|k-1)) + \mathbf{Q}(k) \quad (7.10)$$

where, $\hat{\mathbf{x}}(k|k-1)$ is the predicted state and $\mathbf{M}(k)$ is the predicted error covariance.

Correction or update stage

$$\mathbf{L}'(k) = \mathbf{M}(k)\mathbf{C}^T(\hat{\mathbf{x}}(k|k-1))[\mathbf{C}(\hat{\mathbf{x}}(k|k-1))\mathbf{M}(k)\mathbf{C}^T(\hat{\mathbf{x}}(k|k-1)) + \mathbf{R}(k)]^{-1} \quad (7.11)$$

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{L}'(k)[\mathbf{y}(k) - \mathbf{h}(\hat{\mathbf{x}}(k|k-1))] \quad (7.12)$$

$$\mathbf{P}(k) = (\mathbf{I} - \mathbf{L}'(k)\mathbf{C}(\hat{\mathbf{x}}(k|k-1)))\mathbf{M}(k) \quad (7.13)$$

where, $\mathbf{L}'(k)$ is the Kalman gain, $\hat{\mathbf{x}}(k|k)$ is the updated state estimate, and $\mathbf{P}(k)$ is the update error covariance. The source codes of the developed filters can be seen in Appendix 5.

7.3 Simulation Results

Based on the nonlinear model, N_p linear models are developed to model the nonlinear system during the prediction horizon at each instant, k . The objective function and constraints are formed based on these linear models to have a constrained linear quadratic optimisation problem. The UKF is implemented to estimate the state variables at each instant. At each iteration, the first set of the optimum input vector, $\Delta\hat{\mathbf{u}}(k|k)$, is added to the previous control signal, $\mathbf{u}(k-1)$, and the result is sent to the plant and also to the linear model at $k|k$ instant. The others optimum values are kept for the next sample time, $k+1$, calculation. The block diagram of the output-feedback Newton-type MPC is shown in Figure 7.2. Note that the nonlinear model and linearisation operator have not been shown in Figure 7.2. Figure 7.3 shows the flowchart of the proposed method.

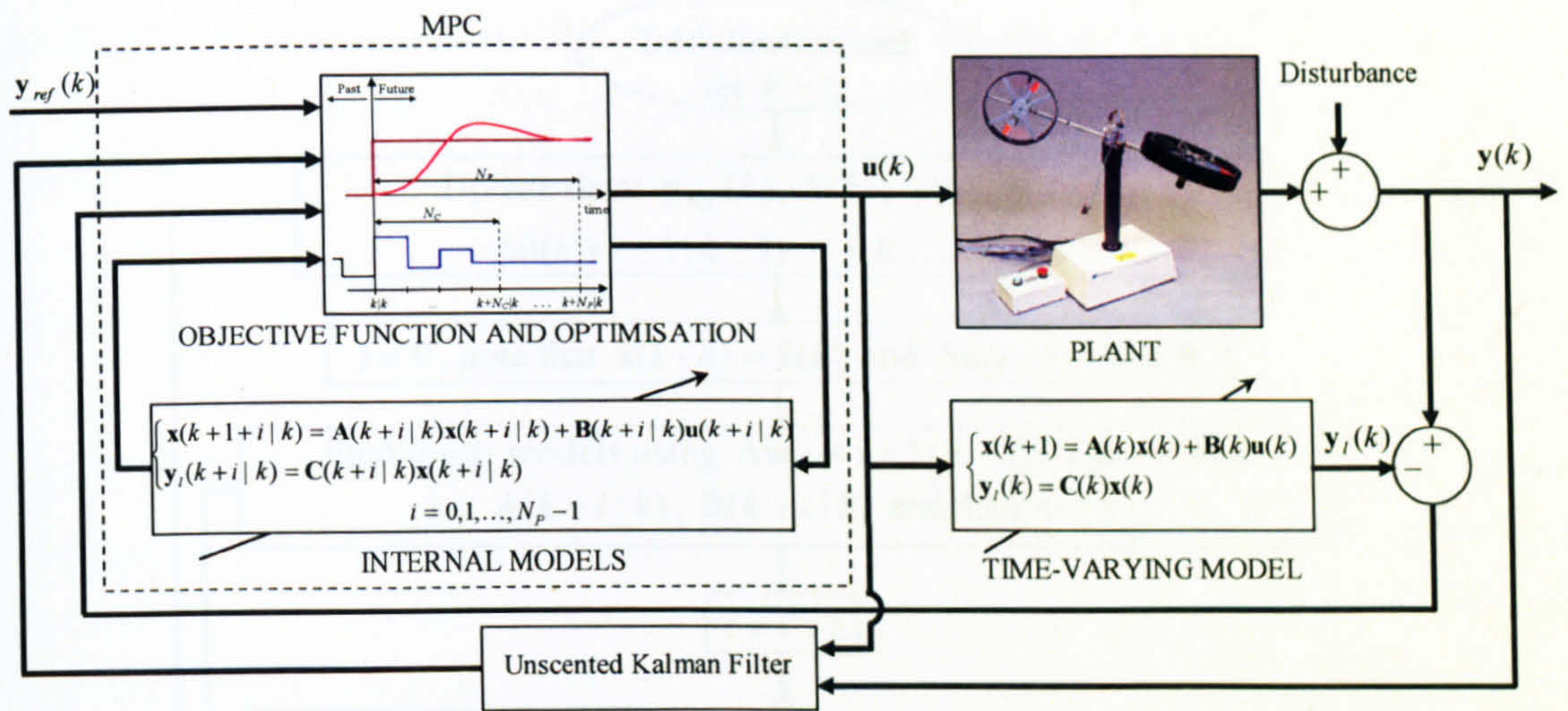


Figure 7.2 Block diagram of the output-feedback Newton-type MPC approach

The parameters of state-feedback MPC presented in the previous chapter are considered for the applied output-feedback nonlinear MPC, see Table 6.1. Again, the sampling time of the model predictive controller is set to be 0.2 seconds and the optimisation approach is chosen to be an active set method. The computational time for a period of 100 seconds simulation with parameters listed in Table 6.1 is 7.3 seconds that means the proposed method can be implemented in real time.

The output-feedback Newton-type MPC developed for the TRMS has been evaluated with a variety of reference signals and the results obtained demonstrate that the controller has a high performance and reliability in the various operating regions. Responses with some of the reference signals used are presented here and the results with similar characteristics are not repeated. The controller has been proven to be reliable under disturbances and various reference signals used. Figures 7.4 and 7.5 show the results of the horizontal (yaw) and the vertical (pitch) angles of the beam with square references.

It is clear from Figures 7.4 and 7.5 that the two channels have significant effect on each other. For instance, at the instant of the 25th second, the yaw angle reference signal has been changed from -1 to 1 radian and subsequently, the yaw angle has followed the reference signal. Here, the pitch angle, due to the mentioned coupling, has been affected by the yaw angle; however, the controller has regained the control and forced the pitch angle to follow the reference trajectory.

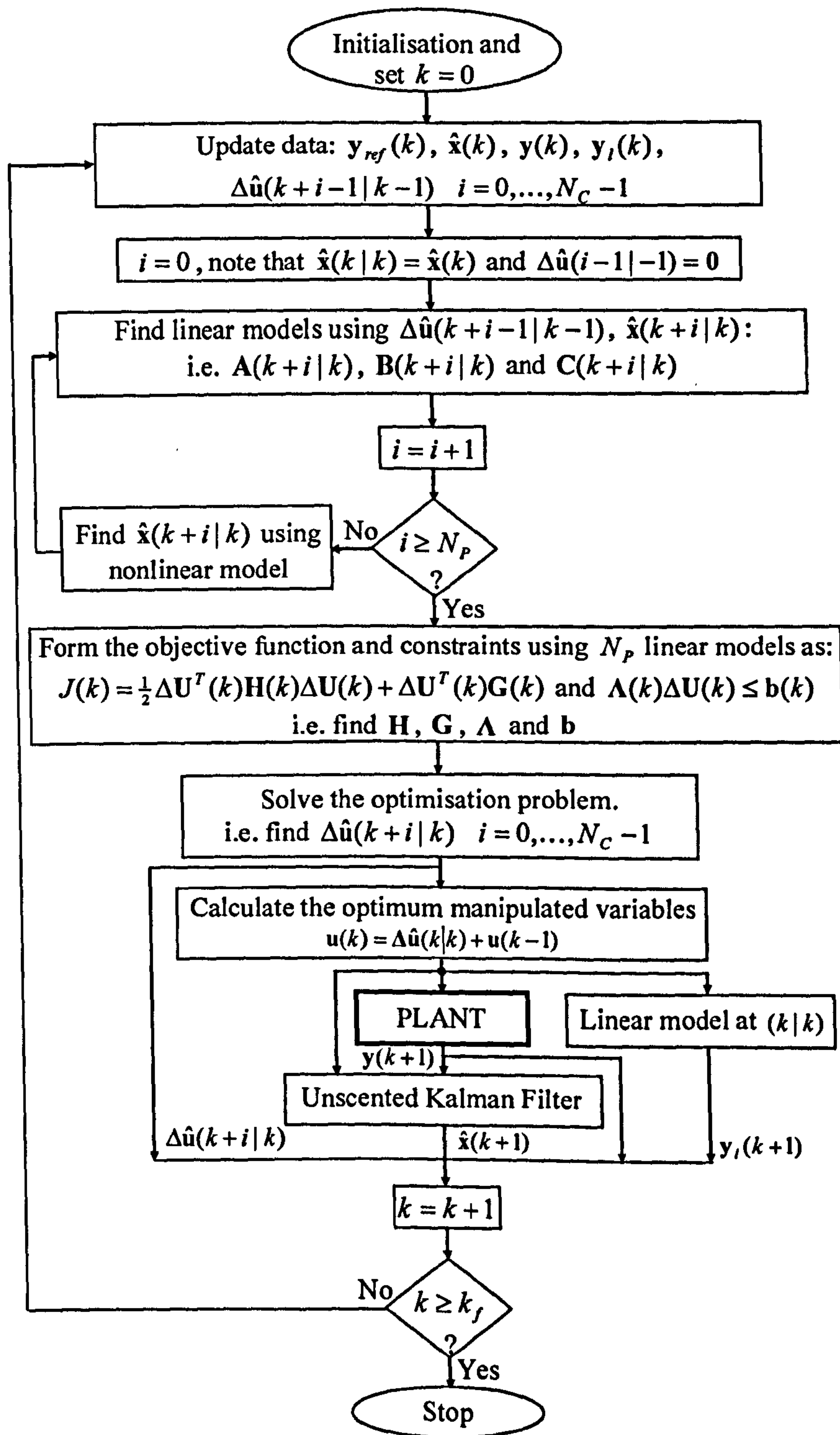


Figure 7.3 Flowchart of the output-feedback Newton-type MPC approach

The overshoots at the instants 50 and 150 seconds of the yaw angle response have been rendered by the step change of the pitch angle reference signal at those instants. As pointed out earlier, this set of reference signals is one of the challenging trajectories in the case of the TRMS and it has been selected deliberately to confirm the performance of the proposed control method. The manipulated signals of the tail and main motors are shown in Figures 7.6 and 7.7 respectively.

The results with square wave references but with swapping the frequencies of both the channels are shown in Figures 7.8 and 7.9 respectively that show the significant effect of the vertical channel on the horizontal one. Again, the controller has regained the control despite the considerable deviation of the horizontal angle. The control signals of the two channels related to this case can be found in Figures 7.10 and 7.11.

Figures 7.12 to 7.15 illustrate the responses of the controller according to square reference signals, but with a non-zero initial position of the beam, and their corresponding manipulated signals. The performance of the controller with sine reference signals has been depicted in Figures 7.16 to 7.17 with a non-zero initial position of the beam which shows the ability of the proposed control system in following a continuously variable reference trajectory. Again, the input signals to the TRMS which force the yaw and pitch angles to follow the reference signals are depicted in Figures 7.18 and 7.19. The results of another case of sine wave reference signals are illustrated in Figures 7.20 to 7.23. Figures 7.24 and 7.25 illustrate the controller performance with respect to sawtooth reference signals. This case is a combination of step change and reference trajectory and clearly the control system is able to track this type of reference signal as well. Input signals to the TRMS in terms of case 6 are presented in Figures 7.26 and 7.27. In order to illustrate the performance of the UKF, a couple of estimated states have been compared with the corresponding actual states. Figure 7.28 depicts the comparison between the actual and estimated S_v . The same comparison in the case of S_h is shown in Figure 7.29.

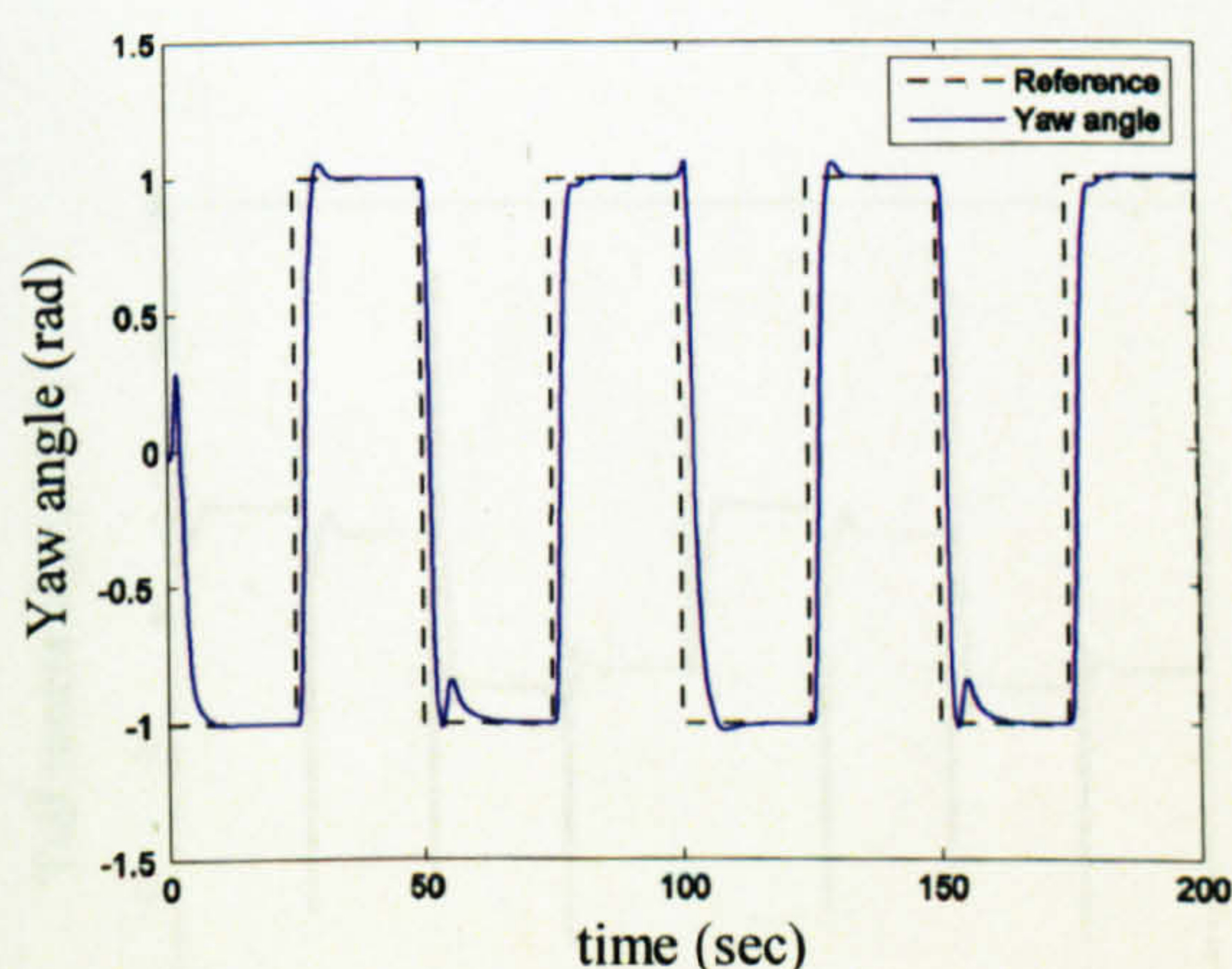


Figure 7.4 Square wave response of the horizontal angle (case 1)

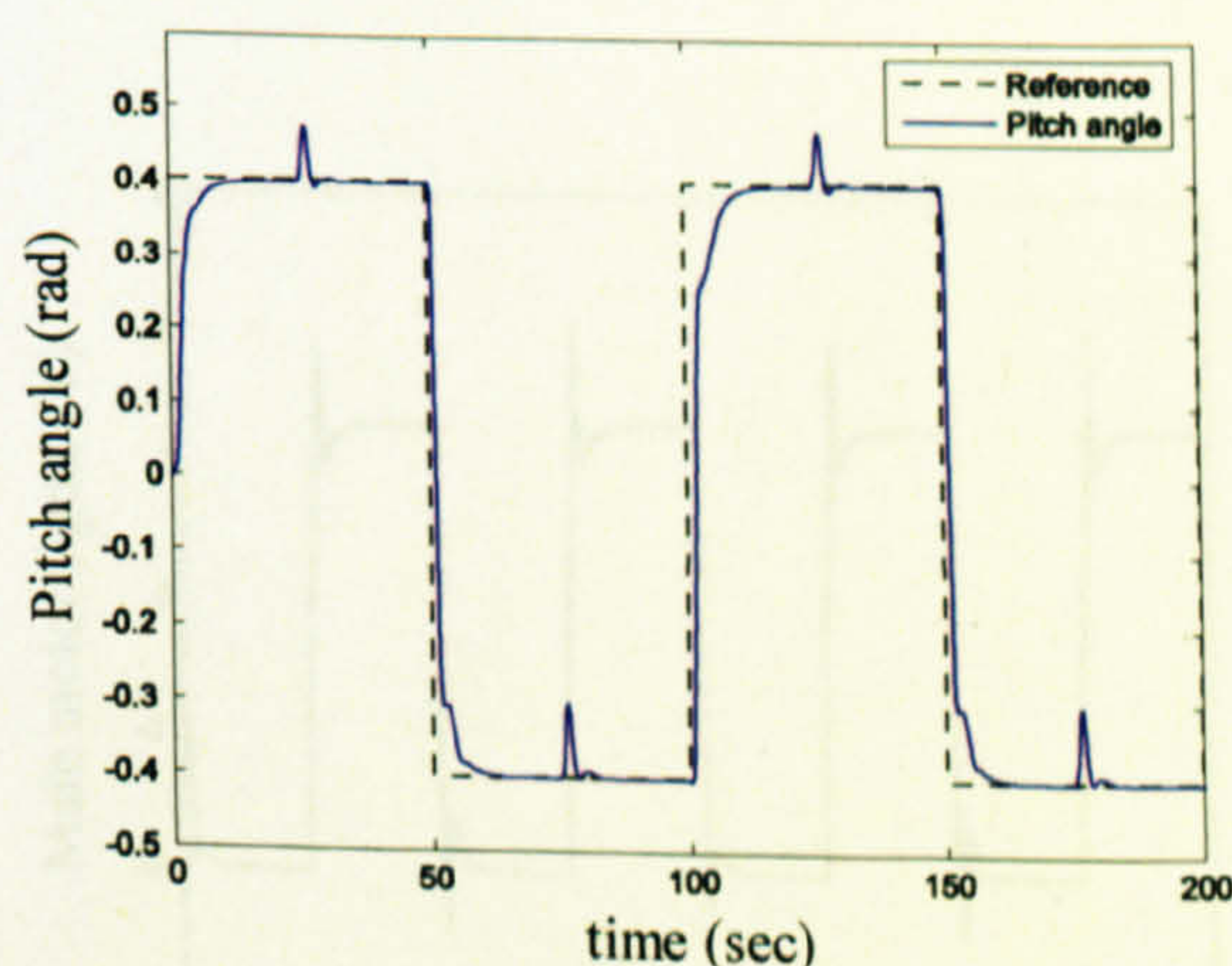


Figure 7.5 Square wave response of the vertical angle (case 1)

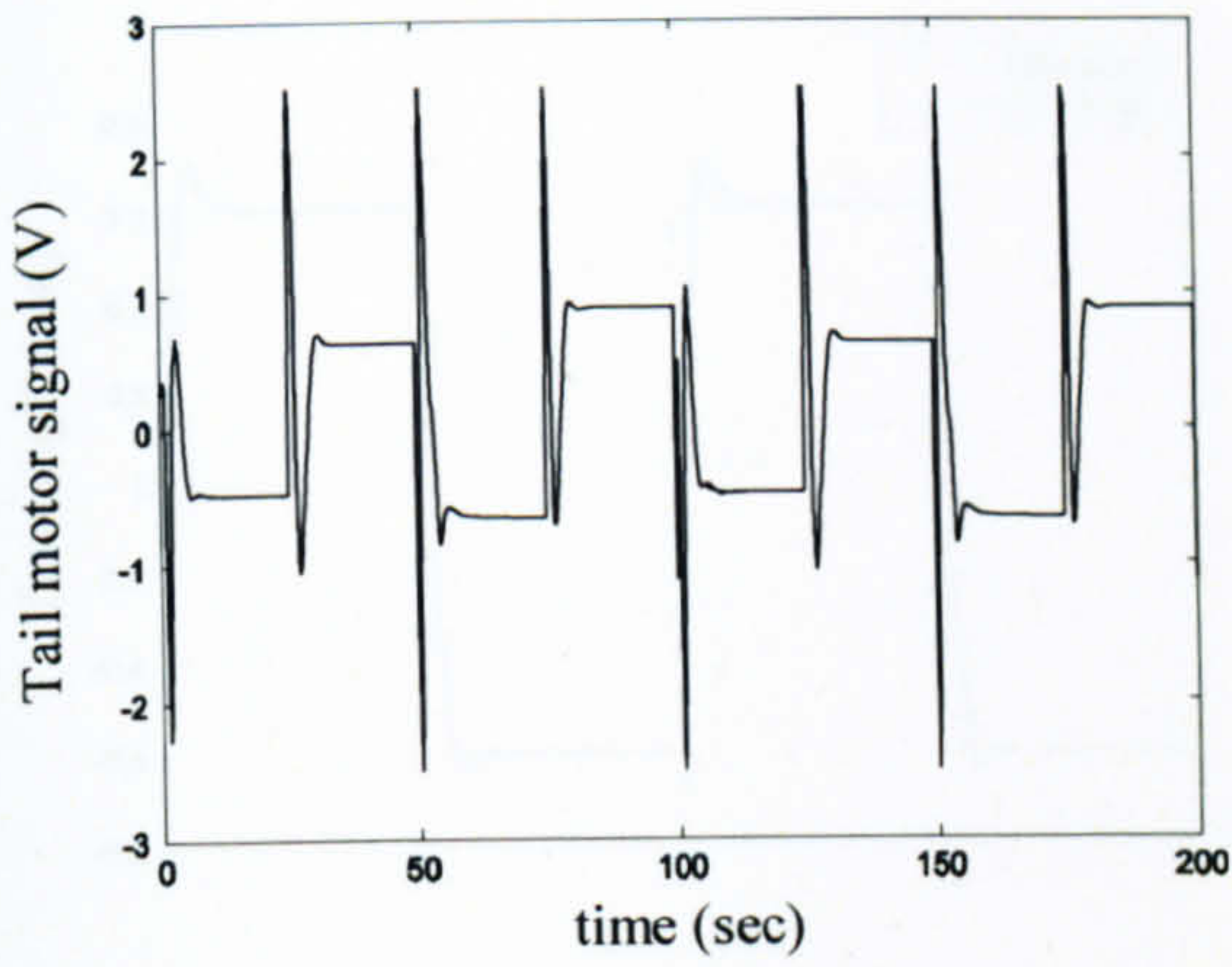


Figure 7.6 Tail motor manipulated signal, U_h , (case 1)

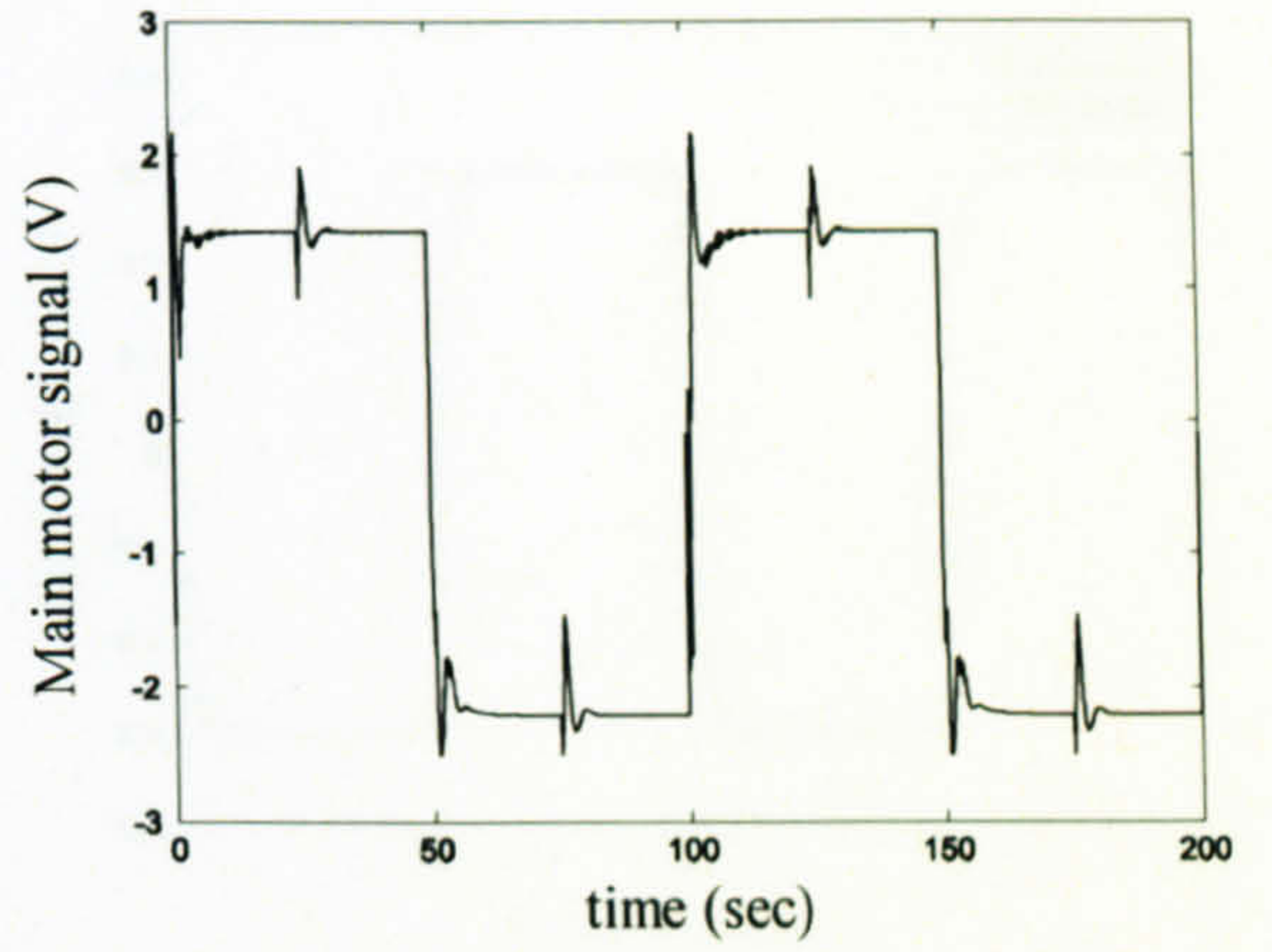


Figure 7.7 Main motor manipulated signal, U_v , (case 1)

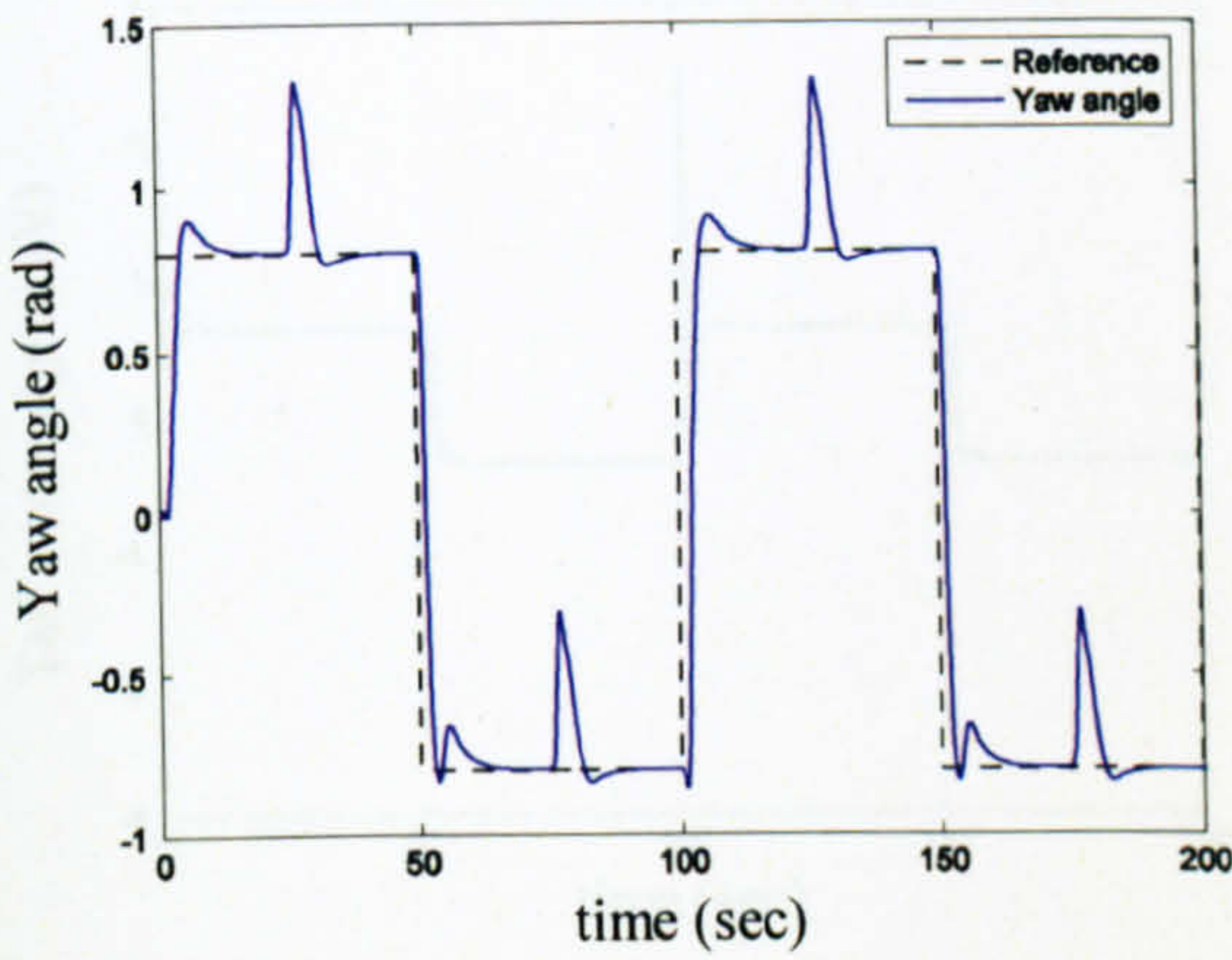


Figure 7.8 Square wave response of the horizontal angle (case 2)

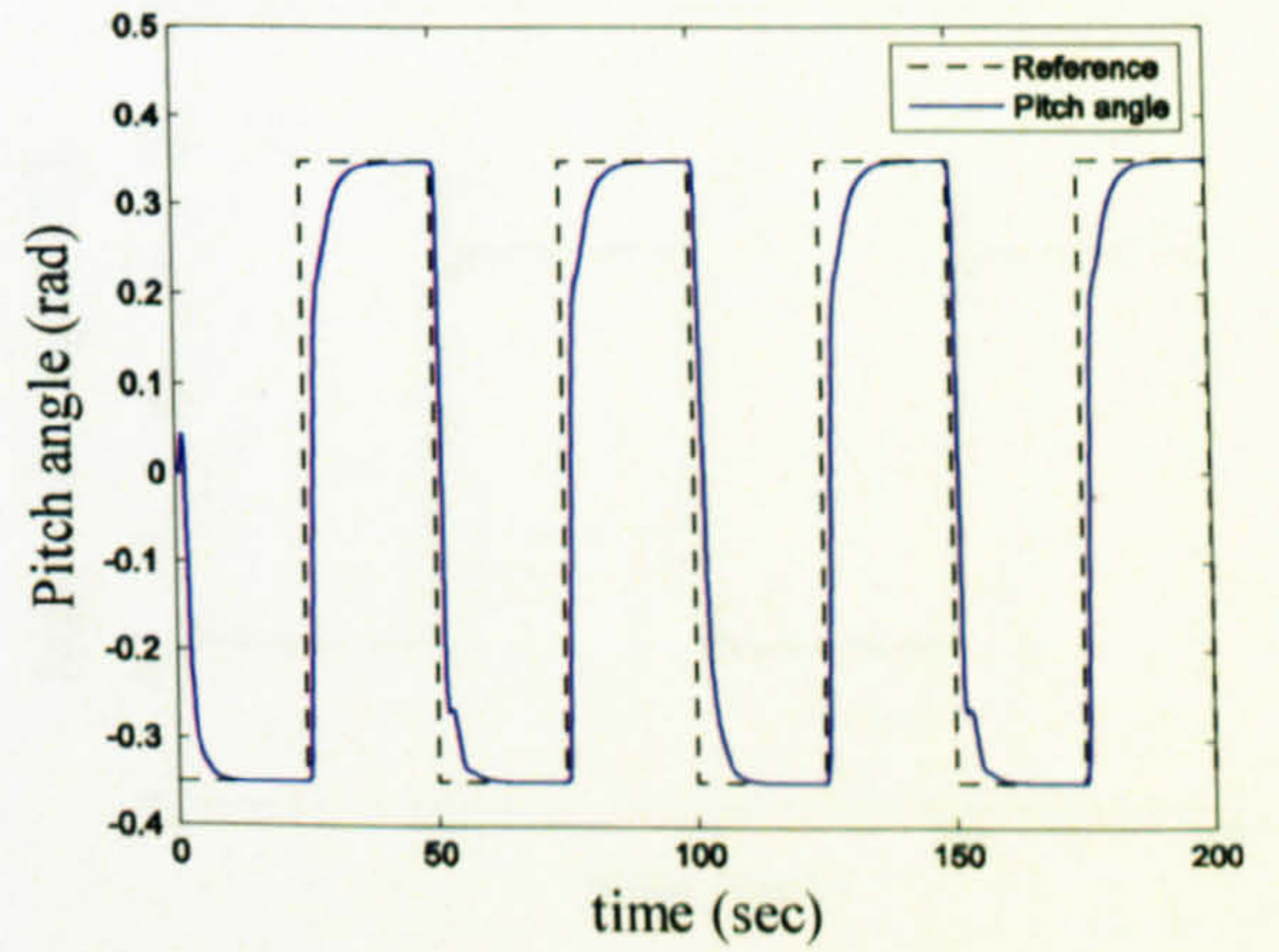


Figure 7.9 Square wave response of the vertical angle (case 2)

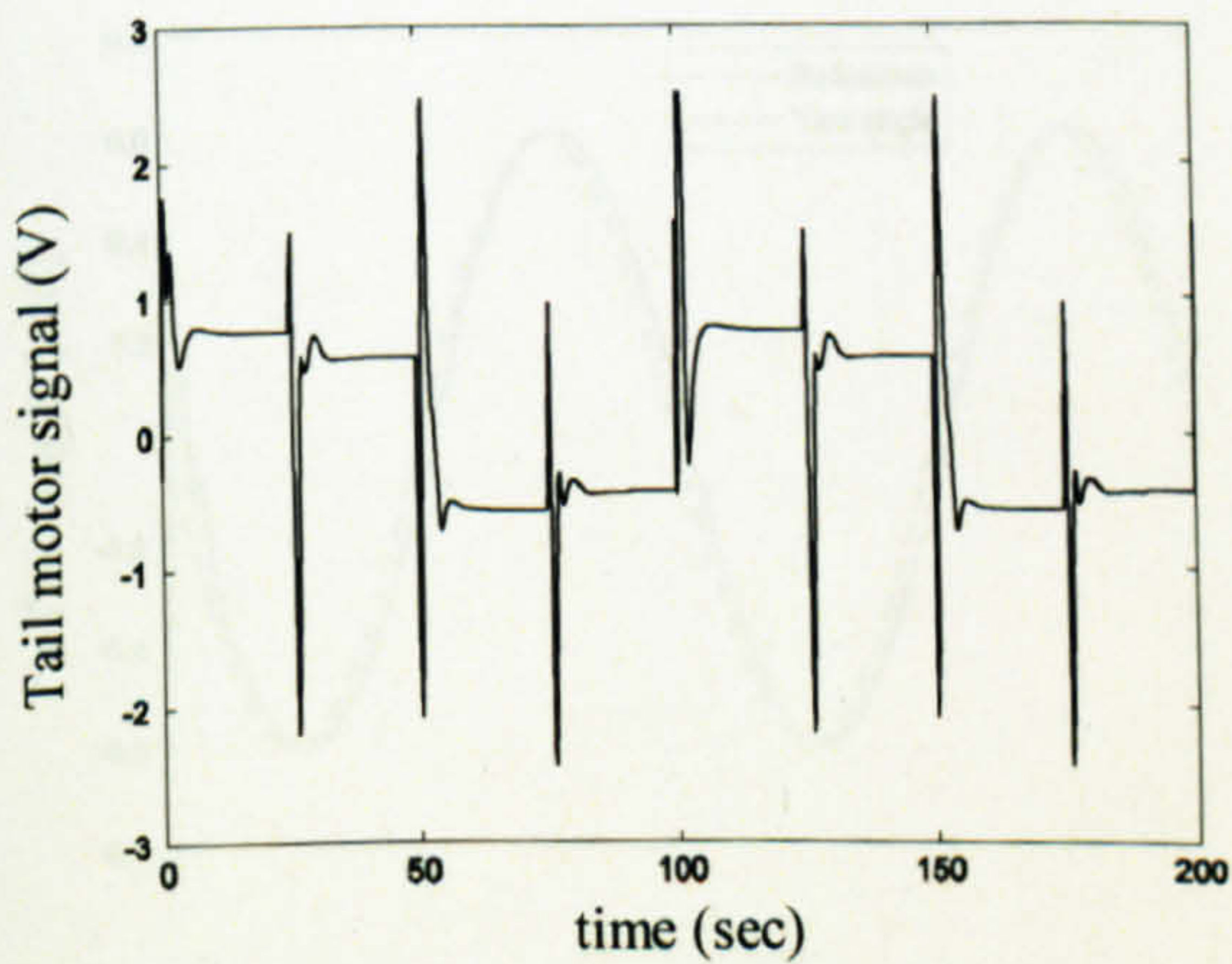


Figure 7.10 Tail motor manipulated signal, U_h , (case 2)

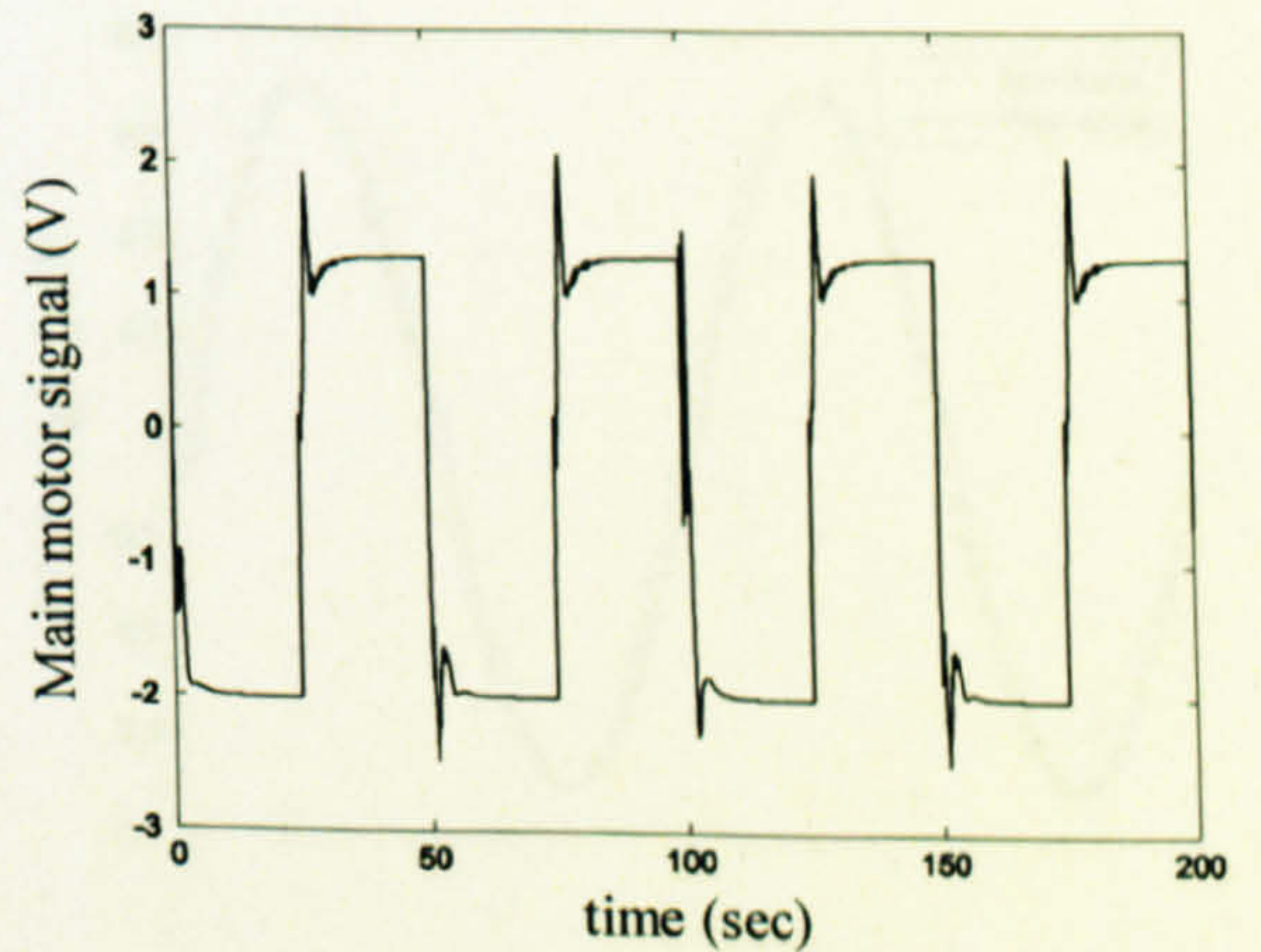


Figure 7.11 Main motor manipulated signal, U_v , (case 2)

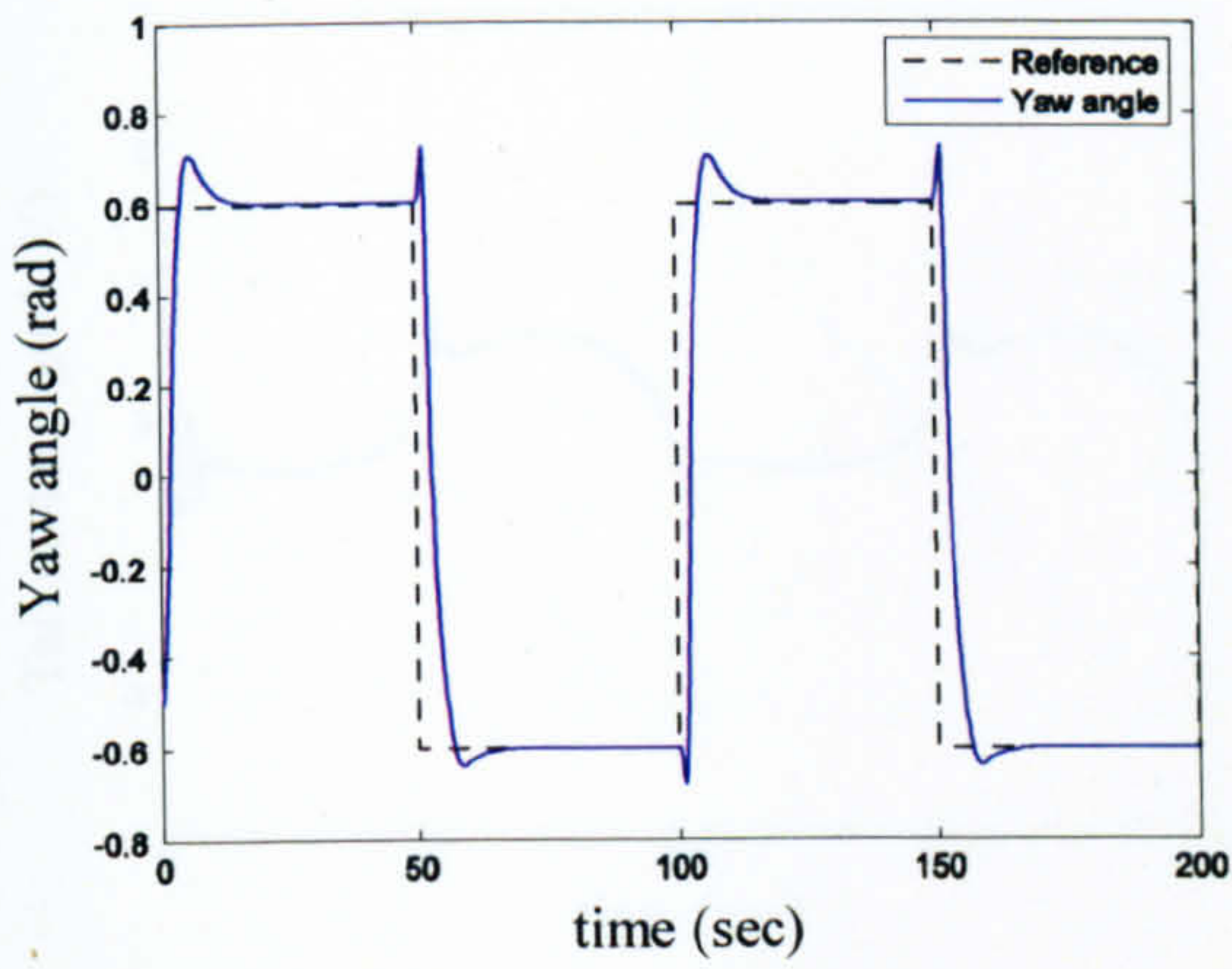


Figure 7.12 Square wave response of the horizontal angle (case 3)

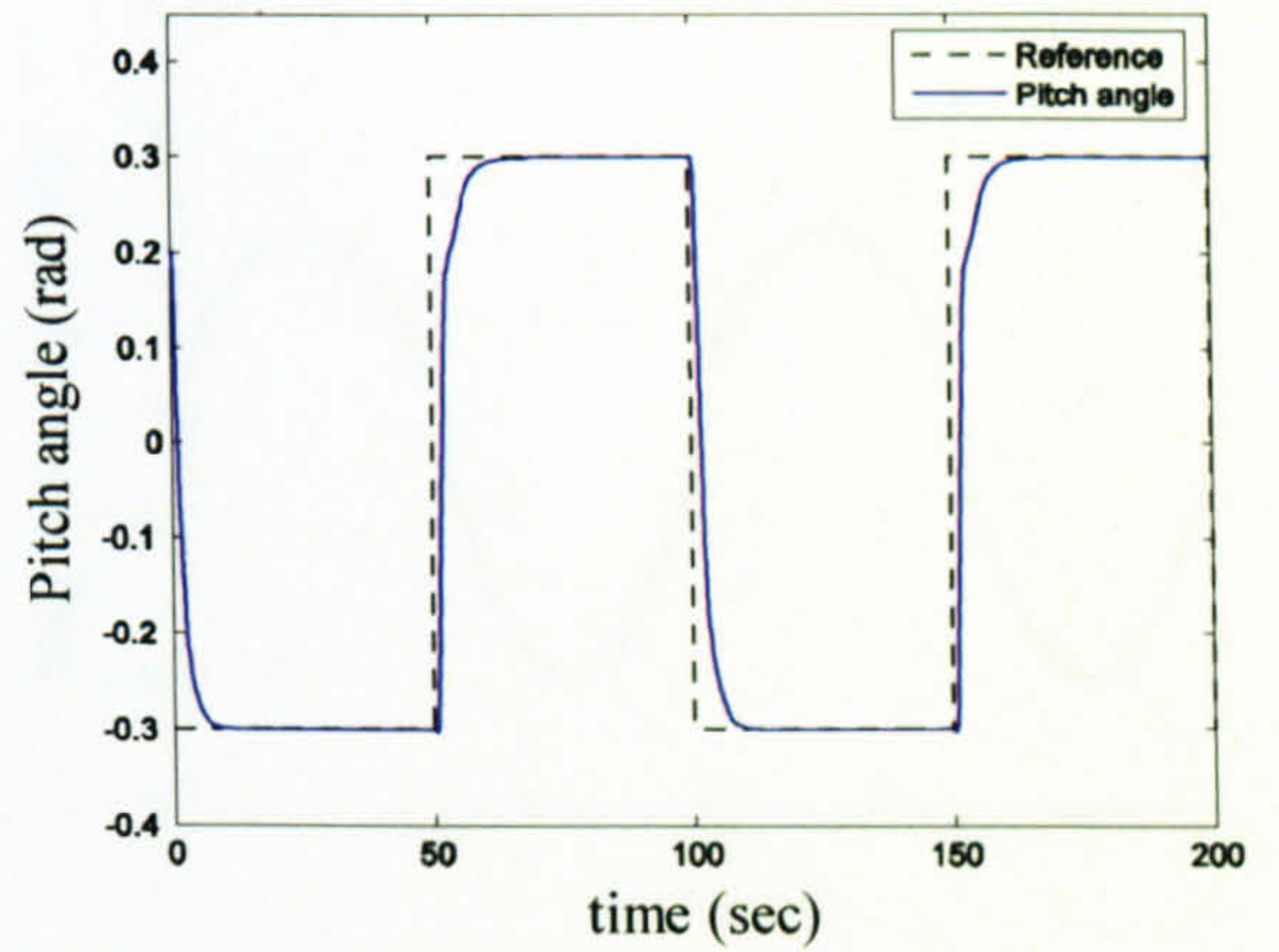


Figure 7.13 Square wave response of the vertical angle (case 3)

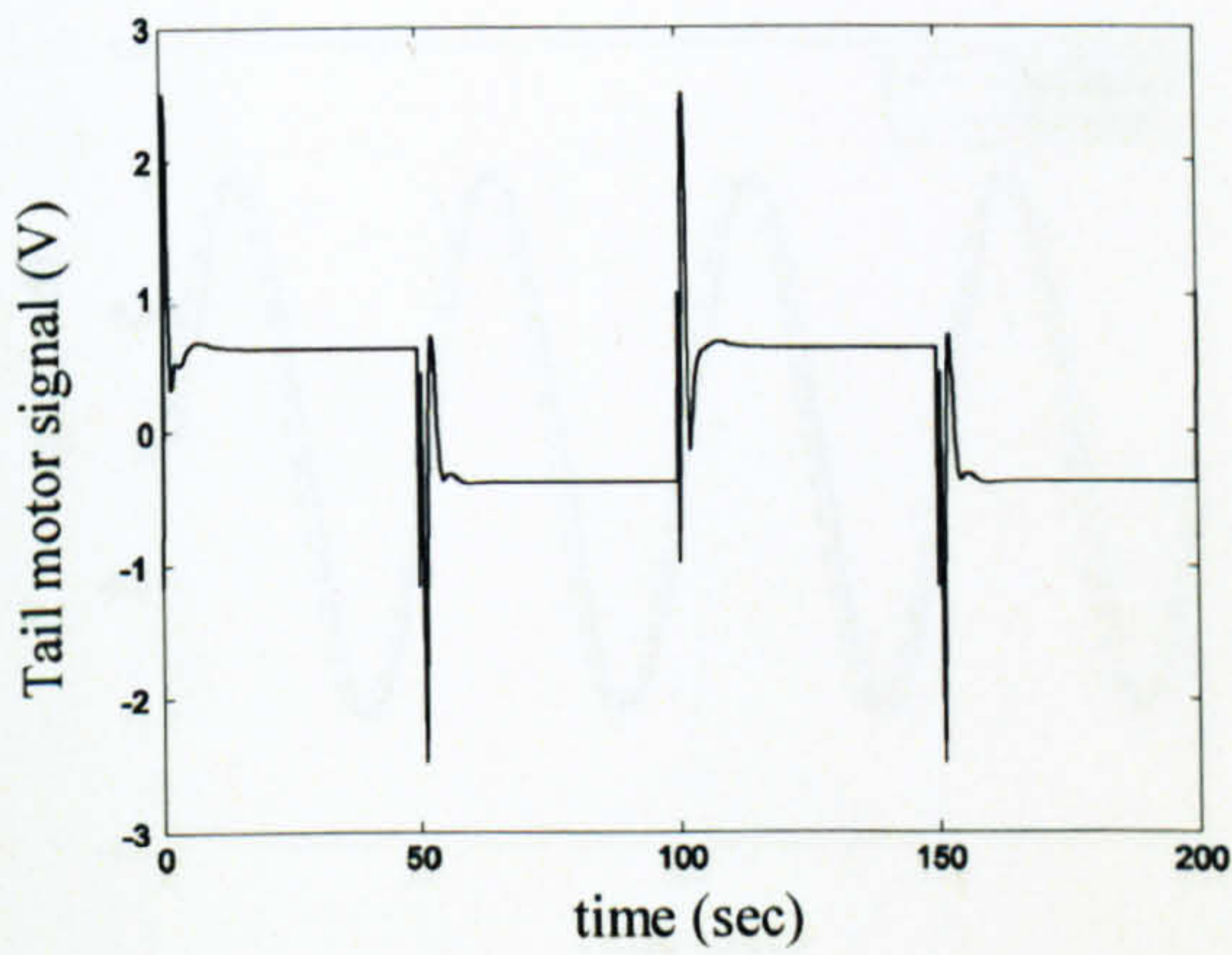


Figure 7.14 Tail motor manipulated signal, U_h , (case 3)

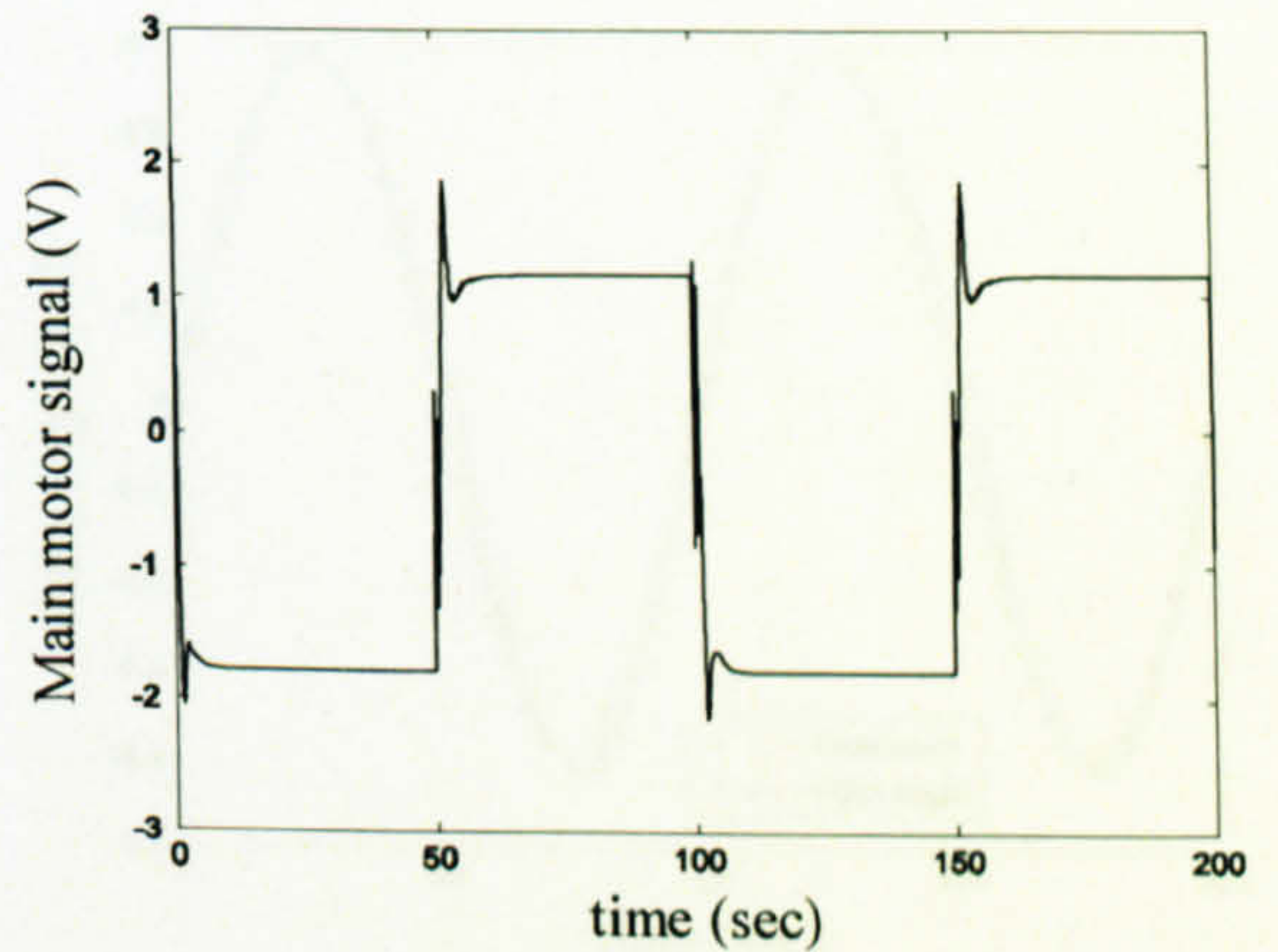


Figure 7.15 Main motor manipulated signal, U_v , (case 3)

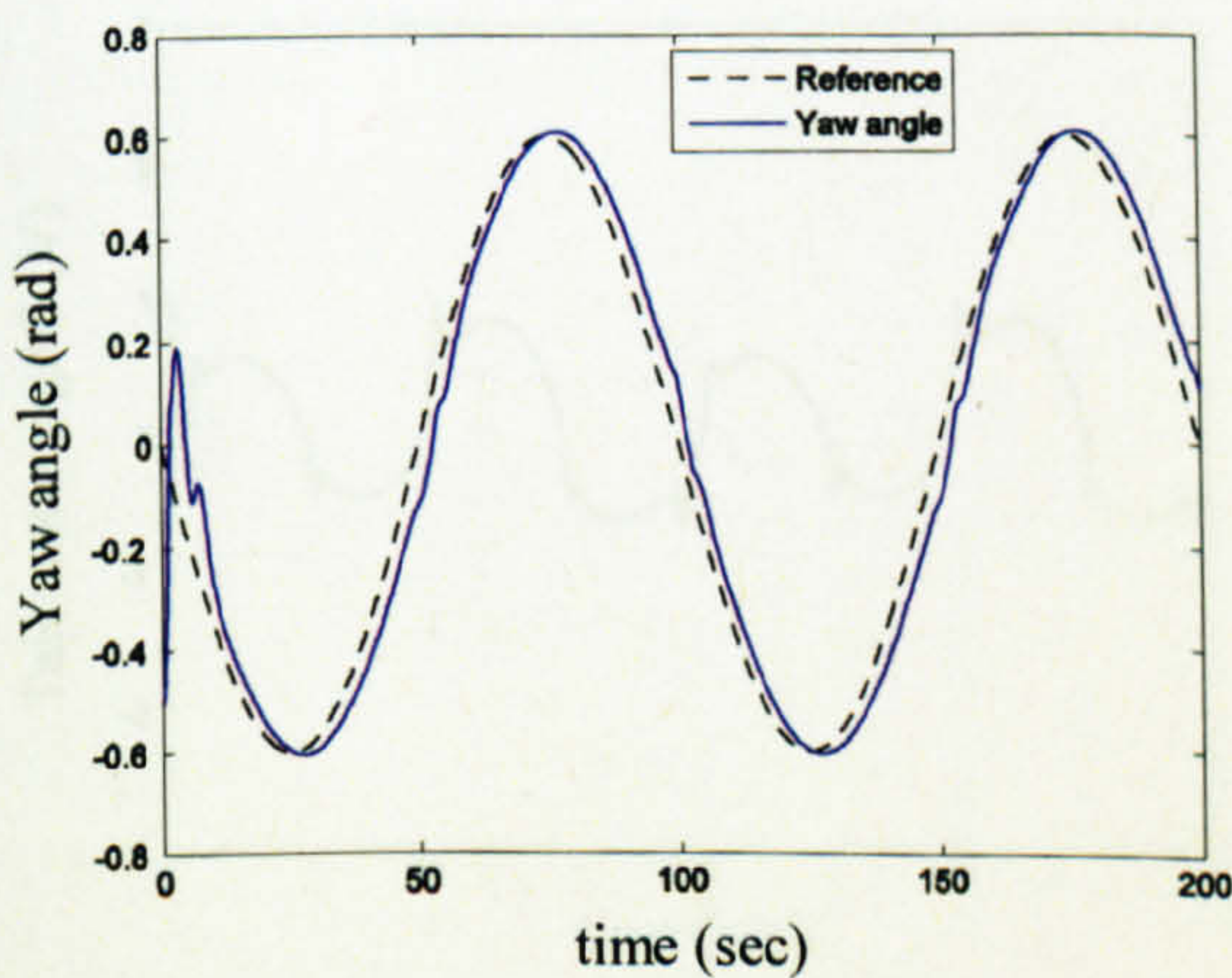


Figure 7.16 Sine wave response of the horizontal angle (case 4)

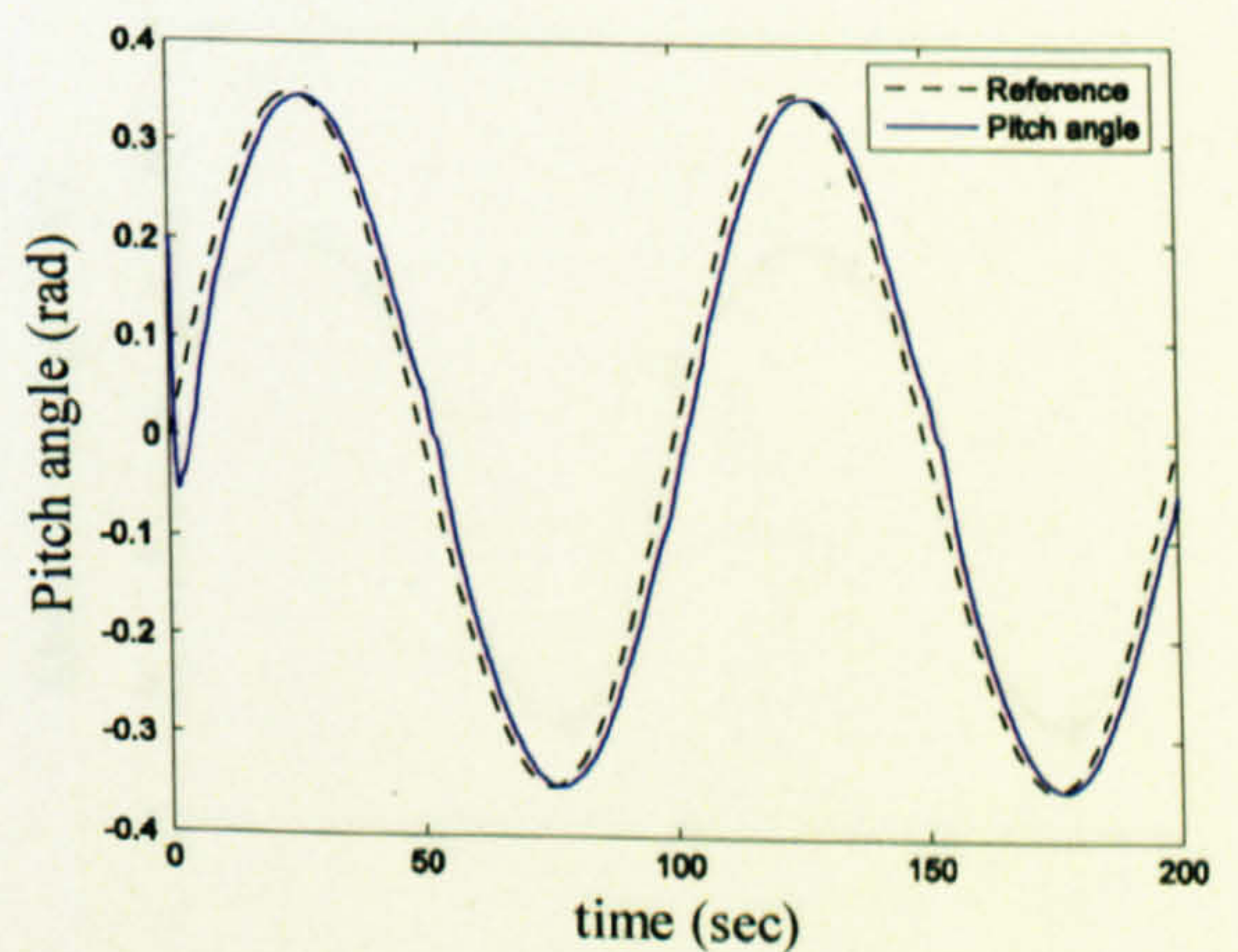


Figure 7.17 Sine wave response of the vertical angle (case 4)

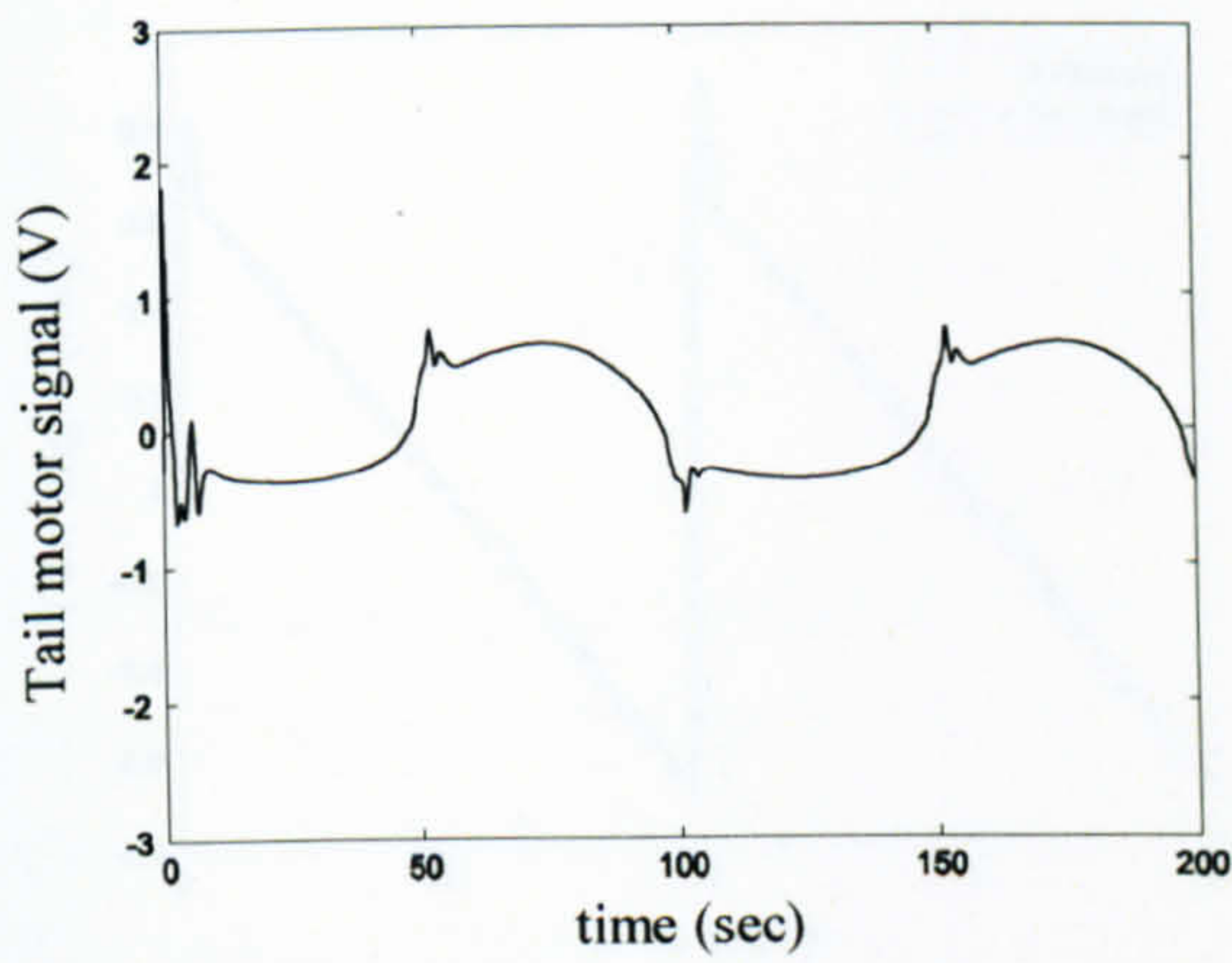


Figure 7.18 Tail motor manipulated signal, U_h , (case 4)

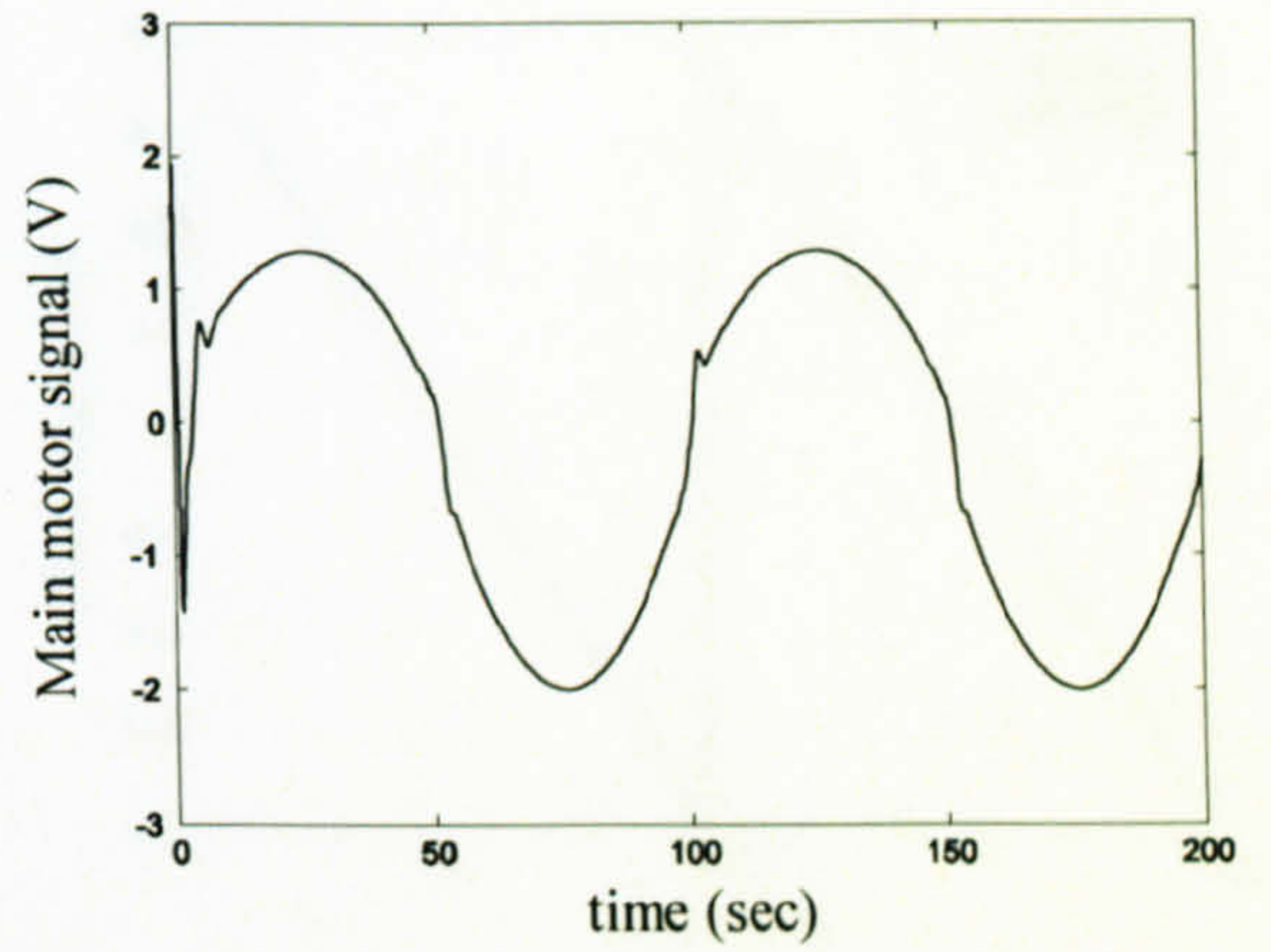


Figure 7.19 Main motor manipulated signal, U_v , (case 4)

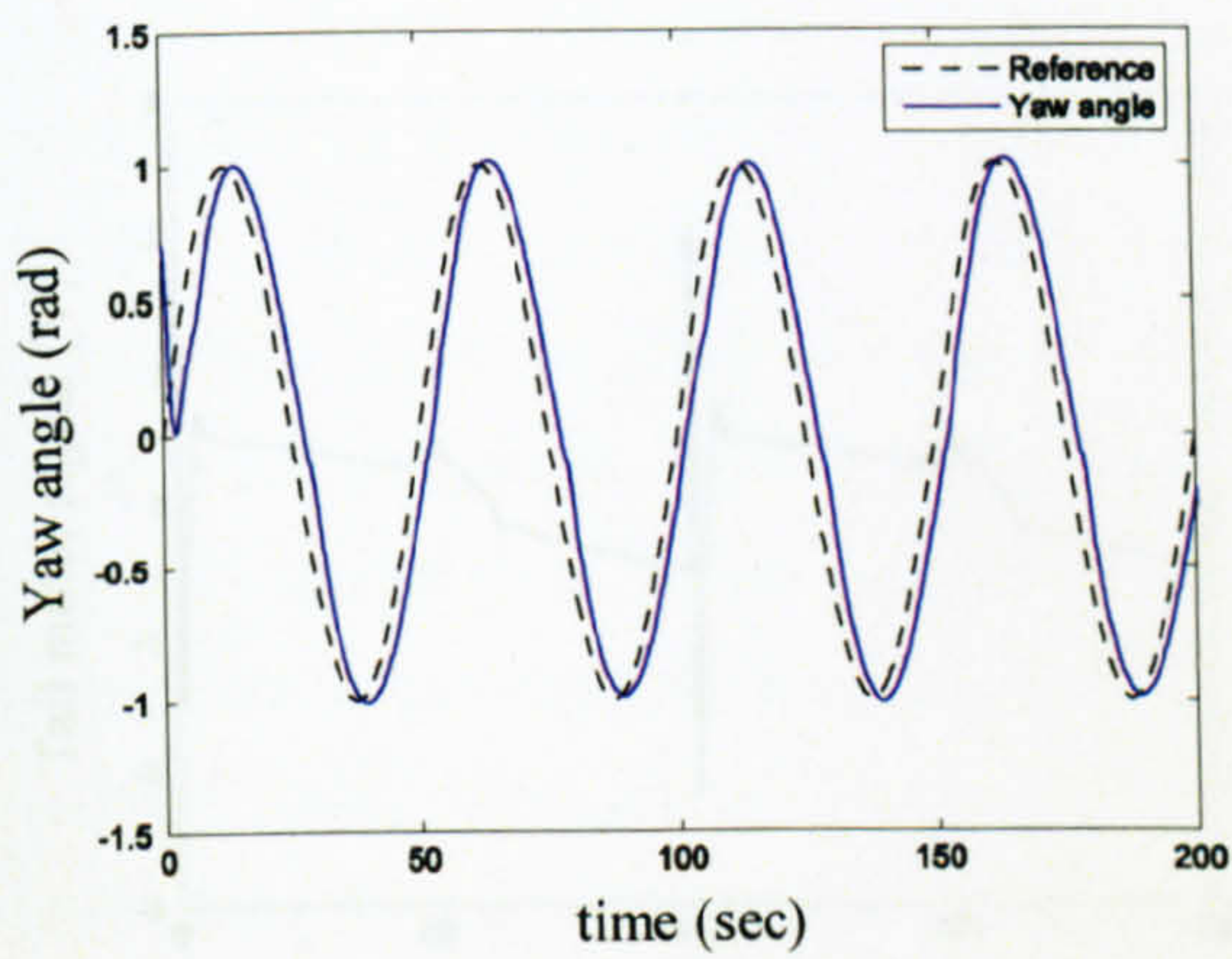


Figure 7.20 Sine wave response of the horizontal angle (case 5)

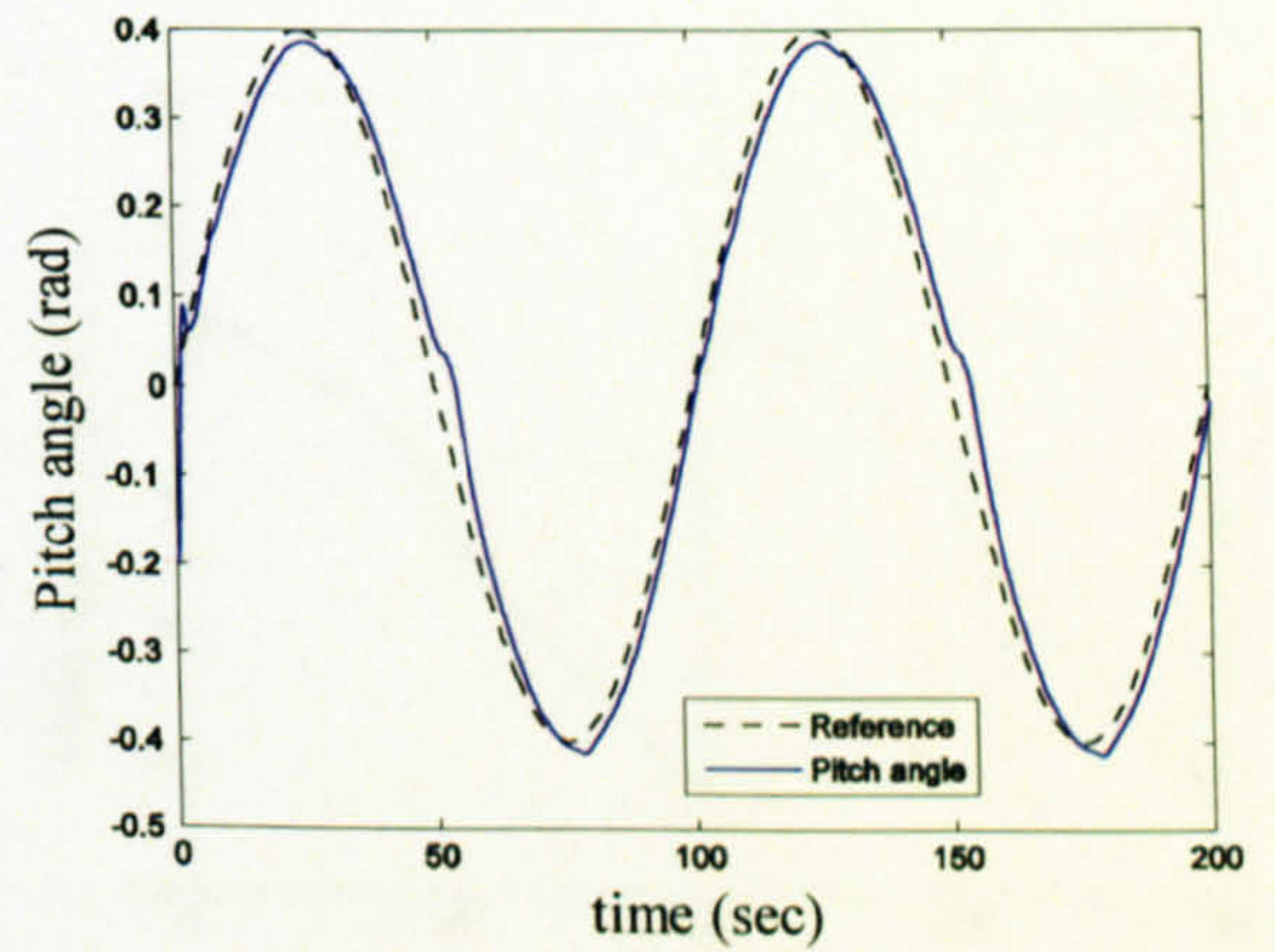


Figure 7.21 Sine wave response of the vertical angle (case 5)

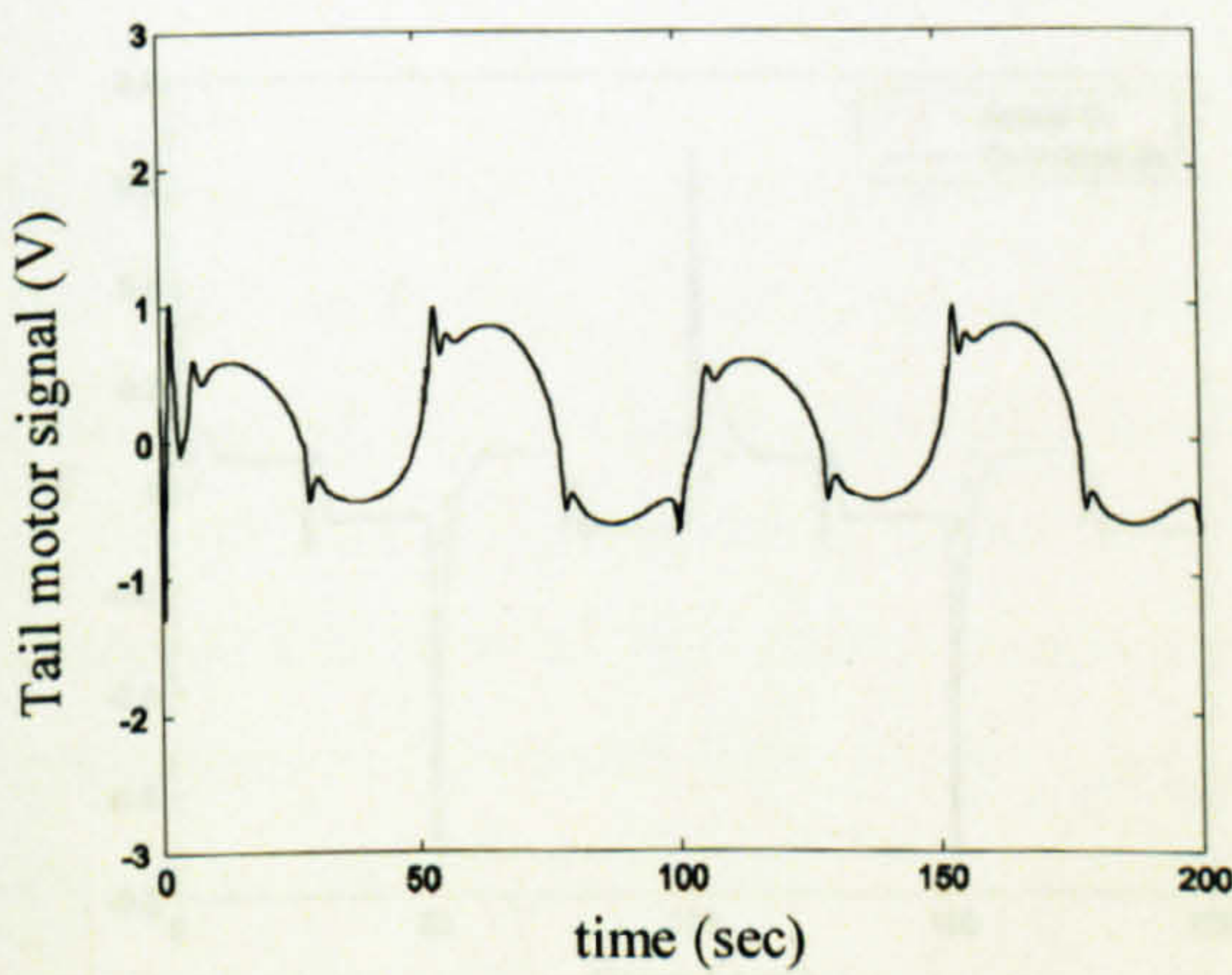


Figure 7.22 Tail motor manipulated signal, U_h , (case 5)

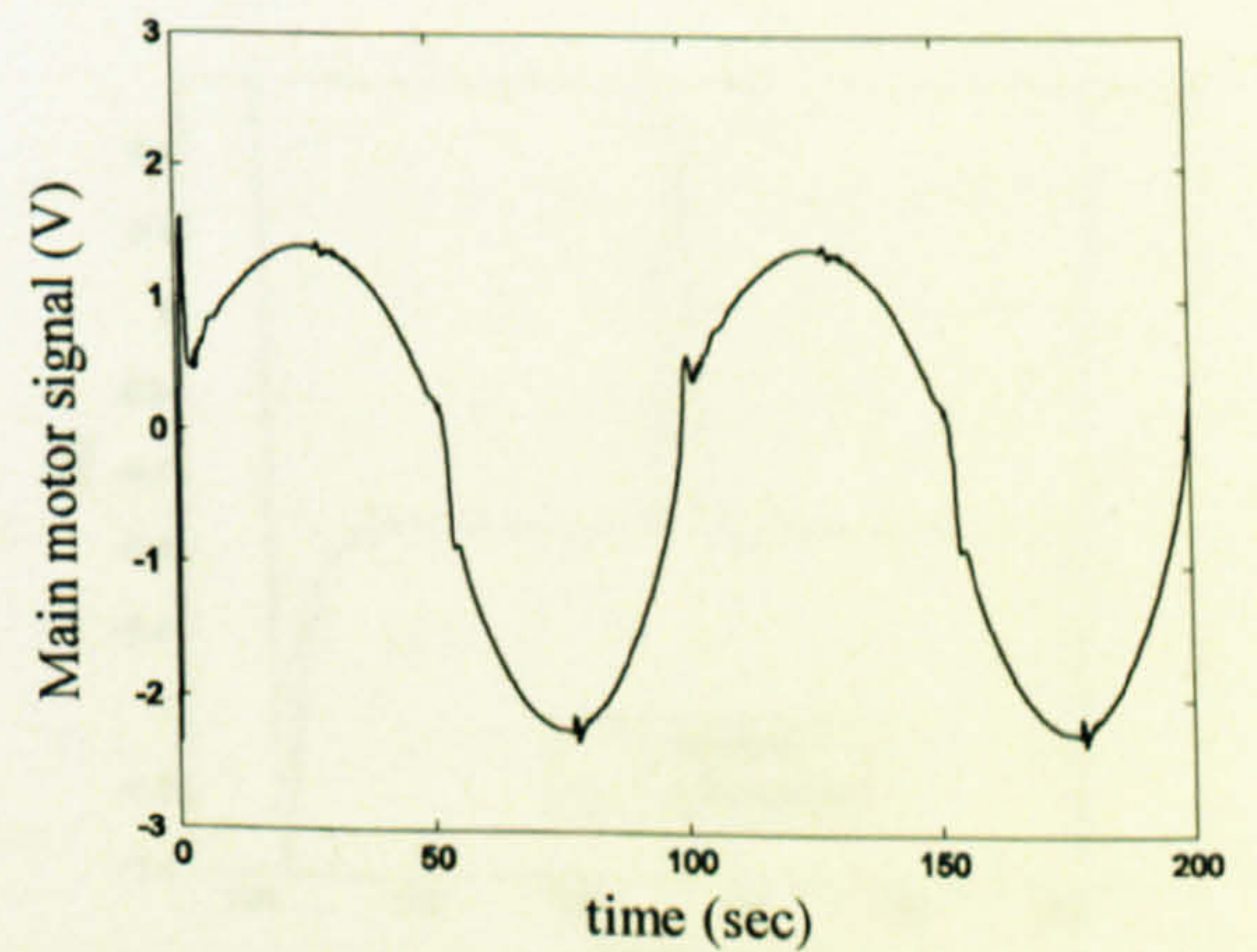


Figure 7.23 Main motor manipulated signal, U_v , (case 5)

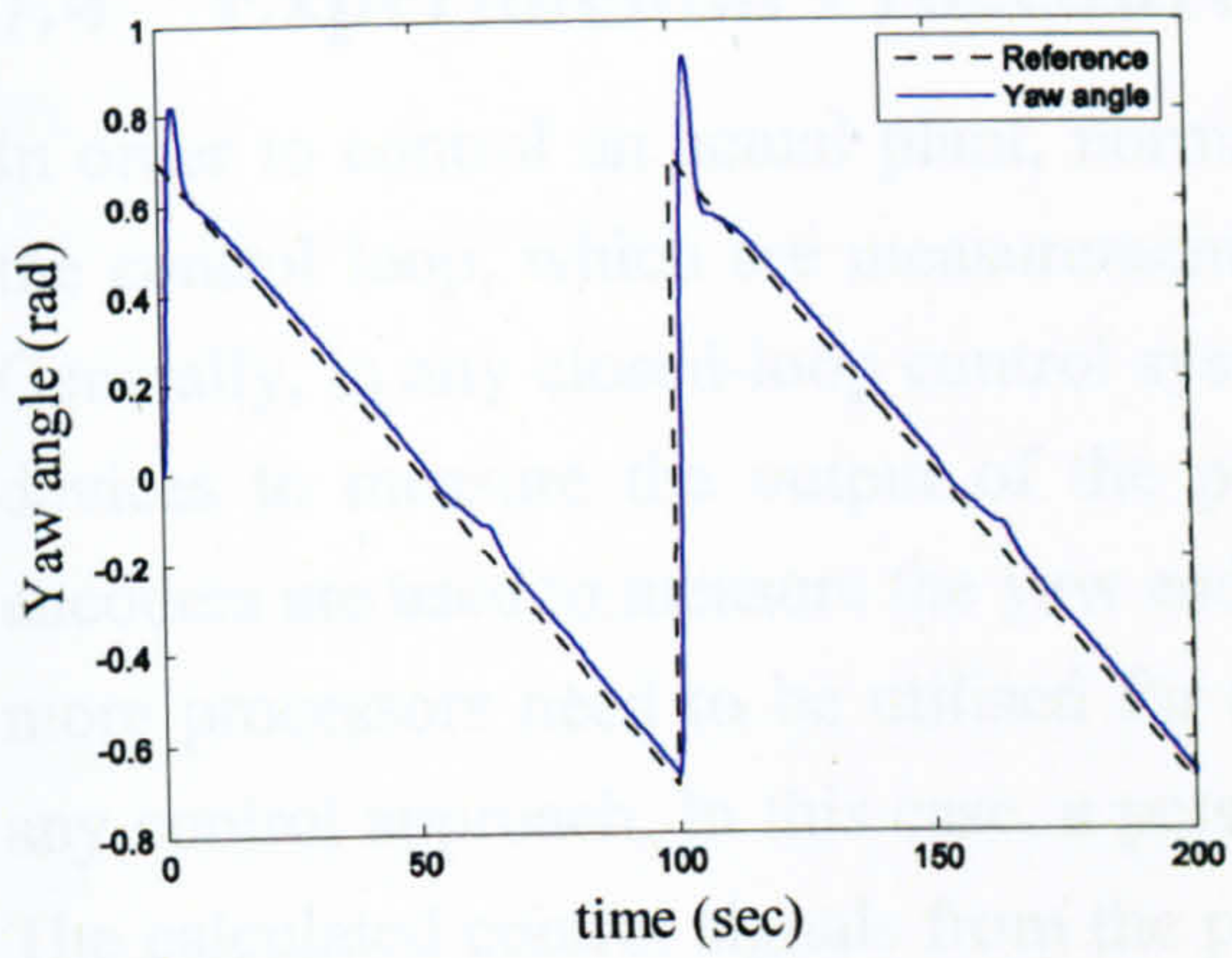


Figure 7.24 Sawtooth wave response of the horizontal angle (case 6)

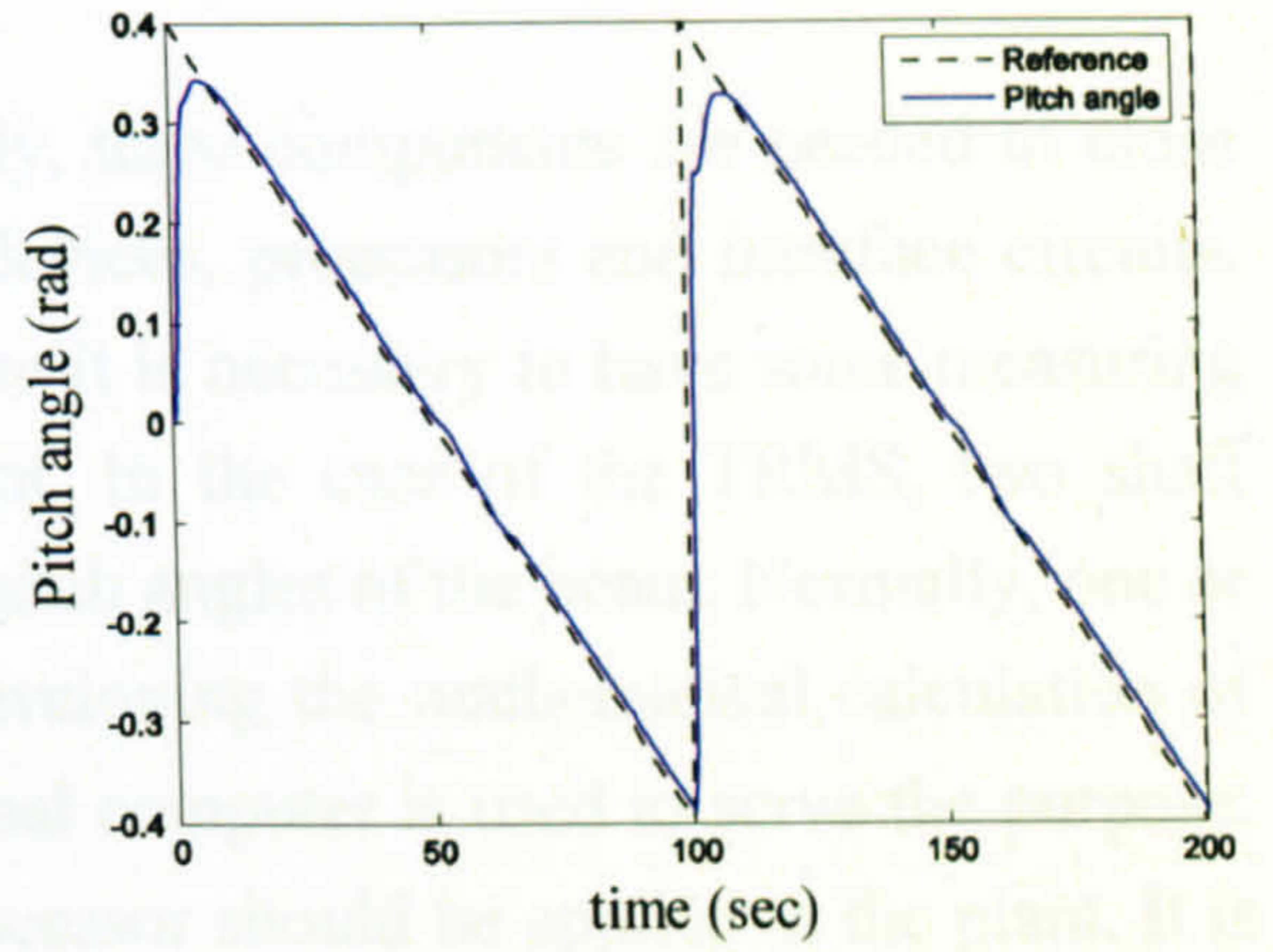


Figure 7.25 Sawtooth wave response of the vertical angle (case 6)

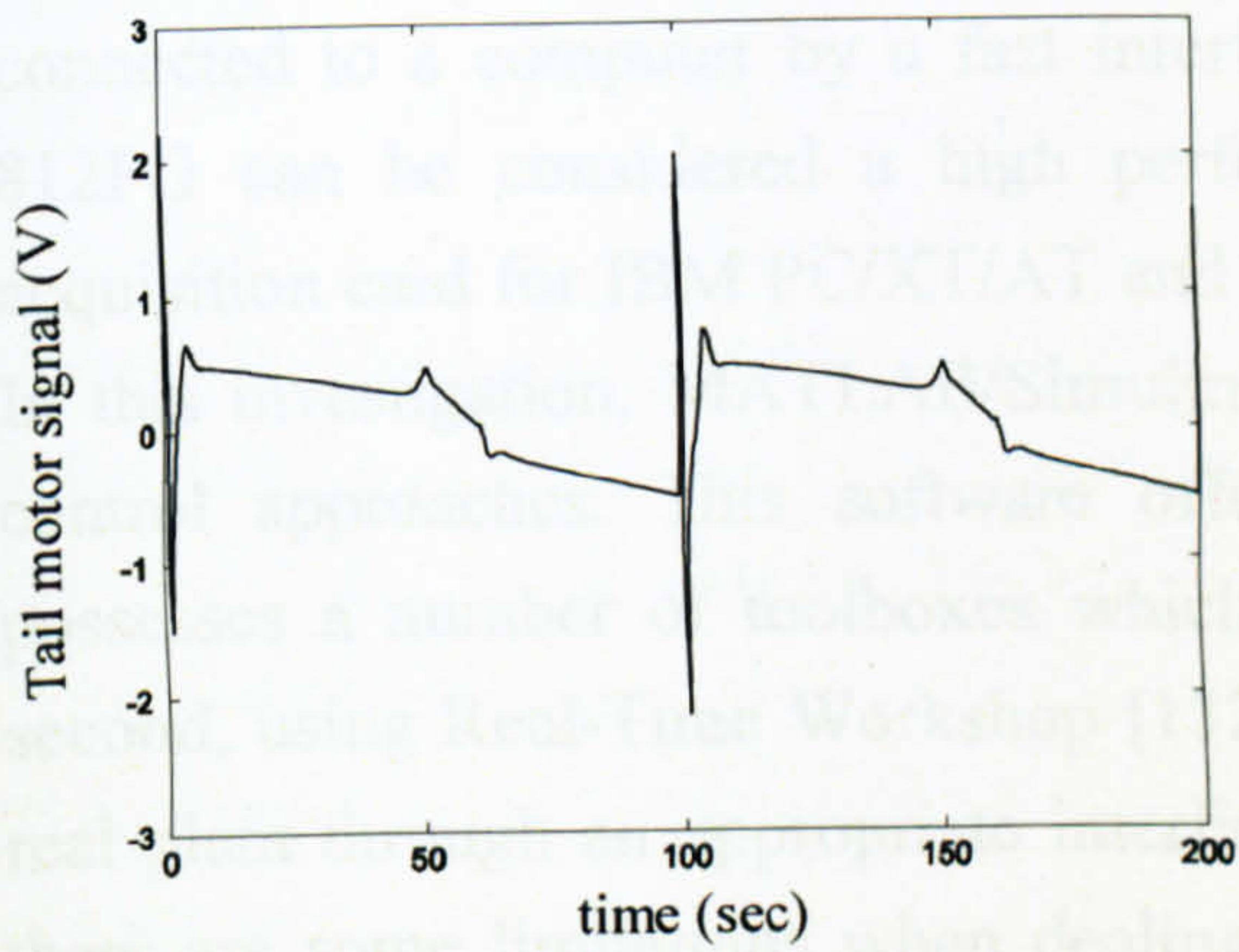


Figure 7.26 Tail motor manipulated signal, U_h , (case 6)

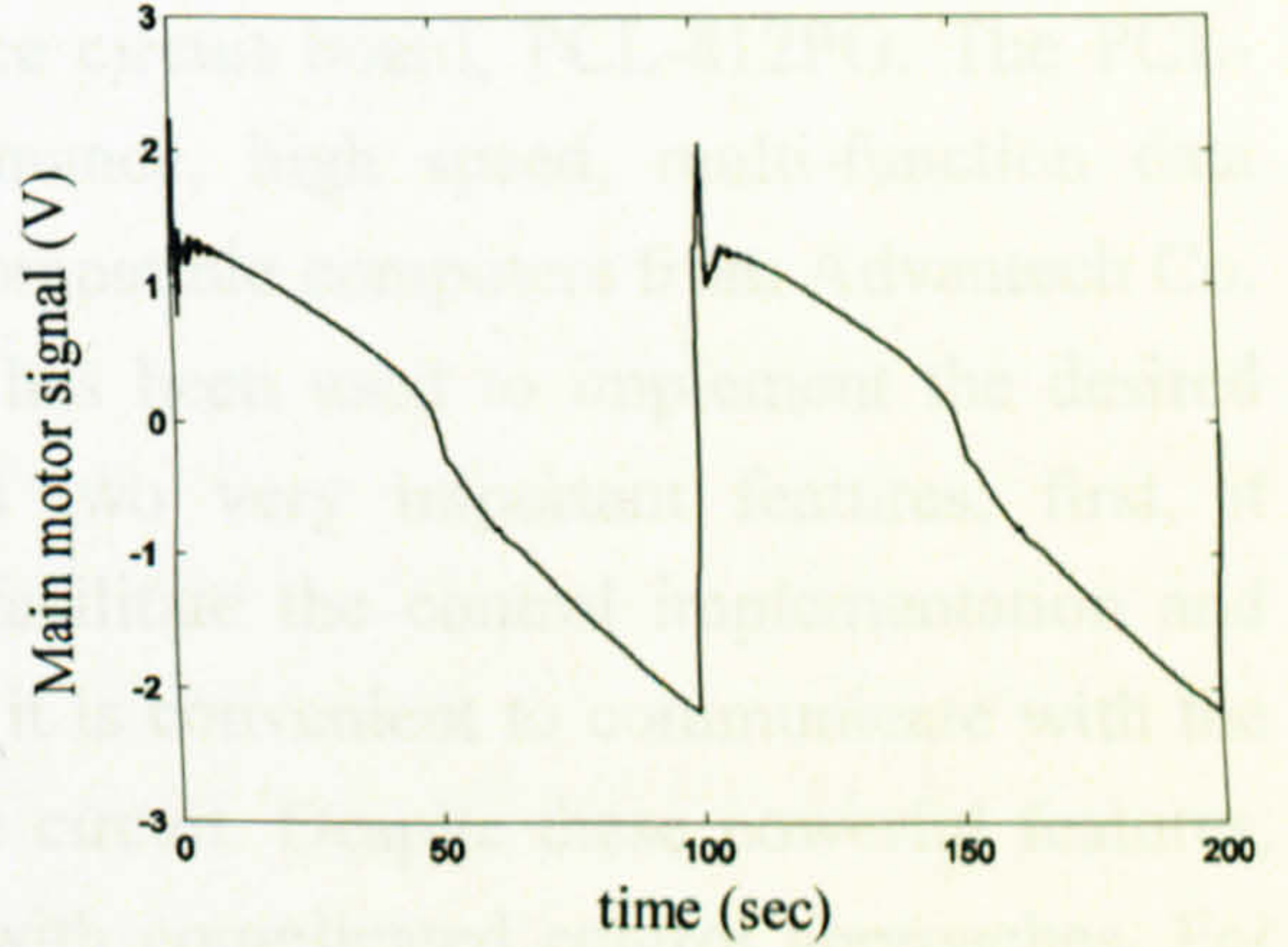


Figure 7.27 Main motor manipulated signal, U_v , (case 6)

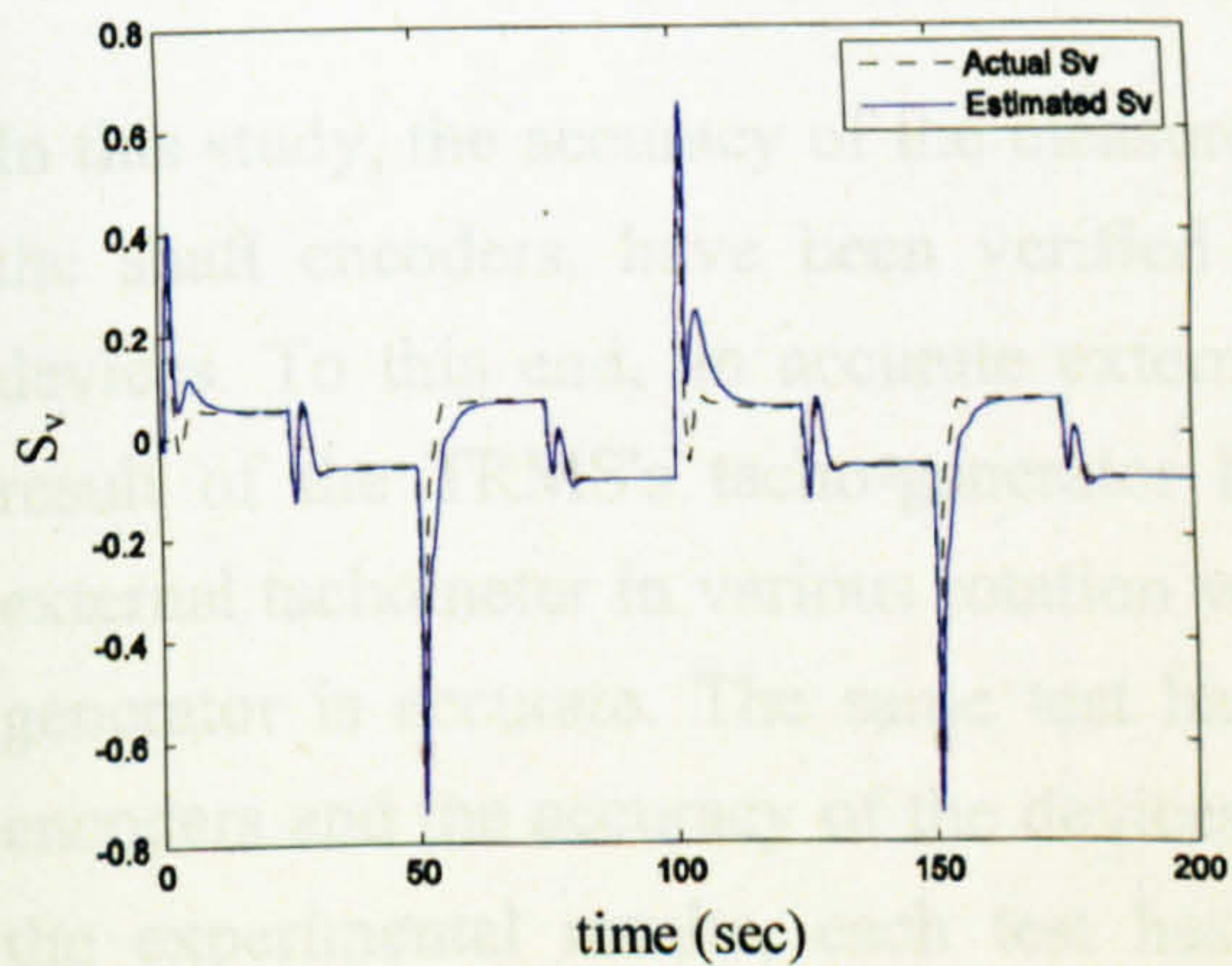


Figure 7.28 The performance of the UKF (S_v)

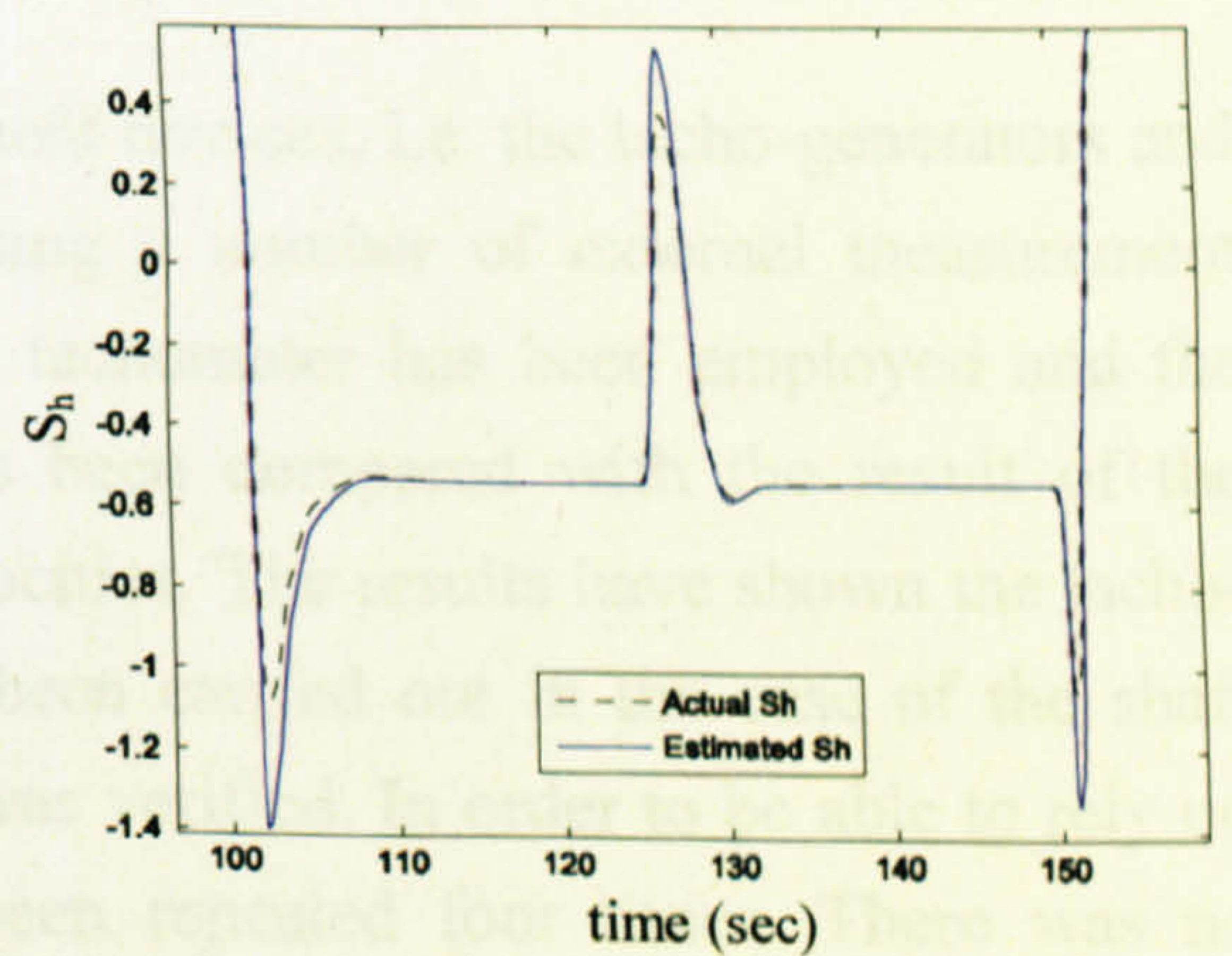


Figure 7.29 The performance of the UKF (S_h)

7.4 Experimental Procedure

In order to control an actual plant, normally, three components are needed to close the control loop, which are measurement devices, processors and interface circuits. Generally, in any closed-loop control system it is necessary to have some measuring devices to measure the output of the plant. In the case of the TRMS, two shaft encoders are used to measure the yaw and pitch angles of the beam. Normally, one or more processors need to be utilised for developing the mathematical calculation of any control approach. In this case, a personal computer is used to serve the purpose. The calculated control signals from the processor should be applied to the plant. It is of great importance to note that the signals from the measurement devices are not normally compatible with the processor and also the control signals from the processor are not in an appropriate level to be directly sent to the plant. Therefore, an interface circuit is necessary to perform these signal adaptations. The real TRMS is connected to a computer by a fast interface circuit board, PCL-812PG. The PCL-812PG can be considered a high performance, high speed, multi-function data acquisition card for IBM PC/XT/AT and compatible computers from Advantech Co. In this investigation, MATLAB/Simulink has been used to implement the desired control approaches. This software offers two very important features; first, it possesses a number of toolboxes which facilitate the control implementation and second, using Real-Time Workshop [132] it is convenient to communicate with the real plant through an appropriate interface circuit. Despite these powerful features, there are some limitations when dealing with complicated control approaches. For instance, Real-Time Workshop cannot handle a Simulink file containing *MATLAB-Fcn* block. Instead of this block, the *S-Function* or *Embedded-MATLAB-Function* tools can be employed which have their own drawbacks.

In this study, the accuracy of the measurement devices, i.e. the tacho-generators and the shaft encoders, have been verified using a number of external measurement devices. To this end, an accurate external tachometer has been employed and the result of the TRMS's tacho-generator has been compared with the result of the external tachometer in various rotation velocities. The results have shown the tacho-generator is accurate. The same test has been carried out in the case of the shaft encoders and the accuracy of the devices was verified. In order to be able to rely on the experimental results, each test has been repeated four times. There was no significant difference observed between the repetitions. The sampling time of the control system is set to be 0.2 seconds.

In the following sections the experimental results of the UKF and the state-dependent EKF are presented and compared, and then the experimental results of the output-feedback MPC are presented.

7.4.1 State Estimator

State variables of the TRMS are estimated using the UKF and the state-dependent EKF. To this end, the open loop system is excited with various input signals and the state variables are estimated and compared with those states which can be measured. Using the measurement devices, one can only measure the yaw and pitch angles and the rotational speed of both the motors; the other two states cannot be measured and therefore cannot be verified. Note that the rotational speed of the motors cannot be measured simultaneously due to the limited number of available ports in the case of the TRMS system and therefore, cannot be used in the control system.

The overall block diagram of the Kalman filters test is depicted in Figure 7.30. The state-dependent EKF has been implemented using Simulink toolbox, and its block diagram can be found in Figure 7.31. 'Predict states' and 'State dependent linearisation' blocks in Figure 7.31 can be formed using the nonlinear and state-dependent models of the TRMS respectively and for the sake of brevity, they are not shown here.

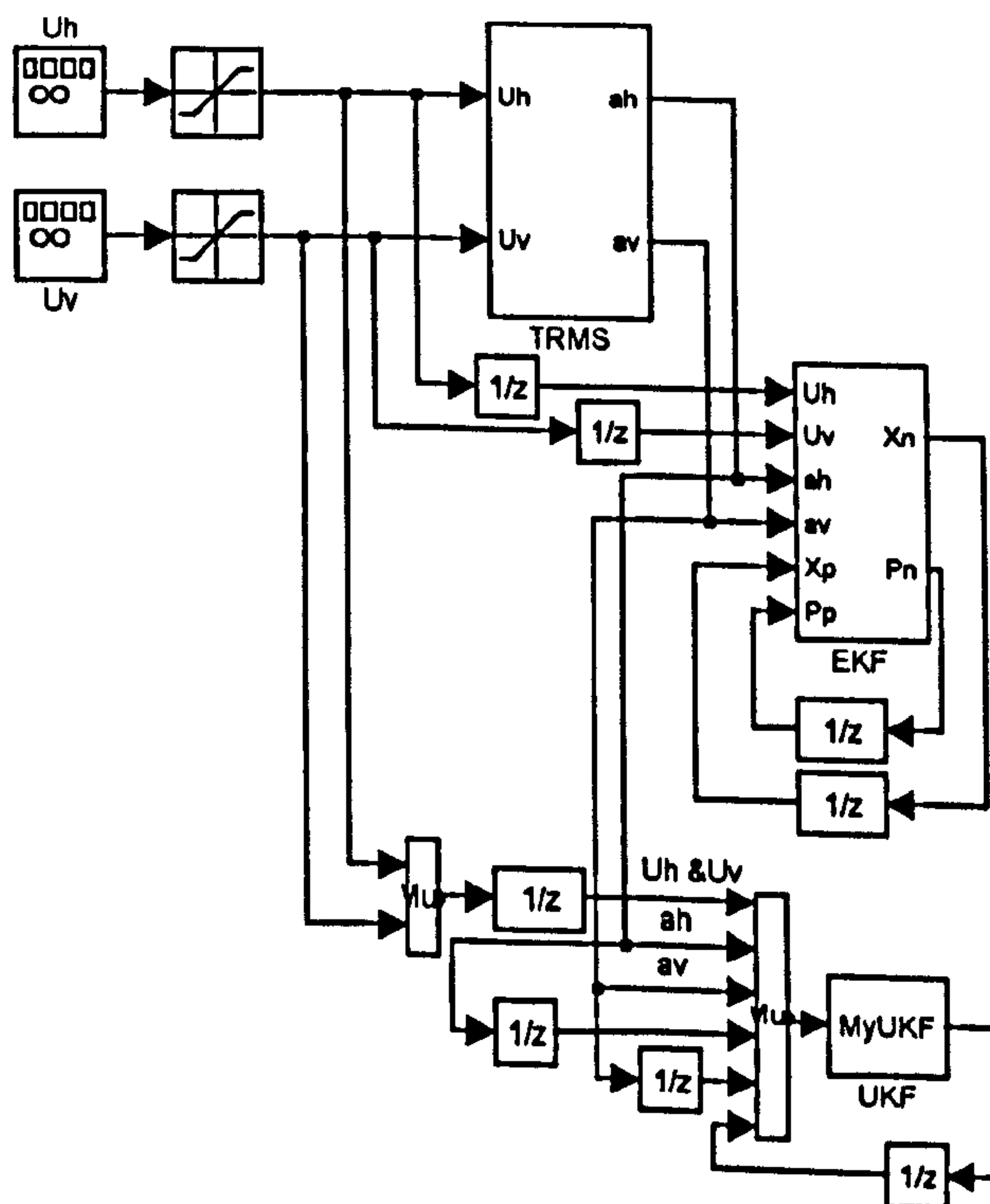


Figure 7.30 The overall block diagram of the UKF and the EKF test

The UKF has been implemented using *S-Function* and its corresponding *Embedded-MATLAB-Function* can be seen in Appendix 5. Note that it is also possible to implement the state-dependent EKF using *Embedded-MATLAB-Function* but due to its comparatively less complexity it is more convenient to use Simulink toolbox. The parameters of the filters are listed in Appendix 1. The results of the filters are illustrated in Figures 7.32 to 7.51 for two cases. In Figures 7.32 and 7.33, the measured yaw angle has been compared with the results of the state-dependent EKF and the UKF respectively. In the case of the pitch angle, one can see the results of the EKF and the UKF estimators in Figures 7.34 and 7.35. The comparative results of the measured rotational speed of the main and the tail rotors with their corresponding estimated values using the EKF and the UKF can be found in Figures 7.36 to 7.39. The other two states cannot be measured and therefore, the results of the estimators are compared with each other in terms of these two inaccessible states. Figures 7.40 and 7.41 show the results of the estimators for the inaccessible states. It is evident from the results that the estimators could estimate the state variables very satisfactorily. However, the estimation of the EKF based on the state-dependent state-space model is slightly better in comparison with that of the UKF.

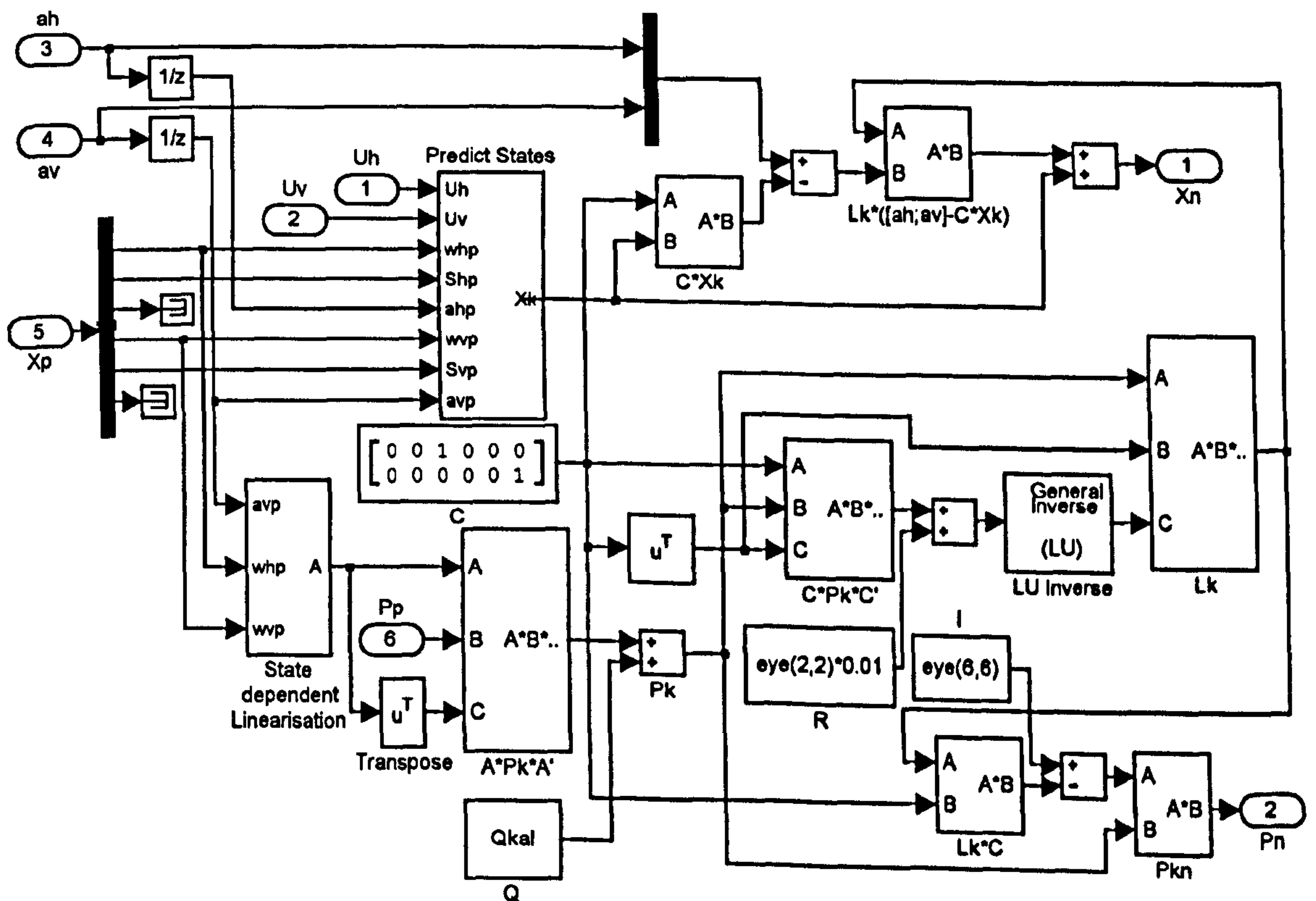


Figure 7.31. Inside the 'EKF' block

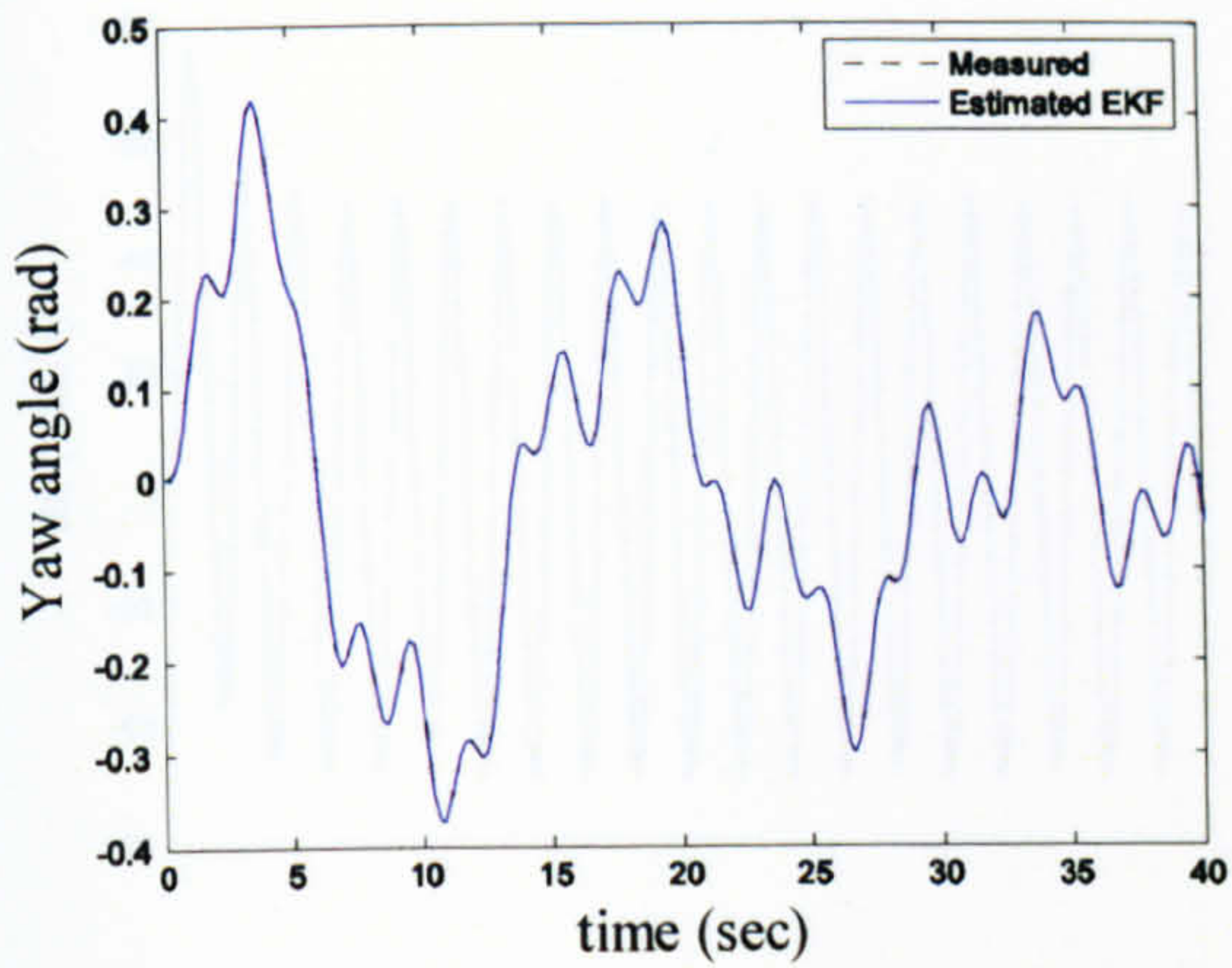


Figure 7.32 Measured and EKF-based estimated yaw angle (case 1)

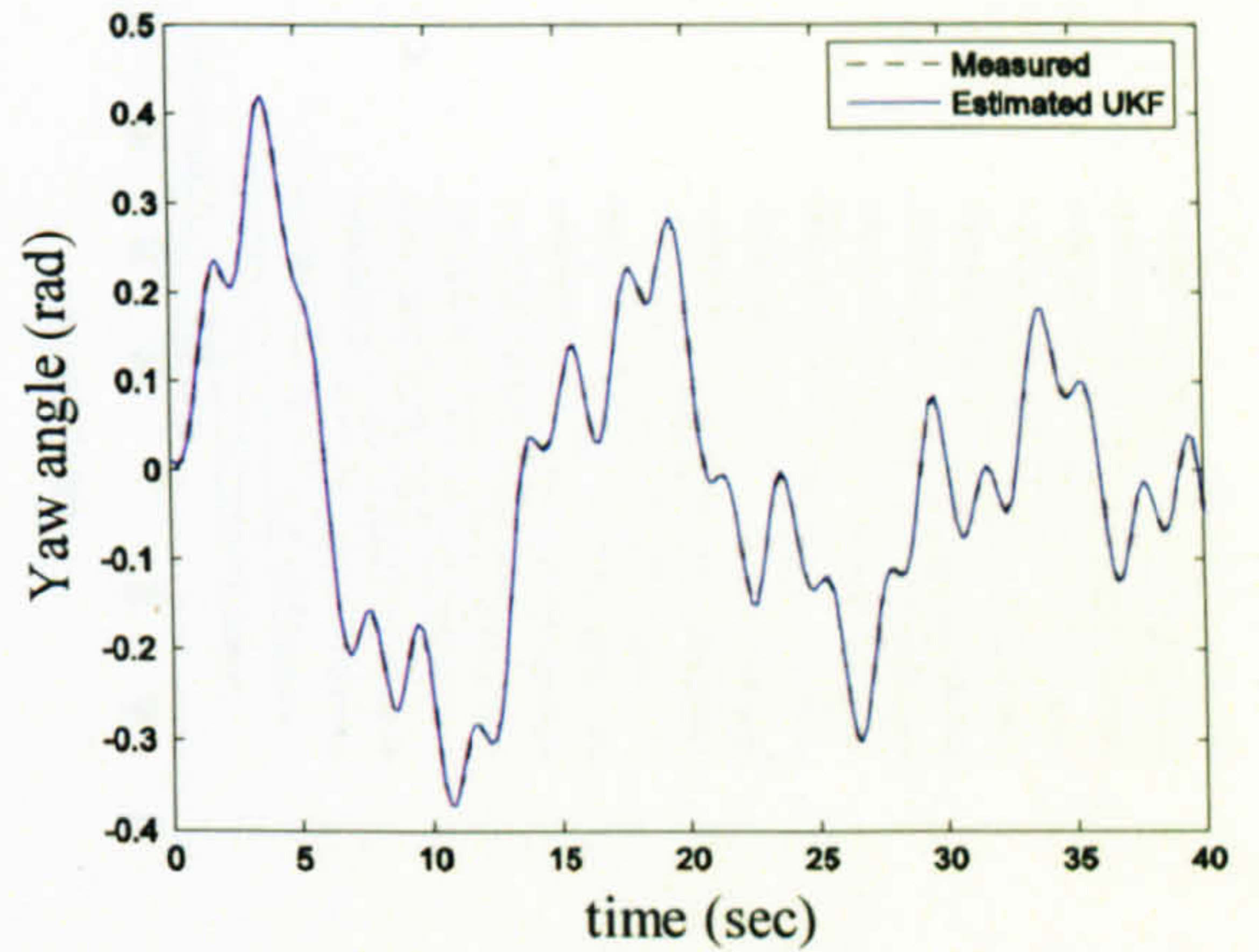


Figure 7.33 Measured and UKF-based estimated yaw angle (case 1)

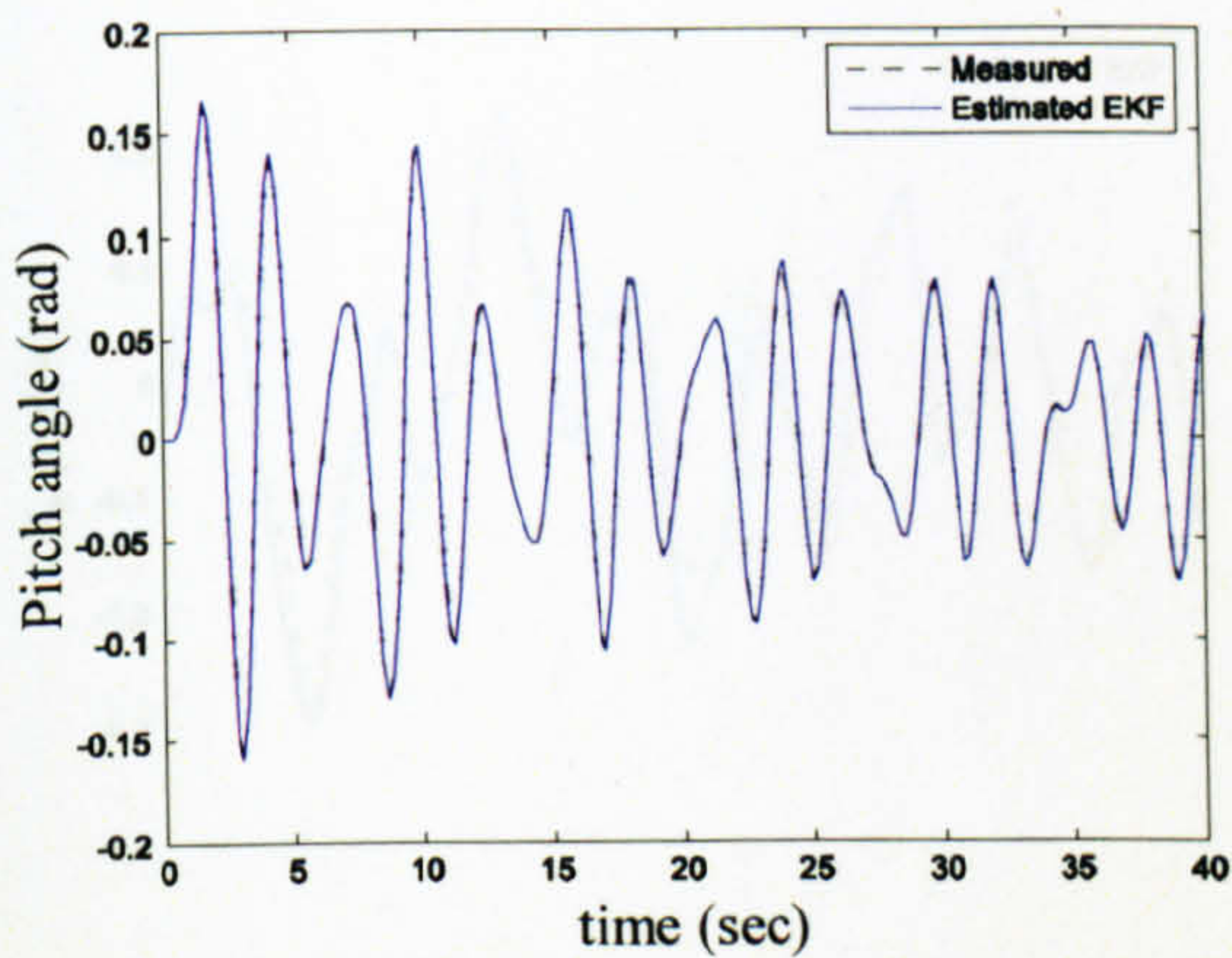


Figure 7.34 Measured and EKF-based estimated pitch angle (case 1)

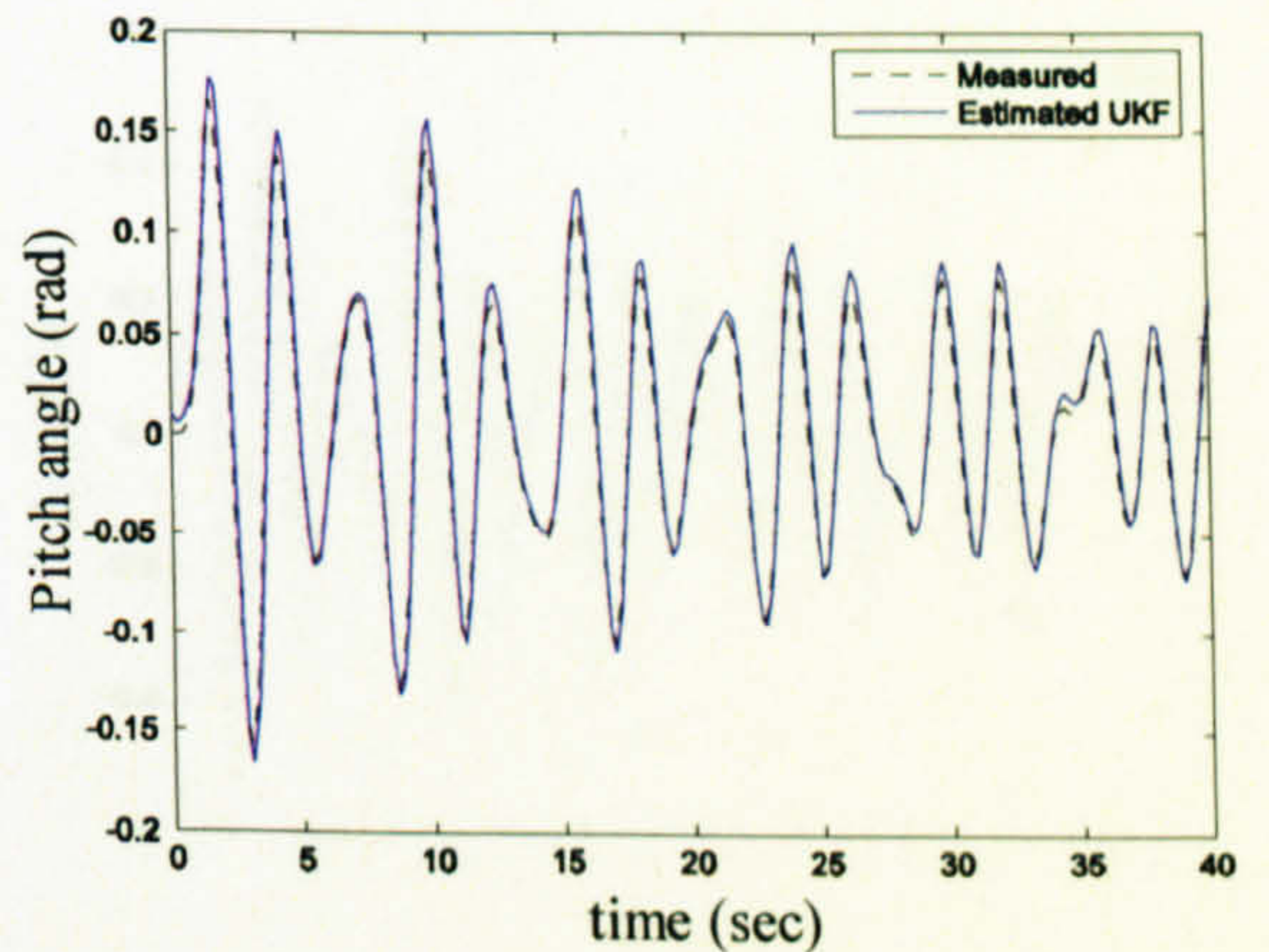


Figure 7.35 Measured and UKF-based estimated pitch angle (case 1)

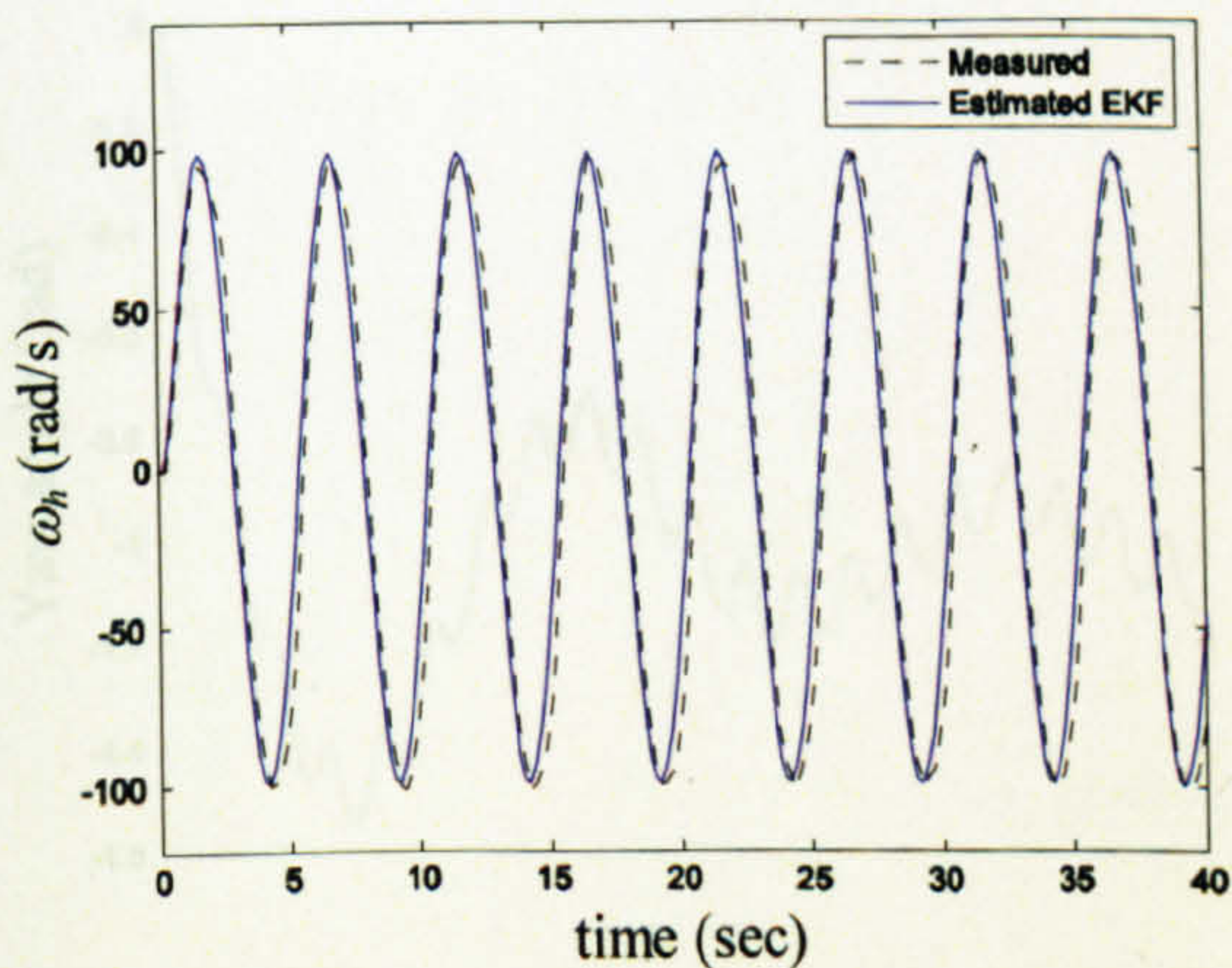


Figure 7.36 Measured and EKF-based estimated ω_h (case 1)

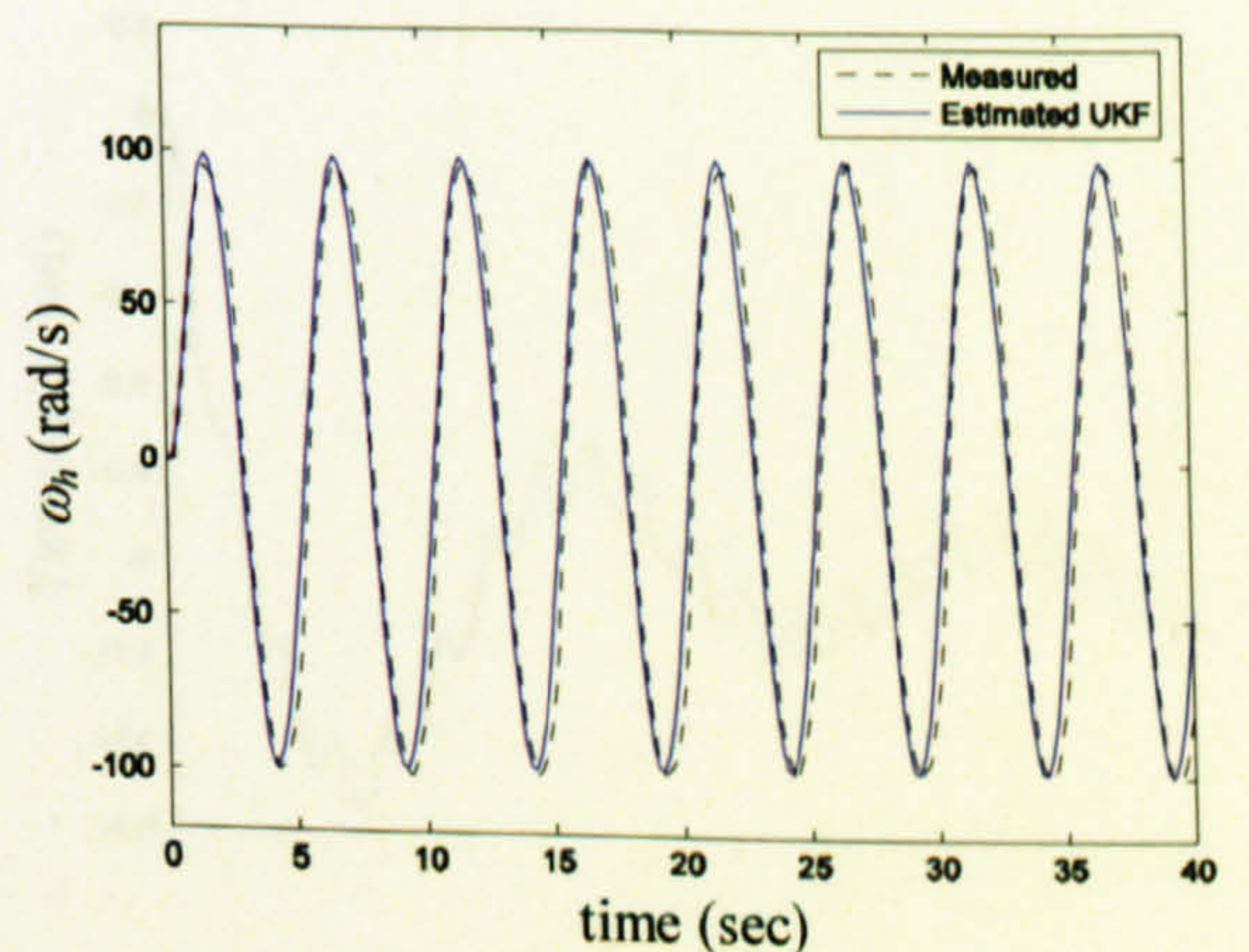


Figure 7.37 Measured and UKF-based estimated ω_h (case 1)

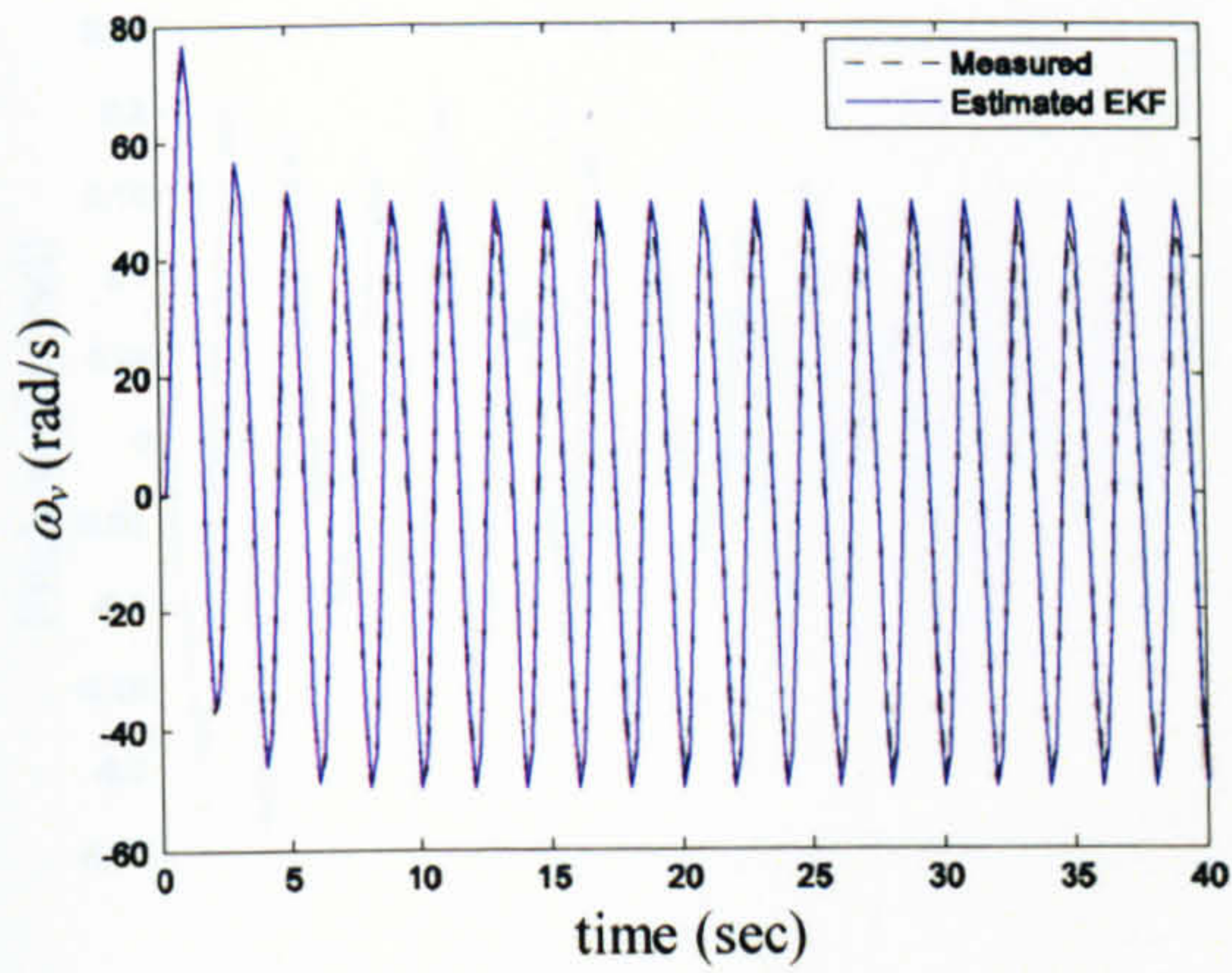


Figure 7.38 Measured and EKF-based estimated ω_v (case 1)

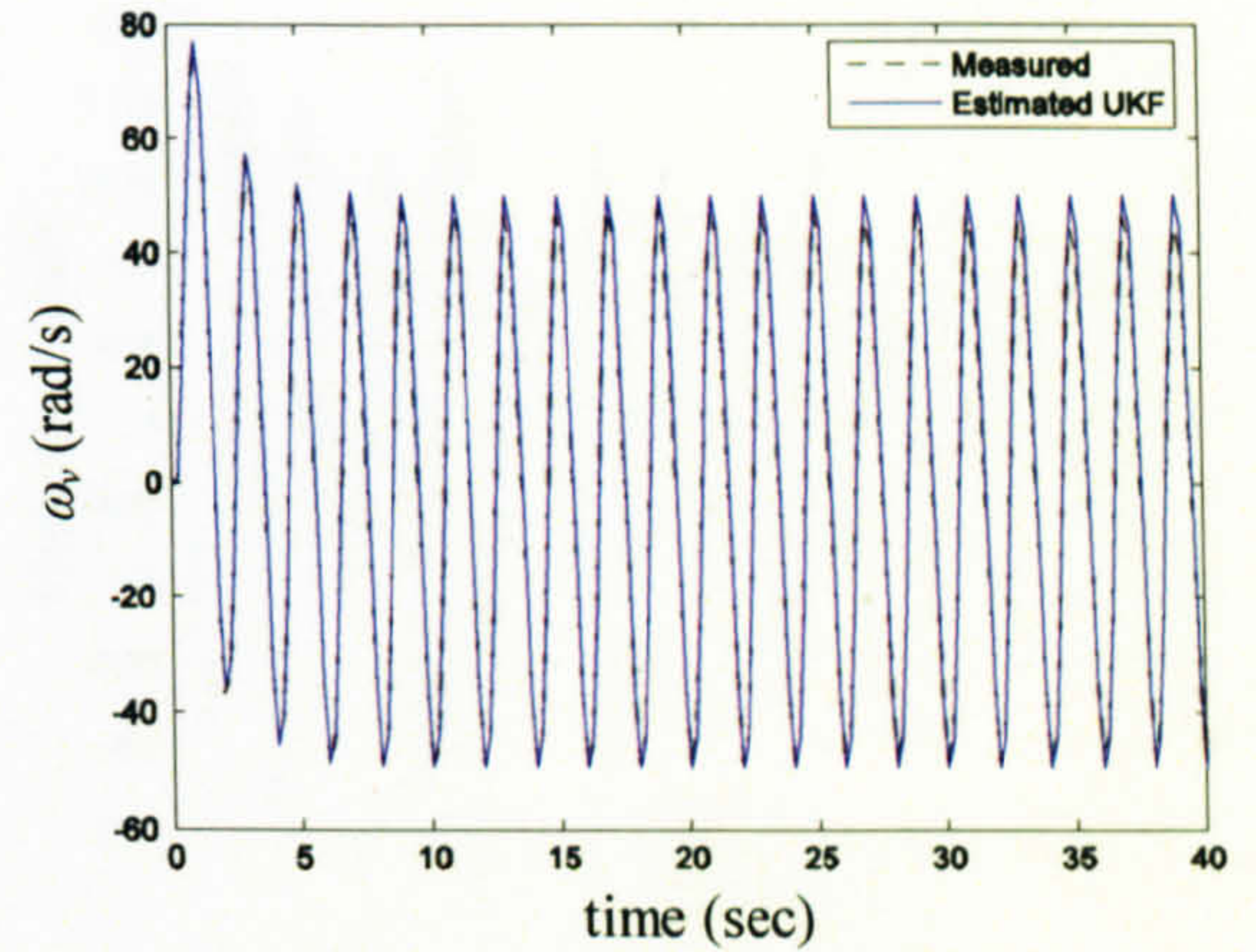


Figure 7.39 Measured and UKF-based estimated ω_v (case 1)

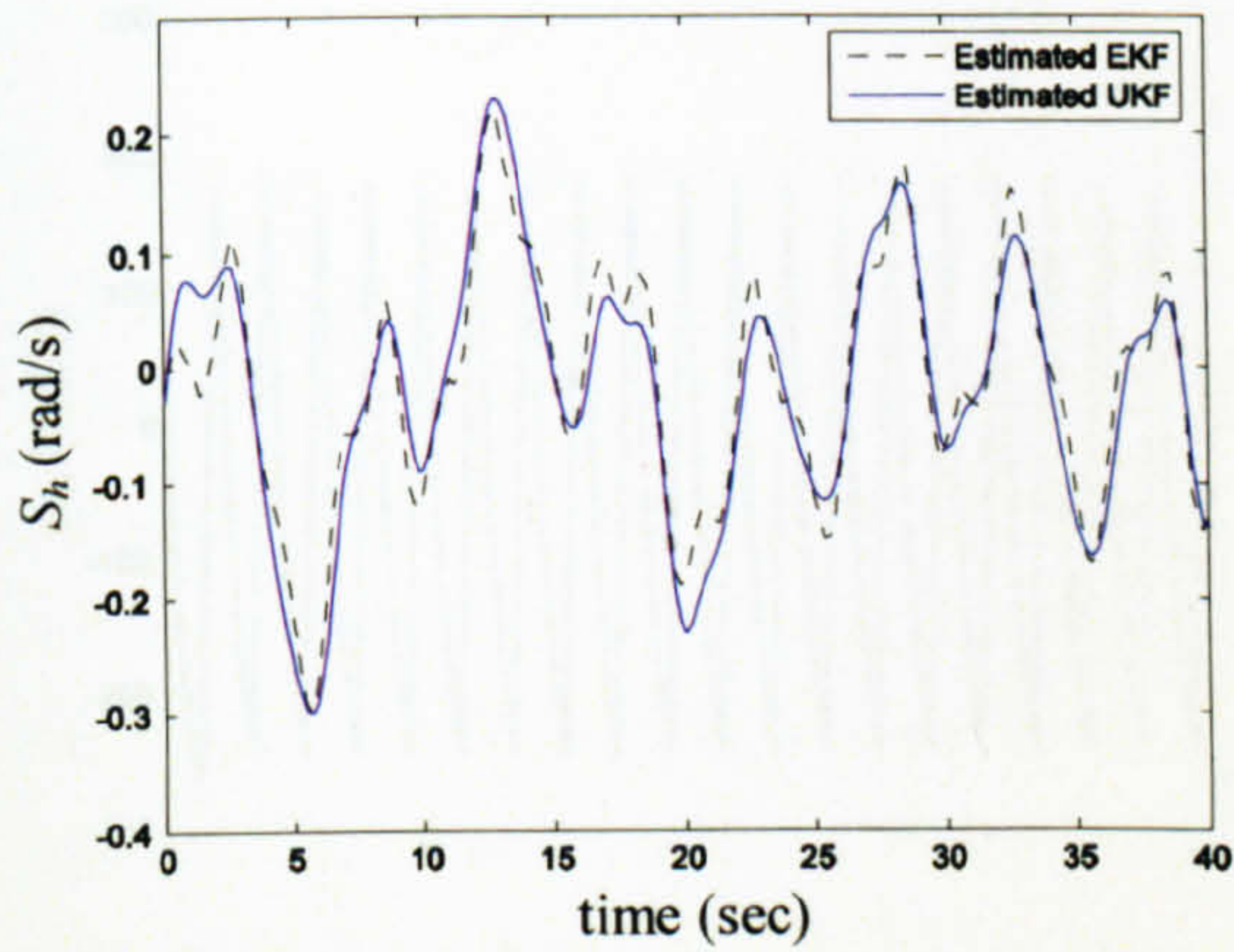


Figure 7.40 EKF and UKF-based estimated S_h (case 1)

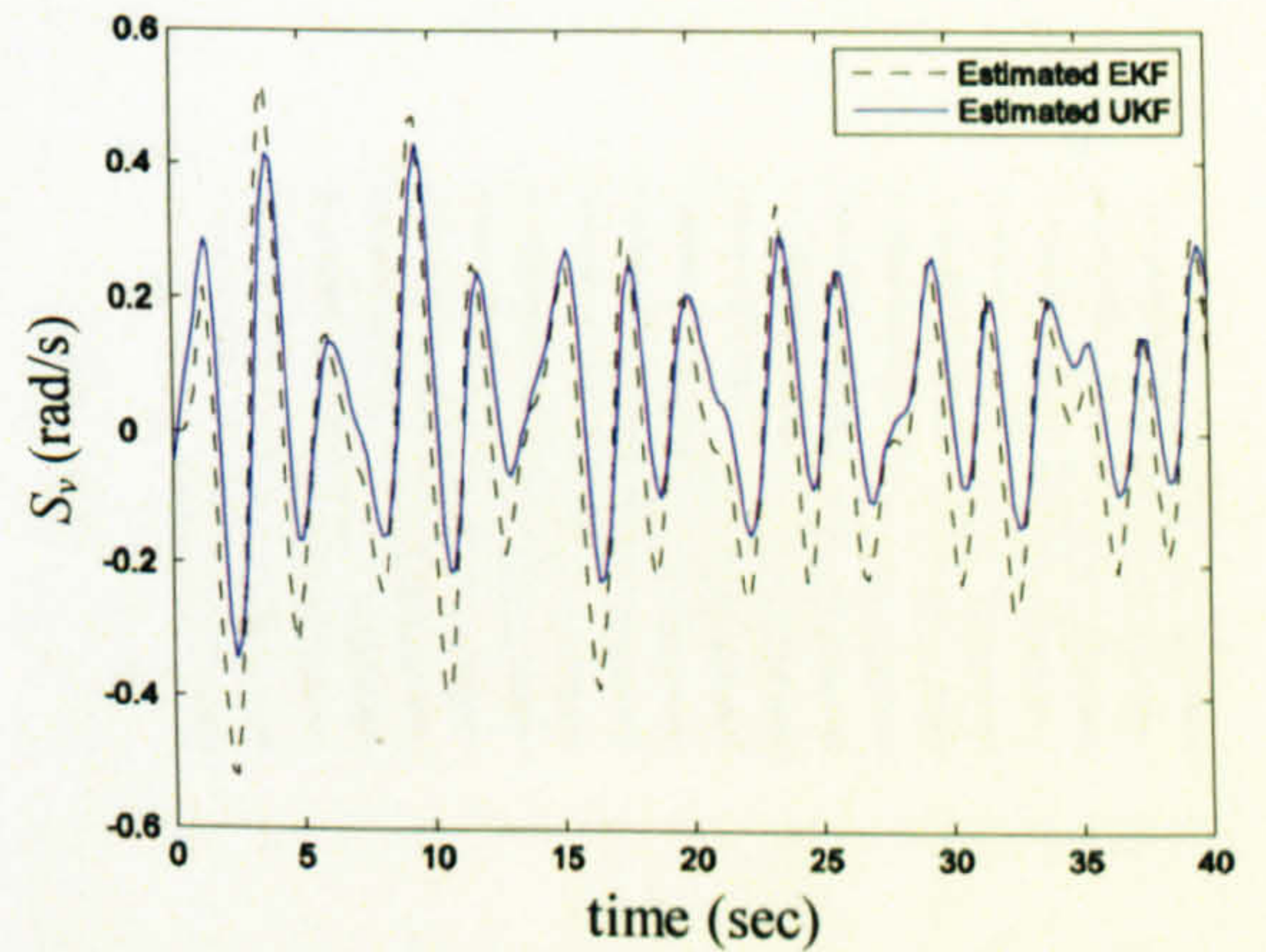


Figure 7.41 EKF and UKF-based estimated S_v (case 1)

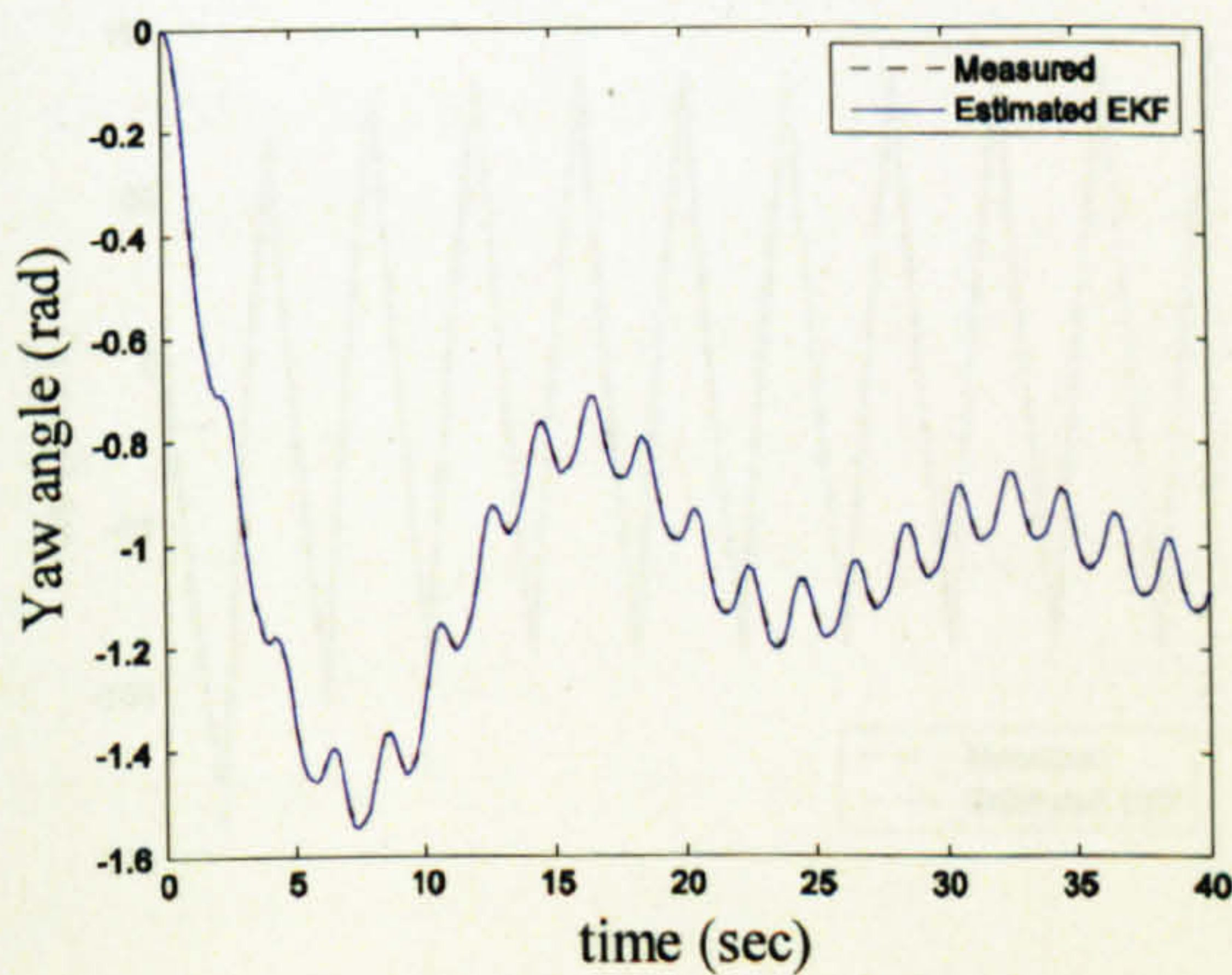


Figure 7.42 Measured and EKF-based estimated yaw angle (case 2)

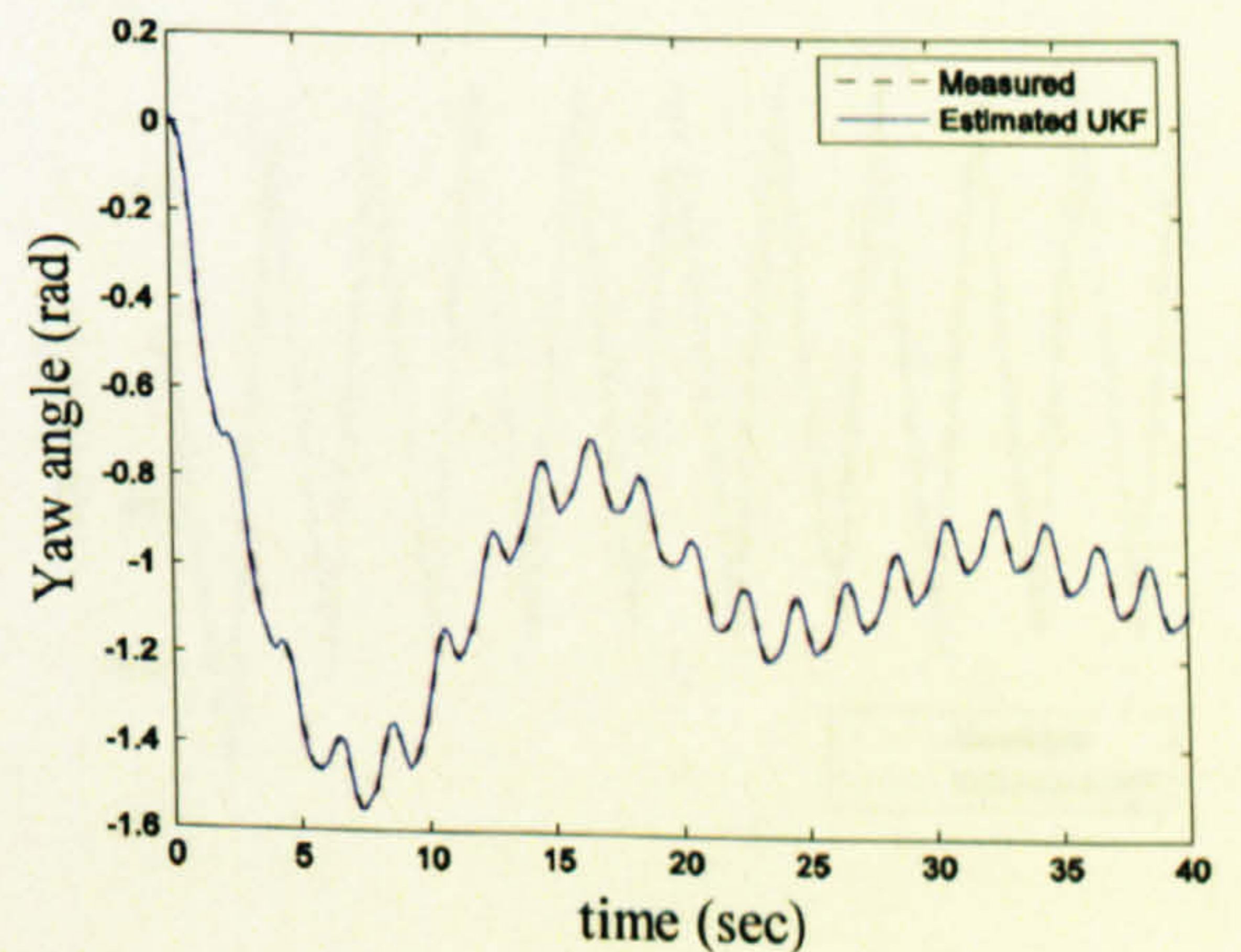


Figure 7.43 Measured and UKF-based estimated yaw angle (case 2)

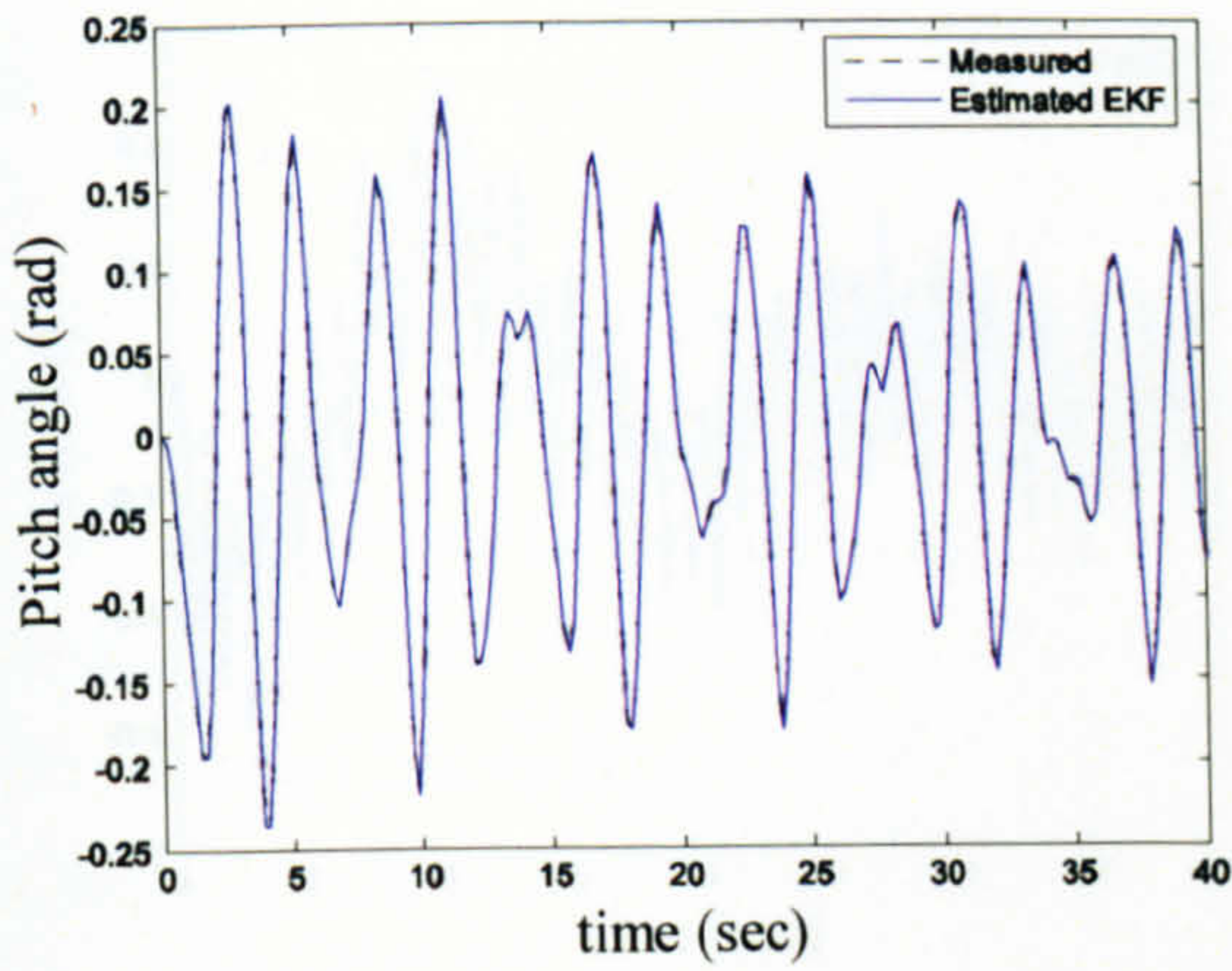


Figure 7.44 Measured and EKF-based estimated pitch angle (case 2)

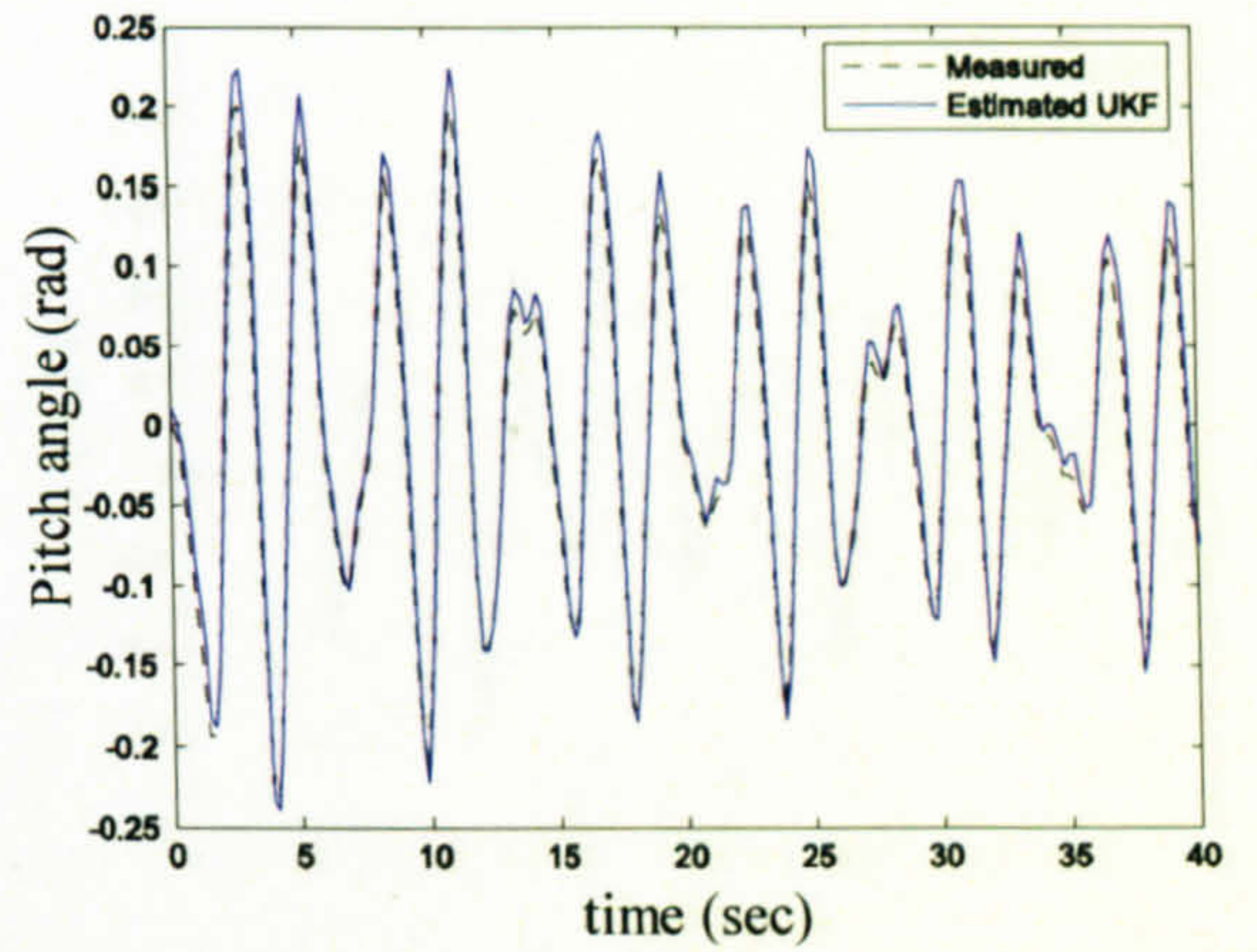


Figure 7.45 Measured and UKF-based estimated pitch angle (case 2)

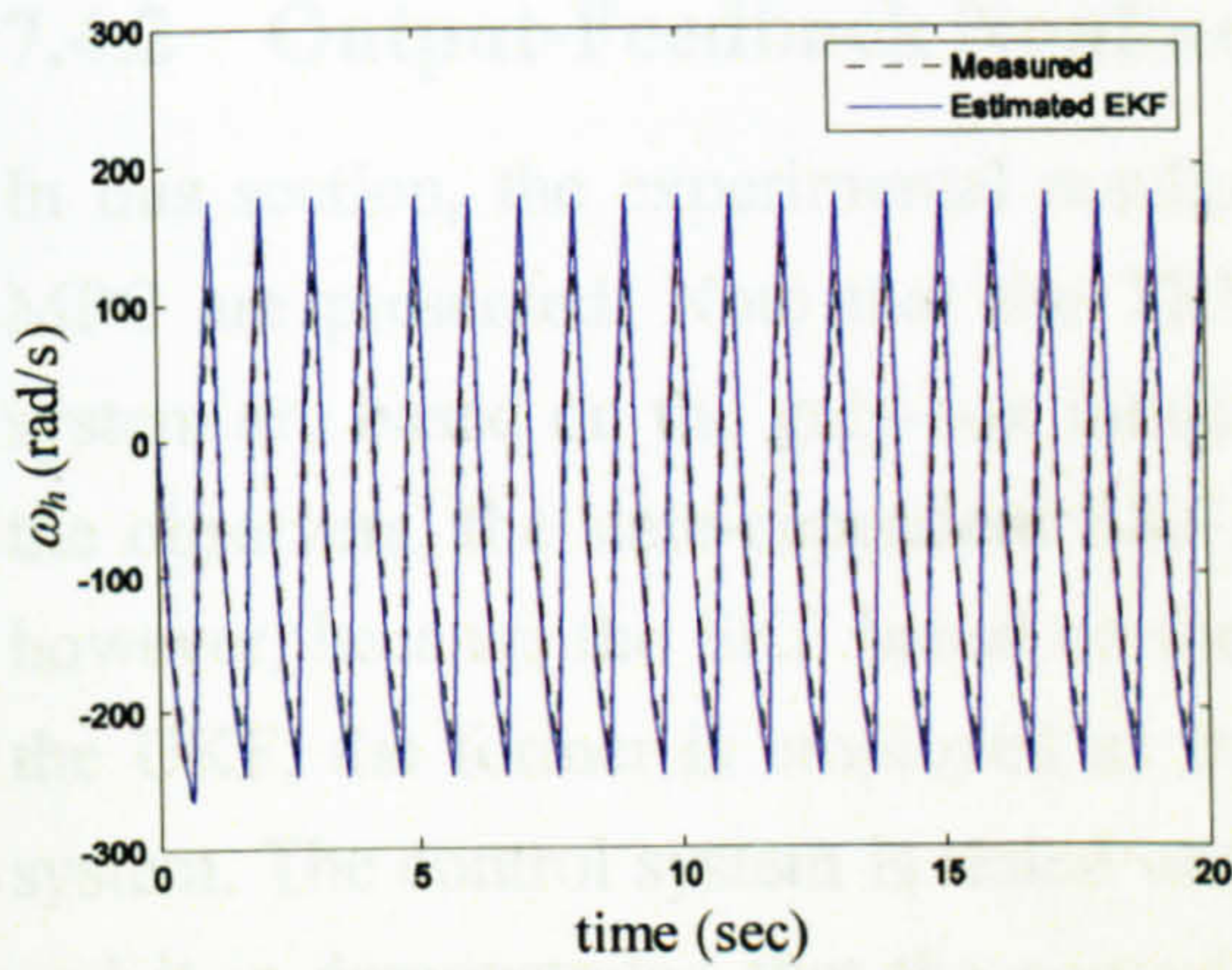


Figure 7.46 Measured and EKF-based estimated ω_h (case 2)

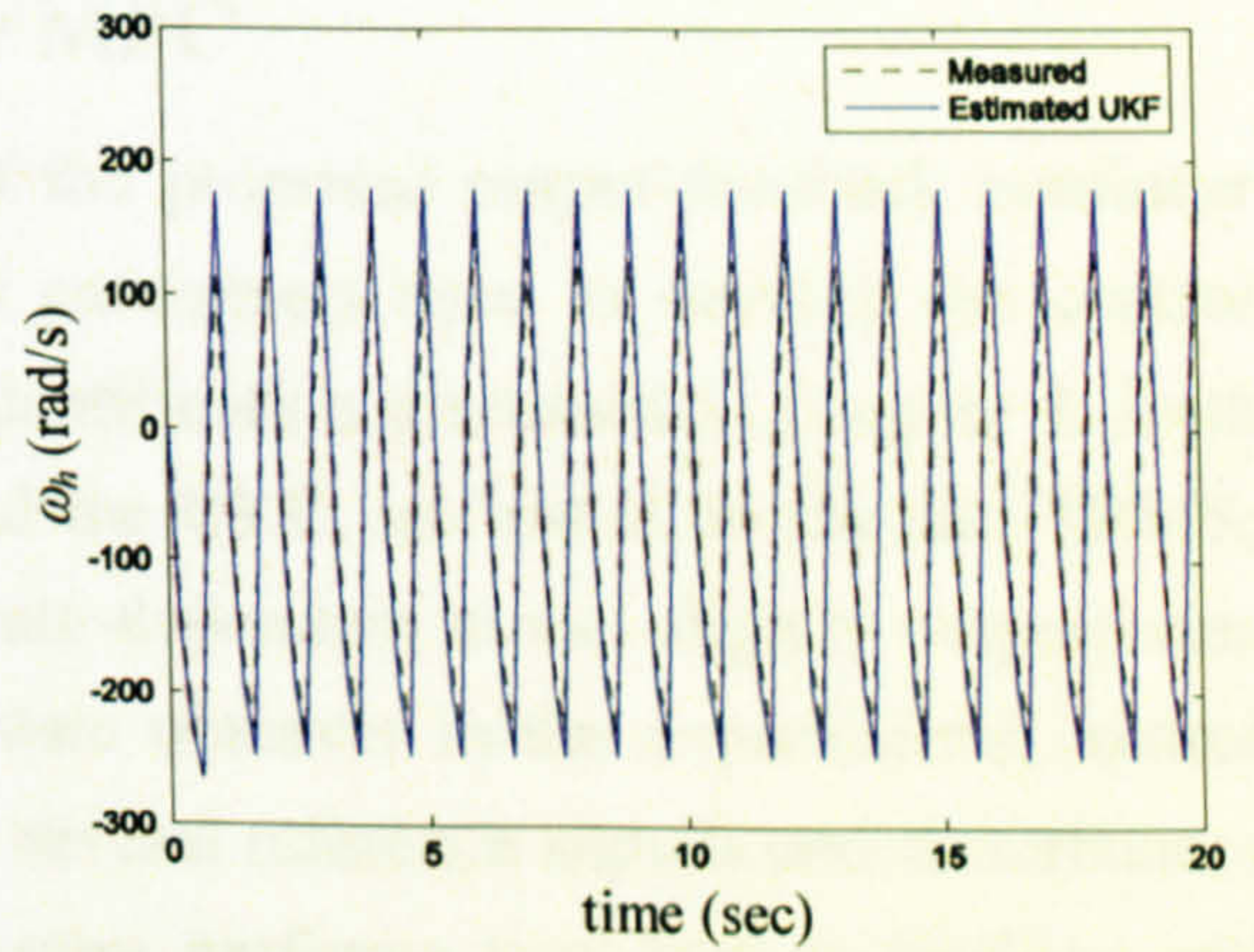


Figure 7.47 Measured and UKF-based estimated ω_h (case 2)

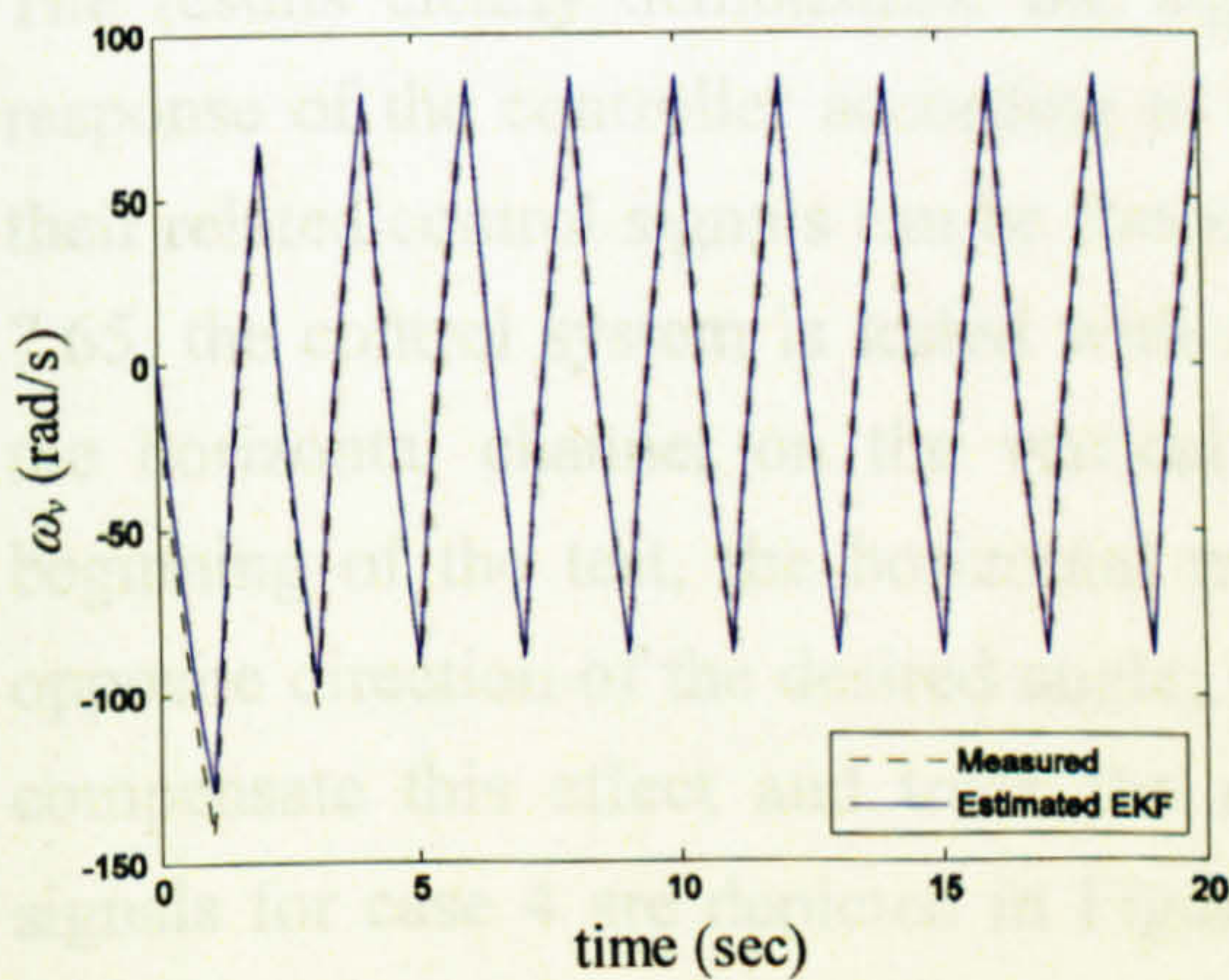


Figure 7.48 Measured and EKF-based estimated ω_v (case 2)

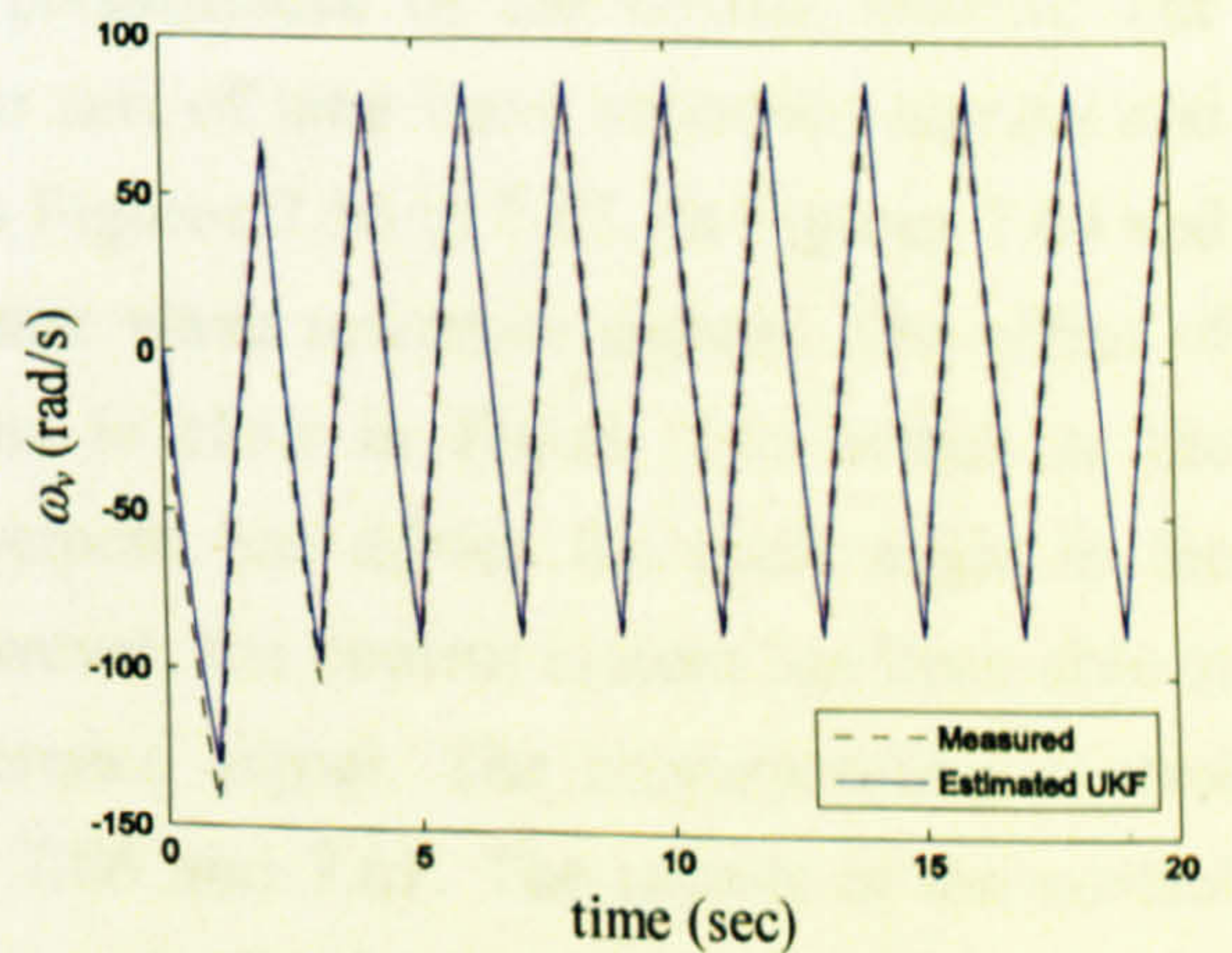


Figure 7.49 Measured and UKF-based estimated ω_v (case 2)

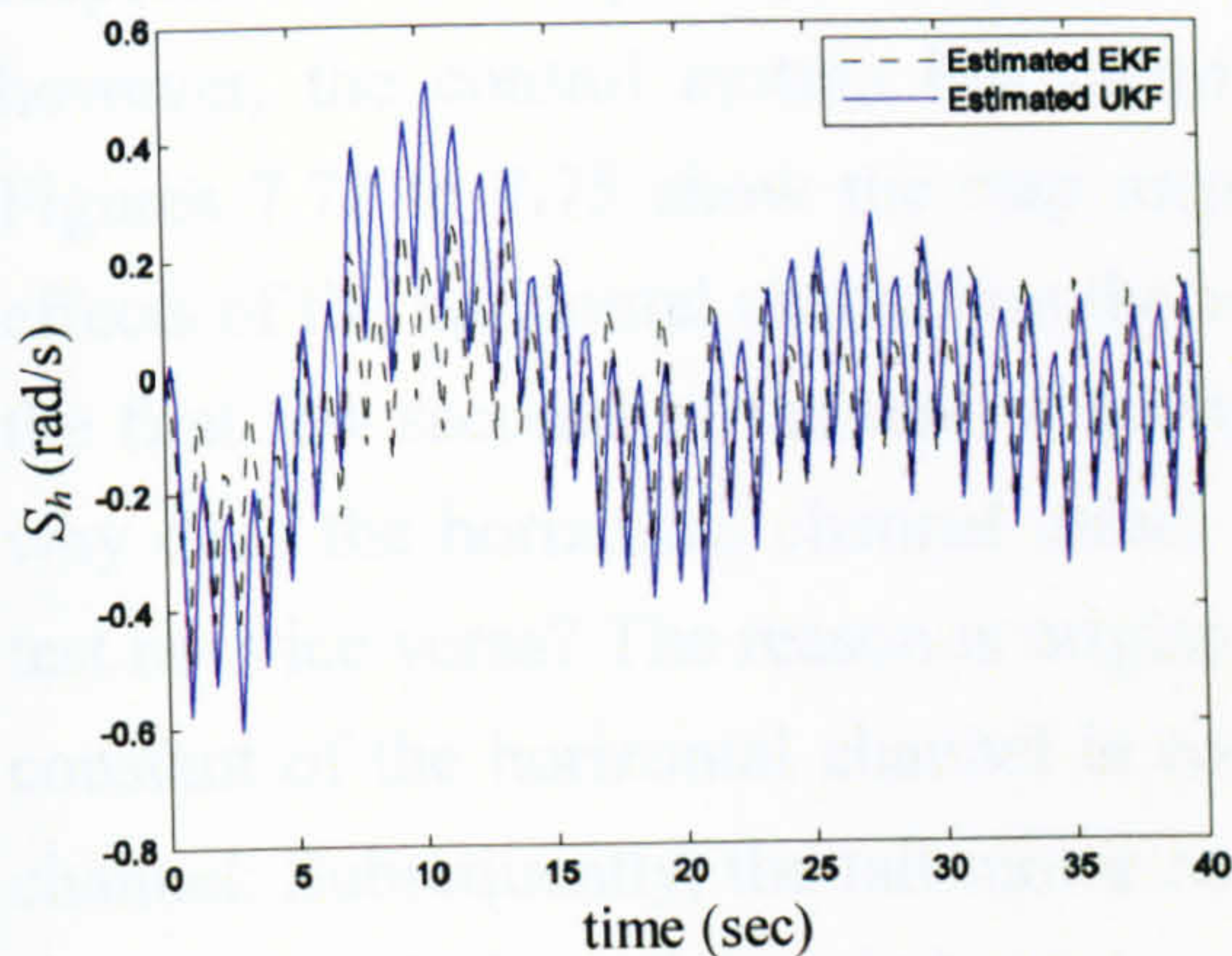


Figure 7.50 EKF and UKF-based estimated S_h (case 2)

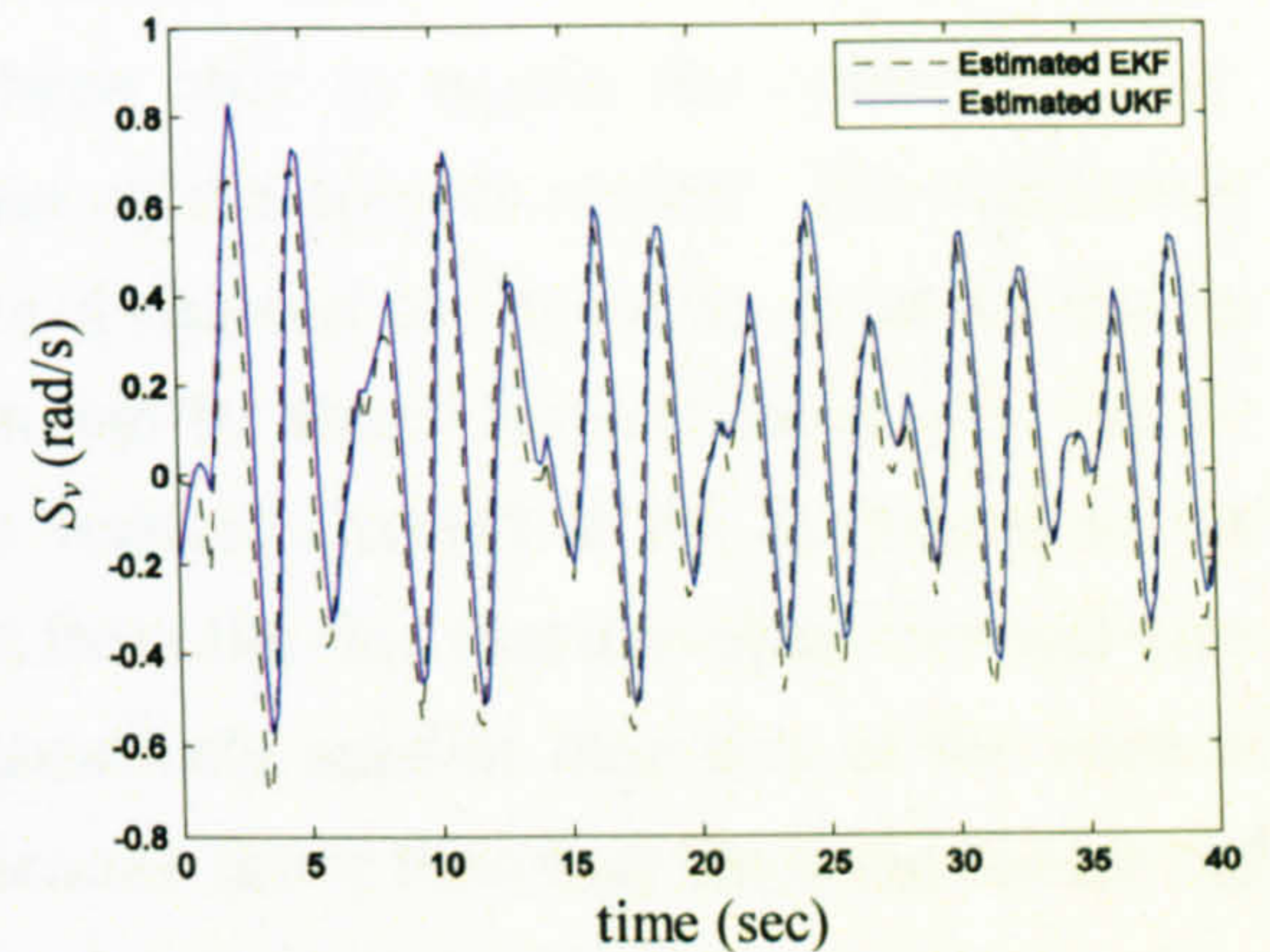


Figure 7.51 EKF and UKF-based estimated S_v (case 2)

7.4.2 Output-Feedback Nonlinear MPC

In this section, the experimental results of the proposed output-feedback nonlinear MPC are presented. Note that the TRMS parameters used to develop the control system are based on the grey-box model identification discussed in Chapter 4. Both the observers, the state-dependent EKF and the UKF, are tested on the real TRMS; however, because the EKF based on the state-dependent model slightly outperforms the UKF, the former is employed as the state observer in the experimental control system. The control system is tested under several reference signals and disturbances and it is demonstrated that the control system performs very well in tracking and disturbance rejection. Figures 7.52 and 7.53 show the response of the control system with respect to a set of sawtooth reference signals. The control signals of the horizontal and vertical channels are illustrated in Figures 7.54 and 7.55 respectively. The results clearly demonstrate the high performance of the control system. The response of the controller according to two sets of sine wave reference signals and their related control signals can be found in Figures 7.56 to 7.63. In Figures 7.64 and 7.65, the control system is tested with square wave reference signals. The effect of the horizontal channel on the vertical one is clear in Figure 7.65 which at the beginning of the test, the horizontal movement has driven the pitch angle in the opposite direction of the desired angle; however, the control system has been able to compensate this effect and track the reference signal. The corresponding control signals for case 4 are depicted in Figures 7.66 and 7.67. The results of the control paradigm in response to a most challenging set of reference signals can be seen in Figures 7.68 to 7.71. In those figures the significant effect of the horizontal channel on the vertical channel can be observed at the instants 25 and 75 seconds in which the yaw angle has been forced to change from -0.5 to 0.5 rad and from 0.5 to -0.5

respectively. Consequently, the pitch angle has deviated from its reference signal; however, the control system has clearly been able to regain the control rapidly. Figures 7.72 to 7.75 show the step response of the control system. The significant effects of the horizontal channel on the vertical channel can again be observed during the first few seconds of the test. A question can be arisen from those results that is: why does the horizontal channel affect the vertical channel at the beginning of the test not vice versa? The reason is originated from the fact that the (approximate) time constant of the horizontal channel is comparatively smaller than that of the vertical channel. Subsequently, the tail motor accelerates faster than that the main motor and causes a deviation of the pitch angle at the beginning of the test, regardless of its reference signal. Figures 7.76 to 7.79 show the step response of the control problem with an external force applied manually to the system in order to test it in the face of a significant disturbance. In this test, an external force has applied to the TRMS and as it is evident the yaw and pitch angles deviate from their desired positions. However, the control system has regained the control and forced the beam to return to its desired angles.

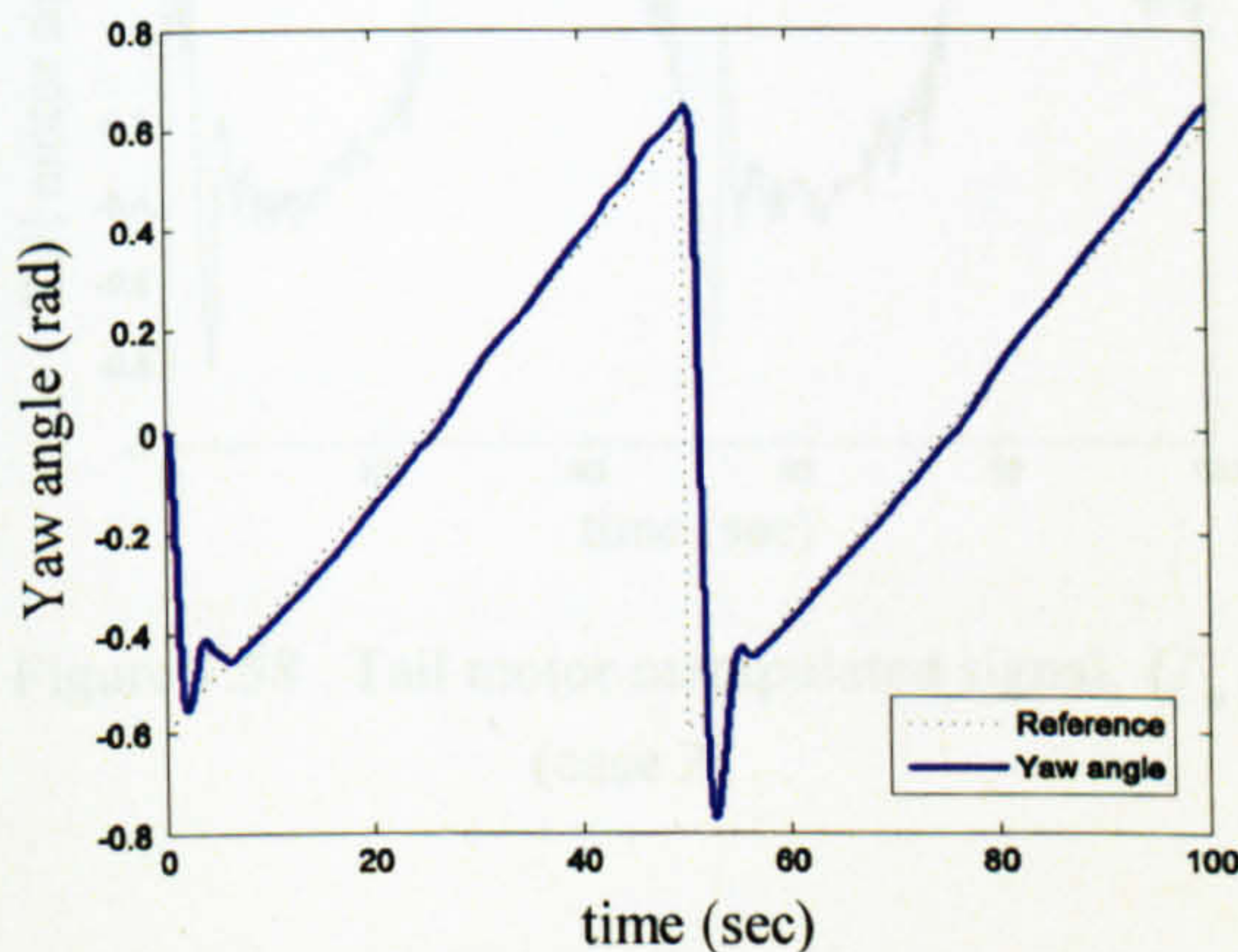


Figure 7.52 Sawtooth wave response of the horizontal angle (case 1)

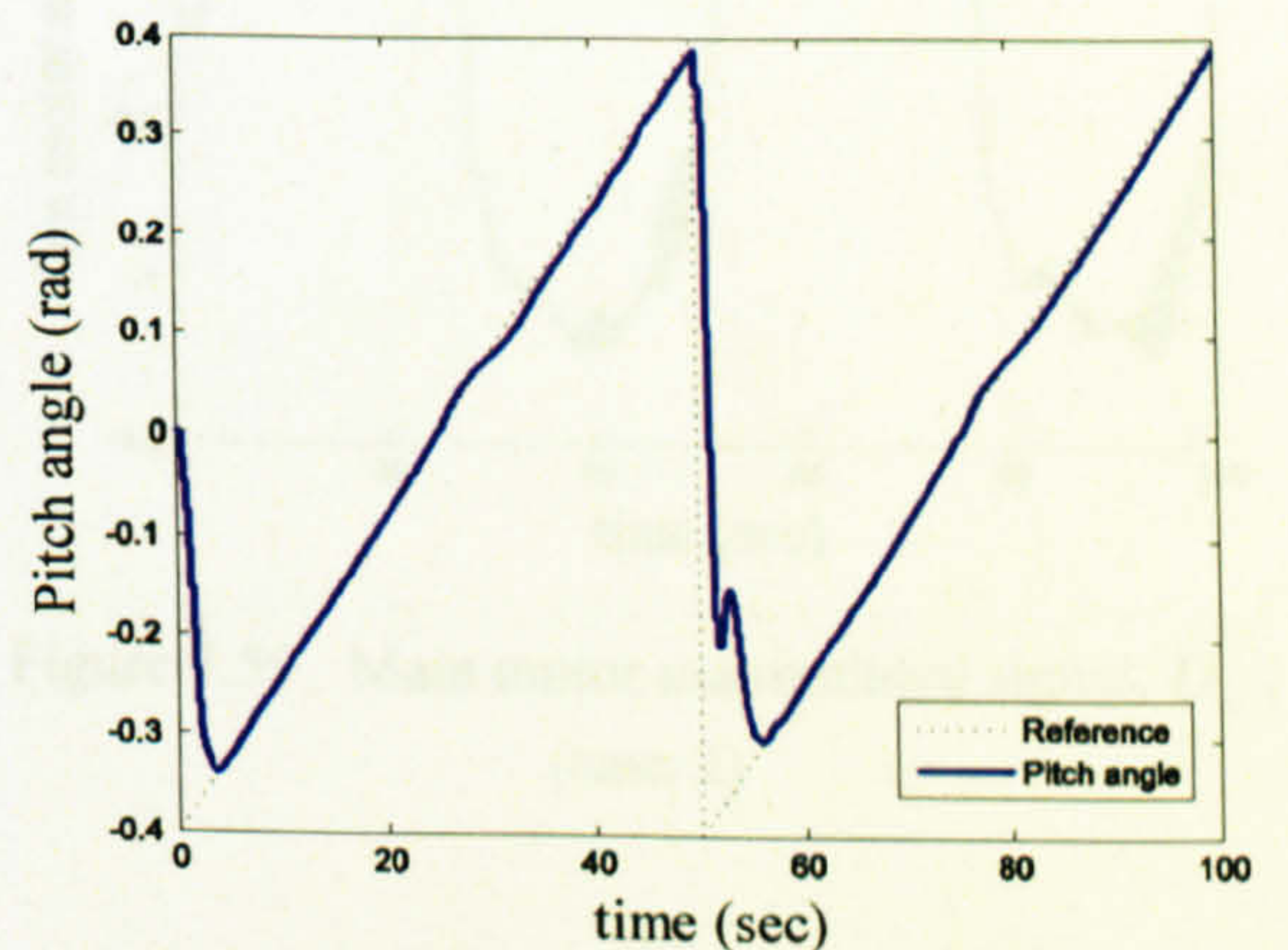


Figure 7.53 Sawtooth wave response of the vertical angle (case 1)

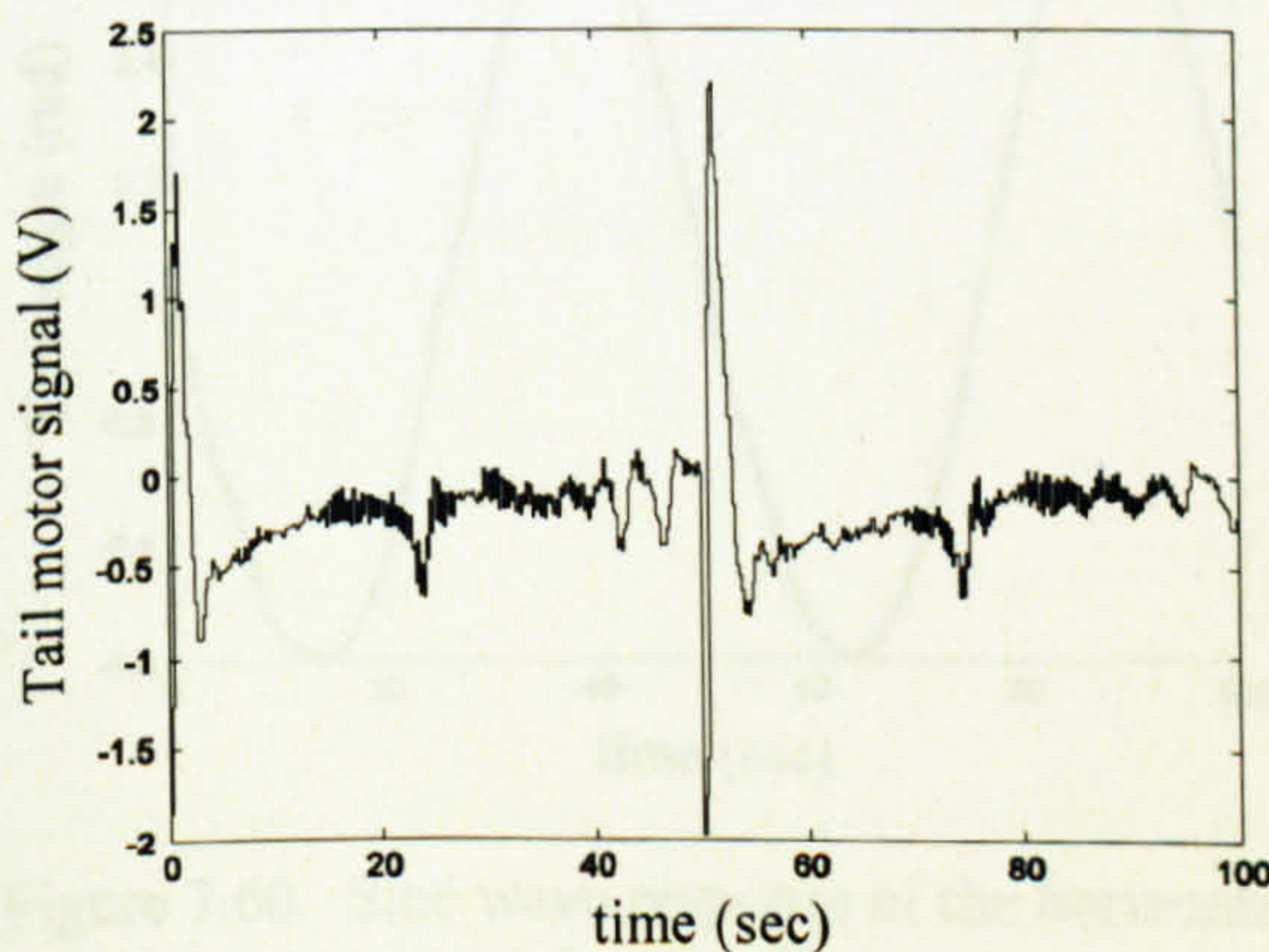


Figure 7.54 Tail motor manipulated signal, U_h , (case 1)

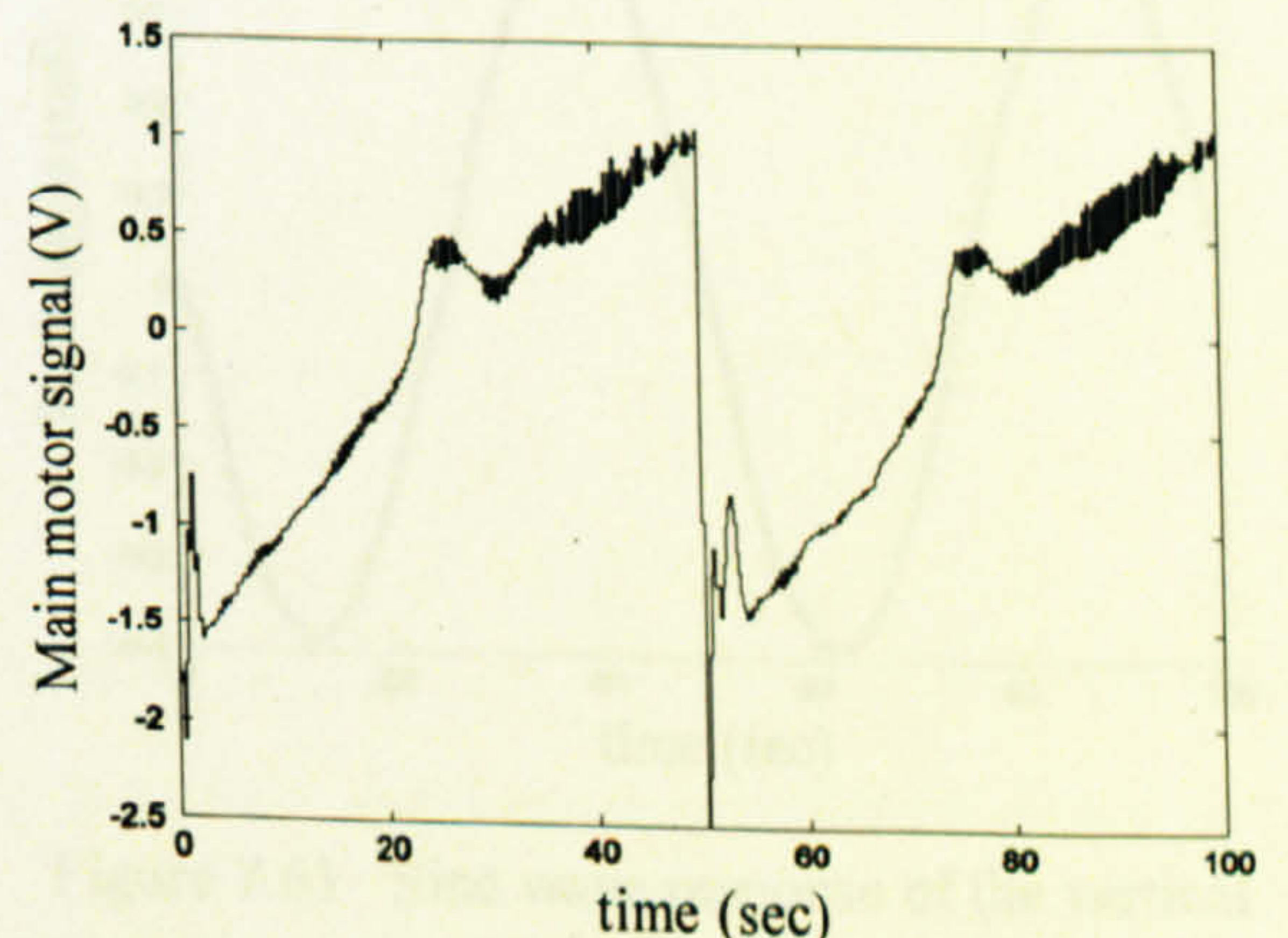


Figure 7.55 Main motor manipulated signal, U_v , (case 1)

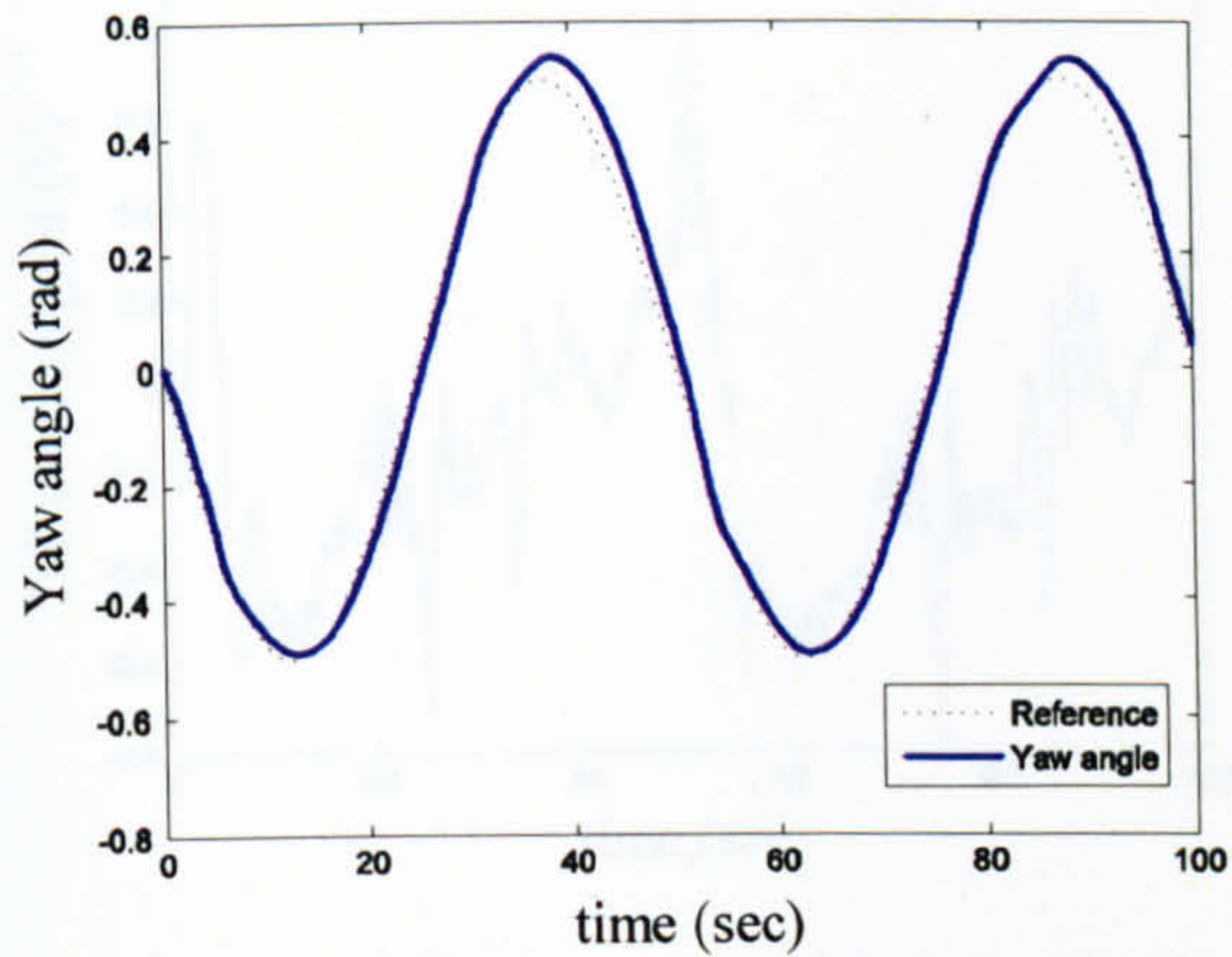


Figure 7.56 Sine wave response of the horizontal angle (case 2)

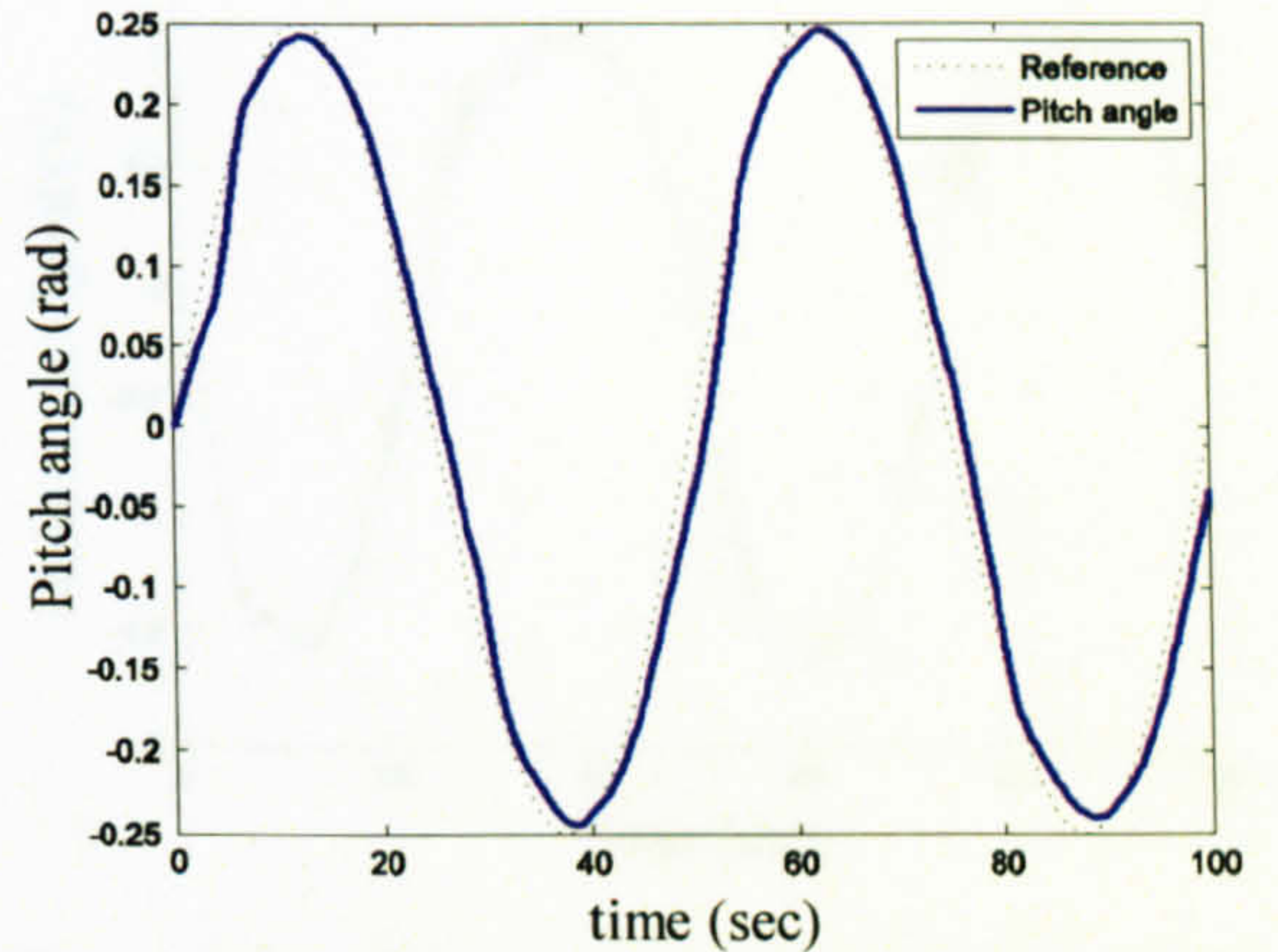


Figure 7.57 Sine wave response of the vertical angle (case 2)

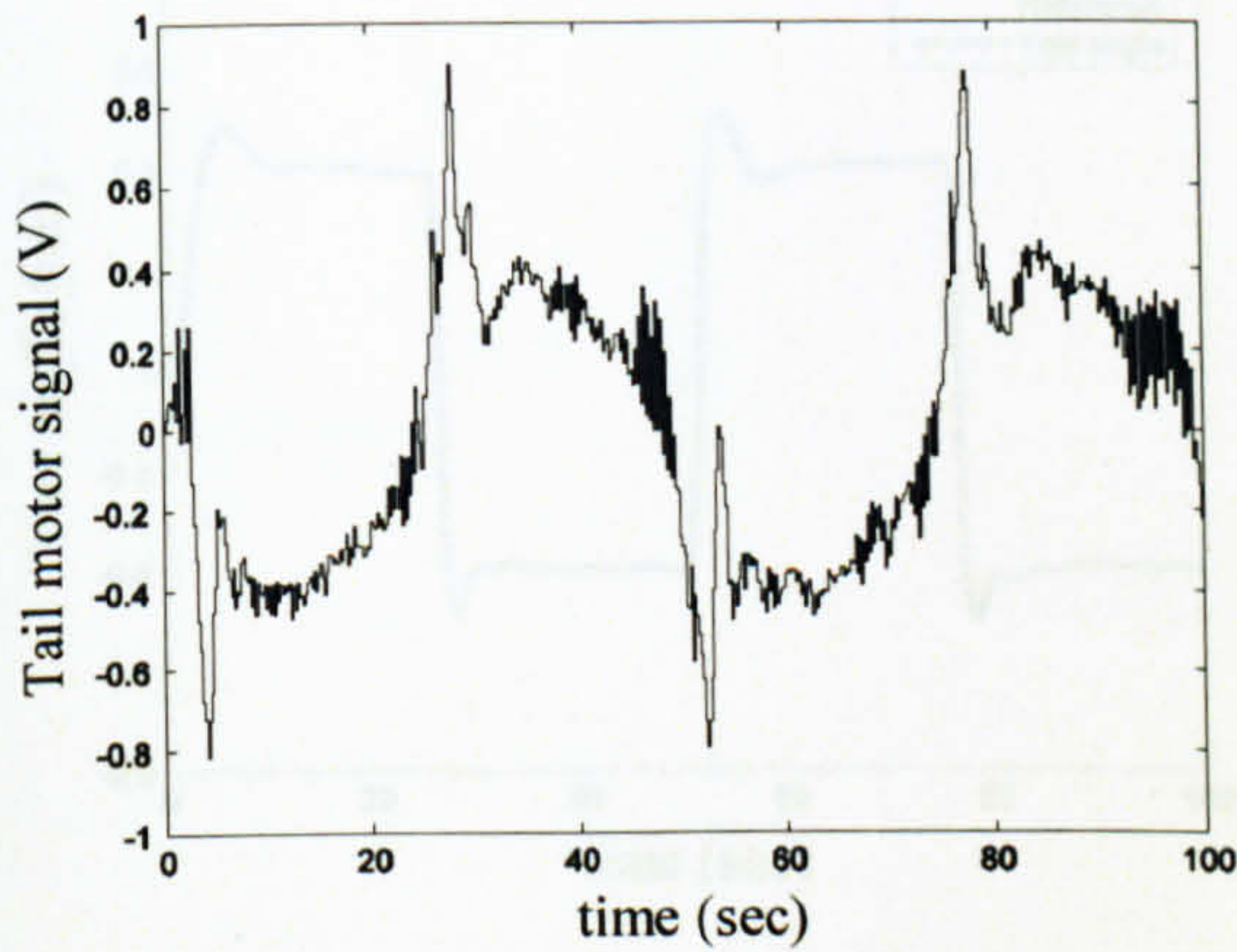


Figure 7.58 Tail motor manipulated signal, U_h , (case 2)

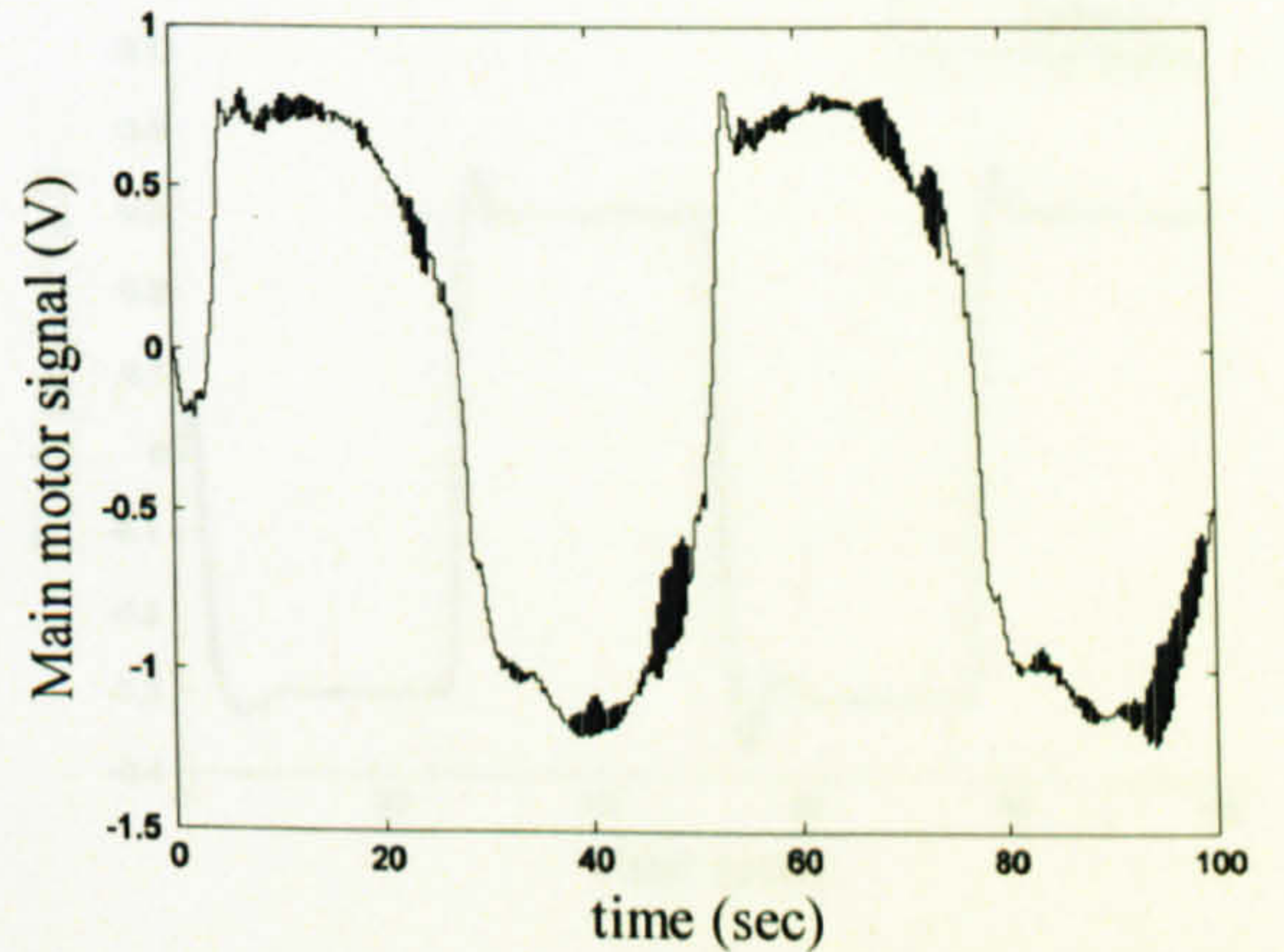


Figure 7.59 Main motor manipulated signal, U_v , (case 2)

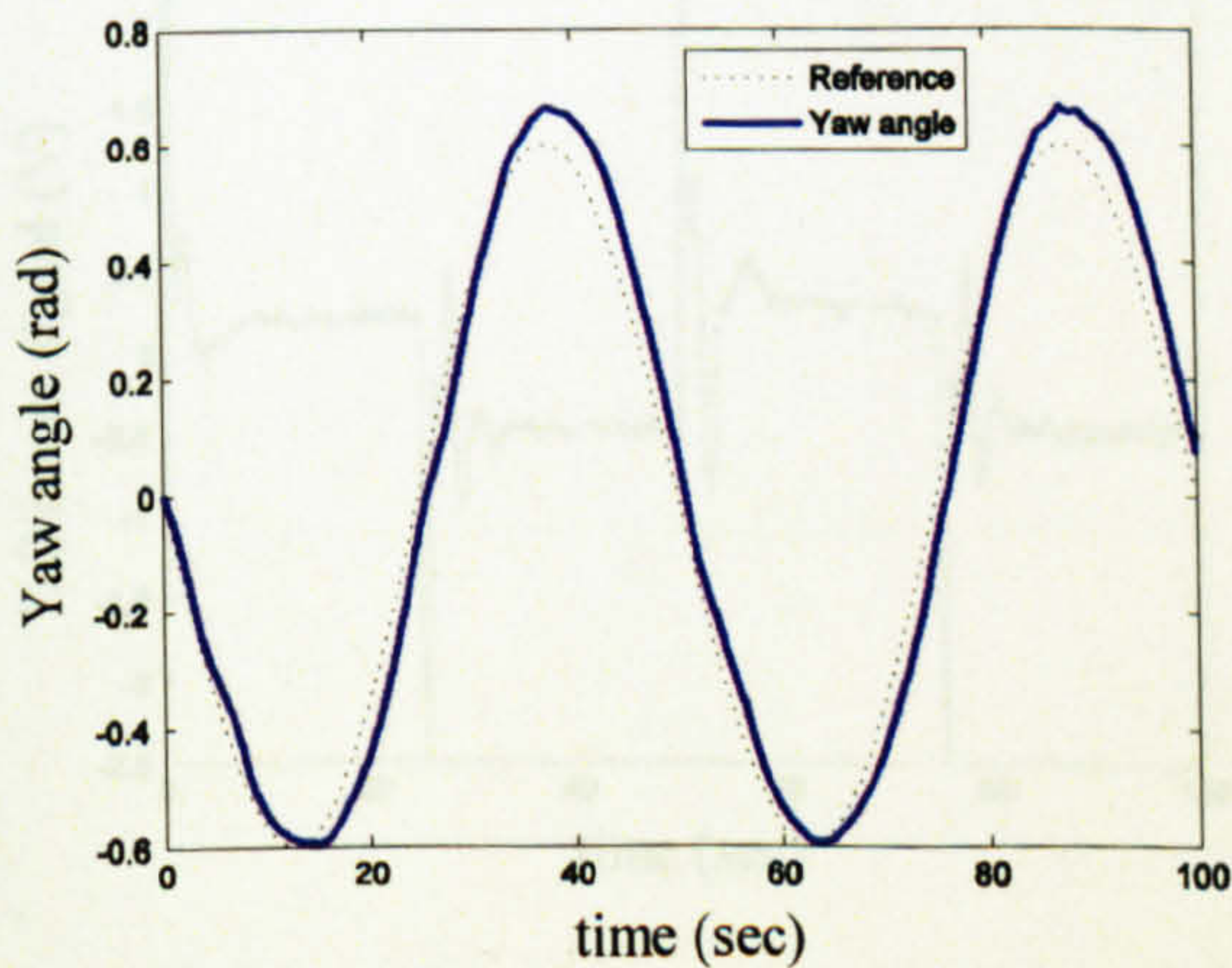


Figure 7.60 Sine wave response of the horizontal angle (case 3)

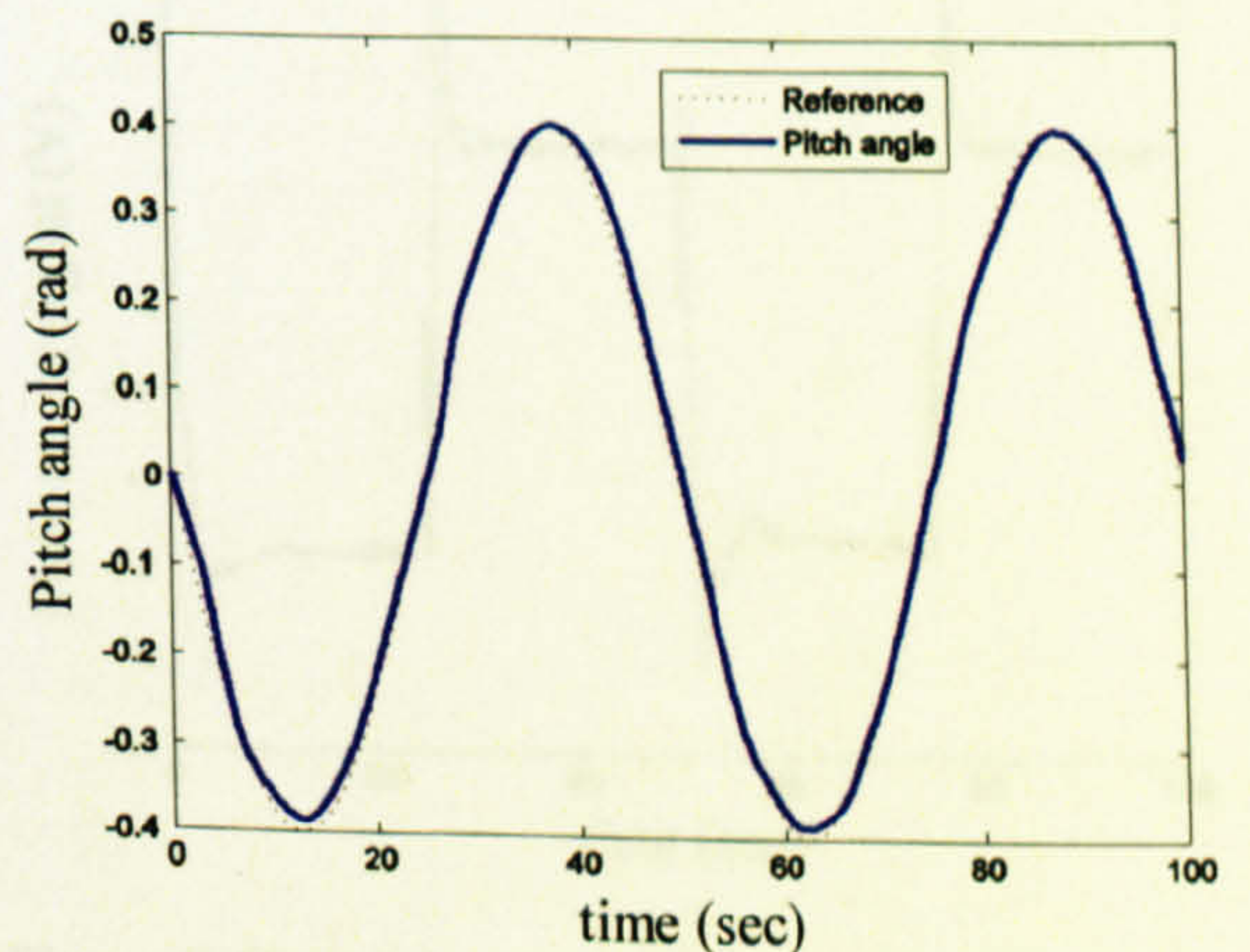


Figure 7.61 Sine wave response of the vertical angle (case 3)

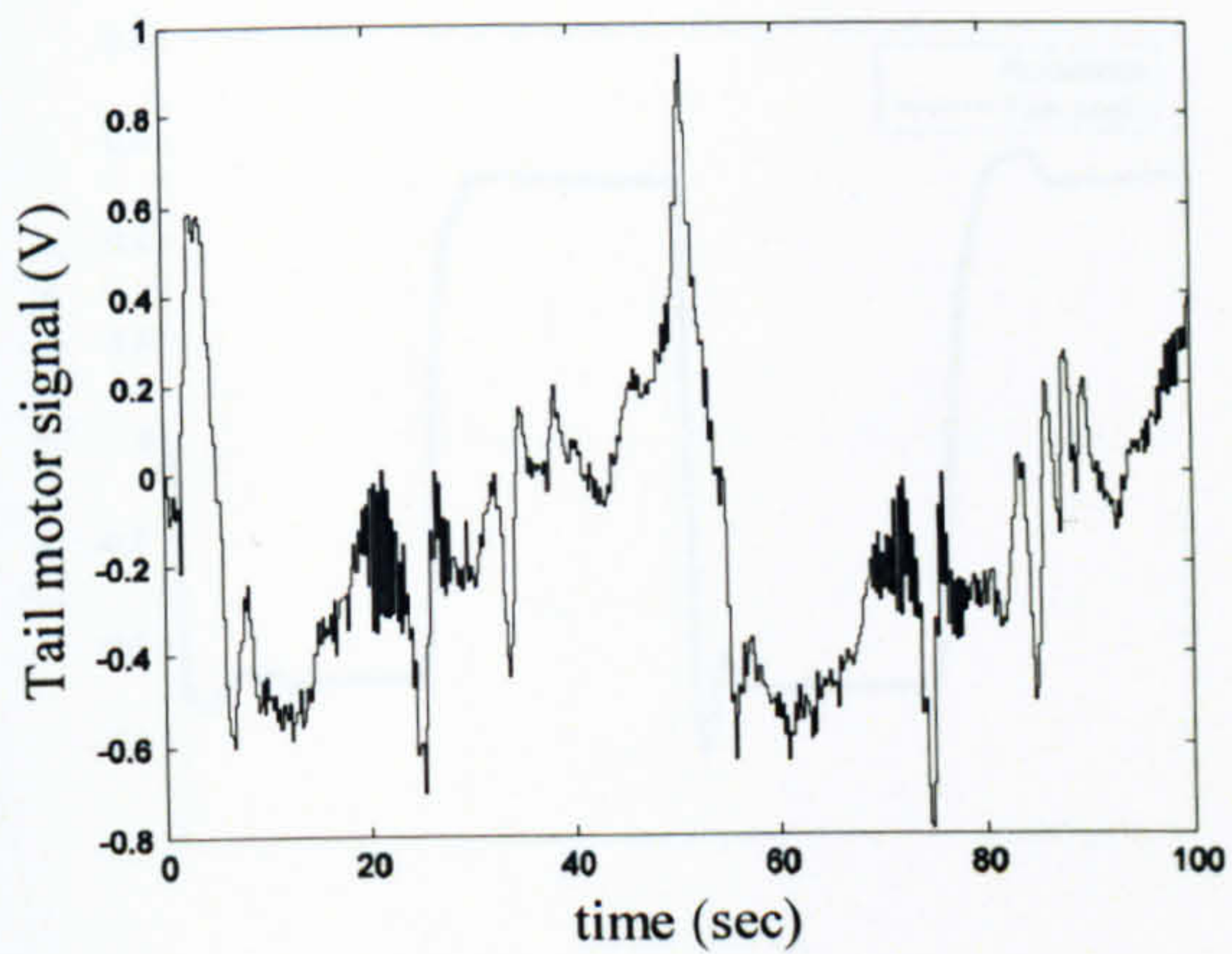


Figure 7.62 Tail motor manipulated signal, U_h , (case 3)

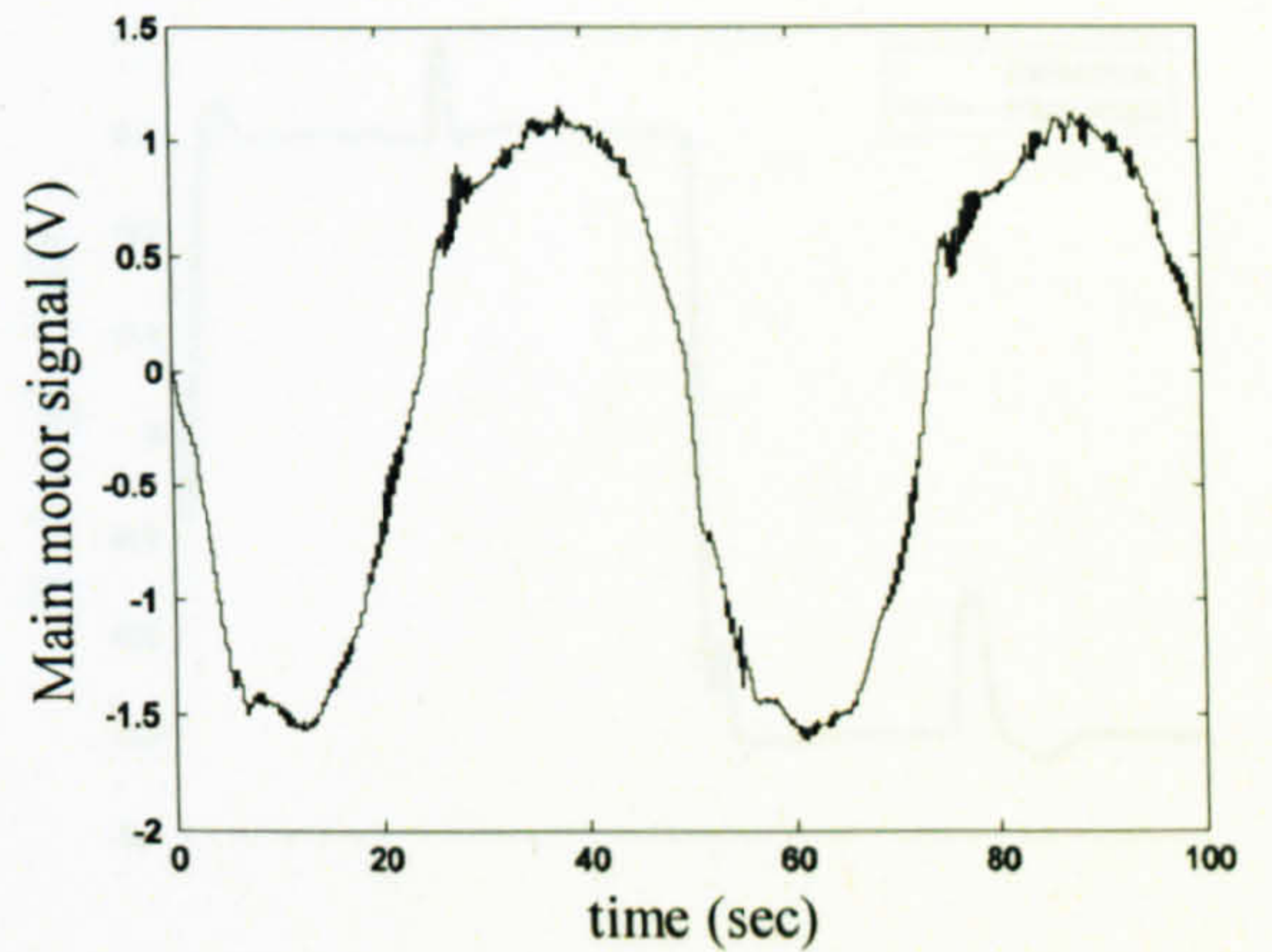


Figure 7.63 Main motor manipulated signal, U_v , (case 3)

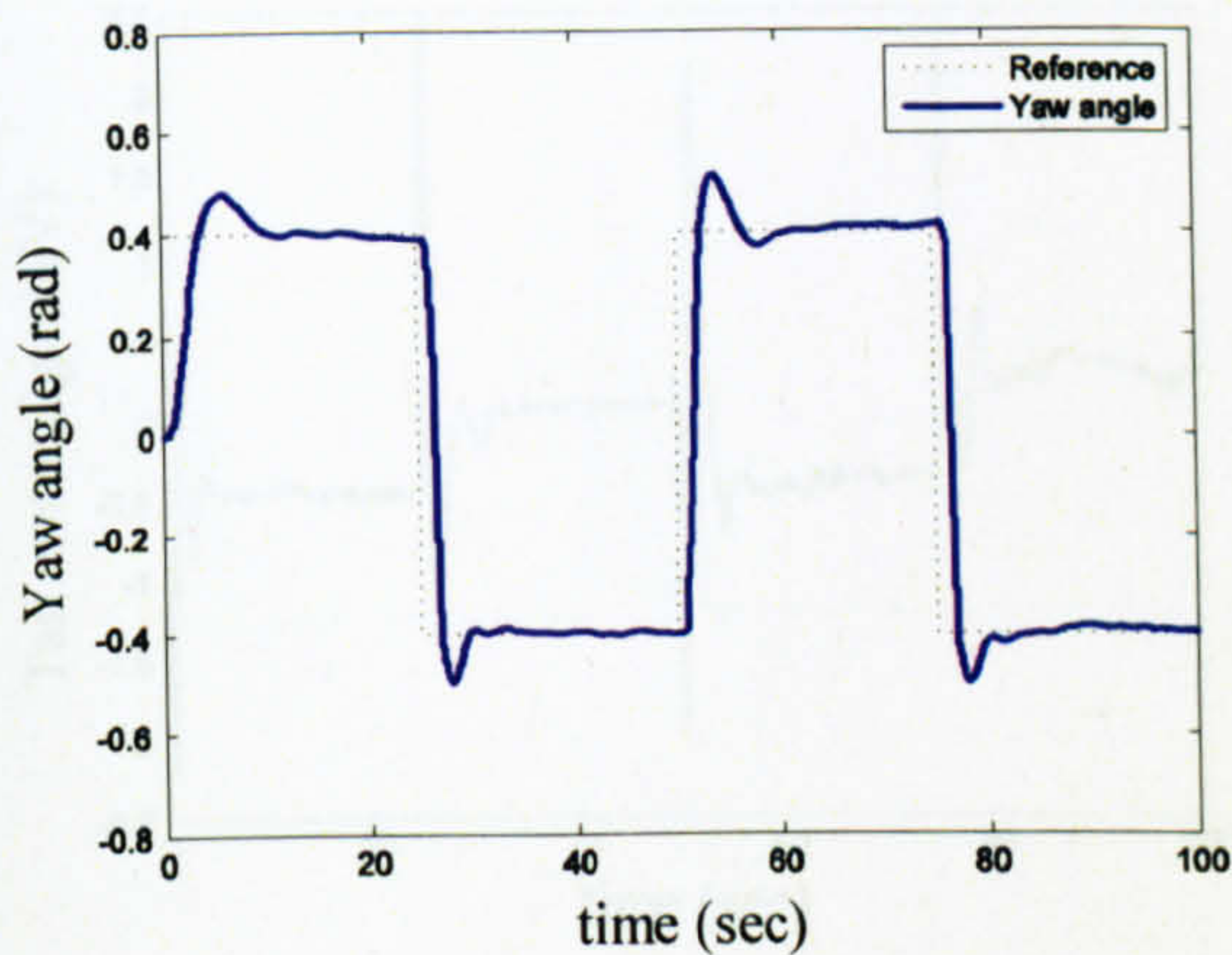


Figure 7.64 Square wave response of the horizontal angle (case 4)

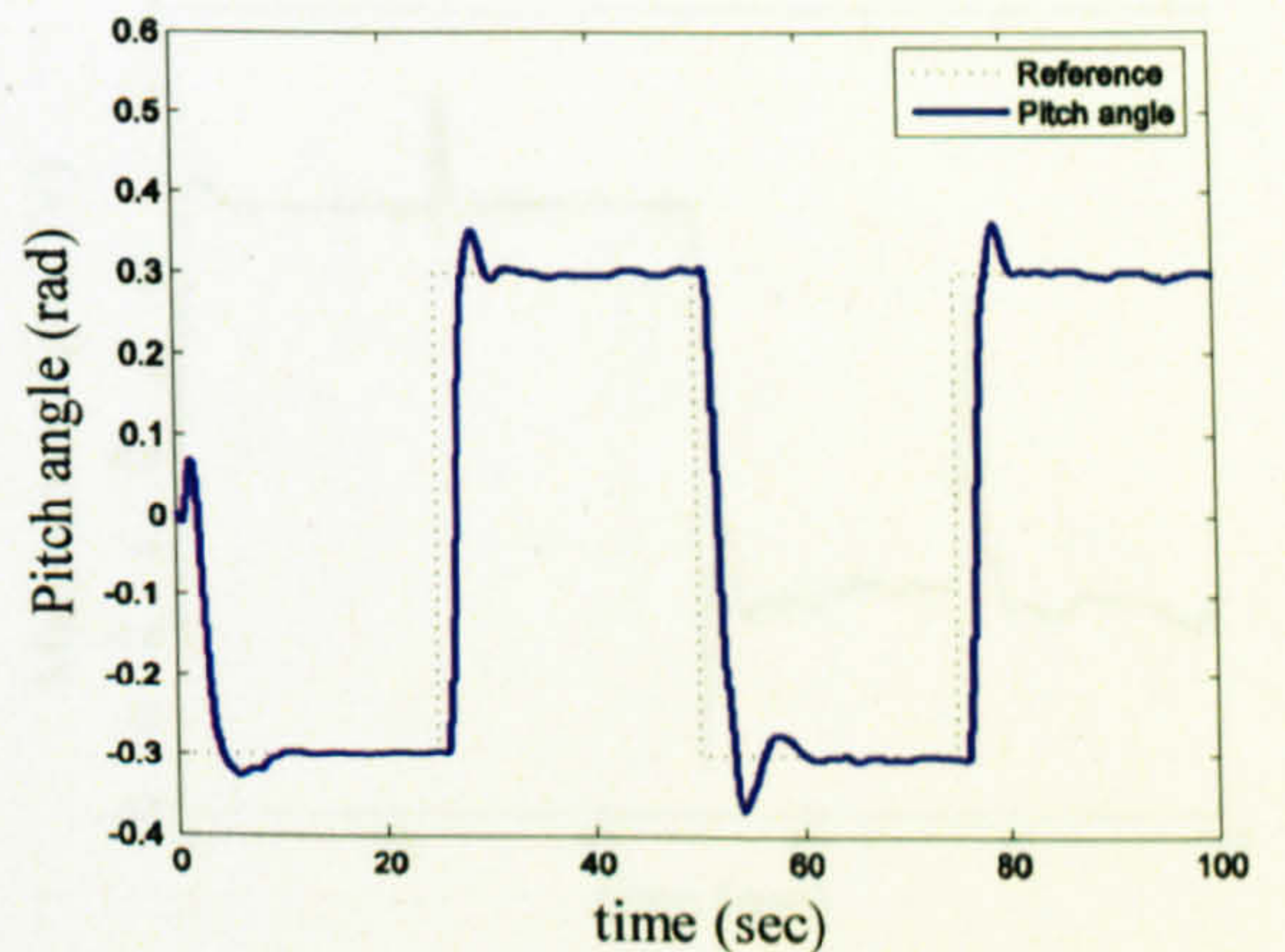


Figure 7.65 Square wave response of the vertical angle (case 4)

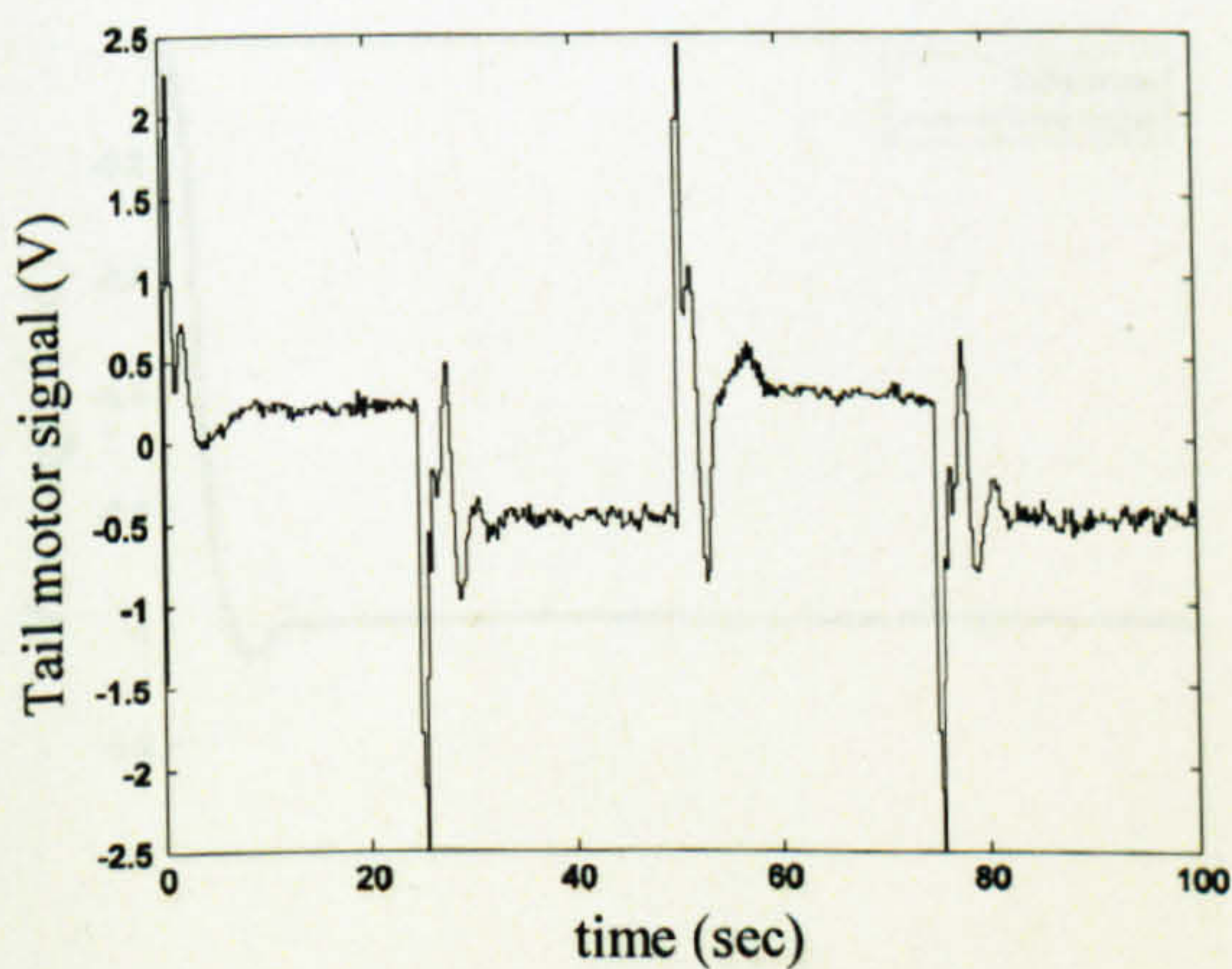


Figure 7.66 Tail motor manipulated signal, U_h , (case 4)

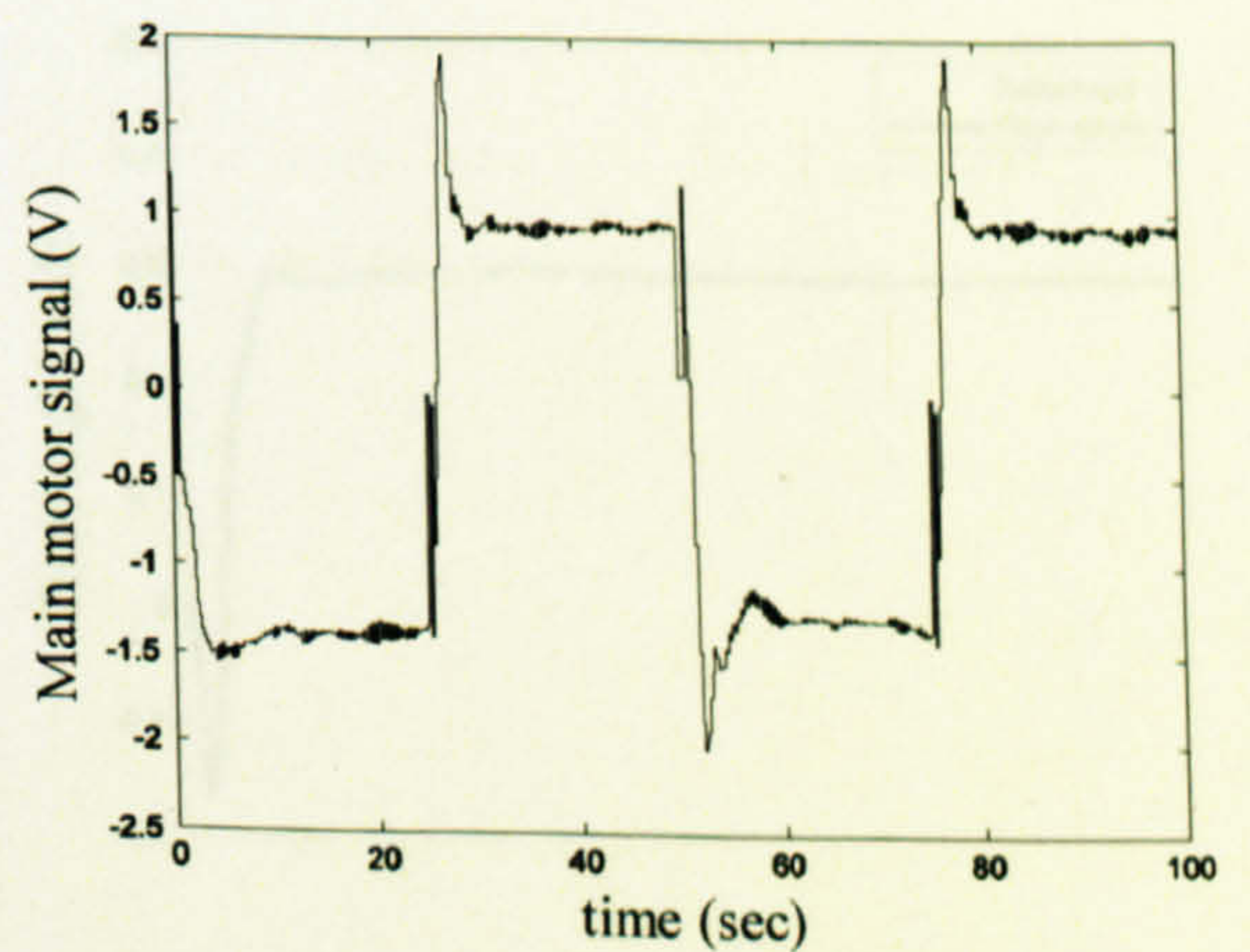


Figure 7.67 Main motor manipulated signal, U_v , (case 4)

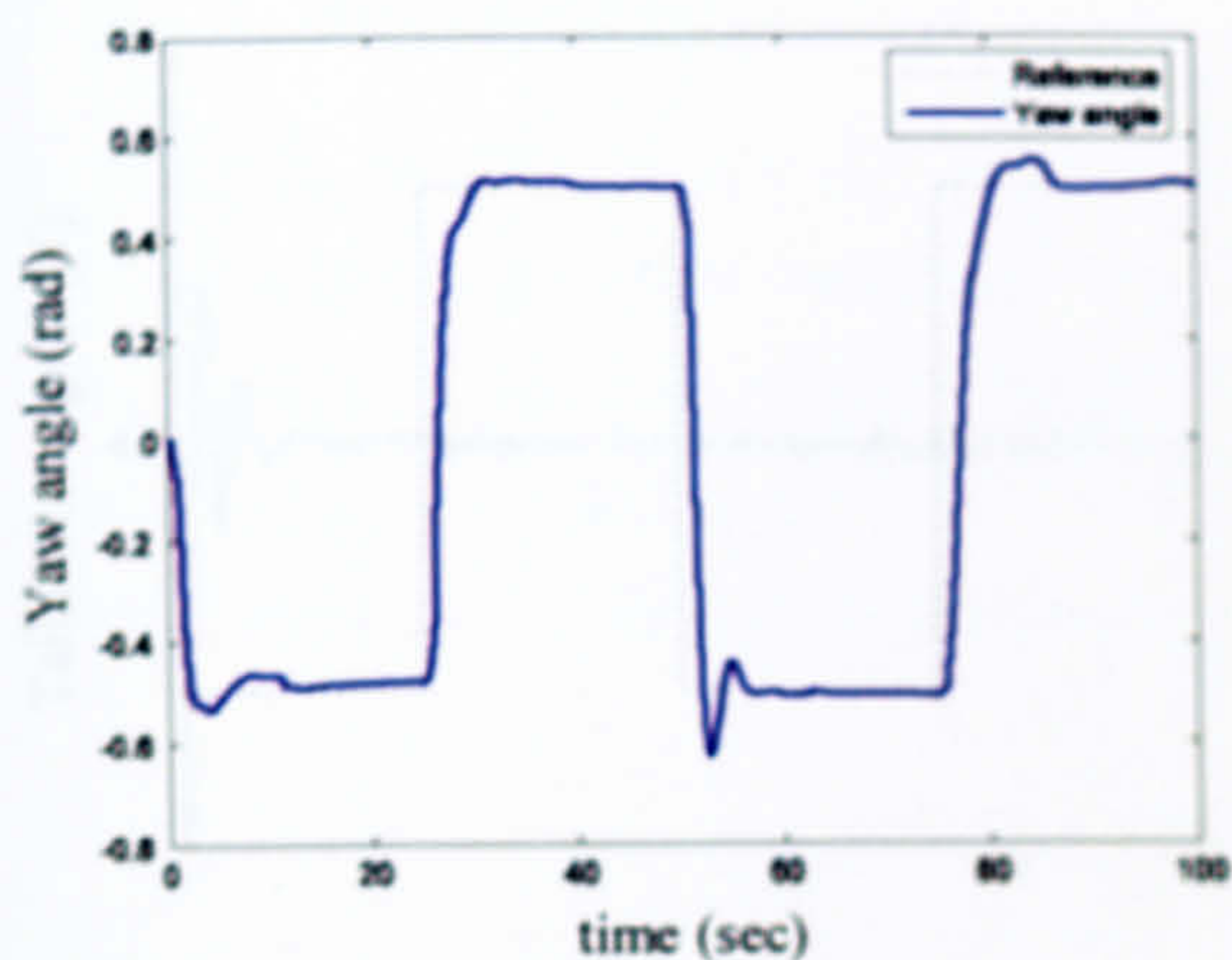


Figure 7.68 Square wave response of the horizontal angle (case 5)

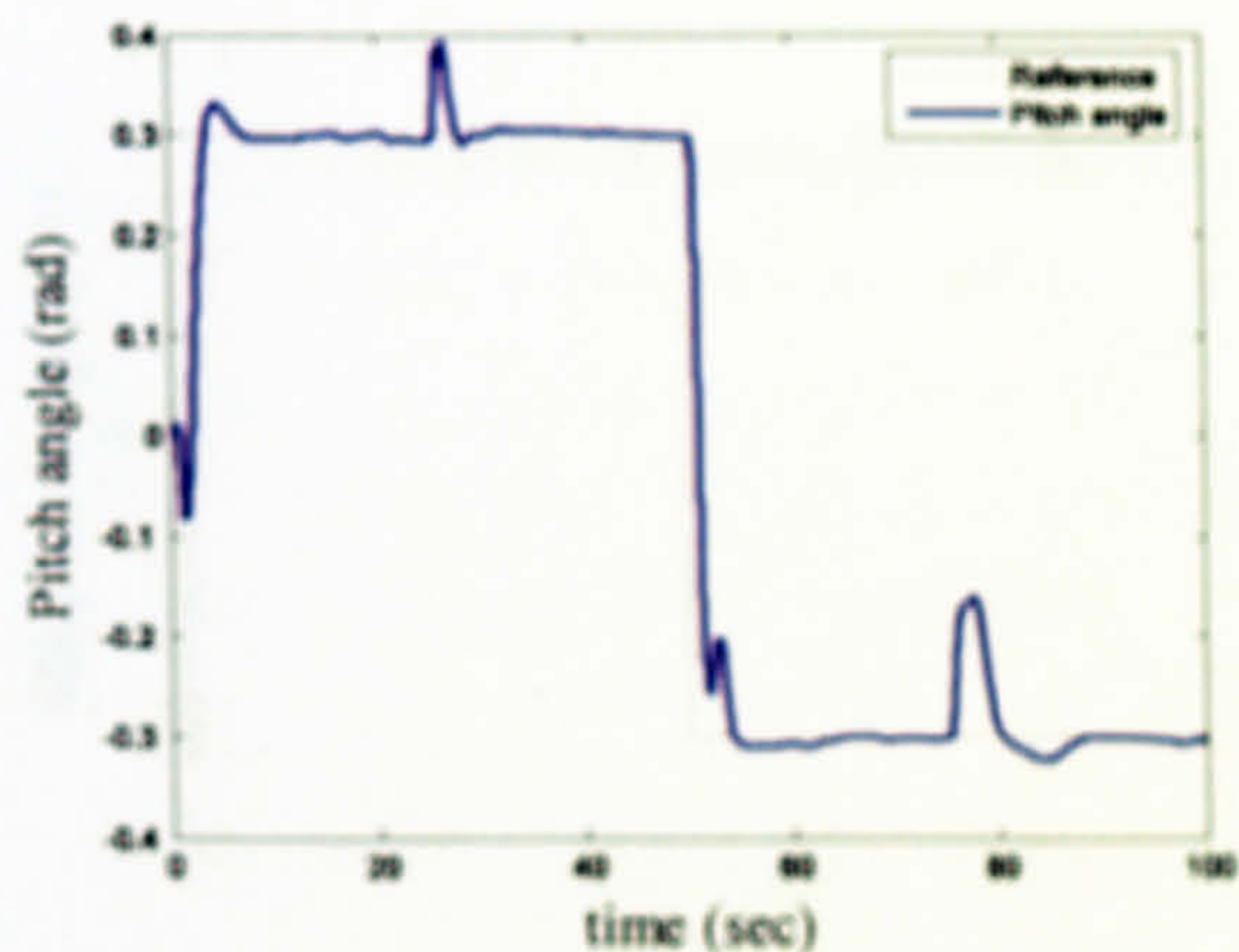


Figure 7.69 Square wave response of the vertical angle (case 5)

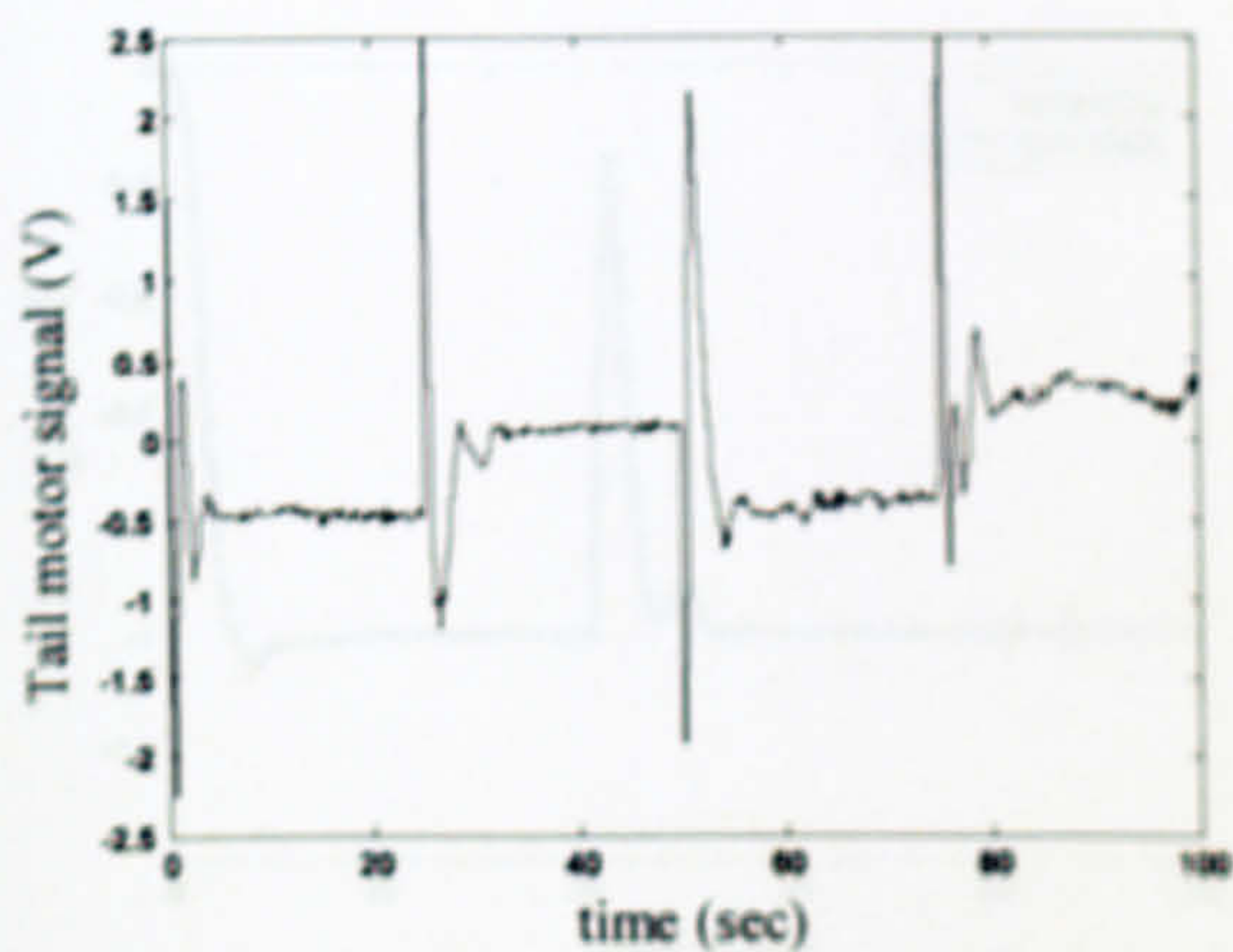


Figure 7.70 Tail motor manipulated signal, U_a , (case 5)

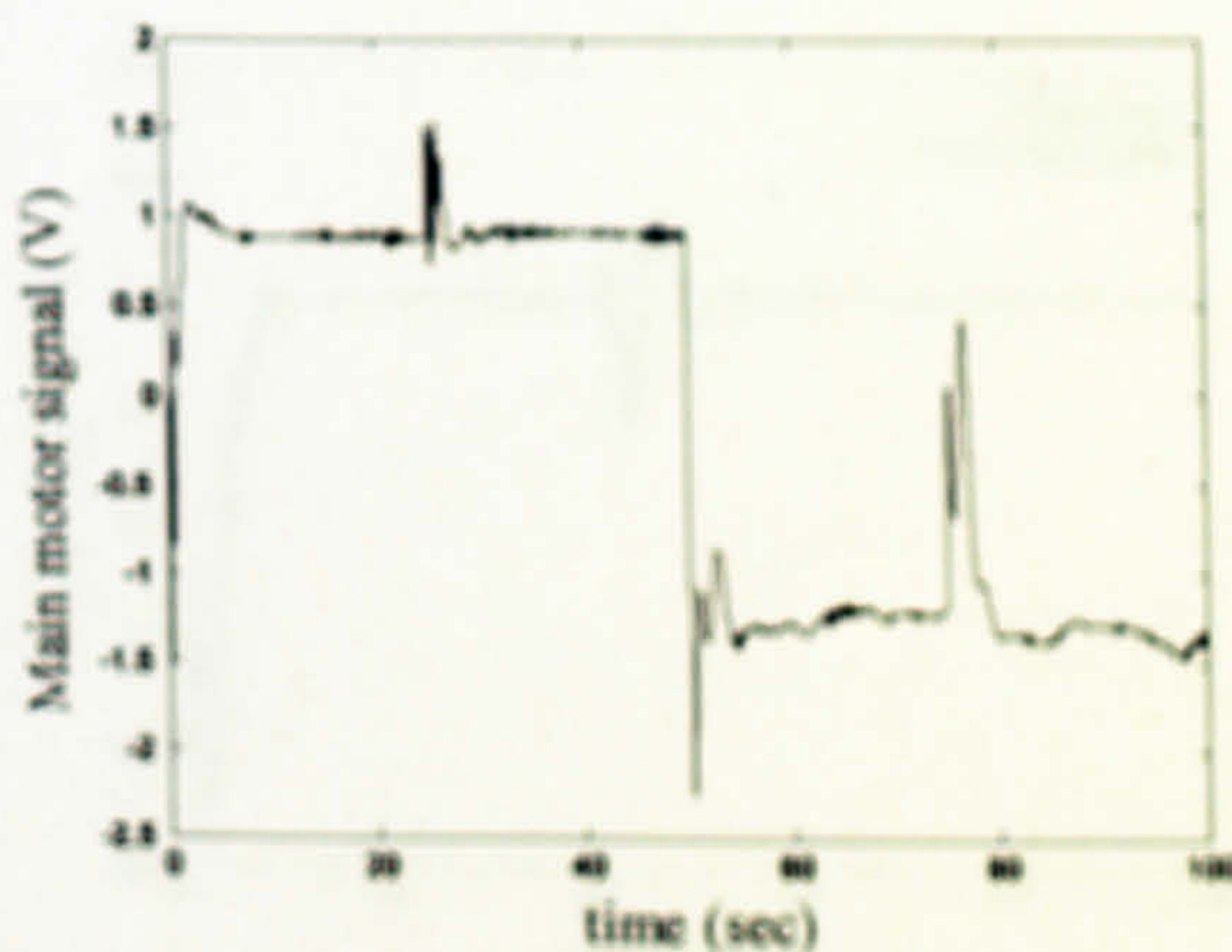


Figure 7.71 Main motor manipulated signal, U_s , (case 5)

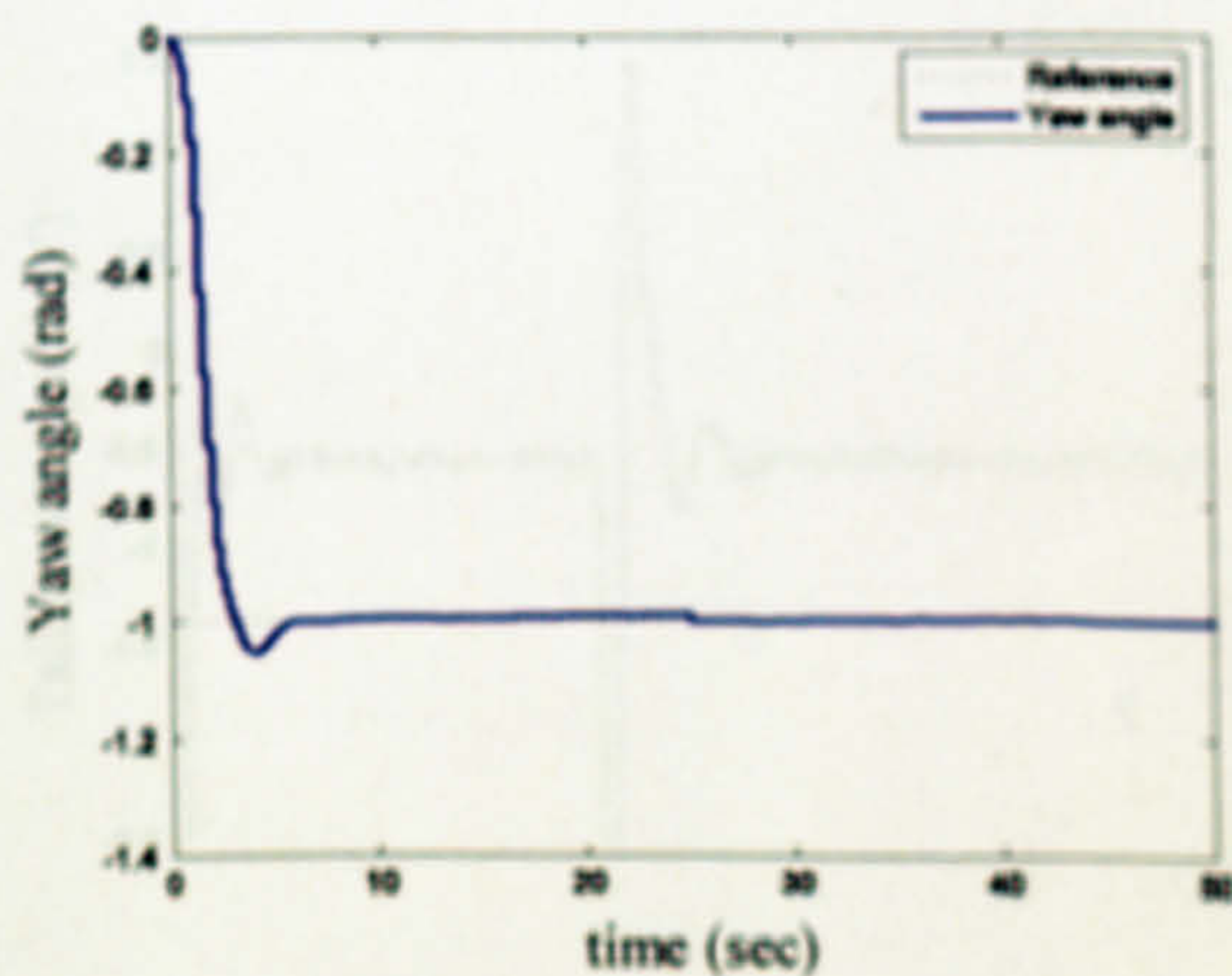


Figure 7.72 Step response of the horizontal angle (case 6)

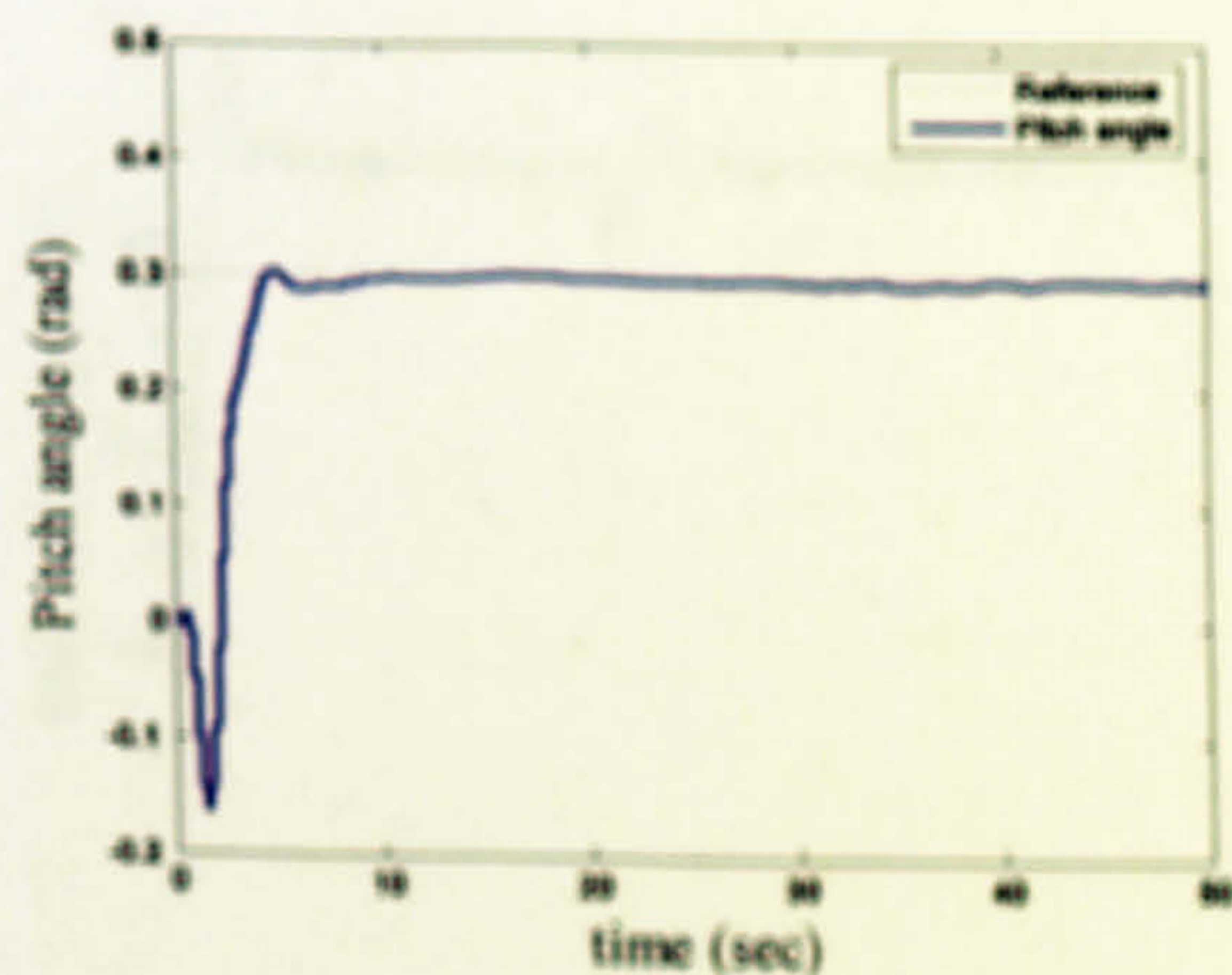


Figure 7.73 Step response of the vertical angle (case 6)

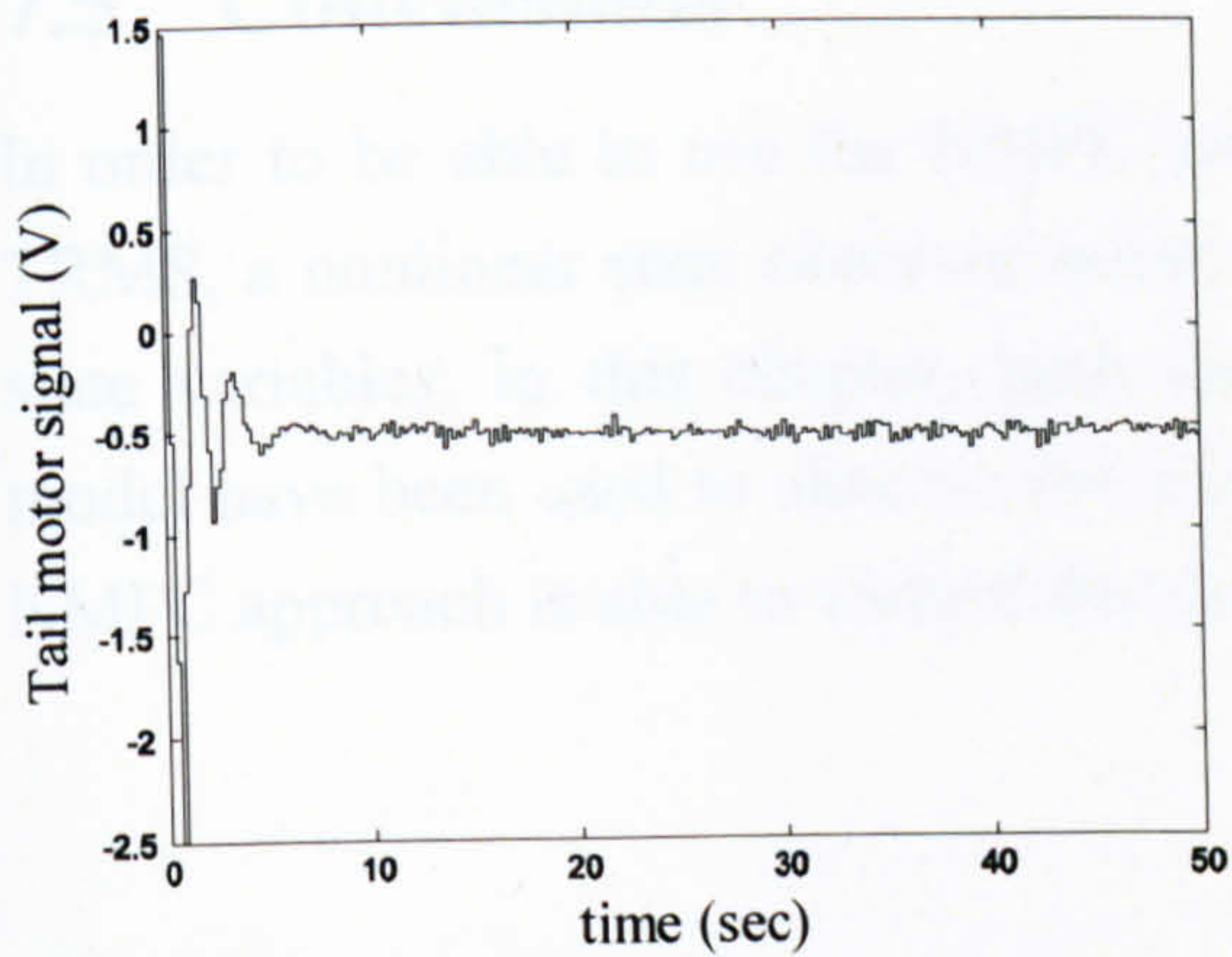


Figure 7.74 Tail motor manipulated signal, U_h , (case 6)

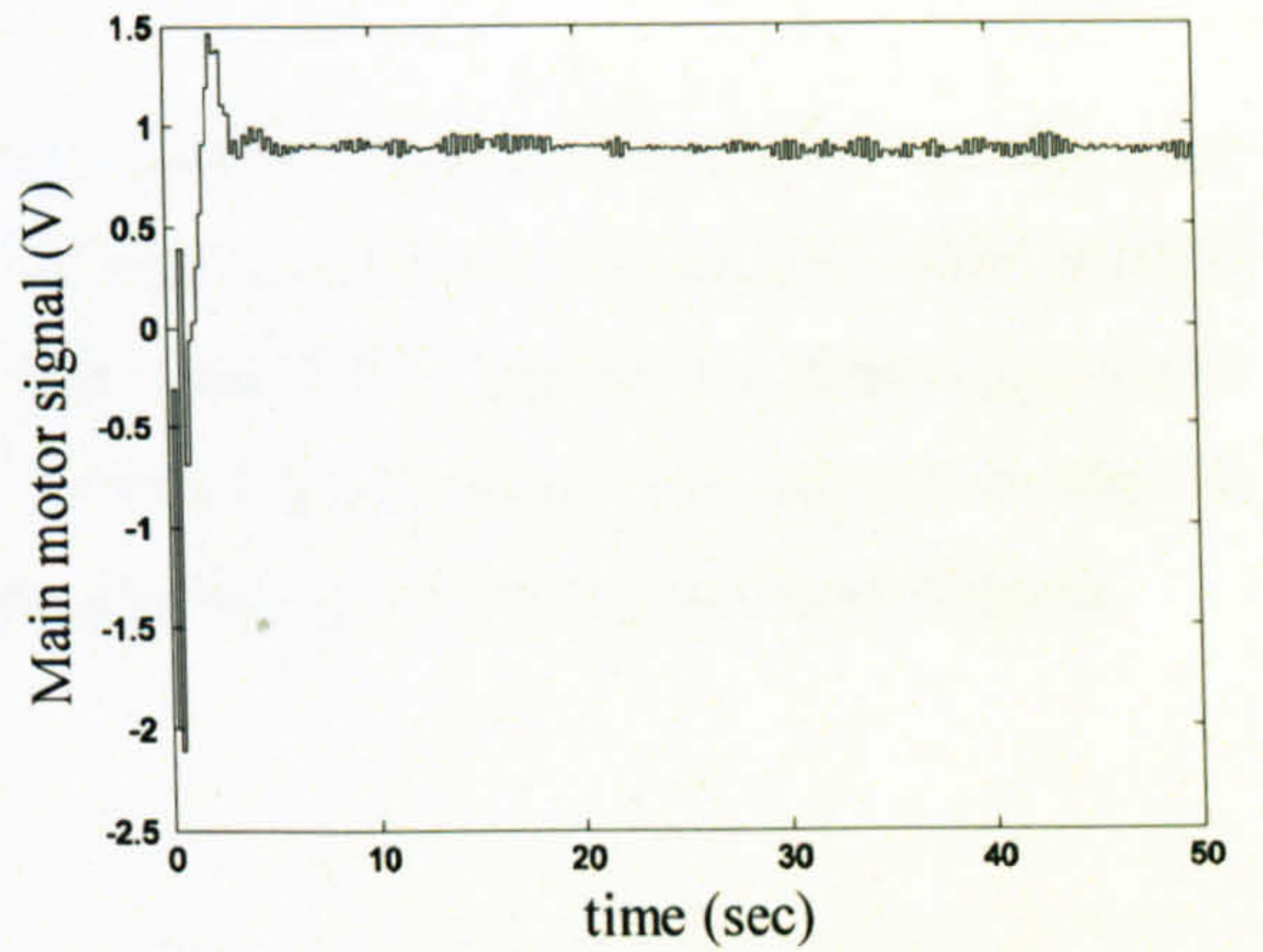


Figure 7.75 Main motor manipulated signal, U_v , (case 6)

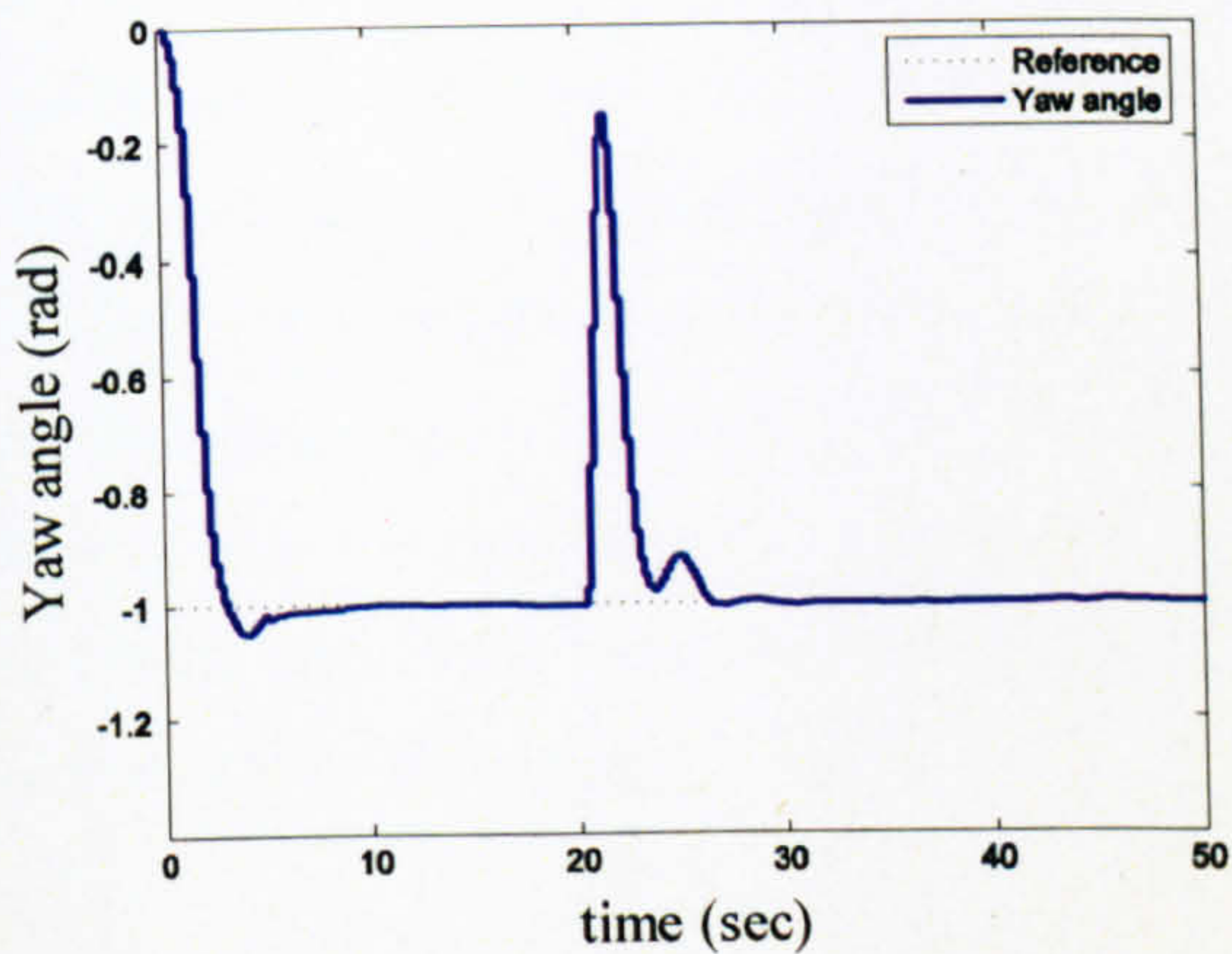


Figure 7.76 Step response of the horizontal angle with external force (case 7)

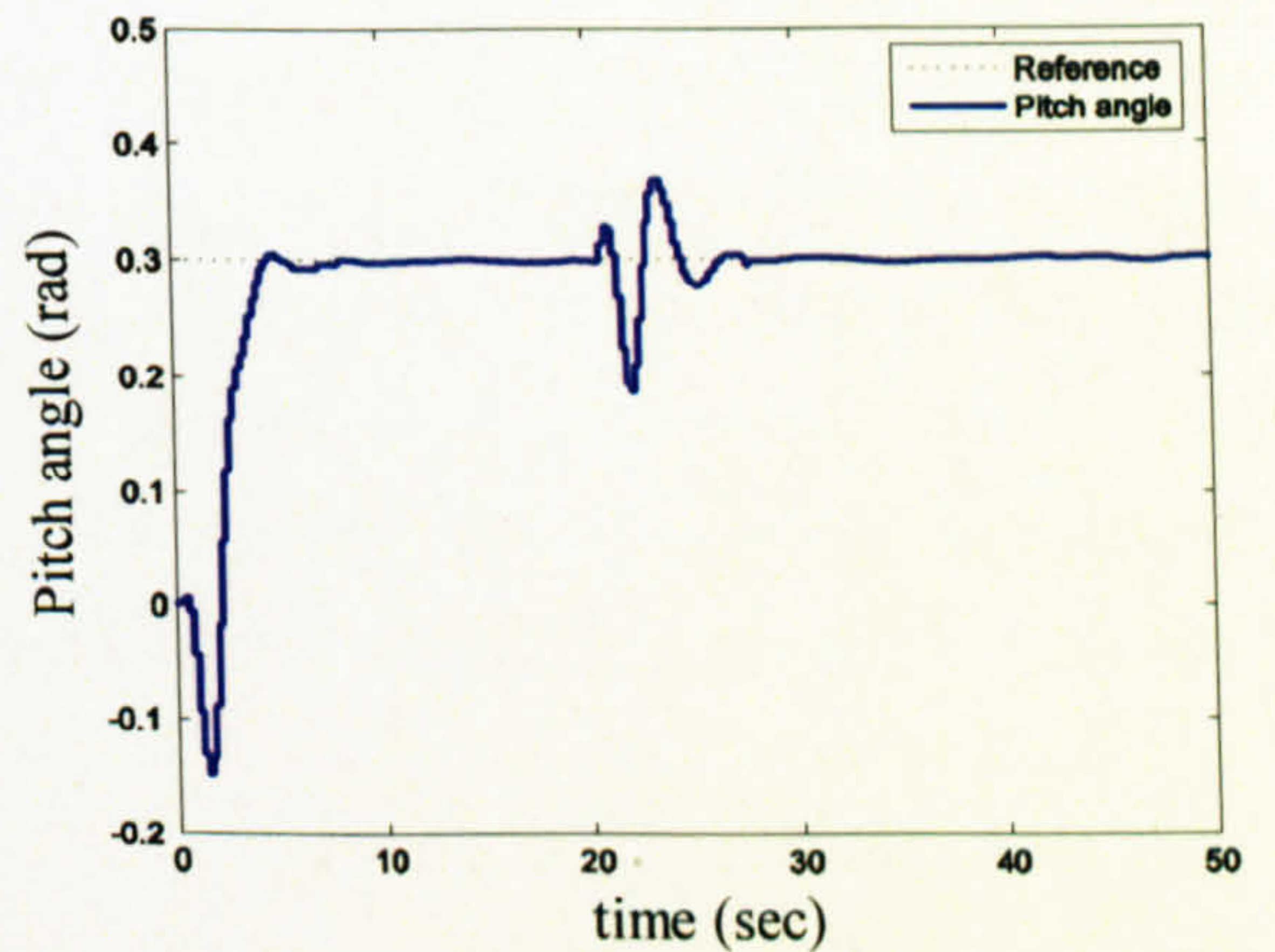


Figure 7.77 Step response of the vertical angle (case 7)

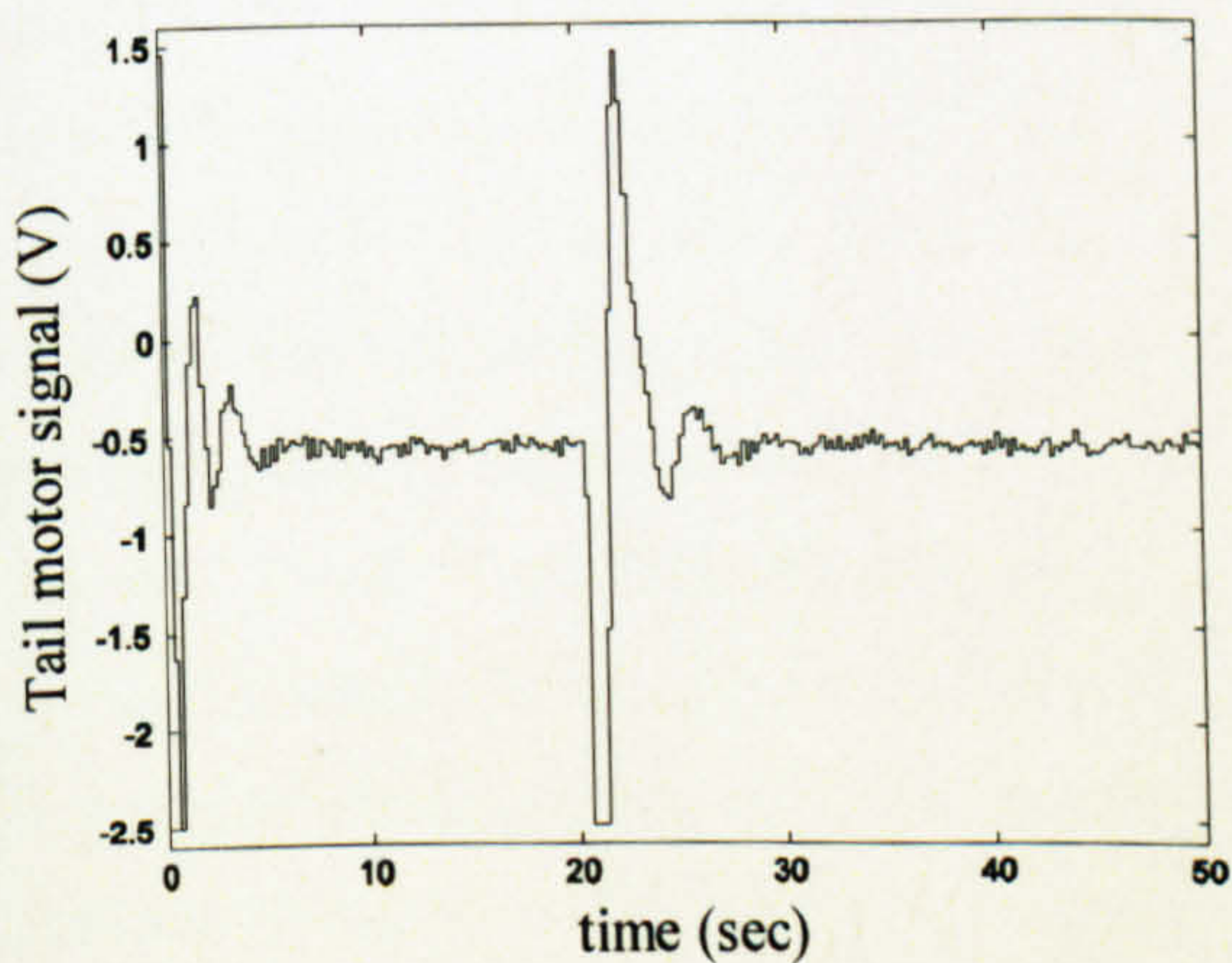


Figure 7.78 Tail motor manipulated signal, U_h , (case 7)

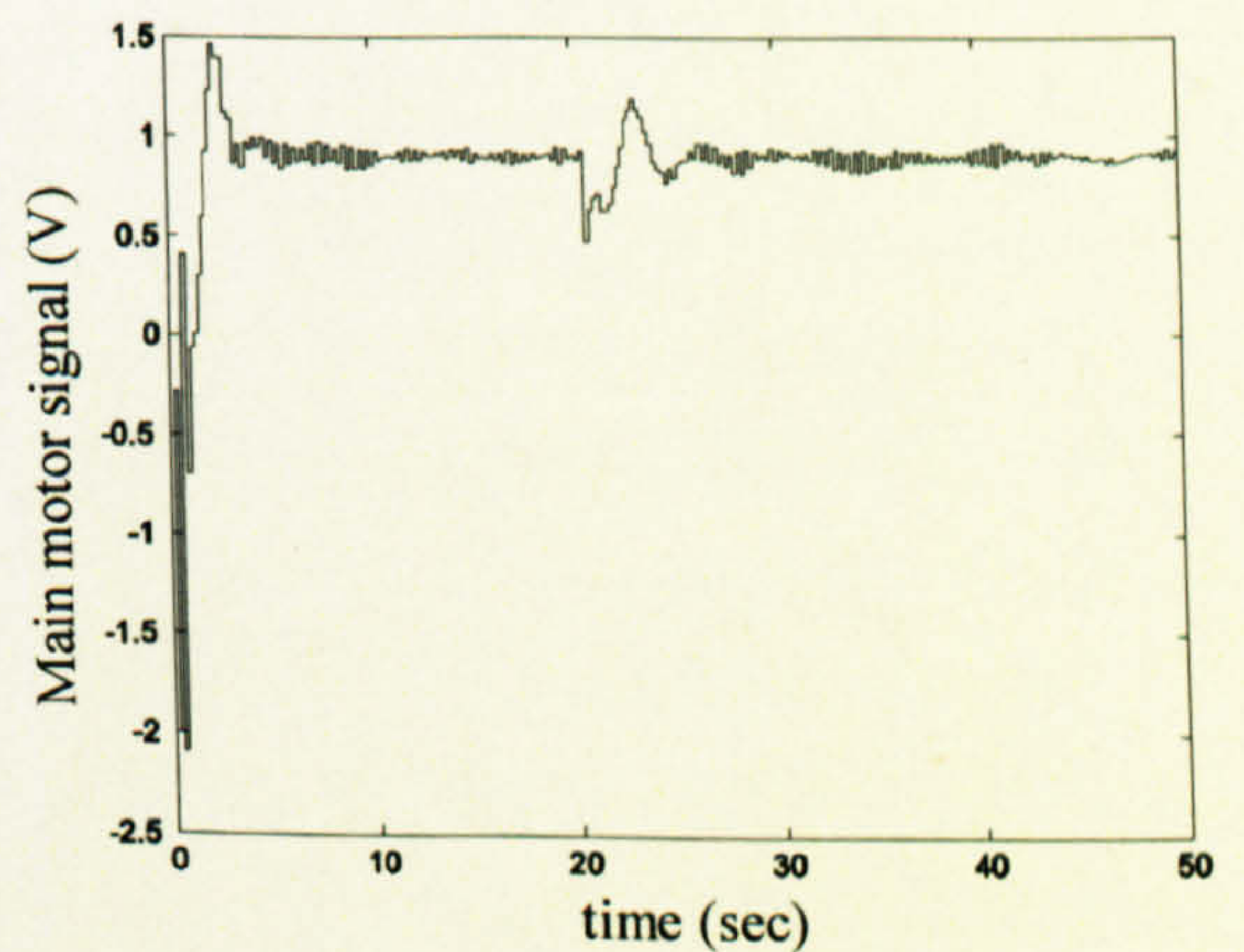


Figure 7.79 Main motor manipulated signal, U_v , (case 7)

7.5 Conclusions

In order to be able to use the NMPC technique developed in Chapter 6 on the real TRMS, a nonlinear state observer needs to be employed to estimate the inaccessible state variables. In this chapter, both the UKF and EKF based on state-dependent model have been used to observe the state vector of the system. The output-feedback NMPC approach is able to control the system according to the predefined criteria.

CHAPTER 8

Robust Model Predictive Control

This chapter investigates a robust MPC approach for the TRMS. The nonlinear dynamic model of the system has been used to design the robust controller. On the basis of the nonlinear model, a polytopic model has been developed considering the nonlinearities and uncertainties of the plant. To this end, a subset of polytopic model is extracted to cover the necessary dynamics of the system in order to make the optimisation problem more feasible. In other words, the subset of polytopic model mitigates the stringent conditions imposed due to robustness. However, some additional conditions on inputs, outputs and states may need to be imposed in order to prevent any violation from the polytope. When the optimisation problem is infeasible, the polytope is redesigned and a new optimisation problem is formulated accordingly. Note that, the robust MPC approach based on LMI has been used as a regulator in this study.

8.1 Introduction

Since any plant-model is an approximation of its corresponding real system, model uncertainty, also known as model mismatch, exists more or less in any case. Also, in stability analysis, a plant-model is assumed to be perfect and so this analysis is not valid anymore in the face of model mismatch. Therefore, robustness of any control system, i.e. satisfying some properties such as stability, constraints fulfilment and performance in the presence of uncertainty, needs to be considered to obtain realistic results. Note that, the discrepancy between robustness analysis and robustness synthesis needs to be clarified. Robustness analysis means checking the robustness of a pre-designed control system; however, robustness synthesis implies designing a control system to be robust. As mentioned in [78], the robustness analysis of a model-based predictive control is more complicated than its robustness synthesis. Here, we focus on robust MPC synthesis, i.e. an MPC is designed in such a way to be robust in terms of stability, performance and constraints fulfilment in the face of uncertainty. Robustness of MPC techniques is a challenging issue because firstly,

current versions of MPC are inherently unable to deal explicitly with model uncertainties, secondly, most investigations on robustness are related to unconstrained systems, and finally, the complexity originated from constraints causes the control system to have high computational burden and therefore, cannot be implementable in real time.

The first stage of robustness synthesis is to define the type of uncertainty according to the *a-priori* knowledge of the system. Although the uncertainty is vague, any specification and characteristic of uncertainty, e.g. being structured and bounded, can be helpful in order to achieve less conservative results. After defining the model uncertainty, one needs to establish the requirements of robust control. In most cases robust stability is the minimum requirement, and other requirements such as robust performance and robust constraints fulfilment may also be necessary to satisfy as in constrained robust MPC algorithms. State availability is the other matter that should be taken into consideration before designing a robust controller. In other words, one has to realise whether a state observer is required (output-feedback paradigm) or state variables are accessible (state-feedback paradigm). It is clear that the combination of separately designed robust controller and robust state observer cannot, in general, guarantee the robustness of the resultant closed-loop system [84], [93]. Therefore, the complexity of an output-feedback robust control design significantly increases compared to a state-feedback robust control.

This work is based on the approach proposed in [85] to robustly control a nonlinear system using MPC. Here, an approach is introduced to find a subset of polytopic model (sub-polytope) in order to form the optimisation problem to be more feasible. This causes to alleviate the severe conditions originated from the robustness criteria; however, some additional conditions may need to be imposed on the optimisation problem.

8.2 Polytopic Model

Consider a nonlinear discrete state-space plant-model,

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) \\ \mathbf{y}(k) &= \mathbf{h}(\mathbf{x}(k)) \end{aligned} \quad (8.1)$$

In general, this nonlinear system can approximately be represented as a linear time-

varying system or more generally as a state-dependent model. A polytopic model can be obtained from the linear time-varying system. Assume a discrete state-space linear time-varying system as,

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}(k)\mathbf{x}(k) \\ [\mathbf{A}(k) | \mathbf{B}(k) | \mathbf{C}(k)] &\in \Omega \end{aligned} \quad (8.2)$$

where Ω is a convex hull and can be defined using its vertices,

$$\Omega = \text{Co}\{[\mathbf{A}_1 | \mathbf{B}_1 | \mathbf{C}_1], [\mathbf{A}_2 | \mathbf{B}_2 | \mathbf{C}_2], \dots, [\mathbf{A}_L | \mathbf{B}_L | \mathbf{C}_L]\} \quad (8.3)$$

where L is the number of vertices, and $[\mathbf{A}_i | \mathbf{B}_i | \mathbf{C}_i]$ is the i -th vertex. It is clear that any system belongs to Ω can be found according to the vertex systems. Figure 8.1 shows the graphical representation of a polytopic uncertainty.

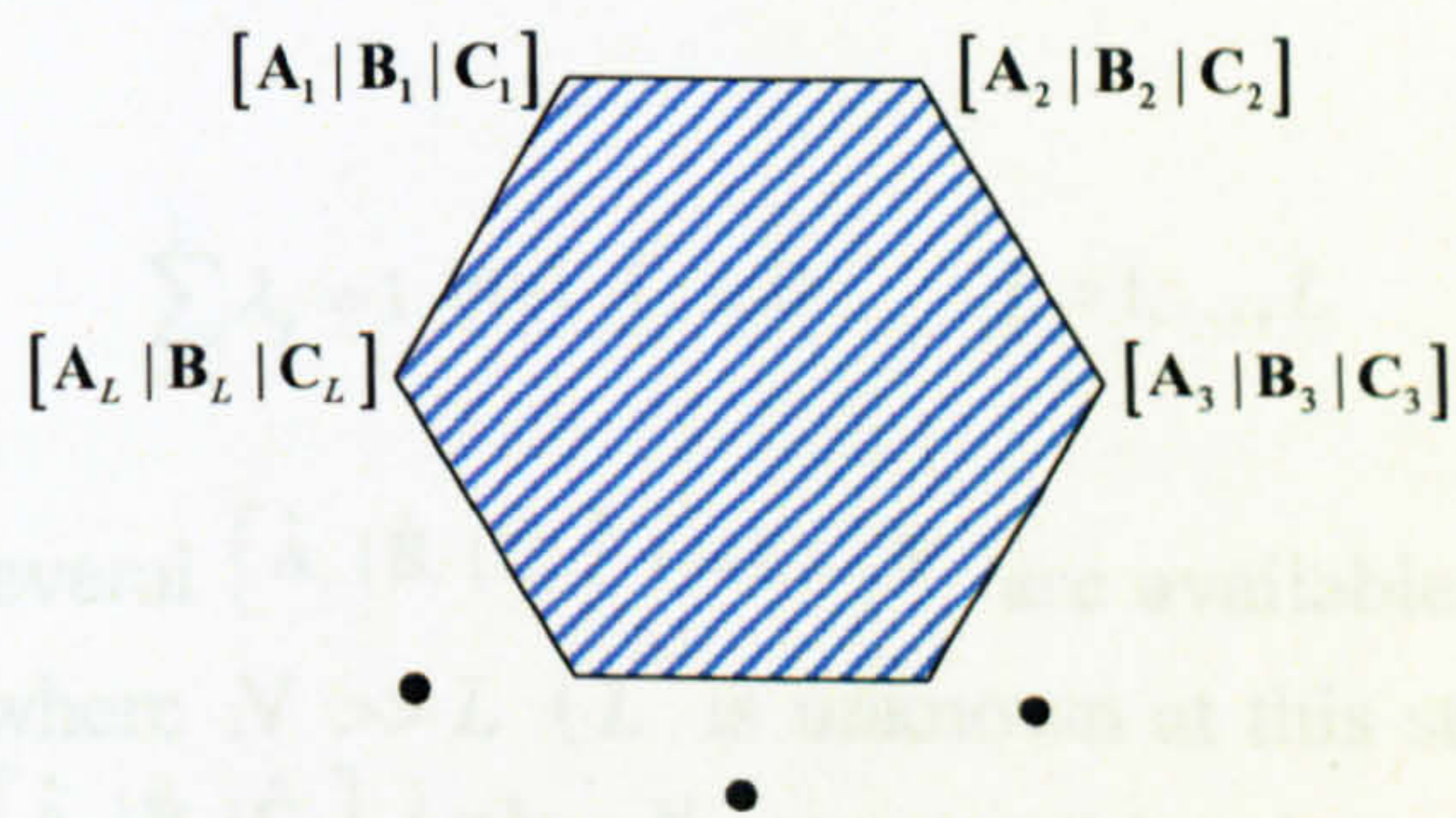


Figure 8.1 The graphical representation of a polytopic uncertainty [85]

Consider the nonlinear state-space system as shown in (8.1), the aim is to find a set of L linear systems as the vertices of the polytope Ω in such a way that all other possible linear systems lie inside this polytope. As pointed out earlier, it is clear that any possible linear model of the nonlinear system can be calculated according to a linear combination of the vertex systems as,

$$[\mathbf{A} | \mathbf{B} | \mathbf{C}] = \sum_{i=1}^L \lambda_i [\mathbf{A}_i | \mathbf{B}_i | \mathbf{C}_i] \quad (8.4)$$

where,

$$\sum_{i=1}^L \lambda_i = 1 \text{ and } \lambda_i \geq 0 \text{ for } i = 1, 2, \dots, L$$

Therefore, it is reasonable to consider the nonlinear system as a polytopic system with known vertices. Any analysis and synthesis on this polytopic system can be valid for the original nonlinear system [85].

The nonlinear system in (8.1) can be linearised around a finite number, N , of operating points in order to cover almost every operation region,

$$\begin{cases} \mathbf{x}(k+1) = \hat{\mathbf{A}}_i \mathbf{x}(k) + \hat{\mathbf{B}}_i \mathbf{u}(k) \\ \mathbf{y}(k) = \hat{\mathbf{C}}_i \mathbf{x}(k) \end{cases} \quad i = 1, \dots, N \quad (8.5)$$

Note that it is also possible to create more linear model due to uncertainties of the plant-model beside those obtained due to nonlinearity of the system. Again, assume all sets of matrices $[\hat{\mathbf{A}}_i | \hat{\mathbf{B}}_i | \hat{\mathbf{C}}_i]$ $i = 1, \dots, N$ lie in a set, Ω . In order to define this polytopic set one needs to find its vertices and introduce the polytope as a convex hull with known vertices as in (8.3). Any $[\hat{\mathbf{A}}_i | \hat{\mathbf{B}}_i | \hat{\mathbf{C}}_i]$ can be found using a linear combination of the vertices $[\mathbf{A}_j | \mathbf{B}_j | \mathbf{C}_j]$ as follows,

$$[\hat{\mathbf{A}}_i | \hat{\mathbf{B}}_i | \hat{\mathbf{C}}_i] = \sum_{j=1}^L \lambda_j [\mathbf{A}_j | \mathbf{B}_j | \mathbf{C}_j] \quad (8.6)$$

subject to,

$$\sum_{j=1}^L \lambda_j = 1 \quad \text{and} \quad \lambda_j \geq 0 \quad j = 1, \dots, L$$

For the time being, several $[\hat{\mathbf{A}}_i | \hat{\mathbf{B}}_i | \hat{\mathbf{C}}_i]$ $i = 1, \dots, N$ are available and the aim is to find $[\mathbf{A}_j | \mathbf{B}_j | \mathbf{C}_j]$ $j = 1, \dots, L$ where $N \gg L$ (L is unknown at this stage). At the first stage of the problem, all $[\hat{\mathbf{A}}_i | \hat{\mathbf{B}}_i | \hat{\mathbf{C}}_i]$ $i = 1, \dots, N$ are considered to be vertices and check whether there is a set of non-negative parameters λ_i summing to one to satisfy the following equation,

$$[\hat{\mathbf{A}}_j | \hat{\mathbf{B}}_j | \hat{\mathbf{C}}_j] = \sum_{\substack{i=1, \\ i \neq j, i \notin O}}^N \lambda_i [\hat{\mathbf{A}}_i | \hat{\mathbf{B}}_i | \hat{\mathbf{C}}_i] \quad j = 1, \dots, N \quad (8.7)$$

where, O is the set of all omitted systems so far. If the problem is feasible for any j then, $[\hat{\mathbf{A}}_j | \hat{\mathbf{B}}_j | \hat{\mathbf{C}}_j]$ must be purged from the set, otherwise it needs to be kept. This procedure should be repeated for all $[\hat{\mathbf{A}}_i | \hat{\mathbf{B}}_i | \hat{\mathbf{C}}_i]$ in order to keep only the vertices of Ω .

8.3 Problem Formulation

To formulate the problem assume each $[\hat{\mathbf{A}}_i | \hat{\mathbf{B}}_i | \hat{\mathbf{C}}_i]$ has q entries therefore, we can form N vectors, $\mathbf{v}_i \in \mathbb{R}^q$ each one corresponding to one $[\hat{\mathbf{A}}_i | \hat{\mathbf{B}}_i | \hat{\mathbf{C}}_i]$,

$$[\hat{\mathbf{A}}_i | \hat{\mathbf{B}}_i | \hat{\mathbf{C}}_i] \rightarrow \mathbf{v}_i = \begin{bmatrix} v_{1i} \\ v_{2i} \\ \vdots \\ v_{qi} \end{bmatrix} \quad i = 1, \dots, N \quad (8.8)$$

At the first iteration the following constrained simultaneous linear equations are considered to check whether they have at least one set of feasible solution,

$$\mathbf{v}_1 = \sum_{i=2}^N \lambda_i \mathbf{v}_i \quad \text{subject to} \quad \sum_{i=2}^N \lambda_i = 1 \quad \text{and} \quad \lambda_i \geq 0 \quad i = 2, \dots, N. \quad (8.9)$$

This system can be rewritten as the following matrix form,

$$\begin{bmatrix} v_{12} & v_{13} & \cdots & v_{1N} \\ v_{22} & v_{23} & \cdots & v_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ v_{q2} & v_{q3} & \cdots & v_{qN} \\ 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \lambda_2 \\ \lambda_3 \\ \vdots \\ \lambda_N \end{bmatrix} = \begin{bmatrix} v_{11} \\ v_{21} \\ \vdots \\ v_{q1} \\ 1 \end{bmatrix} \quad \text{subject to} \quad \lambda_i \geq 0 \quad i = 2, \dots, N \quad (8.10)$$

As stated before, if the system is feasible, then the vector \mathbf{v}_1 is not a vertex of the polytope and has to be removed from the set of vertices. This procedure is repeated for all other \mathbf{v}_i .

Note that the developed polytopic model has some degrees of conservativeness. The number of primarily linear models, N , should be as large as possible to reduce the conservativeness of the polytopic model.

8.4 Robust MPC

Again, consider a plant-model described by polytopic uncertainty. Similar to the robust control of linear systems, one can use min-max method of optimisation to develop a robust MPC paradigm [81].

$$\min_{\substack{\mathbf{u}(k+i|k) \\ i=0,1,\dots,N_C}} \max_{\substack{[\mathbf{A}(k+i)|\mathbf{B}(k+i)|\mathbf{C}(k+i)] \in \Omega \\ i \geq 0}} J_{\infty}(k) \quad (8.11)$$

where N_C is the control horizon and,

$$J_{\infty}(k) = \sum_{i=0}^{\infty} [\mathbf{x}(k+i|k)^T \mathbf{Q}_1 \mathbf{x}(k+i|k) + \mathbf{u}(k+i|k)^T \mathbf{R} \mathbf{u}(k+i|k)]$$

In the above optimisation problem, the worst-case scenario is found over all the possible model realisations, the set Ω , using maximisation and then a set of N_c manipulated variables is obtained to minimise the worse-case scenario. As mentioned by Kothare *et al.* [85], this min-max problem is not tractable despite of convexity. Also as mentioned before, Zheng [83] has shown using a counterexample that the min-max problem alone cannot guarantee robust stability. A remedy has been proposed to derive an upper bound on the robust performance objective and minimise it using a constant state-feedback control law at each sample time [85],

$$\mathbf{u}(k+i|k) = \mathbf{F}(k)\mathbf{x}(k+i|k) \quad i \geq 0 \quad (8.12)$$

Assume a quadratic function $V(\mathbf{x}) = \mathbf{x}^T \mathbf{P} \mathbf{x}$ with $\mathbf{P} > 0$ and $V(\mathbf{0}) = 0$. At each time step one needs to find \mathbf{P} in such a way that the following inequality is satisfied for all $\mathbf{x}(k+i|k)$, $\mathbf{u}(k+i|k)$ with $i \geq 0$ and for any $[\mathbf{A}(k+i)|\mathbf{B}(k+i)|\mathbf{C}(k+i)] \in \Omega$ with $i \geq 0$:

$$\begin{aligned} V(\mathbf{x}(k+i+1|k)) - V(\mathbf{x}(k+i|k)) \leq \\ -[\mathbf{x}(k+i|k)^T \mathbf{Q}_1 \mathbf{x}(k+i|k) + \mathbf{u}(k+i|k)^T \mathbf{R} \mathbf{u}(k+i|k)] \end{aligned} \quad (8.13)$$

In order to have finite robust performance objective function, it is necessary to have $\mathbf{x}(\infty|k) = \mathbf{0}$ which implies $V(\mathbf{x}(\infty|k)) = 0$. Adding up (8.13) from $i = 0$ to $i = \infty$ results to,

$$-V(\mathbf{x}(k|k)) \leq -J_\infty(k) \quad (8.14)$$

and thus the upper bound of the robust performance objective can be expressed as,

$$\max_{\substack{[\mathbf{A}(k+i)|\mathbf{B}(k+i)|\mathbf{C}(k+i)] \in \Omega \\ i \geq 0}} J_\infty(k) \leq V(\mathbf{x}(k|k)) \quad (8.15)$$

As a summary, the single Lyapunov function, $V(\mathbf{x}(k|k)) = \mathbf{x}(k|k)^T \mathbf{P} \mathbf{x}(k|k)$, should be minimised at each time step, k , to obtain optimum $\mathbf{P} > 0$ and state-feedback gain $\mathbf{F}(k)$ in (8.12), subject to (8.13) and other input, output and state constraints. All the mentioned equations can be represented in the form of LMI [133], for more details see [85].

Although the proposed method by Kothare *et al.* [85] is an interesting approach for robust MPC, the condition mentioned in (8.13) is difficult to be satisfied for highly nonlinear practical systems in the case of polytopic uncertainty. In order to overcome this problem, one can form an adaptive polytopic model that is a subset of the main polytopic model and force the system by imposing additional constraints to prevent

any violation from the aforementioned subset. However, finding the optimal subset is not a straightforward task and it severely depends on the control system and also needs deep knowledge about the process. In order to clarify the problem, see Figure 8.2. Assume the regulator is supposed to drive the plant output from an initial position, E , to the origin, O .

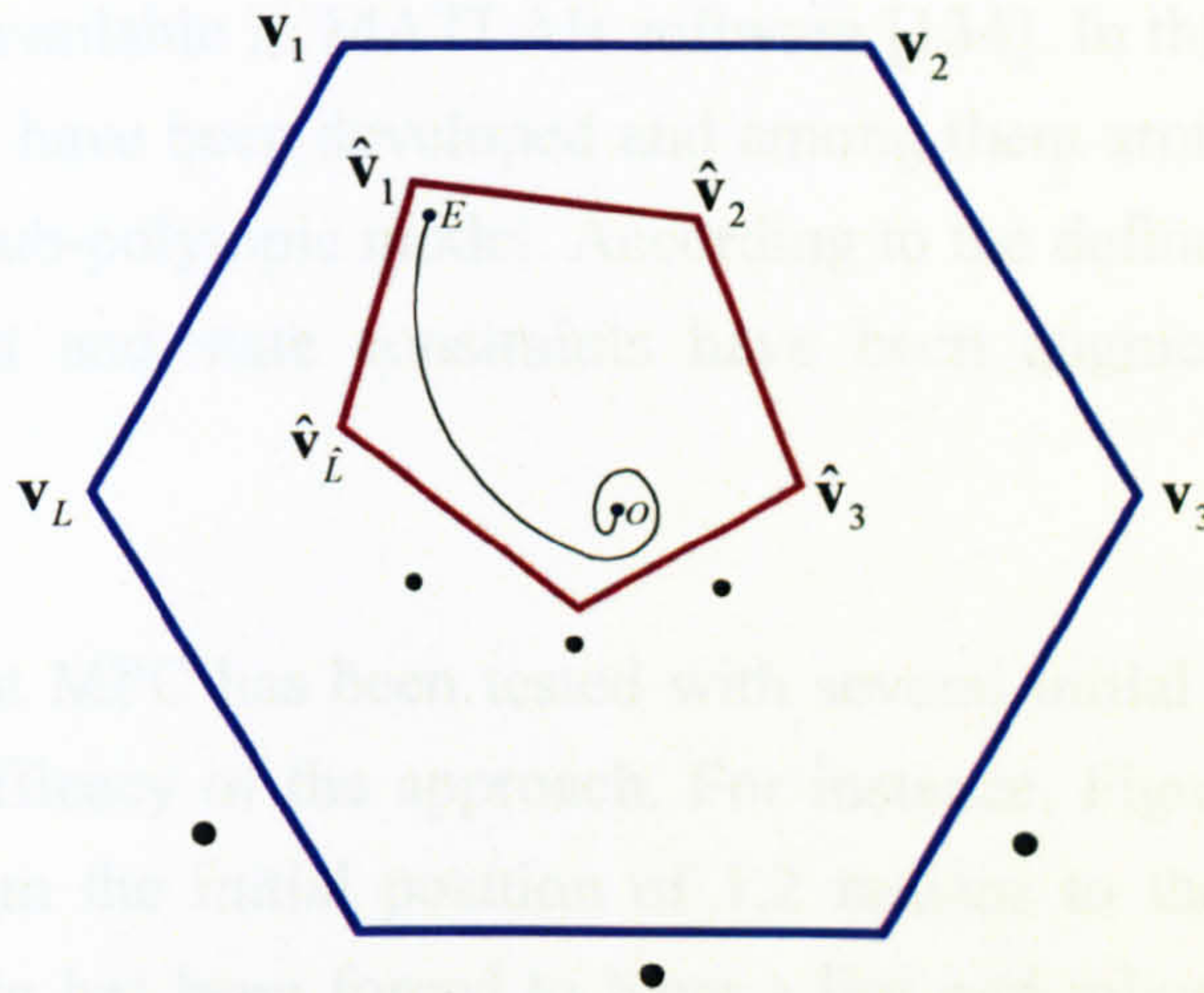


Figure 8.2 The sub-polytopic uncertainty

In Figure 8.2 the main polytopic model is depicted by a hexagon where its vertices are v_i , $i = 1, \dots, L$ and its subset is shown by a pentagon where \hat{v}_i , $i = 1, \dots, \hat{L}$ represent its vertices. Although the trend is not known as an *a-priori* path, it is possible to estimate the subset of polytopic model with some security margin and by imposing additional input, output and/or state constraints force the trend to stay inside the sub-polytope forever. Note that, in the worst case, the subset can be the same as the main polytope.

8.5 Results

The aim of this chapter is to design an infinite horizon, state-feedback robust MPC regulator controller based on polytopic model to be feasible for practical nonlinear systems. The first step of the procedure is to develop a nonlinear model of the plant as accurate as possible. Based on the nonlinear model and the initial values of states and inputs, a large number of linear models are developed to cover any possible region of the future operation points of the system, or in other words path of the inputs, states and outputs, before settling at the origin. According to these linear models, the vertices of the polytopic model are found, as formulated before. Now, the proposed robust MPC approach in [85] that cast into a set of LMI, is used to find

the optimum manipulating signals to satisfy the criteria. Moreover, some additional constraints need to be defined and imposed to the robust MPC approach to prevent any violation from the polytopic model.

Again, the plant has 2 inputs, 2 outputs and 6 states. The sampling time of the controller is set to be 0.2 seconds and the optimisation problem has been solved using LMI toolbox available in MATLAB software [134]. In this case, around 10000 initial linear models have been developed and among them around 140 vertices have been found for the sub-polytopic model. According to the defined operating regions a set of input, output and state constraints have been augmented to the previous constraints.

The proposed robust MPC has been tested with several initial values to validate the effectiveness and efficacy of the approach. For instance, Figure 8.3 shows the yaw angle trajectory from the initial position of 1.2 radians to the origin. At the same time, the pitch angle has been forced to have a fast and robust movement from the initial position of 0.65 radians to the origin as shown in Figure 8.4. The related control signals of both the horizontal and vertical channels have been depicted in Figures 8.5 and 8.6 respectively. The responses of the regulator control with another set of initial values have been shown in Figures 8.7 to 8.10. The initial values of the yaw and pitch angles related to this case are -0.65 and -0.35 respectively. Two other cases with different initial positions are also presented in Figures 8.11 to 8.18 to illustrate various aspects of the regulator controller. In all these cases, the robustness is provided as long as the trajectory lies inside the predefined polytope. If the optimisation problem is not feasible for the defined polytope, the controller needs to find another polytope with some appropriate extension and solve the new optimisation problem. However, if one considers a sufficient security margin when defining the polytope, the optimisation problem is feasible with a high probability.

In the case of the TRMS the optimisation problem is not feasible when the main polytopic model is considered; however, when a proper subset of the main polytopic is used, the optimisation problem has a comparatively higher chance to be feasible.

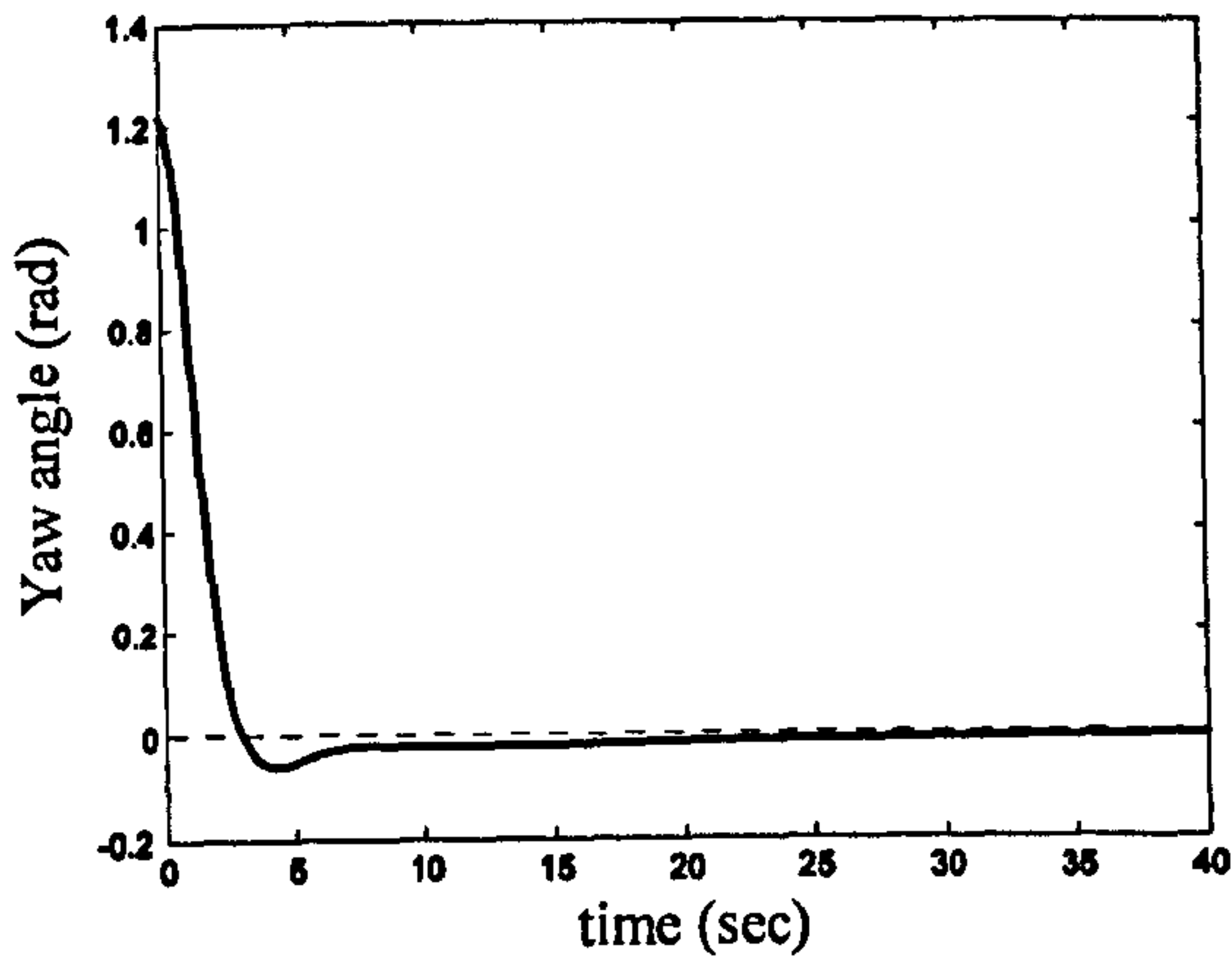


Figure 8.3 Yaw angle response with initial value of 1.2 rad (case 1)

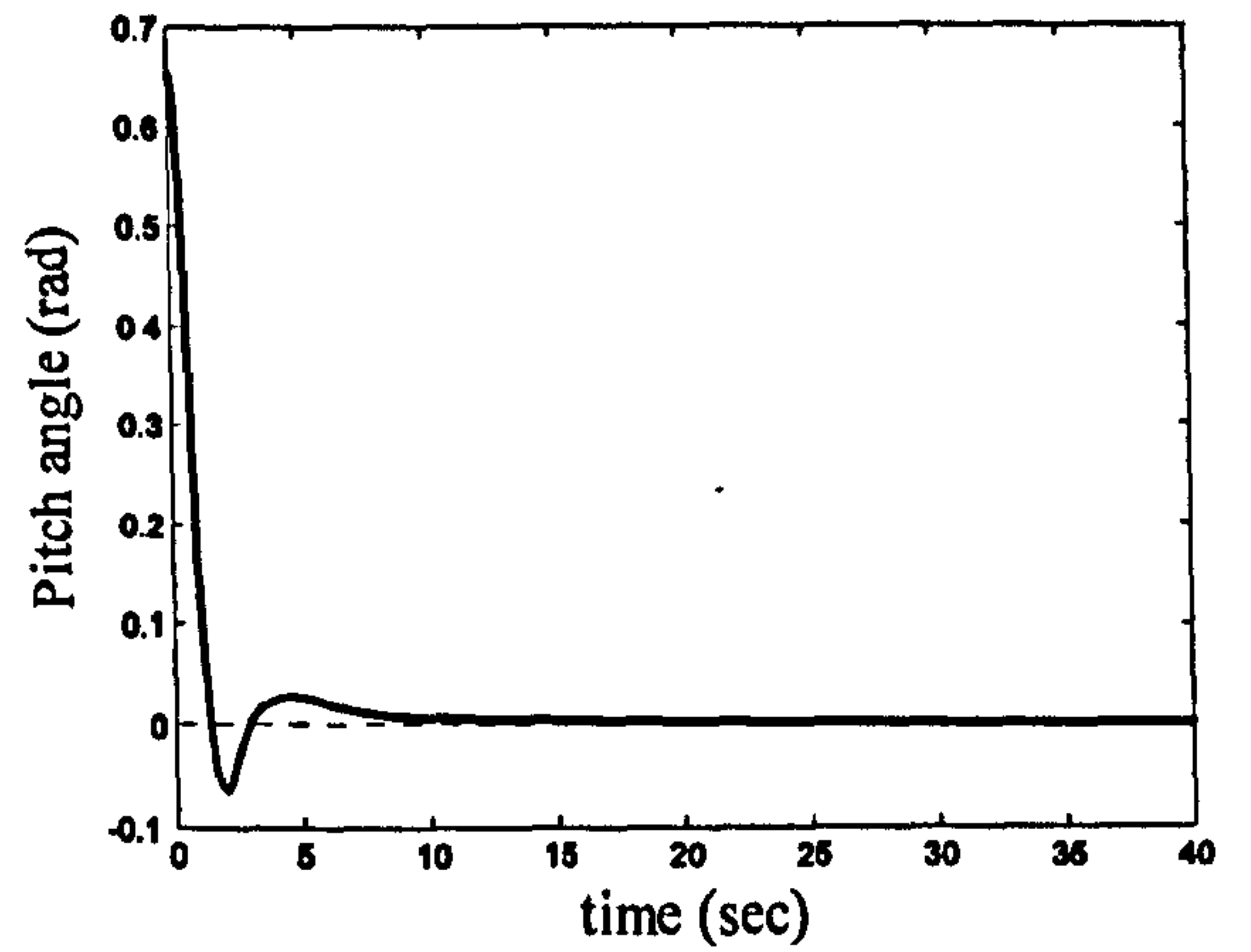


Figure 8.4 Pitch angle with initial value of 0.65 rad (case 1)

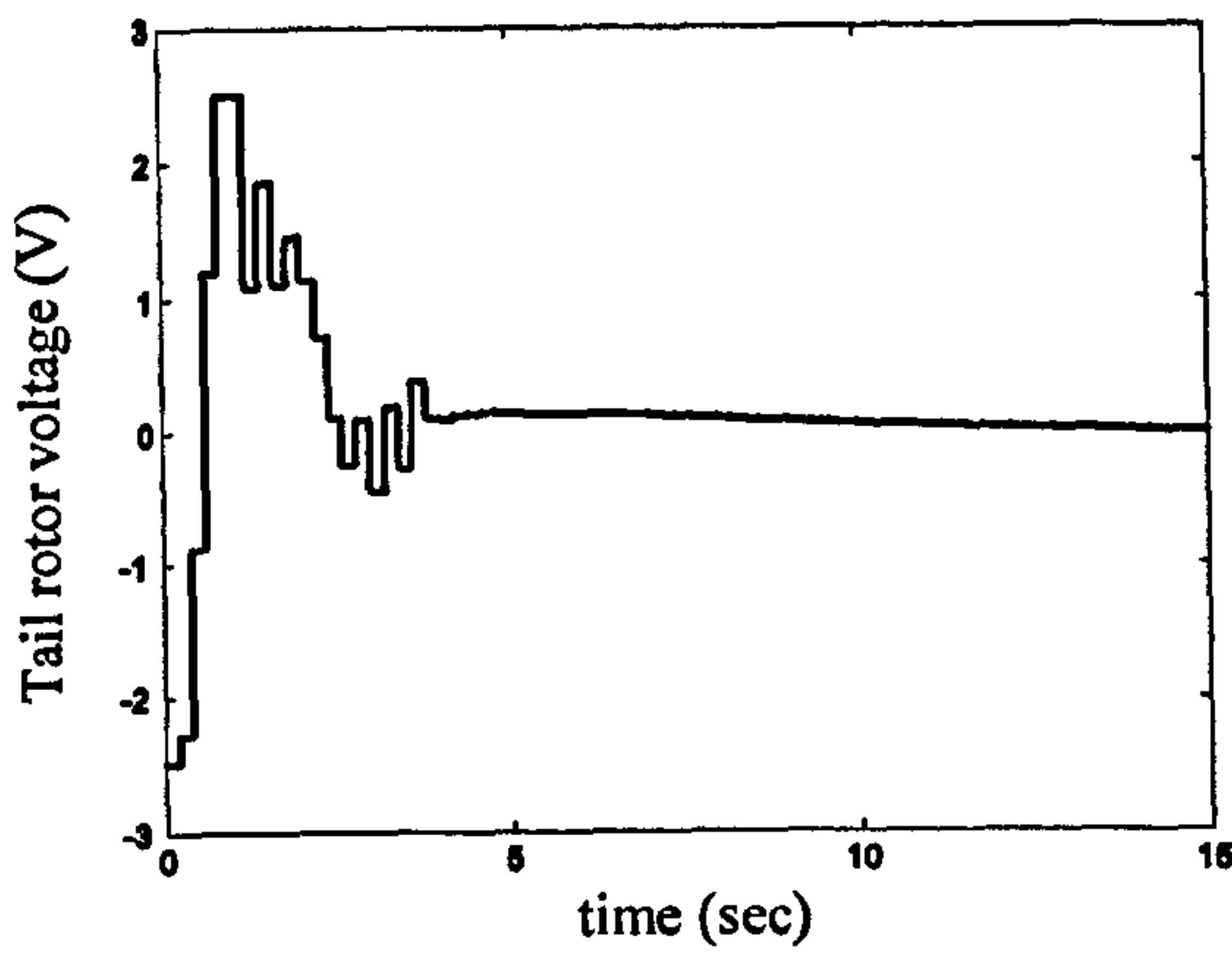


Figure 8.5 Tail rotor applied voltage (case 1)

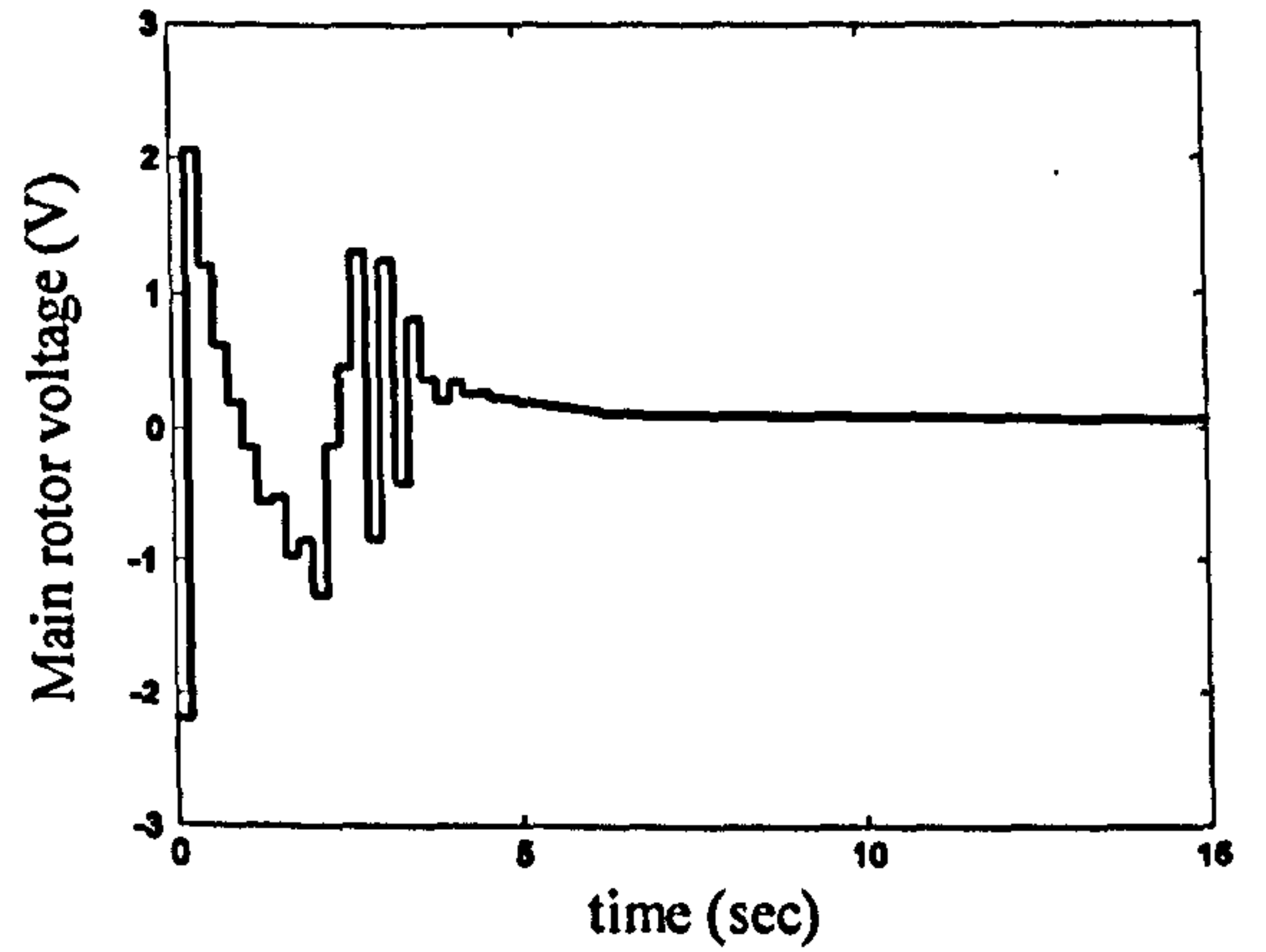


Figure 8.6 Main rotor applied voltage (case 1)

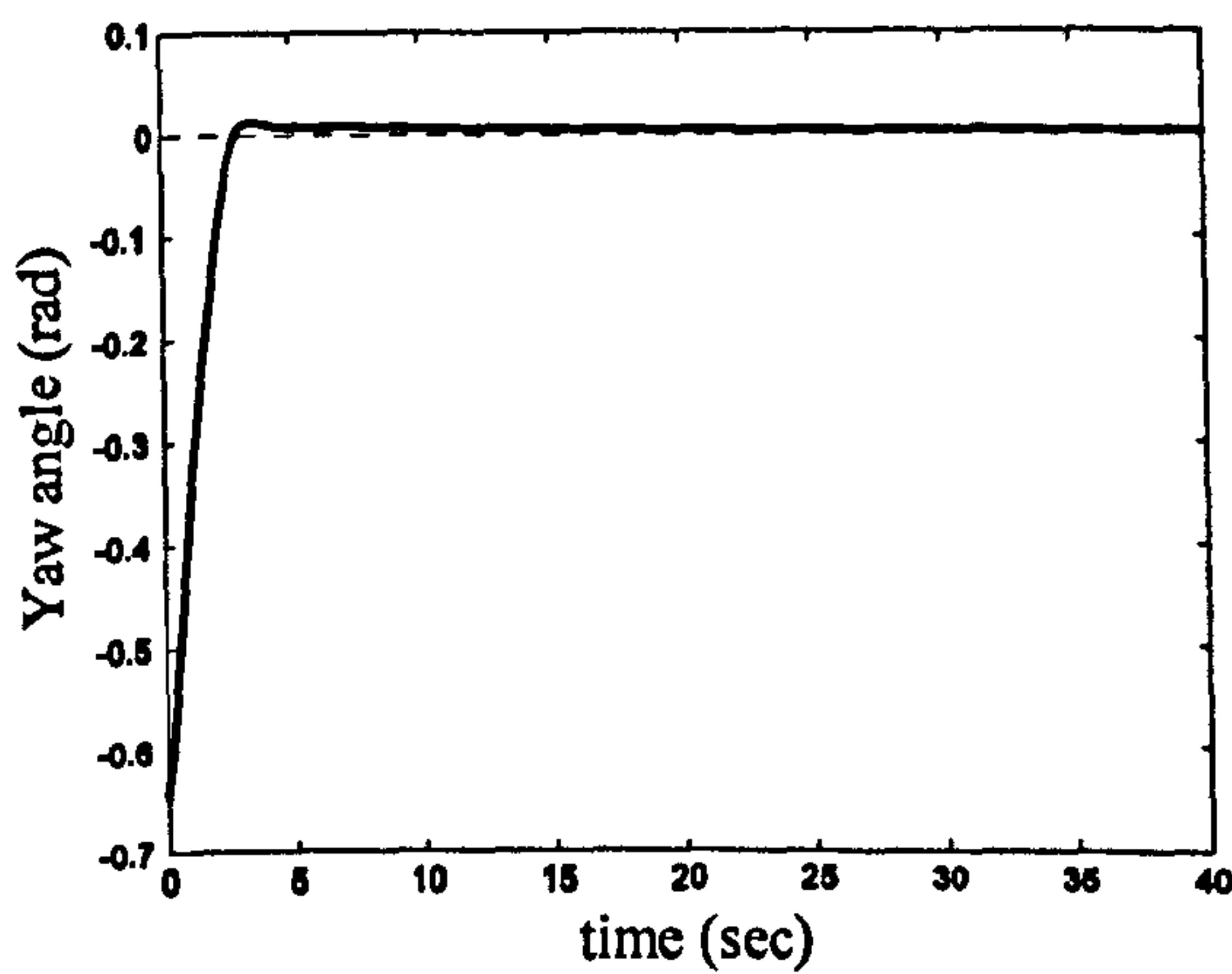


Figure 8.7 Yaw angle response with initial value of -0.65 rad (case 2)

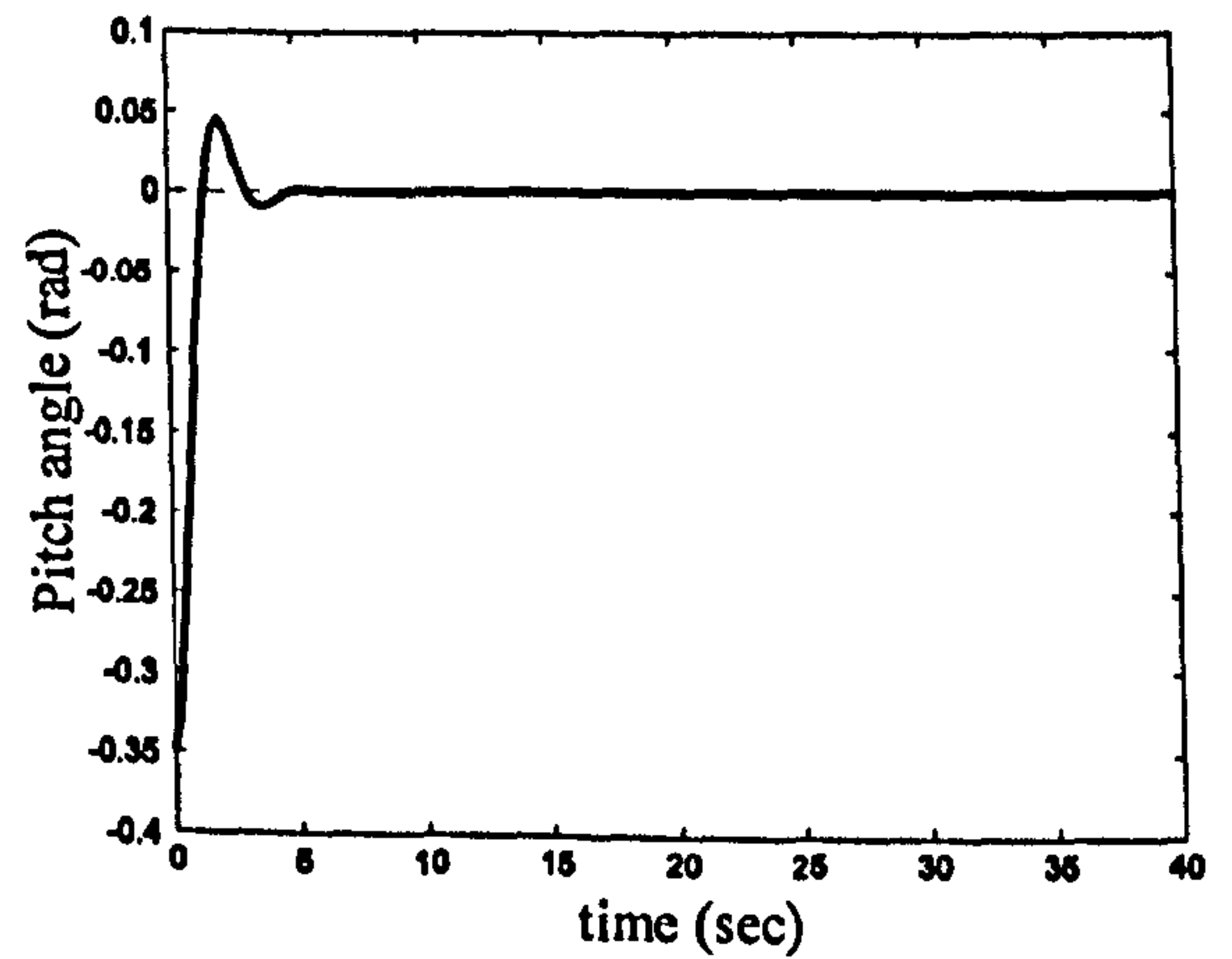


Figure 8.8 Pitch angle with initial value of -0.35 rad (case 2)

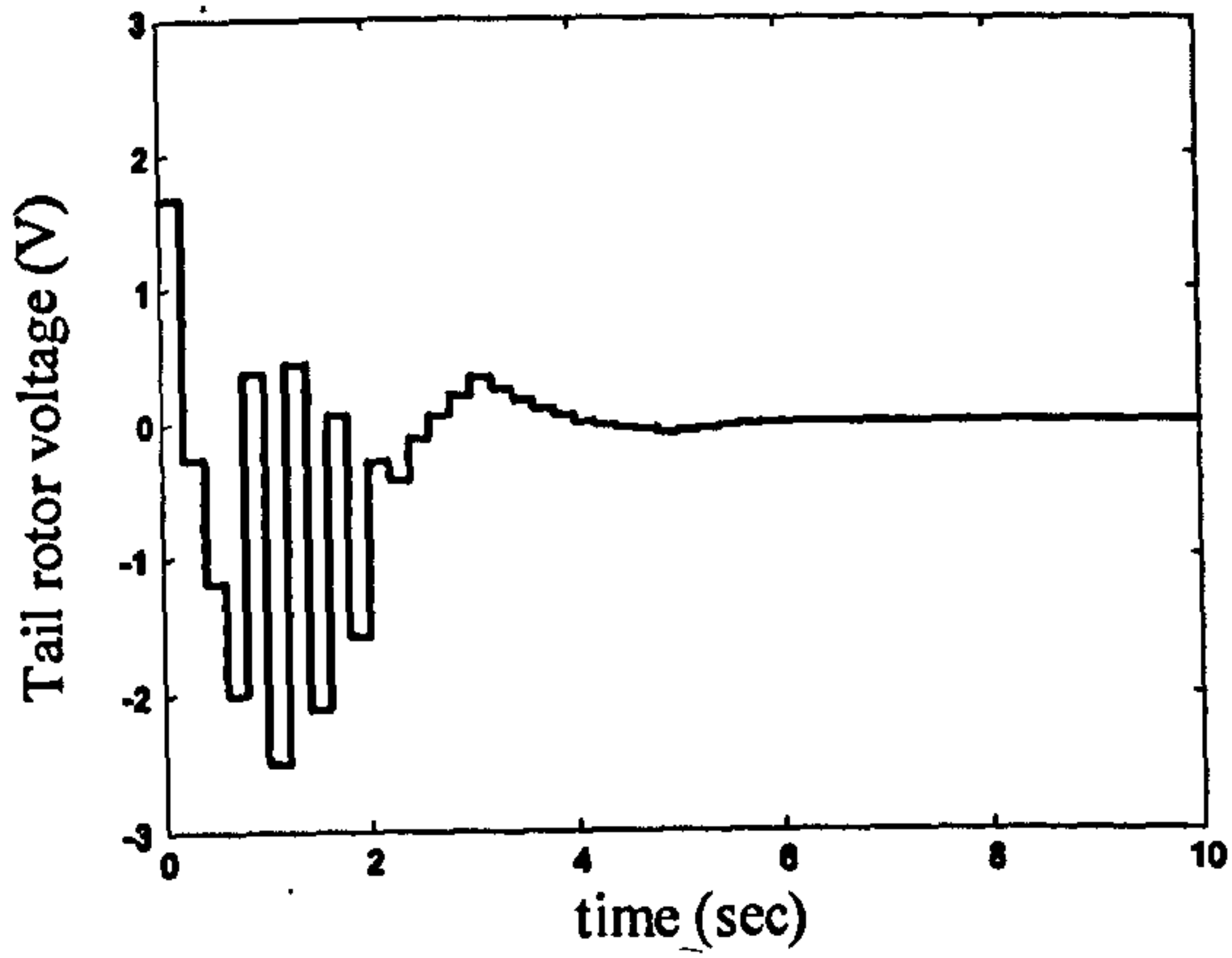


Figure 8.9 Tail rotor applied voltage (case 2)

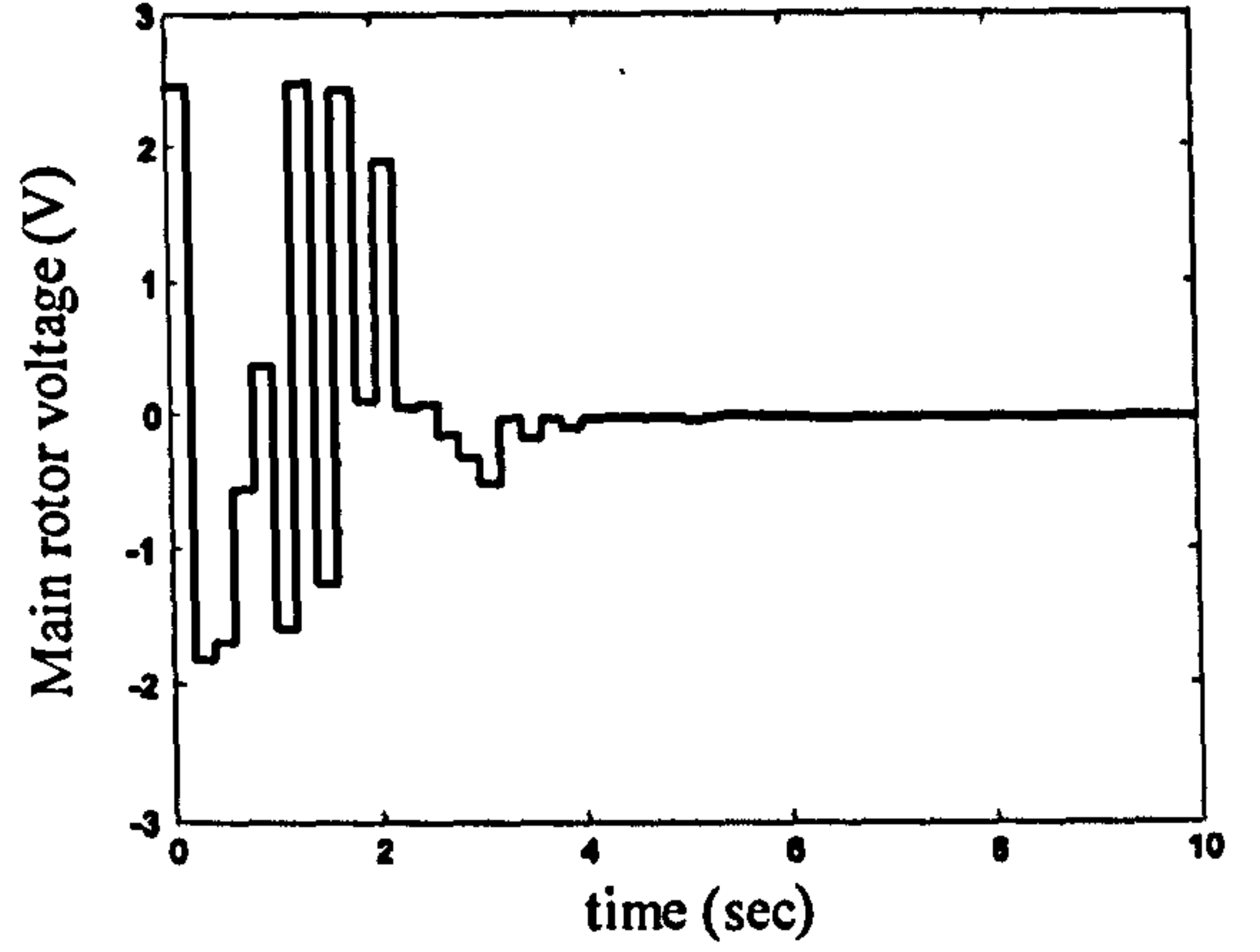


Figure 8.10 Main rotor applied voltage (case 2)

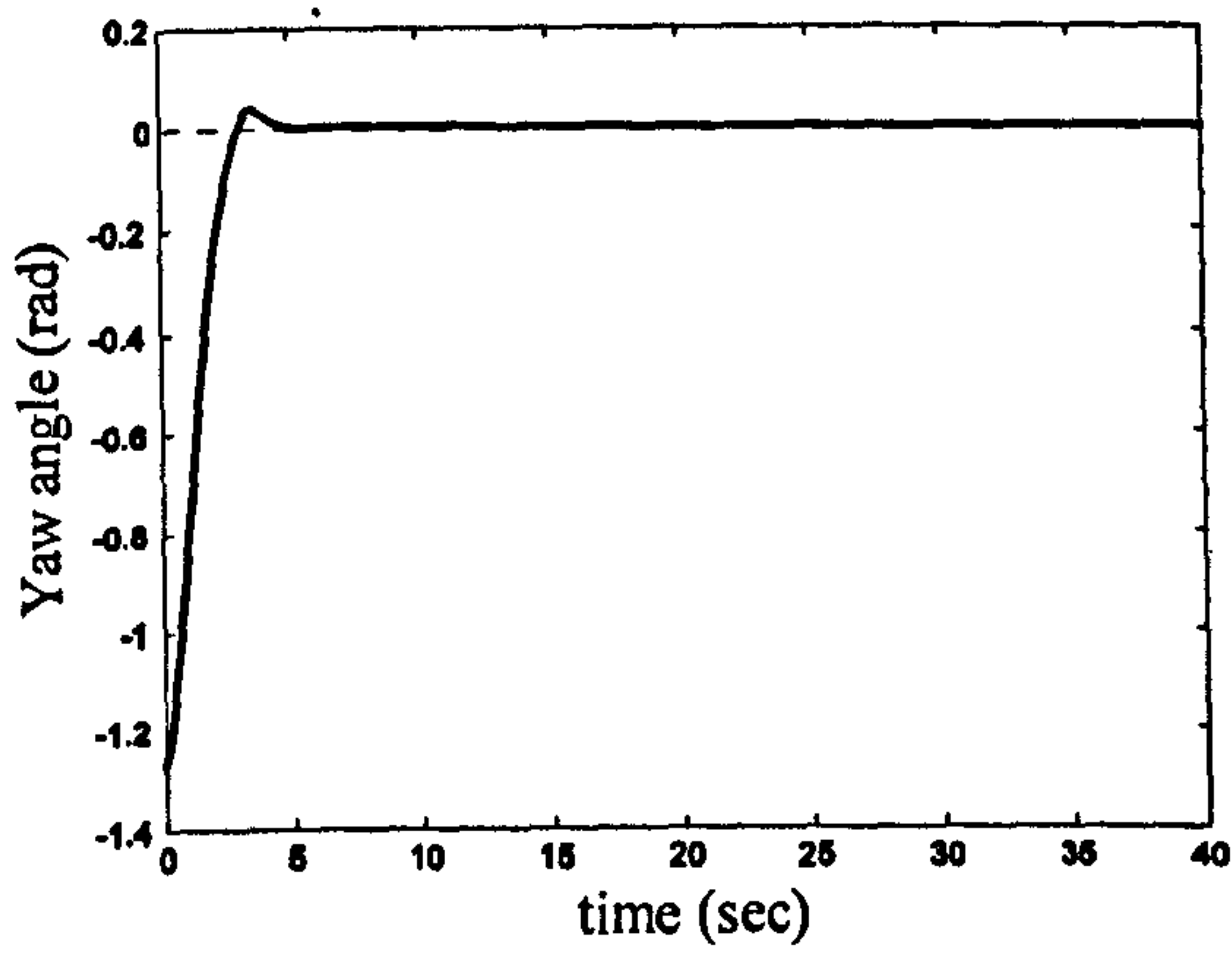


Figure 8.11 Yaw angle response with initial value of -1.27 rad (case 3)

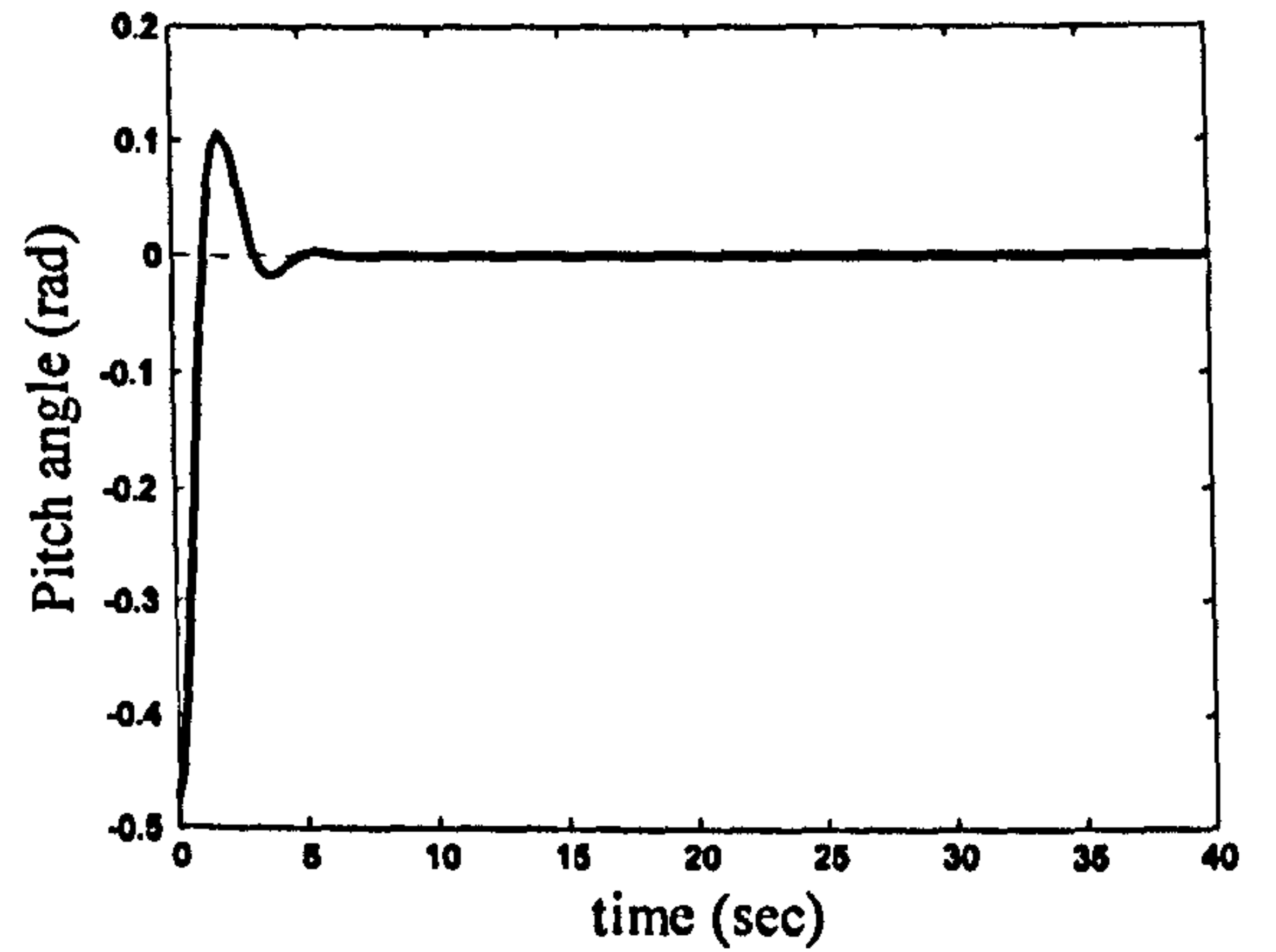


Figure 8.12 Pitch angle with initial value of -0.47 rad (case 3)

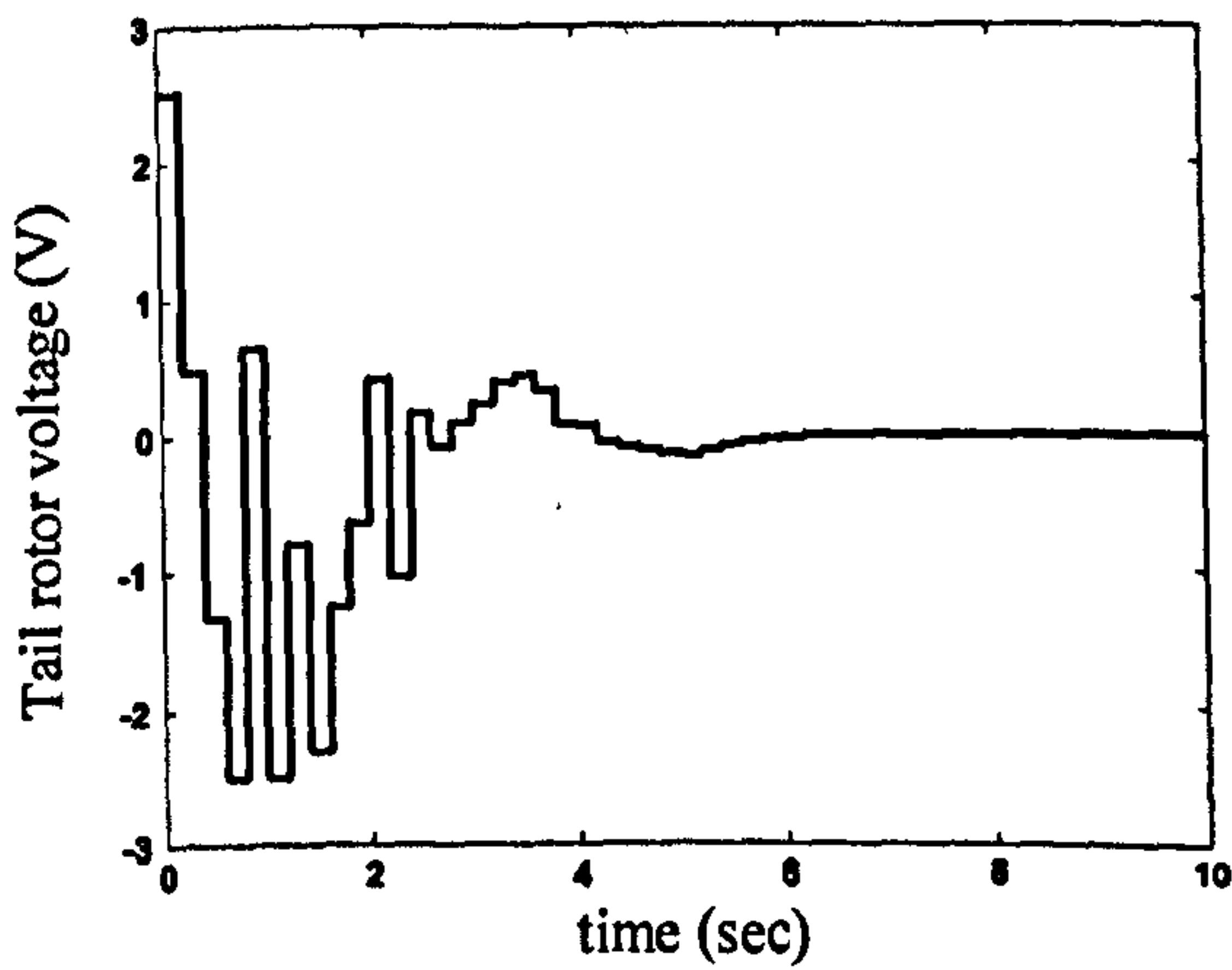


Figure 8.13 Tail rotor applied voltage (case 3)

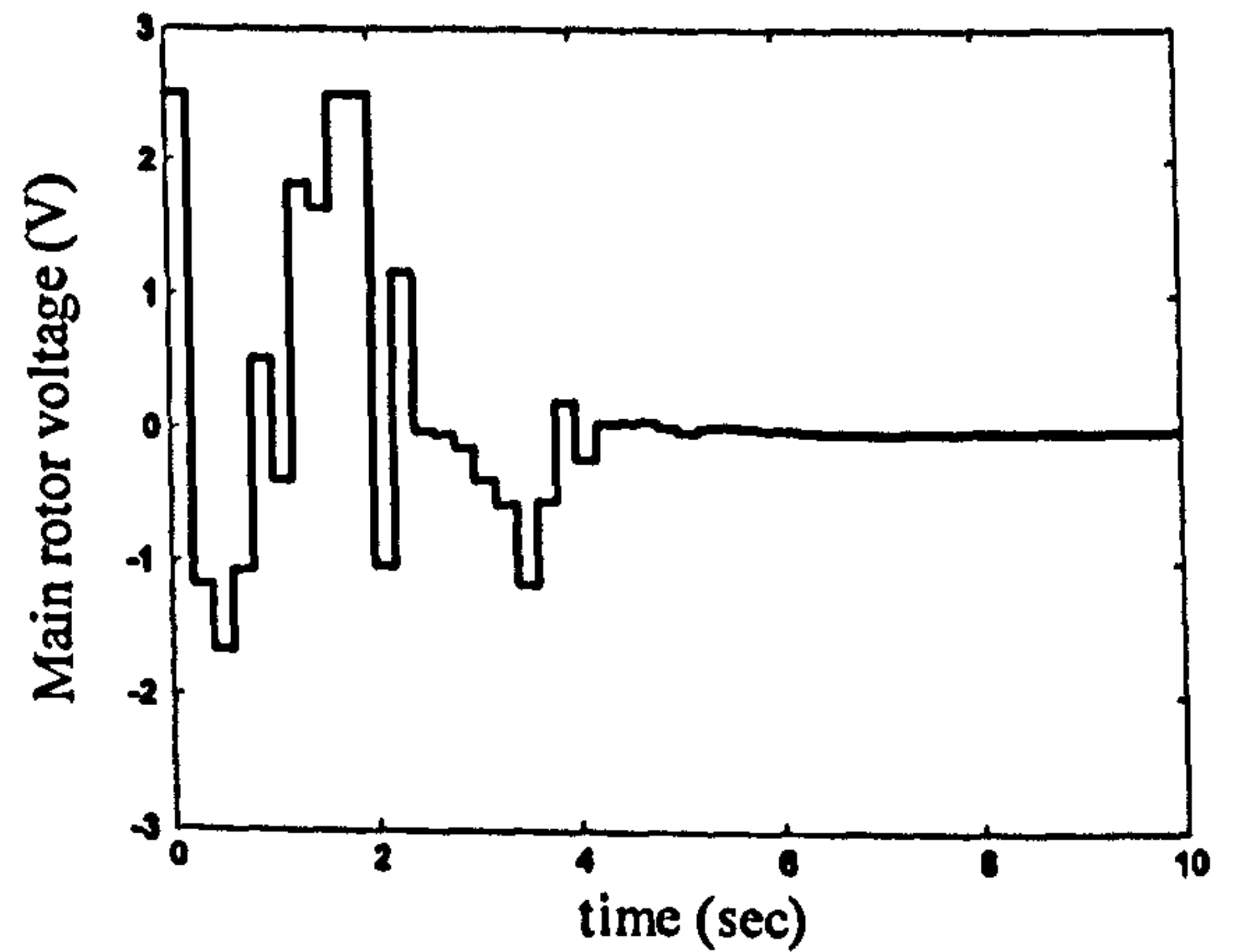


Figure 8.14 Main rotor applied voltage (case 3)

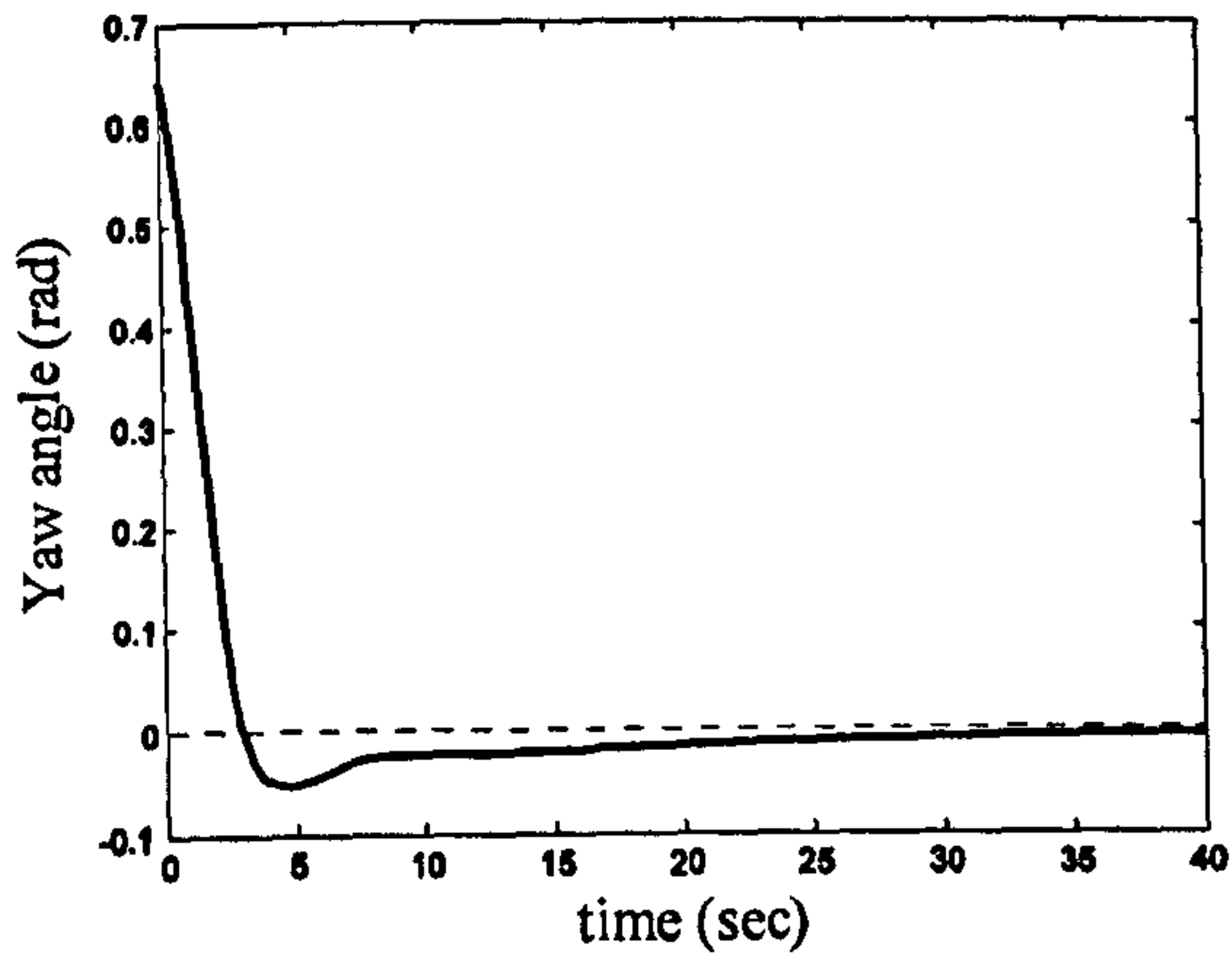


Figure 8.15 Yaw angle response with initial value of 0.64 rad (case 4)

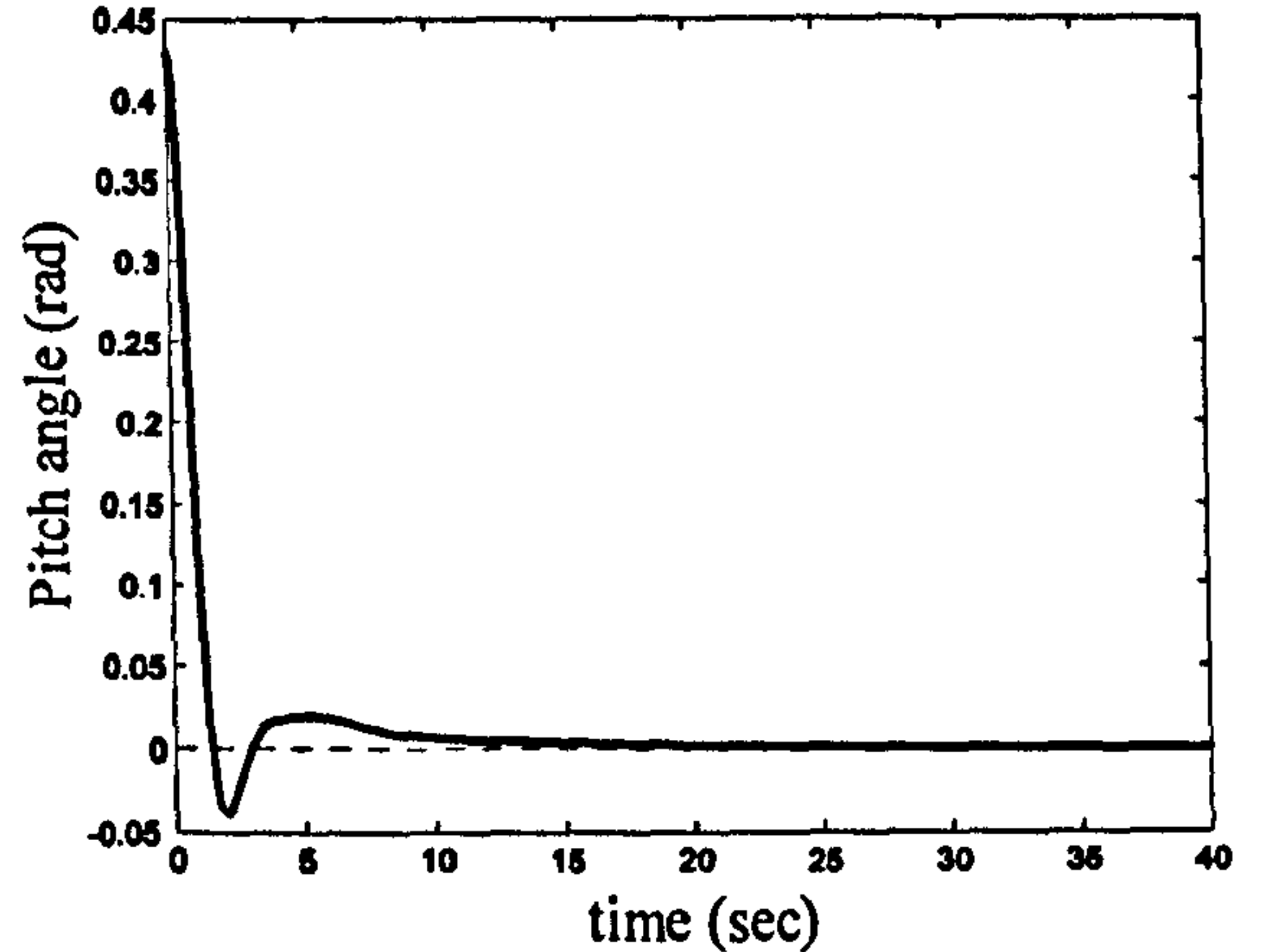


Figure 8.16 Pitch angle with initial value of 0.43 rad (case 4)

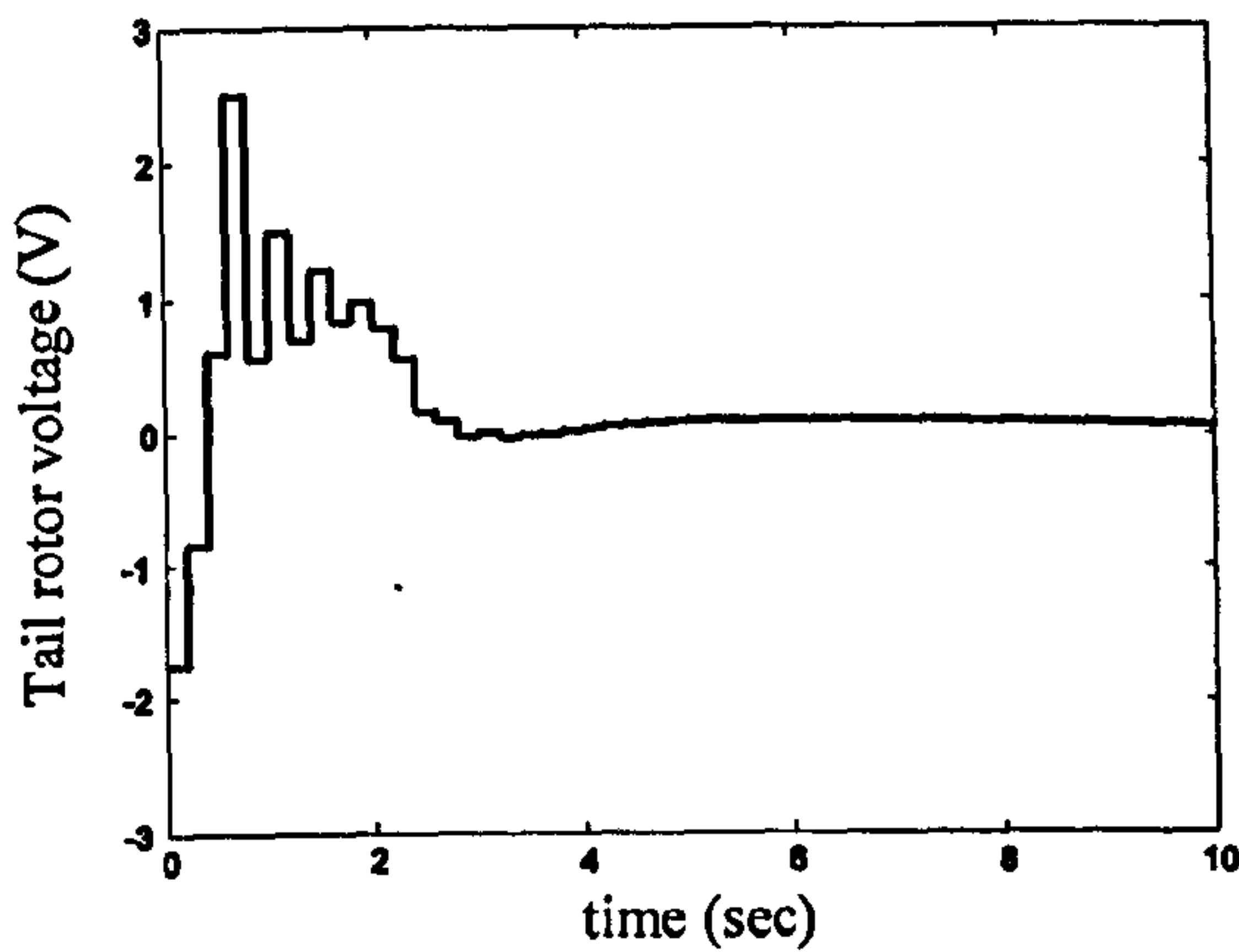


Figure 8.17 Tail rotor applied voltage (case 4)

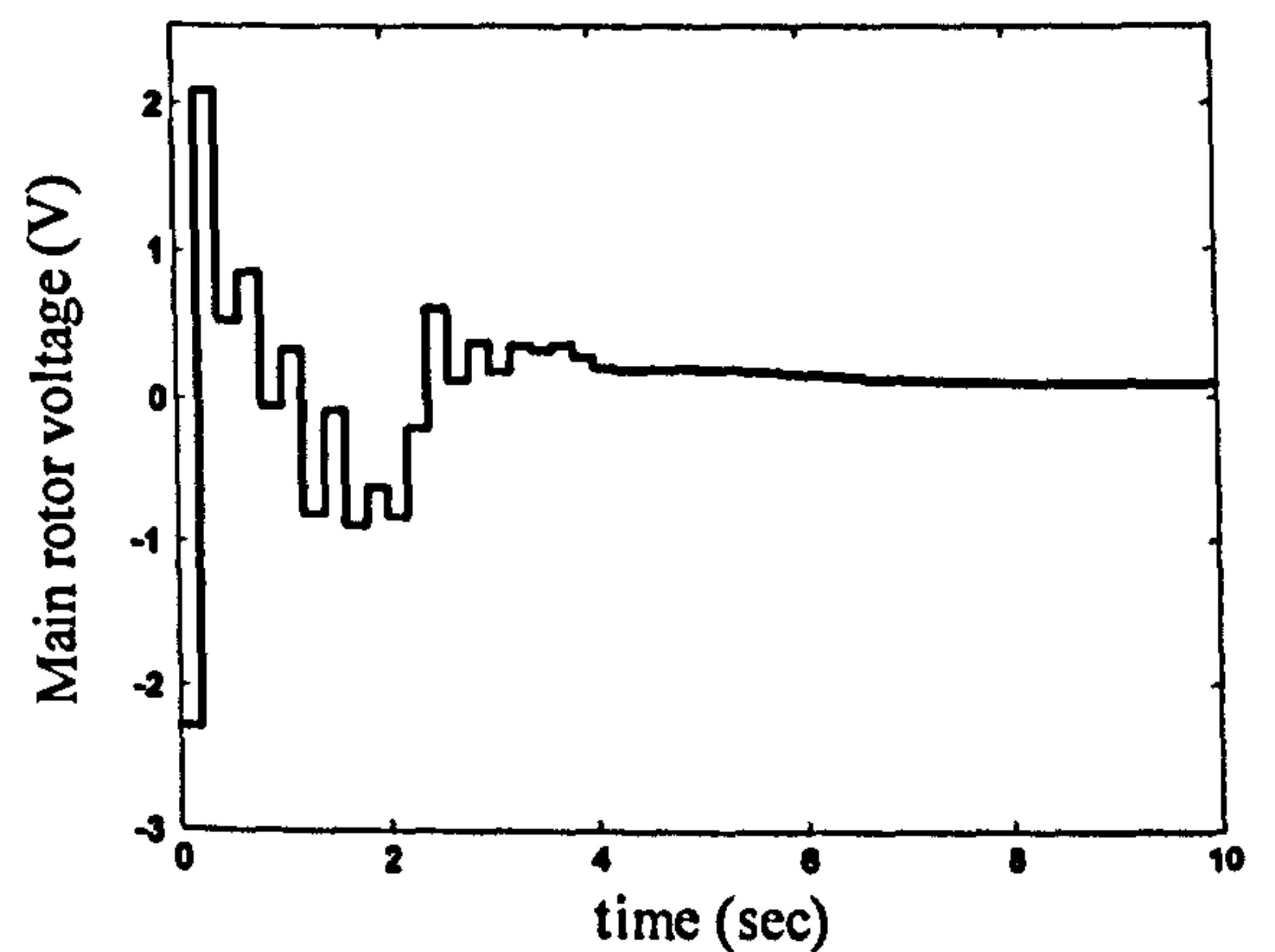


Figure 8.18 Main rotor applied voltage (case 4)

8.6 Conclusions

In this chapter, a robust MPC approach based on a polytopic model has been presented for the nonlinear system. The polytopic model has been developed based on the linearisation of the nonlinear model and finding the vertices of the convex hull. By using a subset of the polytopic model, a method has been proposed to improve the feasibility of the robust MPC method.

CHAPTER 9

Conclusions and Future Work

9.1 Conclusions

The TRMS is an agile, MIMO, nonlinear and cross-coupled system and it is perceived as a challenging system in the field of control engineering from the perspective of modelling and control design. It can be used as a test-bed to design and implement state-of-the-art control approaches for real aerodynamic systems such as UAVs.

Modelling is the first and most important part of any model-based control design such as MPC, and the model accuracy is vital for having an accurate control system. White-box modelling approaches need full knowledge of the plant and can be achieved by energy and mass equations in the case of aerodynamic systems. The most difficult and time-consuming stage of analytical modelling is parameter identification. Overall, physical modelling of a system necessitates *a-priori* information about the process and it is also time-consuming. In this research, the TRMS has been accurately modelled by the Newton and Euler-Lagrange-based approaches in terms of its 1DOF and 2DOF dynamics. The models have been evaluated using both the overlapping and PSD tests and found to be adequate in terms of MSE and frequency response. The comparison between the two analytical modelling approaches has been presented in Table 2.4 which shows that the Lagrange-based approach is more accurate than the Newton one. Although the Lagrangian model is more precise, it is also more complex which is undesirable from the control design viewpoint.

Contrary to that of white-box modelling, the so-called black-box modelling paradigms do not require *a-priori* insight into the plant; however, it requires an ample amount of mined input-output data to entirely represent the dynamics of the system. The most cumbersome and time-consuming stage of the empirical modelling approaches is data mining, especially in the case of MIMO systems. NNs are deemed to be one of the most promising tools in the case of nonlinear black-box modelling.

In this investigation, a sufficient amount of data is extracted from the real TRMS to empirically model the system in terms of its 1DOF and 2DOF dynamics using feedforward MLP networks. Various training algorithms have been employed to train the networks, and it is experimentally observed that the Levenberg-Marquardt outperforms its rivals from the training and test points of view. The performance comparison of the training algorithms in terms of 2DOF dynamics can be found in Table 3.3. Having the input-output data available, the black-box model is easy and fast to develop. Although the black-box models with efficient learning techniques are more accurate compared to the analytical models, they suffer from some disadvantages such as being valid merely in the range of training data and loss of the process structure.

As pointed out earlier, each of the aforementioned approaches has its own advantages and disadvantages. Nonetheless, the combination of white- and black-box modelling paradigms leads to a hybrid scheme well-known as grey-box modelling. In fact, any model identification technique which simultaneously involves both white- and black-box modelling approaches can be put in the grey-box modelling category. Therefore, grey-box modelling is not a unique method and the concept covers a broad scope of modelling approaches. For instance, it is convenient to derive the analytical equations of the plant by white-box theory and subsequently the black-box notion is utilised to identify the parameters of the plant-model. To this end, the parameters of the analytical Newton-based model have been retuned using an evolutionary optimisation technique, the GA, since the generic gradient-based optimisation approaches are not able to handle such problems. A GA seems to be a consistent candidate in order to find the optimum parameters. The improvement of the grey-box model has been validated by overlapping tests. Table 4.1 presents the comparison between the grey-box and Newtonian white-box models in terms of their MSE.

Although significant attempts have been made on the control of nonlinear systems, there is still considerable demand for research on nonlinear control, stability and robustness. Amongst diverse types of nonlinear control approaches such as nonlinear quadratic regulator [135], nonlinear quadratic Gaussian [136], and exact linearisation control [17], [137]-[139] MPC has received a great deal of attention. This is due to several rationales: first, the MPC can be tailored for almost any nonlinear multivariable constrained system; second, it can be combined with other powerful techniques [140], [141] to obtain more positive features. Finally, its stability and more generally its robustness can be guaranteed through the use of some well-known

methods [71]. In this investigation, a nonlinear model predictive controller has been developed and implemented on the TRMS. The proposed method is based on the so-called multistep Newton-type MPC; however, the present method possesses several advantages over the original method. Instead of the Taylor-expansion-based linearisation, a state-dependent technique has been utilised to develop the linear models at each sample time. Moreover, the formulation is able to use the control signal changes instead of the control signal itself. The desired performance specifications, mentioned in the aims and objectives, have been met by the developed control approach. For instance, the steady state error of the closed-loop system is almost zero, the rise time of the control system is within the desired limits and the system is able to continuously follow a reference trajectory.

Generally, stability analysis of nonlinear systems is a non-trivial problem. In other words, checking the stability of a designed nonlinear control system (note that even a constrained linear MPC for a linear plant is a nonlinear control system) using available tools, such as the Lyapunov and invariant set criteria, cannot normally be guaranteed. However, during the controller design, it is usually feasible to force the overall control system to be stable by some additional consideration. It can be concluded that stability synthesis is more convenient than stability analysis. In the case of MPC algorithms, terminal state equality constraints, terminal cost approach, terminal inequality constraints (dual mode), terminal contractive constraints, control Lyapunov function, infinite horizon method, and quasi-infinite horizon [71], [72] are the most common techniques for providing the stability of an MPC system. In this investigation, the terminal state equality constraints technique has been employed to guarantee the stability of the proposed nonlinear system. Nevertheless, the stability of the control system has also been checked with various challenging reference signals and disturbances.

Dealing with advanced control paradigms such as an MPC necessitates the accessibility of the full state variables. However, from a practical point of view, full state variables are not generally available for measurement in the case of most of the industrial processes due to cost limitations, physical restrictions and so on. The most interesting technique of state estimation for linear systems was introduced by Kalman [128] and was called the Kalman filtering. The expansion of the paradigm to nonlinear systems is referred to as EKF; however, it does not exhibit as convincing performance as its linear counterpart, especially when dealing with a highly nonlinear system. A UKF is believed to outperform an EKF owing to the fact that the linearisation step in the latter approach is omitted in the former [129]. An EKF based

on state-dependent state-space equations is also introduced which exhibits a reasonable performance as UKF. Both the UKF and the state-dependent EKF have been utilised in this study to efficiently observe the full state variables of the TRMS. The consolidation of the state estimator with the proposed nonlinear state-feedback MPC leads to a nonlinear output-feedback MPC. The control approach has successfully been implemented on the TRMS. The state observers employed are able to estimate the state vector despite the highly nonlinear nature of the system. The results of the output-feedback control system show that the system is stable in the face of the coupling effect between the two channels, external disturbance force and step change. Also the control system is able to accurately follow any reference trajectory. However, the guaranteed stability of the state-feedback MPC provided by the terminal equality constraints technique cannot be utilised in the case of the output-feedback MPC since, the prerequisite condition for stability, i.e. the plant-model perfection, is not held anymore.

Nominal performance and stability refer to the performance and stability of a system when the plant-model is perfect and free of any type of uncertainty. However, from the perspective of practical systems, model mismatch exists in almost every case. Therefore, in addition to nominal performance and stability, robust performance and stability need to be considered when dealing with a practical system. In other words, the control system has to be stable and needs to exhibit reasonable performance, while any unpredictably small deviation from pre-designed parts occurs, as well as exhibiting stability and good performance in the absence of uncertainty. Basically, any information and knowledge about uncertainty, such as its structure (e.g. which parameters may deviate from its nominal value) and its boundedness (the maximum deviation may possibly occur), can lead to less conservative results of the control system. In robust synthesis the wider bound of uncertainty implies the high probability of infeasibility. In this study, a polytopic uncertainty has been developed for the TRMS on the basis of the nonlinear Newtonian model. Generally, a polytopic model of a nonlinear system can be represented by a number of linear models. Instead of considering all these linear models, one can form a convex hull which covers all the linear models. The convex hull is mathematically characterised by the linear models located on its vertices. Having developed the polytopic uncertainty, one needs to synthesise a robust model predictive controller. At each instant, a state-feedback control law has been designed to minimise a worst-case scenario of an infinite horizon objective function subject to input and output constraints. Based on an unknown quadratic Lyapunov function, the min-max optimisation problem has been reduced to a convex optimisation problem engaging LMI; the unknown

parameter of the Lyapunov function is optimally found as a part of the consolidated optimisation problem. The technique here is the improved version of the robust MPC proposed in [85]. The Lyapunov inequality used for finding the upper bound of the worst-case scenario is stringent for practical system and cannot be satisfied. Therefore, to relieve the optimisation problem, a subset of the main polytope is found which depends on the initial values of the system. Moreover, by imposing extra constraints, the system trajectory is forced to stay inside the subset forever. It renders the optimisation problem more feasible. The robust state-feedback MPC approach has been implemented on the TRMS to validate the efficacy of the regulator. The results show that the regulator is robust in terms of performance and stability.

9.2 Suggestions for Future Work

In this investigation, the TRMS has been modelled analytically using both the Newtonian and Lagrangian methods. The TRMS can also be modelled using other methodologies for deriving the equation of motion, such as Kane's equations of motion [142]. Kane's method implements the concept of generalised speeds (quasi-velocity coordinates) to represent motion. This implementation allows focusing on the motion aspects of dynamic systems rather than just on the configuration. It is especially suitable for dealing with non-holonomic constraints.

Further work can be carried out on the empirical modelling of the TRMS in terms of its 2DOF dynamics using other type of NNs such as wavelet-based NNs [143]. Wavelet-based NNs topologies are similar to that of RBF networks and have only one hidden layer; however, they are different in terms of the type of activation functions and learning algorithms. The type of activation functions, scaling and wavelet functions, should be selected according to the system to be modelled. Among various types of wavelet functions such as, Meyer, Morlet, Mexican Hat, and Daubechies, the one that is most suited to the application needs to be chosen. The location of the basis functions should be found in such a way to be consistent with the multi-resolution analysis. Wavelet-based NNs support multi-resolution construction and can guarantee convergence [143]. Figure 9.1 depicts a wavelet-based NN construction with p inputs and a single output.

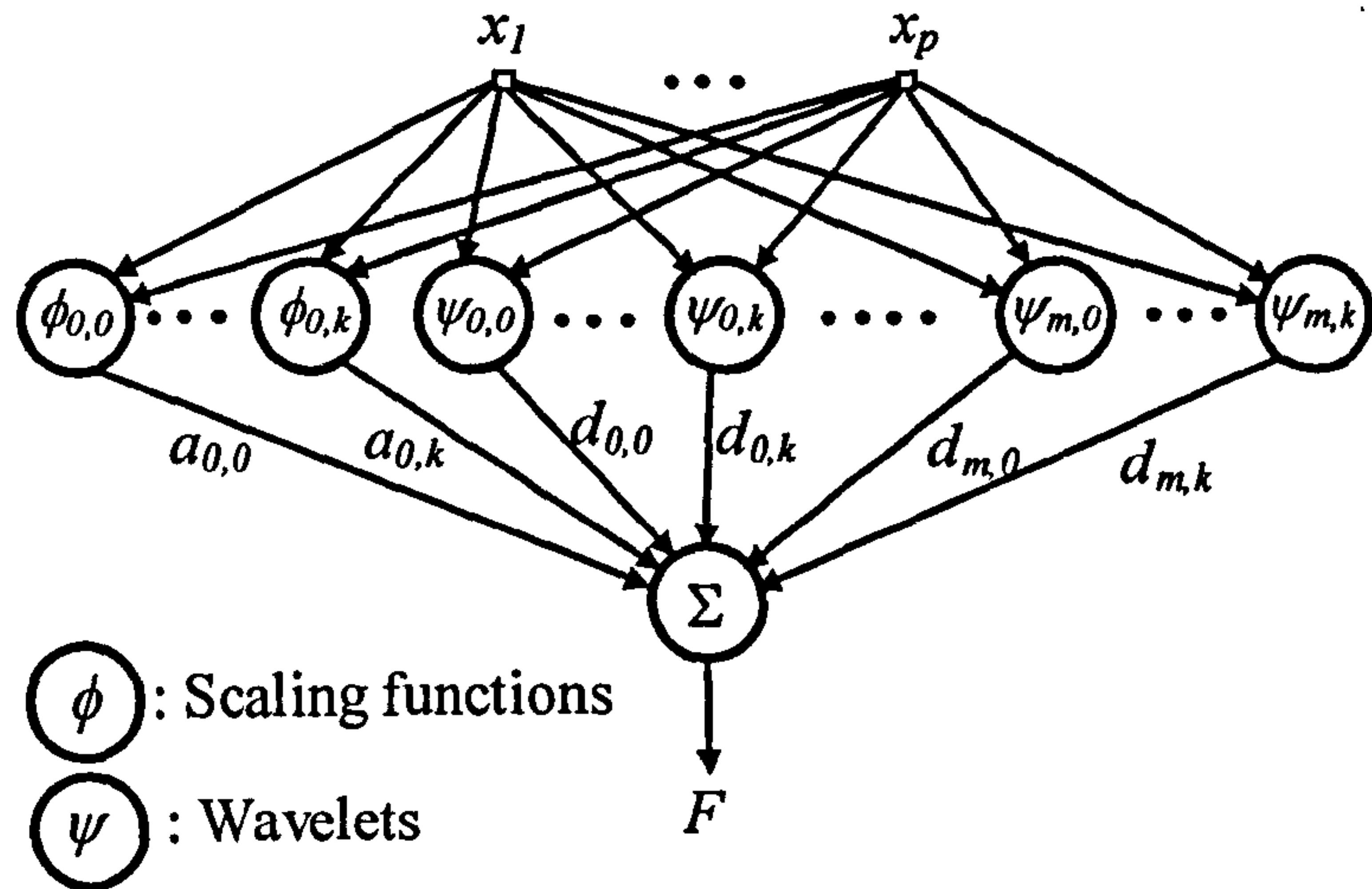


Figure 9.1 A wavelet-based network with p inputs and a single output

Following the success of biologically inspired approaches to optimisation, championed by GAs, a number of promising meta-heuristic search and optimisation algorithms have recently been reported in the literature. These include ant colony, particle swarm, simulated annealing and harmony search. All these techniques can be used in the grey-box modelling approach of the TRMS. In the real world, ants can find the shortest path from a source of food to their nest. The ant colony system is based on these ants' natural foraging behaviour. Dorigo *et al.* [144] have proposed this optimisation method and utilised it in some optimisation problem such as the travelling salesman problem [145]. The particle swarm optimisation is a stochastic optimisation approach, modelling the biological behaviour of members in a swarm of insects or a school of fish that allows them to determine a desirable path [146], [147]. Simulated annealing is a generic probabilistic optimisation method invented by Kirkpatrick *et al.* [148]. The method is inspired by the annealing process in metallurgy. Another meta-heuristic algorithm, mimicking the improvisation process of music players, has recently been developed and named as harmony search (HS) [149]-[151]. It has been successful in a wide variety of optimisation problems [152], [153], offering several advantages with respect to traditional optimisation techniques. The HS algorithm imposes fewer mathematical requirements and does not require initial value settings of the decision variables. As the HS algorithm uses stochastic random searches, derivative information is also unnecessary. The HS algorithm generates new vectors, after considering all of the existing vectors, whereas the GA only considers the two parent vectors. These features increase the flexibility of the HS algorithm to produce better solutions.

It is suggested that the stability of the output-feedback Newton-type MPC is

guaranteed. In this respect, one can use a nonlinear high-gain observer [154] as explained in [74]. Other nonlinear observers, such as those mentioned in [155]-[158], can also be utilised to estimate the state variables of the TRMS.

The development of an output-feedback robust MPC for the TRMS is recommended. With this aim, one can refer to [84], [93], [94] in which a consolidated observer and state-feedback MPC has been suggested to provide the robust stability and performance of the control system.

Exact linearisation MPC [140], [141] is an interesting subject that may be implemented on the TRMS. Roughly speaking, exact linearisation [137] tries to cancel the nonlinearities in a nonlinear system in such a way that the closed-loop dynamics appears in a linear form. It is a transformation for mapping a nonlinear system into a linear one without any approximation which occurs, for instance, in the Taylor-expansion-based linearisation approach.

Other types of stability paradigms which are applicable to MPC algorithms can be employed to guarantee the stability of the proposed MPC approach. For example, dual model stability criterion [70] is suggested for use in guaranteeing the stability of the proposed system as an alternative method to terminal equality constraints.

The proposed MPC method is perceived to be applicable to other practical aerodynamic systems, such as UAVs. Due to the fast response of aerodynamic systems, the multistep Newton-type MPC method is deemed to be implementable; however, an accurate plant-model is a prerequisite for the control approach.

NN-based MPC methods [2] are considered as nonlinear MPC paradigms, which use the NN-based empirical model of a system, and solve a nonlinear optimisation problem due to the nonlinearities stemming from the nonlinear activation functions and multi-layer networks. Therefore, the development of an NN-based MPC is recommended for the TRMS based on the developed NNARX model.

As mentioned earlier, the TRMS has a pair of restrictors for each channel, and if the beam of the TRMS hits one of these restrictors, a torque exerts to the beam and forces it to move in the opposite direction. In order to model this phenomenon, the so-called hybrid dynamics theory can be involved. Hybrid dynamics are a

combination of continuous-time dynamical systems and discrete-time dynamical systems [159], [160]. The impact phenomenon is one of the examples which can be modelled using hybrid dynamics concepts. Continuous-time equations of motion are employed to describe the behaviour of a hybrid system in between the impacts. During the impacts a set of discrete-time equations can be used. The control of hybrid systems needs the requirements according to their nature [159], [160].

Bibliography

- [1] Feedback Co., *Twin Rotor MIMO System 33-220 user manual*, 1998.
- [2] M. Norgaard, O. Ravn, N. K. Poulsen, L. K. Hansen, *Neural Networks for Modelling and control of dynamic systems*, 3rd Edition, Springer-Verlag, London, 2003.
- [3] G. R. Fowles, *Analytical Mechanics*, CBS College Publishing, Fourth Edition, 1986.
- [4] J. Richalet, A. Rault, J. L. Testud, J. Papon, "Algorithmic Control of Industrial Processes", *IFAC Symposium on Identification and System Parameter Estimation*, 1976, pp. 1119-1167.
- [5] J. Richalet, A. Rault, J. L. Testud, J. Papon, "Model Predictive Heuristic Control: Application to Industrial Processes", *Automation*, 14, pp. 413-428, 1978.
- [6] C. R. Cutler, B. L. Ramaker, "Dynamic Matrix Control - A Computer Control Algorithm", *Automatic Control Conference*, San Francisco, California, 1980.
- [7] C. R. Cutler, "Dynamic Matrix Control of Imbalanced Systems", *ISA transactions*, Vol. 21, No. 1, 1982, pp. 1-6.
- [8] C. R. Cutler, A. Morshedi, J. Haydel, "An Industrial Perspective on Advanced Control", *AIChE Annual Meeting*, Washington DC, 1983.
- [9] E. F. Camacho, C. Bordons, *Model Predictive Control*, Springer-Verlag, London, 2004.
- [10] S. M. Ahmad, M. H. Shaheed, A. J. Chipperfield, M. O. Tokhi, "Nonlinear Modelling of a Twin Rotor MIMO System using Radial Basis Function Networks", *IEEE National Aerospace and Electronics Conference*, 2000, pp. 313-320.
- [11] S. M. Ahmad, A. J. Chipperfield, M. O. Tokhi, "Dynamic Modelling and Optimal Control of a Twin Rotor MIMO System", *IEEE National Aerospace and Electronics Conference*, 2000, pp. 391-398.
- [12] M. H. Shaheed, "Performance Analysis of 4 Types of Conjugate Gradient Algorithm in the Nonlinear Dynamic Modelling of a TRMS using Feedforward Neural Networks", *IEEE International Conference on Systems, Man and Cybernetics*, 2004, pp. 5985-5990.

- [13] F. M. Aldebrez, I. Z. Mat Darus, M. O. Tokhi, "Dynamic Modelling of a Twin Rotor System in Hovering Position", *International Symposium on Control, Communications and Signal Processing*, 2004, pp. 823-826.
- [14] I. Z. Mat Darus, F. M. Aldebrez, M. O. Tokhi, "Parametric Modelling of a Twin Rotor System Using Genetic Algorithms", *International Symposium on Control, Communications and Signal Processing*, 2004, pp. 115-118.
- [15] M. Rao, S. K. Biswas, B. P. Butz, D. G. Miller, "Dynamic Modelling and Feedback Control of a Side-by-side Rotor Tandem Helicopter", *Proceedings of the IEEE International Symposium on Intelligent Control Systems*, Philadelphia, PA, September, 1990, pp. 780-785.
- [16] X. Wu, R. Ignatov, G. Muenst, A. Imaev, J. J. Zhu, "A Nonlinear Flight Controller Design for a UFO by Trajectory Linearization Method, Part I: Modelling", *Proceedings of the 34th Southeastern Symposium on System Theory*, 2002. pp. 97-102.
- [17] V. Mistler, A. Benalleque, N. K. Msirdi, "Exact Linearization and Non-interacting Control of a 4 Rotors Helicopter via Dynamic Feedback", *10th IEEE International Workshop on Robot and Human Interactive Communication*, Sep. 2001, pp. 586–593.
- [18] R. Mahony, T. Hamel, A. Dzul, "Hover Control via Lyapunov Control for an Autonomous Model Helicopter", *IEEE Conference on Decision and Control*, Vol. 4, 1999, pp. 3490-3495.
- [19] P. Castillo, A. Dzul, R. Lozano, "Real-Time Stabilization and Tracking of a Four-Rotor Mini Rotorcraft", *IEEE Transactions on Control Systems Technology*, Vol. 12, No. 4, July 2004, pp. 510-516.
- [20] Y. I. Somov, O. Y. Polyntsev, "Nonlinear Dynamics and Control of a Wind-milling Gyroplane Rotor", *International Conference Physics and Control*, August, Vol. 1, 2003, pp. 151- 156.
- [21] J. C. Avila-Vilchis, B. Brogliato, R. Dzul, R. Lozano, "Nonlinear Modelling and Control of Helicopters", *Automatica*, Vol. 39, No. 9, 2003, pp. 1583-1596.
- [22] J. D. Cerro, J. Valero, A. Barrientos, "Identification of a Small Unmanned Helicopter Model using Genetic Algorithms", *IEEE International Conference on Intelligent Robots and Systems*, August 2005, pp. 3360–3365.
- [23] D. Song-Yun, Z. Ji-Hong, H. Chun-Hua, "Mathematical Modelling of a UAV during Take-off with Nose-Wheel off the Ground", *World Automation Congress*, Vol. 15, June 2004, pp. 359-364.

- [24] M. Jun, S. I. Roumeliotis, G. S. Sukhatme, "State Estimation of an Autonomous Helicopter using Kalman Filtering", *IEEE International Conference on Intelligent Robots and Systems*, Vol. 3, Oct. 1999, pp. 1346–1353.
- [25] N. R. Kristensen, H. Madsen, S. B. Jorgensen, "A Method for Systematic Improvement of Stochastic Grey-box Models", *Computers and Chemical Engineering*, Vol. 28, 2004, pp. 1431-1449.
- [26] J. Spannar, P. Wide, B. Sohlberg, "Estimation of a Non-Measurable Variable by Using Grey Box Modelling", *IEEE Con. on Instr. and Measurement Tech., Budapest*, May 2001, pp. 778-783.
- [27] K. C. Tan, Y. Li, P. J. Gawthrop, A. Glidle, "Evolutionary Grey-box Modelling for Practical Systems", *IEE Conference on Genetic Algorithm in Engineering Systems*, Sep. 1997, pp. 369-375.
- [28] R. K. Pearson, M. Pottmann, "Grey-box Identification of Block-oriented Nonlinear Models", *Journal of Process Control*, Vol. 10, 2000, pp. 301-315.
- [29] Q. Xiong, A. Jutan, "Grey-box Modelling and Control of Chemical Processes", *Chemical Engineering Science*, Vol. 57, 2002, pp. 1027-1039.
- [30] Y. Oussar, G. Dreyfus, "How to be a Grey Box: Dynamic Semi-physical Modelling", *Neural Networks*, Vol. 14, 2001, pp. 1161-1172.
- [31] B. Sohlberg, "Hybrid Grey-box Modelling of a Pickling Process", *Control Engineering Practice*, Vol. 13, 2005, pp. 1093-1102.
- [32] C. S. Ong, J. J. Huang, G. H. Tzeng, "Model Identification of ARIMA Family using Genetic Algorithms", *Applied Mathematics and Computation*, Vol. 164, 2005, pp. 885-912.
- [33] W. D. Chang, "Nonlinear System Identification and Control using a Real-coded Genetic Algorithm", *Applied Mathematical Modelling*, Vol. 31, 2007, pp. 541-550.
- [34] N. R. Kristensen, H. Madsen, S. B. Jorgensen, "Parameter Estimation in Stochastic Grey-box Models", *Automatica*, Vol. 40, 2004, pp. 225-237.
- [35] B. Sohlberg, "Grey Box Modelling for Model Predictive Control of a Heating Process", *Journal of Process Control*, Vol. 13, 2003, pp. 225-238.
- [36] S. J. Qin, T. A. Badgwell, "A Survey of Industrial Model Predictive Control Technology", *Control Engineering Practice*, Vol. 11, 2003, pp. 733-764.
- [37] P. Poignet, M. Gautier, "Nonlinear Model Predictive Control of a Robot Manipulator", *IEEE AMC Conference*, 2000, pp. 401-406.

- [38] W. Wroblewski, "Implementation of a Model Predictive control Algorithm for a 6dof Manipulator-Simulation Results", *IEEE 4th International Workshop on Robotic and Control*, 2004, pp. 209-212.
- [39] D. H. Shim, H. J. Kim, S. Sastry, "Decentralized Nonlinear Model Predictive Control of Multiple Flying Robots", *IEEE Conference on Decision and Control*, 2003, pp. 3621-3626.
- [40] H. J. Kim, D. H. Shim, S. Sastry, "Flying Robots: Modelling, Control and Decision Making", *IEEE International Conference on Robotics and Automation*, 2002, pp. 66-71.
- [41] A. Vivas, P. Poignet, F. Pierrot, "Predictive Functional Control for a Parallel Robot", *IEEE International Conference on Intelligent Robotics and systems*, 2003, pp. 2785-2790.
- [42] H. Kawati, Y. Kawati, M. Fujita, "Visual Feedback Control of Planar Manipulators Based on Nonlinear Receding Horizon Control Approach", *IEEE American Control Conference*, 2001, pp. 763-768.
- [43] D. Gu, H. Hu, "A Stabilizing Receding Horizon Regulator for Nonholonomic Mobile Robots", *IEEE Transactions on Robotics*, Vol. 21, No. 5, 2005, pp. 1022-1028.
- [44] F. Lizarralde, E. V. L. Nunes, L. Hsu, J. T. Wen, "Mobile Robot Navigation using Sensor Fusion", *IEEE International Conference on Robotic and Automation*, 2003, pp. 458-463.
- [45] B. Vanek, T. Peni, J. Bokor, G. Balas, "Practical Approach to Real-time Trajectory Tracking of UAV Formations", *IEEE American Control Conference*, 2005, pp. 122-127.
- [46] B. Aufderheide, B. W. Bequette, "A Variably Tuned Multiple Model Predictive Controller Based on Minimal Process Knowledge", *American Control Conference*, Arlington, June 2001, pp. 3490-3495.
- [47] B. Aufderheide, V. Prasad, B. W. Bequette, "A Comparison of Fundamental Model-Based and Multiple Model Predictive Control", *IEEE Conference on Decision and Control*, Orlando, Florida USA, December 2001, pp. 4863-4868.
- [48] D. Dougherty, D. Cooper, "A Practical Multiple Model Adaptive Strategy for Multivariable Model Predictive Control", *Control Engineering Practice*, Vol. 11, 2003, pp. 649-664.
- [49] Z. Wan, M. V. Kothare, "Efficient Scheduled Stabilizing Output Feedback Model Predictive Control for Constrained Nonlinear Systems", *IEEE*

- Transactions on Automatic Control*, Vol. 49, No. 7, July 2004, pp. 1172-1177.
- [50] C. Yu, R. J. Roy, H. Kaufman, B. W. Bequette, "Multiple-Model Adaptive Predictive Control of Mean Arterial Pressure and Cardiac Output", *IEEE Transactions on Biomedical Engineering*, Vol. 39, No. 8, 1992, pp. 765-778.
- [51] R. R. Rao, B. Aufderheide, B. W. Bequette, "Experimental Studies on Multiple-Model Predictive Control for Automated Regulation of Hemodynamic Variables", *IEEE Transactions on Biomedical Engineering*, Vol. 50, No. 3, March 2003, pp. 277-288.
- [52] H. Fukushima, T. H. Kim, T. Sugie, "Adaptive Model Predictive Control for a Class of Constrained Linear Systems Based on the Comparison Model", *Automatica*, Vol. 43, 2007, pp. 301-308.
- [53] T. H. Kim, T. Sugie, "A New Adaptive Parameter Estimation Algorithm for Robust Constrained Predictive Control", *IEEE Conference on Decision and Control, and European Control Conference*, Seville, Spain, December 2005, pp. 284-289.
- [54] B. Zhang, W. Zhang, "Adaptive Predictive Functional Control of a Class of Nonlinear Systems", *ISA Transactions*, Vol. 45, No. 2, April 2006, pp. 175-183.
- [55] T. J. J. van den Boom, B. De Schutter, G. Schullerus, V. Krebs, "Adaptive Model Predictive Control for Max-Plus-Linear Discrete Event Input-Output Systems", *IEE Proc.-Control Theory Applications*, Vol. 151, No. 3, May 2004, pp. 339-346.
- [56] M. O' Brien, A. W. Ordys, "Nonlinear Predictive Control of a Wastewater Model", *UKAAC Control*, Glasgow, UK, August 2006, pp. 63-68.
- [57] W.C. Li, L.T. Biegler, "Multistep, Newton-type Control Strategies for Constrained Nonlinear Processes", *Chemical Engineering Research and Design*, Vol. 67, 1989, pp. 562-577.
- [58] N.M.C. De Oliveira, L.T. Biegler, "An Extension of Newton-type Algorithms for Nonlinear Process Control", *Automatica*, Vol. 31, No. 2, 1995, pp. 281-286.
- [59] M. Diehl, H. G. Bock, J. P. Schlöder, "A Real-time Iteration Scheme for Nonlinear Optimization in Optimal Feedback Control", *SIAM Journal on Control and Optimization*, Vol. 43, No. 5, 2005, pp. 1714-1736.
- [60] M. Diehl, R. Findeisen, F. Allgöwer, H. G. Bock, J. P. Schlöder, "Nominal Stability of Real-time Iteration Scheme for Nonlinear Model Predictive

- Control", *IEE Proc.-Control Theory Application*, Vol. 152, No. 3, May 2005, pp.296-308.
- [61] R. Bitmead, M. Gevers, V. Wertz, *Adaptive Optimal Control: The Thinking Man's GPC*, Prentice Hall, 1990.
- [62] S. S. Keerthi, E. G. Gilbert, "Optimal Infinite-Horizon Feedback Laws for a General Class of Constrained Discrete-Time Systems: Stability and Moving-Horizon Approximations", *Journal of Optimization Theory and Applications*, Vol. 57, No. 2, May 1988, pp. 265-293.
- [63] C. C. Chen, L. Shaw, "On Receding Horizon Feedback Control", *Automatica*, Vol. 18, 1982, pp. 349-352.
- [64] W. H. Kwon, A. M. Bruckstein, T. Kailath, "Stabilizing State Feedback Design via the Moving Horizon Method", *International Journal of Control*, Vol. 37, No. 3, 1983, pp. 631-643.
- [65] D. Q. Mayne, H. Michalska, "Receding Horizon Control of Nonlinear Systems", *IEEE Transactions on Automatic Control*, Vol. 35, July 1990, pp. 814-824.
- [66] J. B. Rawlings, K. R. Muske, "The Stability of Constrained Receding Horizon Control", *IEEE Transactions on Automatic Control*, Vol. 38, No. 10, Oct. 1993, pp. 1512-1516.
- [67] H. Chen, F. Allgöwer, "A Quasi-Infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability", *Automatica*, Vol. 34, No. 10, 1998, pp. 1205-1217.
- [68] P. O. M. Scokaert, D. Q. Mayne, J. B. Rawlings, "Suboptimal Model Predictive Control (Feasibility Implies Stability)", *IEEE Transactions on Automatic Control*, Vol. 44, No. 3, March 1999, pp. 648-654.
- [69] G. De Nicolao, L. Magni, R. Scattolini, "On the Robustness of Receding Horizon Control with Terminal Constraints", *IEEE Transactions on Automatic Control*, Vol. 41, No. 3, March 1996, pp. 451-453.
- [70] H. Michalska, D. Q. Mayne, "Robust Receding Horizon Control of Constrained Nonlinear Systems", *IEEE Transactions on Automatic Control*, Vol. 38, No. 11, Nov. 1993, pp. 1623-1633.
- [71] D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. M. Scokaert, "Constrained Model Predictive Control: Stability and Optimality", *Automatica*, Vol. 36, 2000, pp. 789-814.
- [72] G. De Nicolao, L. Magni, R. Scattolini, "Stability and Robustness of Nonlinear

- Receding Horizon Control", in *Nonlinear Model Predictive Control*, Editors: F. Allgöwer and A. Zheng, Birkhauser Verlag, 2000, Vol. 26, Progress in Systems and Control Theory, pp. 3-22.
- [73] A. Jadbabaie, J. Yu, J. Hauser, "Unconstrained Receding-Horizon Control of Nonlinear Systems", *IEEE Transactions on Automatic Control*, Vol. 46, No. 5, May 2001, pp. 776-783.
- [74] L. Imsland, R. Findeisen, E. Bullinger, F. Allgöwer, B. A. Foss, "A Note on Stability, Robustness and Performance of Output Feedback Nonlinear Model Predictive Control", *Journal of Process Control*, Vol. 13, 2003, pp. 633-644.
- [75] H. Michalska, D. Q. Mayne, "Moving Horizon Observers and Observer-Based Control", *IEEE Transactions on Automatic Control*, Vol. 40, No. 6, June 1995, pp. 995-1006.
- [76] R. Findeisen, M. Diehl, T. Burner, F. Allgöwer, H. G. Bock, J. P. Schlöder, "Efficient Output Feedback Nonlinear Model Predictive Control", *American Control Conference*, Anchorage, AK, 2002, pp. 4752-4757.
- [77] S. L. O. Kothare, M. Morari, "Contractive Model Predictive Control for Constrained Nonlinear Systems", *IEEE Transactions on Automatic Control*, Vol. 45, No. 6, June 2000, pp. 1053-1071.
- [78] A. Bemporad, M. Morari, "Robust Model Predictive Control: a Survey", in: A. Vicino, G. Zappa, A. Garulli, A. Tesi (Eds.), *Robustness in Identification and Control*, Vol. 245, Springer-Verlag, London LTD, Godalming, 1999, pp. 207-226.
- [79] A. A. Jalali, V. Nadimi, "A Survey on Robust Model Predictive Control from 1999-2006", *International Conference on Computational Intelligence for Modelling Control and Automation*, Sydney, Australia, Nov. 2006, pp. 207-212.
- [80] L. Magni, R. Sepulchre, "Stability Margins of Nonlinear Receding-Horizon Control via Inverse Optimality", *Systems & Control Letters*, Vol. 32, 1997, pp. 241-245.
- [81] P. J. Campo, M. Morari, "Robust Model Predictive Control", *American Control Conference*, Vol. 2, 1987, pp. 1021-1026.
- [82] H. S. Witsenhausen, "A Minmax Control Problem for Sampled Linear Systems", *IEEE Transactions on Automatic Control*, Vol. 13, No. 1, 1968, pp. 5-21.
- [83] Z. Q. Zheng, *Robust Control of Systems Subject to Constraints*, PhD

- dissertation, California Institute of Technology, Pasadena, CA, USA, 1995.
- [84] D. Q. Mayne, S. V. Rakovic, R. Findeisen, F. Allgöwer, "Robust Output Feedback Model Predictive Control of Constrained Linear Systems", *Automatica*, Vol. 42, 2006, pp. 1217-1222.
- [85] M. V. Kothare, V. Balakrishnan, M. Morari, "Robust Constrained Model Predictive Control using Linear Matrix Inequalities", *Automatica*, Vol. 32, No. 10, 1996, pp. 1361-1379.
- [86] F. A. Cuzzola, J. C. Geromel, M. Morari, "An Improved Approach for Constrained Robust Model Predictive Control", *Automatica*, Vol. 38, 2002, pp. 1183-1189.
- [87] W. J. Mao, "Robust Stabilization of Uncertain Time-varying Discrete Systems and Comments on 'An Improved Approach for Constrained Robust Model Predictive Control'", *Automatica*, Vol. 39, 2003, pp. 1109-1112.
- [88] B. Ding, Y. Xi, S. Li, "A Synthesis Approach of On-line Constrained Robust Model Predictive Control", *Automatica*, Vol. 40, 2004, pp. 163-167.
- [89] Z. Wan, M. V. Kothare, "An Efficient Off-line Formulation of Robust Model Predictive Control using Linear Matrix Inequalities", *Automatica*, Vol. 39, 2003, pp. 837-846.
- [90] P. O. M. Scokaert, D. Q. Mayne, "Min-Max Feedback Model Predictive Control for Constrained Linear Systems", *IEEE Transactions on Automatic Control*, Vol. 43, No. 8, 1998, pp. 1136-1142.
- [91] M. C. de Oliveira, J. Bernussou, J. C. Geromel, "A New Discrete-time Robust Stability Condition", *Systems & Control Letters*, Vol. 37, 1999, pp. 261-265.
- [92] Y. J. Wang, J. B. Rawlings, "A New Robust Model Predictive Control Method I: Theory and Computation", *Journal of Process Control*, Vol. 14, 2004, pp. 231-247.
- [93] Z. Wan, M. V. Kothare, "Robust Output Feedback Model Predictive Control using Off-line Linear Matrix Inequality", *Journal of Process Control*, Vol. 12, 2002, pp. 763-774.
- [94] B. Ding, Y. Xi, M. T. Cychowski, T. O' Mahony, "A Synthesis Approach for Output Feedback Robust Constrained Model Predictive Control", *Automatica*, 2007, In Press.

- [95] M. Cannon, V. Deshmukh, B. Kouvaritakis, "Nonlinear model predictive control with polytopic invariant sets", *Automatica*, Vol. 39, 2003, pp. 1487-1494.
- [96] G. Karer, B. Zupančič, "Modelling and Identification of a Laboratory Helicopter", *Proceedings 5th Mathematical Modelling (MATHMOD) Vienna*, Vienna University of Technology, Austria, 2006.
- [97] T. L. McClelland, D. E. Rumelhart, *et al.*, *Parallel Distributed Processing*, The MIT Press, Cambridge, 1986.
- [98] P. J. Werbo, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, Ph.D. Thesis, Harvard University, Cambridge, 1974.
- [99] S. Haykin, L. Li, "Nonlinear and Adaptive Prediction of Non-stationary Signals", *IEEE Transactions on Signal Processing*, Vol. 43, No. 2, 1995, pp. 526-535.
- [100] A. N. Michel, J. A. Farrell, "Associative Memories via Artificial Neural Networks", *IEEE Control Systems Magazine*, April 1990, pp. 6-17.
- [101] T. Kohonen, *Self-Organization and Associative Memory*, 2nd Ed., Springer-Verlag, Berlin, 1987.
- [102] G. A. Carpenter, S. Grossberg, "ART2 Self-organization of State Category Recognition Codes for Analog Input Patterns", *Applied Optics*, Vol. 26, No. 23, 1987, pp. 4919-4930.
- [103] G. A. Carpenter, S. Grossberg, D. S. Rosen, "Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System", *Neural Networks*, Vol. 4, 1991, pp. 759-771.
- [104] J. J. Hopfield, D. W. Tank, "Neural Computation of Decisions in Optimization Problems", *Biological Cybernetics*, Vol. 52, 1985, pp. 141-154.
- [105] W. S. McCulloch, W. H. Pitts, "A Logical Calculus of the Ideas Imminent in Nervous Activity", *Bulletin of Mathematical Biophysics*, Vol. 5, 1943, pp. 115-133.
- [106] M. F. Moller, "A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning", *Neural Networks*, Vol. 6, 1993, pp. 525-533.
- [107] M. T. Hagan, H. B. Demuth, M. H. Beale, *Neural Network Design*, Boston, MA: PWS Publishing, 1996.

- [108] M. Riedmiller, H. Braun, "A Direct Adaptive Method for Faster Back-propagation Learning: The RPROP Algorithm", *Proceedings of the IEEE International Conference on Neural Networks*, 1993.
- [109] R. Fletcher, C. M. Reeves, "Function Minimization by Conjugate Gradients", *Computer Journal*, Vol. 7, 1964, pp. 149-154.
- [110] M. J. D. Powell, "Restart Procedures for the Conjugate Gradient Method", *Mathematical Programming*, Vol. 12, 1977, pp. 241-254.
- [111] E. M. L. Beale, "A Derivation of Conjugate Gradients", in *F.A. Lootsma, ed., Numerical Methods for Nonlinear Optimization*, London: Academic Press, 1972.
- [112] J. E. Dennis, R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Englewood Cliffs, NJ: Prentice-Hall, 1983.
- [113] R. Battiti, "First and Second Order Methods for Learning: Between Steepest Descent and Newton's Method", *Neural Computation*, Vol. 4, No. 2, 1992, pp. 141-166.
- [114] R. Battiti, F. Masulli, "BFGS Optimization for Faster and Automated Supervised Learning", *International Neural Network Conference*, Paris, July 1990, pp. 757-760.
- [115] M. T. Hagan, M. Menhaj, "Training Feedforward Networks with the Marquardt Algorithm", *IEEE Transactions on Neural Networks*, Vol. 5, No. 6, 1994, pp. 989-993.
- [116] D. J. C. MacKay, "Bayesian Interpolation", *Neural Computation*, Vol. 4, No. 3, 1992, pp. 415-447.
- [117] F. D. Foresee, M. T. Hagan, "Gauss-Newton Approximation to Bayesian Regularization", *Proceedings of the 1997 International Joint Conference on Neural Networks*, 1997, pp. 1930-1935.
- [118] S. A. Billings, W. S. N. Voon, "Correlation Based Model Validity Tests for Non-linear Models", *International Journal of Control*, 44(1), 1986, pp. 235-244.
- [119] S. A. Billings, Q. M. Zhu, "Nonlinear Model Validation using Correlation Tests", *International Journal of Control*, 60(6), 1994, pp. 1107-1120.
- [120] H. B. Demuth, M. Beale, M. T. Hagan, *The Neural Networks Toolbox*, Mathworks Inc. USA, 2006.

- [121] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Kluwer Academic Publishers, Boston, MA, 1989.
- [122] Mathworks Inc., *Genetic Algorithm and Direct Search Toolbox*, USA, 2006.
- [123] J.-J. E. Slotine, W. Li, *Applied Nonlinear Control*, Prentice-Hall, Englewood Cliffs, NJ, 1991.
- [124] H. J. C. Huijberts, H. Nijmeijer, "Nonlinear Chaos Control and Synchronization", pp. 721-749 in: E. Schoell, H. G. Schuster, *handbook of Chaos Control*, Second Completely Revised and Enlarged Edition, Wiley-VCH, Weinheim, Germany, 2007.
- [125] H. K. Khalil, *Nonlinear Systems* (3rd Ed.), Prentice Hall, Upper Saddle River, 2002.
- [126] V. A. Yakubovich, "Matrix Inequalities Method in Stability Theory for Nonlinear Control Systems: I. Absolute Stability of Forced Vibrations", *Automation and Remote Control*, Vol. 7, 1964, pp. 905-917.
- [127] J. M. Maciejowski, *Predictive Control with Constraints*, Prentice Hall, 2002.
- [128] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", *Transactions of the ASME - Journal of Basic Engineering*, Vol. 82, pp. 35-45, 1960.
- [129] S. Julier, J. Uhlmann, H. F. Durrant-Whyte, "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators", *IEEE Transactions on Automatic Control*, Vol. 45, No. 3, 2000, pp. 477-482.
- [130] S. Julier, J. Uhlmann, "Unscented Filtering and Nonlinear Estimation", *Proceeding of the IEEE*, Vol. 92, No. 3, 2004, pp. 401-422.
- [131] Y. Wu, D. Hu, M. Wu, X. Hu, "Unscented Kalman Filtering for Additive Noise Case: Augmented versus Nonaugmented", *IEEE Signal Processing Letters*, Vol. 12, No. 5, 2005, pp. 357-360.
- [132] MathWorks Inc., *Real-Time Workshop User's Guide*, USA, 2002.
- [133] S. Boyd, L. El Ghaoui, E. Feron, V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM, Philadelphia.
- [134] P. Gahinet, A. Nemirovski, A. J. Laub, M. Chilali, *LMI Control Toolbox*, Mathworks Inc., USA, 1995.
- [135] B. C Ding, S. Y. Li, "Design and Analysis of Constrained Nonlinear Quadratic Regulator", *ISA Transactions*, Vol. 42, 2003, pp. 251-258.

- [136] S. I. Han, J. S. Kim, "Nonlinear Quadratic Gaussian Control with Loop Transfer Recovery", *Mechatronics*, Vol. 13, No. 3, 2003, pp. 273-293.
- [137] L. R. Hunt, R. Su, G. Meyer, "Global Transformations of Nonlinear Systems", *IEEE Transactions on Automatic Control*, Vol. AC-28, No. 1, 1983, pp. 24-31.
- [138] G. O. Guardabassi, S. M. Savaresi, "Approximate Linearization via Feedback-an Overview", *Automatica*, Vol. 37, 2001, pp. 1-15.
- [139] K. Robenack, "Automatic Differentiation and Nonlinear Controller Design by Exact Linearization", *Future Generation Computer Systems*, Vol. 21, 2005, pp. 1372-1379.
- [140] W. H. Chen, D. J. Ballance, P. J. Gawthrop, "Optimal Control of Nonlinear Systems: A Predictive Control Approach", *Automatica*, Vol. 39, 2003, pp. 633-641.
- [141] W. H. Chen, D. J. Ballance, P. J. Gawthrop, J. J. Gribble, J. O'Reilly, "Nonlinear PID Predictive Controller", *IEE Proc.-Control Theory Application*, Vol. 146, No. 6, 1999, pp. 603-611.
- [142] T. R. Kane, D. A. Levinson, *Dynamics: Theory and Applications*, 1st ed., McGraw-Hill Series in Mechanical Engineering, McGraw-Hill, New York, 1985.
- [143] A. A. Safavi, J. A. Romagnoli, "Application of Wavelet-based Neural Networks to the Modelling and Optimisation of an Experimental Distillation Column", *Engineering Application of Artificial Intelligence*, Vol. 10, No. 3, 1997, pp. 301-313.
- [144] M. Dorigo, V. Maniezzo, A. Colomi, "Ant System: Optimization by a Colony of Cooperating Agents", *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, Vol. 26, No. 1, 1996, pp. 29-41.
- [145] M. Dorigo, L. M. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Prob.", *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, 1997, pp. 53-66.
- [146] J. Kennedy, and R. Eberhart, "Particle Swarm Optimization", in Proc. of the *IEEE International Conference on Neural Networks*, Piscataway, NJ, 1995, pp. 1942-1948.
- [147] F. V. D. Bergh, A. P. Engelbrecht, "A Cooperative Approach to Particle Swarm Optimization", *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, June 2004, pp. 225-239.

- [148] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, "Optimization by Simulated Annealing", *Science*, Vol. 220, No. 4598, 1983, pp. 671-680.
- [149] Z.W. Geem, J. H. Kim, and G. V. Loganathan, "A New Heuristic Optimization Algorithm: Harmony Search", *Simulation*, Vol. 76, No. 2, 2001, pp. 60-68.
- [150] K. S. Lee and Z. W. Geem, "A New Meta-heuristic Algorithm for Continues Engineering Optimization: Harmony Search Theory and Practice", *Comput. Meth. Appl. Mech. Eng.*, Vol. 194, 2004, pp. 3902-3933.
- [151] S. L. Kang and Z.W. Geem, "A New Structural Optimization Method Based on the Harmony Search Algorithm", *Computers & Structures*, Vol. 82, No. 9-10, 2004, pp. 781-798.
- [152] J. H. Kim, Z. W. Geem and E. S. Kim, "Parameter Estimation of the Nonlinear Muskingum Model using Harmony Search", *Journal of the American Water Resources Association*, Vol. 37, No. 5, 2001, pp. 1131-1138.
- [153] Z. W. Geem, C. Tseng, and Y. Park, "Harmony Search for Generalized Orienteering Problem: Best Touring in China", *Springer Lecture Notes in Computer Science*, No. 3612, 2005, pp. 741-750.
- [154] J. P. Gauthier, I. Kupka, *Deterministic Observation Theory and Applications*, Cambridge University Press, Cambridge, UK, 2001.
- [155] A. J. Krener, M. Xiao, "Nonlinear Observer Design in the Siegel Domain", *SIAM Journal on Control & Optimization*, Vol. 41, No. 3, 2002, pp. 932-953.
- [156] A. J. Krener, M. Xiao, "Erratum: Nonlinear Observer Design in the Siegel Domain", *SIAM Journal on Control & Optimization*, Vol. 43, No. 1, 2004, pp. 377-378.
- [157] K. Kazantzis, C. Kravaris, "Nonlinear Observer Design using Lyapunov's Auxiliary Theorem", *Systems & Control Letter*, Vol. 34, 1998, pp. 241-247.
- [158] V. Andrieu, L. Praly, "On the Existence of a Kazantzis-Kravaris/Luenberger Observer", *SIAM Journal on Control & Optimization*, Vol. 45, 2006, pp. 432-456.
- [159] P. J. Antsaklis, A. Nerode, "Hybrid Control Systems: An Introductory Discussion to the Special Issue", *IEEE Transactions on Automatic Control*, Vol. 43, No. 4, 1998, pp. 457-460.

- [160] M. S. Branicky, V. S. Borkar, S. K. Mitter, "A Unified Framework for Hybrid Control: Model and Optimal Control Theory", *IEEE Transactions on Automatic Control*, Vol. 43, No. 1, 1998, pp. 31-45.
- [161] J. S. Milton, J. C. Arnold, *Introduction to Probability and Statistics: Principles and Applications for Engineering and the Computing Sciences*, 3rd edition, McGraw-Hill, 1995.
- [162] P. Zarchan, H. Musoff, *Fundamentals of Kalman Filtering: A Practical Approach*, American Institute of Aeronautics and Astronautics, Inc., 2005.

Appendix 1 The TRMS Parameters

Tables A1.1 and A1.2 present the measured data of the vertical and horizontal interface circuits respectively.

Table A1.1: The relationship between U_v and V_v

No.	U_v	V_v	No.	U_v	V_v	No.	U_v	V_v
1	-2.5	-20.44	18	-0.8	-7.62	35	0.9	8.80
2	-2.4	-20.02	19	-0.7	-6.82	36	1.0	9.59
3	-2.3	-19.32	20	-0.6	-6.02	37	1.1	10.35
4	-2.2	-18.52	21	-0.5	-5.20	38	1.2	11.16
5	-2.1	-17.74	22	-0.4	-4.35	39	1.3	11.93
6	-2.0	-16.98	23	-0.3	-3.46	40	1.4	12.71
7	-1.9	-16.20	24	-0.2	-2.46	41	1.5	13.49
8	-1.8	-15.43	25	-0.1	-1.13	42	1.6	14.27
9	-1.7	-14.65	26	0	0.35	43	1.7	15.06
10	-1.6	-13.87	27	0.1	1.80	44	1.8	15.87
11	-1.5	-13.09	28	0.2	3.00	45	1.9	16.69
12	-1.4	-12.33	29	0.3	3.80	46	2.0	17.50
13	-1.3	-11.55	30	0.4	4.80	47	2.1	18.32
14	-1.2	-10.78	31	0.5	5.62	48	2.2	19.06
15	-1.1	-10.00	32	0.6	6.42	49	2.3	19.69
16	-1.0	-9.20	33	0.7	7.22	50	2.4	19.90
17	-0.9	-8.42	34	0.8	8.01	51	2.5	20.13

Table A1.2: The relationship between U_h and V_h

No.	U_h	V_h	No.	U_h	V_h	No.	U_h	V_h
1	-2.5	-15.38	18	-0.8	-5.90	35	0.9	6.69
2	-2.4	-14.84	19	-0.7	-5.31	36	1.0	7.26
3	-2.3	-14.29	20	-0.6	-4.71	37	1.1	7.83
4	-2.2	-13.74	21	-0.5	-4.09	38	1.2	8.40
5	-2.1	-13.19	22	-0.4	-3.44	39	1.3	8.96
6	-2.0	-12.65	23	-0.3	-2.72	40	1.4	9.52
7	-1.9	-12.10	24	-0.2	-1.85	41	1.5	10.08
8	-1.8	-11.54	25	-0.1	-0.79	42	1.6	10.63
9	-1.7	-10.98	26	0	0	43	1.7	11.18
10	-1.6	-10.43	27	0.1	1.27	44	1.8	11.74
11	-1.5	-9.87	28	0.2	2.24	45	1.9	12.29
12	-1.4	-9.32	29	0.3	3.02	46	2.0	12.84
13	-1.3	-8.75	30	0.4	3.69	47	2.1	13.39
14	-1.2	-8.19	31	0.5	4.33	48	2.2	13.94
15	-1.1	-7.62	32	0.6	4.93	49	2.3	14.49
16	-1.0	-7.05	33	0.7	5.53	50	2.4	15.04
17	-0.9	-6.48	34	0.8	6.11	51	2.5	15.59

The parameters of the TRMS are listed in Table A1.3.

Figure A1.1 shows the measured horizontal aerodynamic force against the rotational velocity of the tail motor and the corresponding model. Although the maximum rotational speed of the tail motor is 837 rad/sec at no load condition, the motor with propeller can rotate at the maximum rate of 400 rad/sec due to the electrical current limitation. The measured data of the horizontal channel is listed in Table A1.4.

Table A1.3: The parameters of the TRMS

Parameter	Value	Parameter	Value
l_t (m)	0.282	R_a (Ω)	8.00
l_m (m)	0.246	L_a (H)	8.60×10^{-4}
l_b (m)	0.290	$k_a \cdot \phi$ (Nm/A)	2.02×10^{-2}
l_{cb} (m)	0.276	J_{mr} (kg m^2)	2.16×10^{-4}
r_{ms} (m)	0.155	J_{tr} (kg m^2)	3.14×10^{-5}
r_{ts} (m)	0.100	B_{tr} ($\text{kg m}^2/\text{s}$)	2.30×10^{-5}
m_{tr} (kg)	0.221	B_{mr} ($\text{kg m}^2/\text{s}$)	4.50×10^{-5}
m_{mr} (kg)	0.236	k_{thp} ($\text{Nms}^2/\text{rad}^2$)	5.00×10^{-8}
m_{cb} (kg)	0.068	k_{thn} ($\text{Nms}^2/\text{rad}^2$)	4.20×10^{-8}
m_t (kg)	0.015	k_{tvp} ($\text{Nms}^2/\text{rad}^2$)	5.60×10^{-7}
m_m (kg)	0.014	k_{tvn} ($\text{Nms}^2/\text{rad}^2$)	5.10×10^{-7}
m_b (kg)	0.022	k_{fhp} (Ns^2/rad^2)	1.84×10^{-6}
m_{ts} (kg)	0.119	k_{fhn} (Ns^2/rad^2)	2.20×10^{-6}
m_{ms} (kg)	0.219	k_{fvp} (Ns^2/rad^2)	1.57×10^{-5}
k_g (ms/rad)	2.00×10^{-1}	k_{fvn} (Ns^2/rad^2)	8.30×10^{-6}
k_t	2.60×10^{-5}	k_m	2.00×10^{-4}

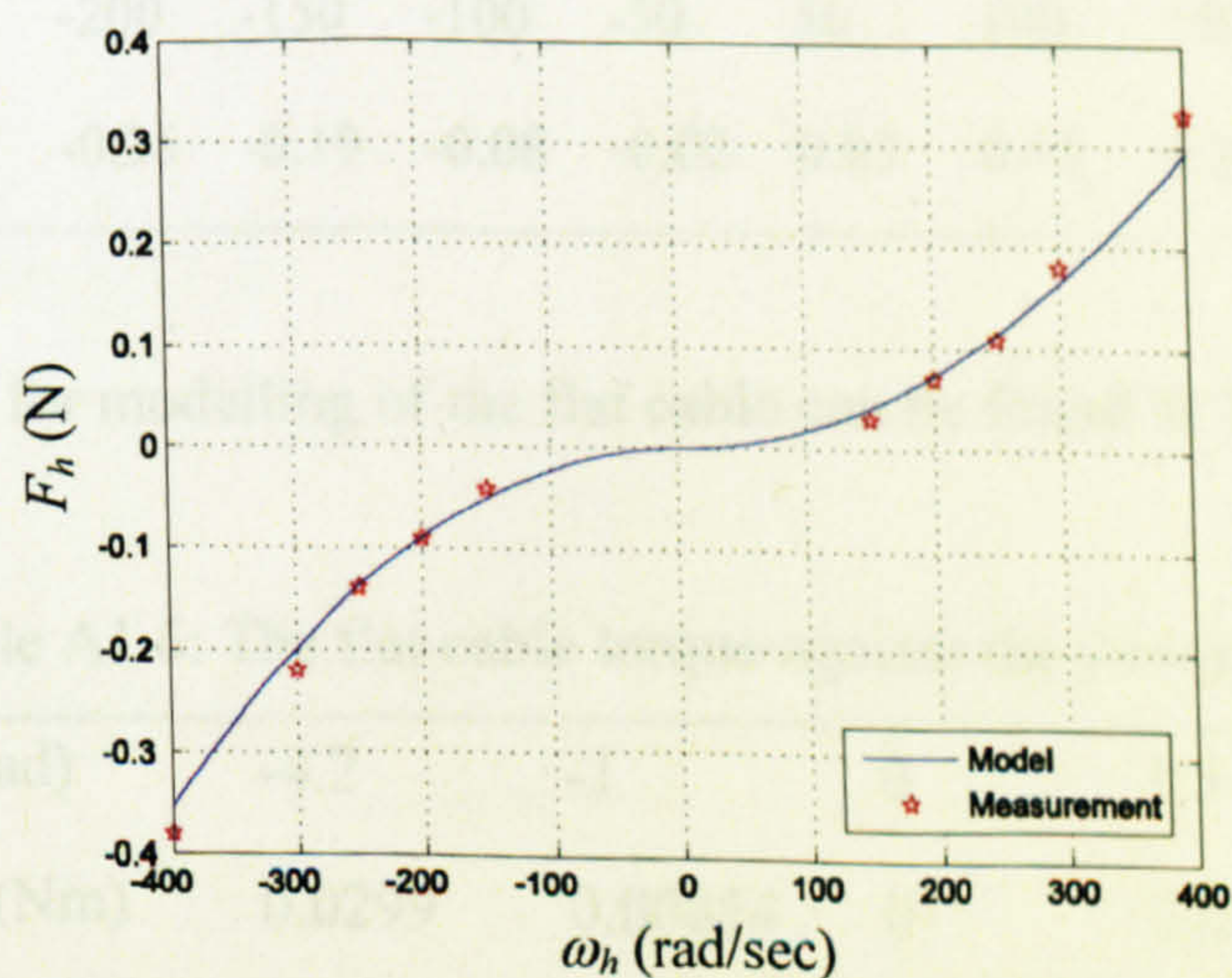


Figure A1.1: The aerodynamic force due to the tail rotor versus its rotational velocity

Table A1.4: The aerodynamic force of the tail rotor against its velocity

ω_h (rad/s)	-400	-300	-250	-200	-150	150	200	250	300	400
F_h (N)	-0.38	-0.22	-0.14	-0.09	-0.04	0.03	0.07	0.11	0.18	0.33

Similar results for the vertical channel are presented in Figure A1.2 where the maximum rotational speed of the main rotor with propeller is 250 rad/sec. As mentioned before, the figures are not symmetric. The measured data of the vertical channel is listed in Table A1.5.

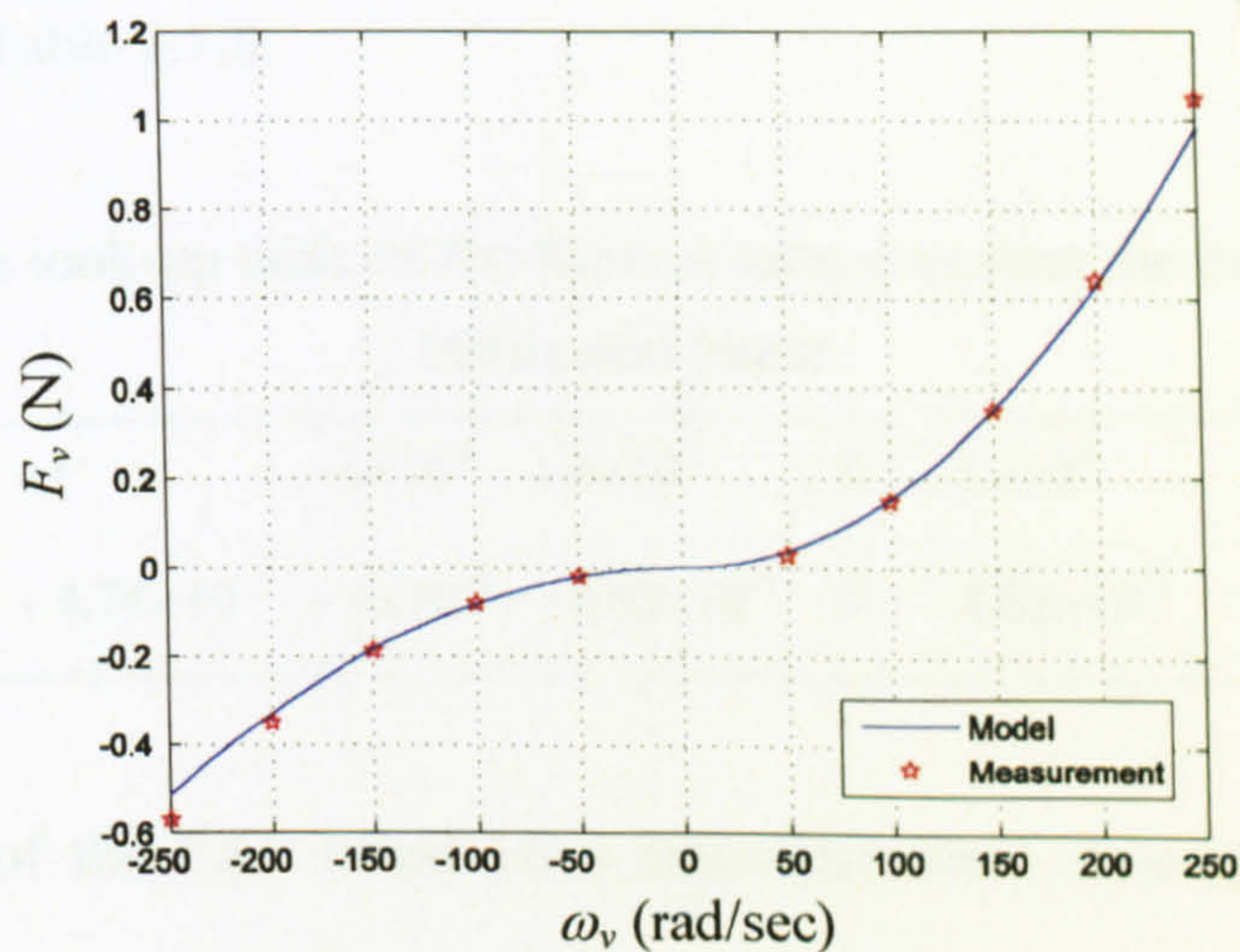


Figure A1.2: The aerodynamic force due to the main rotor versus the rotational velocity of the main rotor

Table A1.5: The aerodynamic force of the main rotor against its velocity

ω_v (rad/s)	-250	-200	-150	-100	-50	50	100	150	200	250
F_v (N)	-0.57	-0.35	-0.19	-0.08	-0.02	0.03	0.15	0.35	0.64	1.05

The measured data for modelling of the flat cable can be found in Table A1.6.

Table A1.6: The flat cable torque against the yaw angle

α_h (rad)	-4.2	-1	0	1.5
M_{h3} (Nm)	0.0299	0.00854	0	0.0128

The look-up table of the friction torque against the beam speed in the vertical plane is shown in Table A1.7.

Table A1.7: The look-up table of the friction torque against the beam speed in the vertical plane

Ω_v (rad/s)	-2	-7×10^{-4}	-5×10^{-4}	0	5×10^{-4}	7×10^{-4}	2
$M_{fric,v}$ (Nm)	-2.77×10^{-3}	-1.509×10^{-4}	-1.51×10^{-4}	0	1.51×10^{-4}	1.509×10^{-4}	2.77×10^{-3}

The look-up table of the friction torque against the beam speed in the horizontal plane is found in Table A1.8.

Table A1.8: The look-up table of the friction torque against the beam speed in the horizontal plane

Ω_h (rad/s)	-1	-1×10^{-4}	-5×10^{-5}	0	5×10^{-5}	1×10^{-4}	1
$M_{fric,h}$ (Nm)	-4.74×10^{-3}	-4×10^{-5}	-4.02×10^{-5}	0	4.02×10^{-5}	4×10^{-5}	4.74×10^{-3}

The parameters of the EKF based on a state-dependent state-space model are as follows,

$$\mathbf{R} = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 \end{bmatrix}$$

The parameters of the UKF are,

$$E[\mathbf{w}(k)] = \begin{bmatrix} 1 \\ 0.1 \\ 0.032 \\ 1 \\ 0.1 \\ 0.032 \end{bmatrix}, \quad E[\mathbf{v}(k)] = \begin{bmatrix} 0.032 \\ 0.032 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} 0.4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.4 \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} 0.4 & 0 \\ 0 & 0.4 \end{bmatrix}, \mathbf{P}(-1) = \mathbf{I}_{6 \times 6}, \alpha = 0.001, \beta = 2, \kappa = 0,$$

Table A1.9 lists the optimum parameters from the GA-based grey-box identification approach. For convenient, the parameters of the white-box approach are also listed for comparison.

Table A1.9: The optimum parameters from the GA-based grey-box identification

Parameter	White-box	Grey-box
J_{mr} (kg m ²)	2.16×10^{-4}	2.01×10^{-4}
J_{tr} (kg m ²)	3.14×10^{-5}	3.14×10^{-5}
B_{tr} (kg m ² /s)	2.30×10^{-5}	2.30×10^{-5}
B_{mr} (kg m ² /s)	4.50×10^{-5}	2.97×10^{-5}
k_{thp} (Nms ² /rad ²)	5.00×10^{-8}	5.00×10^{-8}
k_{thn} (Nms ² /rad ²)	4.20×10^{-8}	4.41×10^{-8}
k_{tvp} (Nms ² /rad ²)	5.60×10^{-7}	5.60×10^{-7}
k_{tvn} (Nms ² /rad ²)	5.10×10^{-7}	5.10×10^{-7}
k_{fhp} (Ns ² /rad ²)	1.84×10^{-6}	2.14×10^{-6}
k_{fhn} (Ns ² /rad ²)	2.20×10^{-6}	1.91×10^{-6}
k_{fvp} (Ns ² /rad ²)	1.57×10^{-5}	1.95×10^{-5}
k_{fvn} (Ns ² /rad ²)	8.30×10^{-6}	1.10×10^{-5}
k_m (-)	2.00×10^{-4}	2.30×10^{-4}
k_t (-)	2.60×10^{-5}	2.60×10^{-5}
k_g (ms/rad)	2.00×10^{-1}	1.84×10^{-1}
J_v (kg m ²)	6.49×10^{-2}	5.26×10^{-2}
k_{vfv} (Nms/rad)	1.31×10^{-3}	5.48×10^{-3}
k_{vfh} (Nms/rad)	4.70×10^{-3}	4.91×10^{-3}
k_{chp} (Nm/rad)	8.54×10^{-3}	5.60×10^{-3}
k_{chn} (Nm/rad)	8.54×10^{-3}	5.60×10^{-3}

Appendix 2 Equation Derive

A2.1 Centrifugal Force

Assume a particle with mass m connected to a light beam of length l is rotating about the z -axis with the angular velocity of Ω , and suppose the angle of the beam with the horizontal plane is α , as shown in Figure A2.1. The centrifugal force according to the shown direction can be calculated as follows.

$$F = ma \quad (\text{A2.1})$$

$$a = -\Omega^2 r = -\Omega^2 l \cos \alpha \quad (\text{A2.2})$$

$$F = -m\Omega^2 l \cos \alpha \quad (\text{A2.3})$$

Note that a is the centrifugal acceleration. Therefore, the centrifugal force and the centrifugal torque with specific direction are respectively expressed as,

$$F_c = F \sin \alpha = -ml\Omega^2 \cos \alpha \sin \alpha \quad (\text{A2.4})$$

$$M_c = F_c l = -ml^2 \Omega^2 \cos \alpha \sin \alpha \quad (\text{A2.5})$$

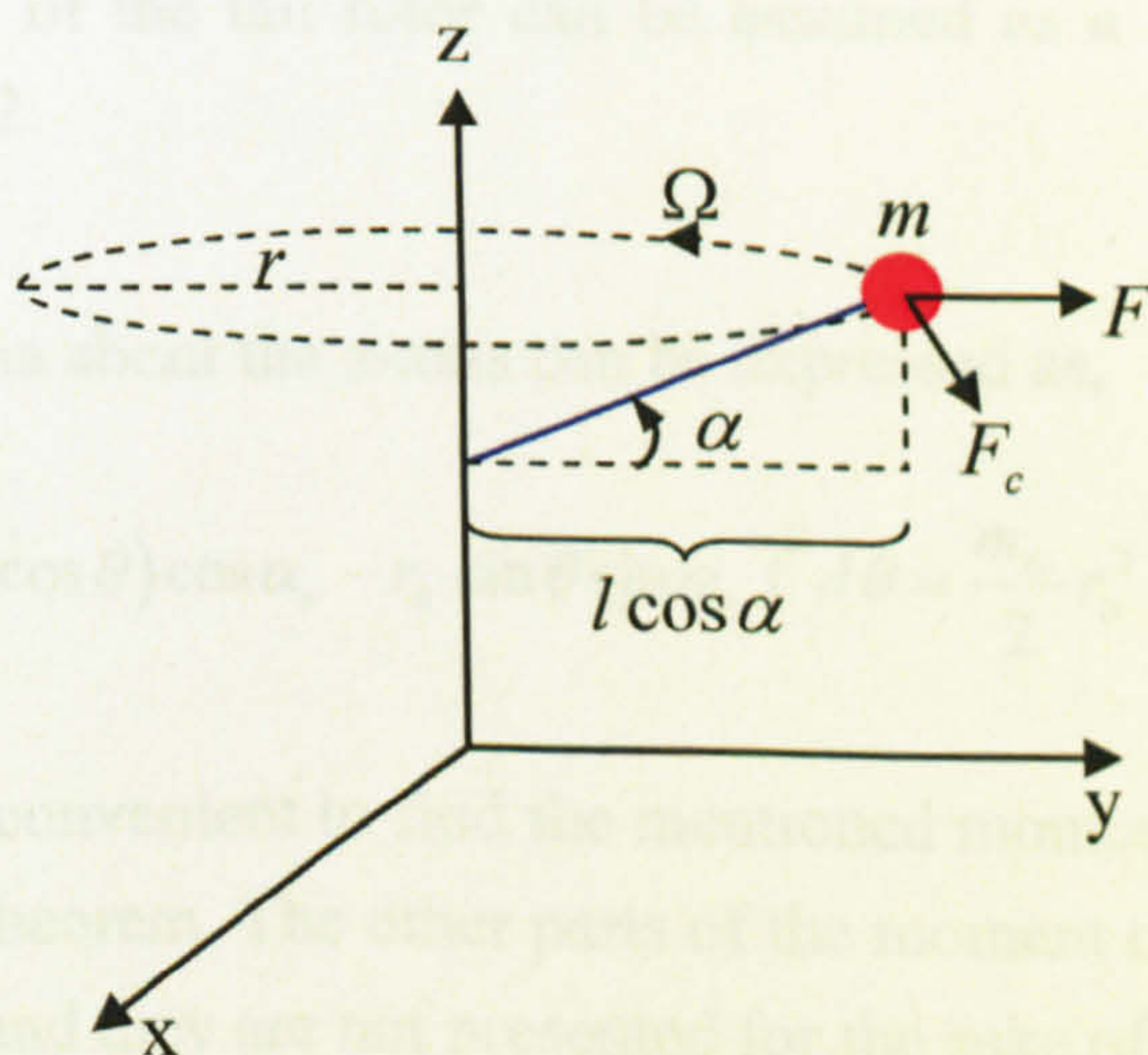


Figure A2.1: Centrifugal force of a particle

where,

r : Radius of rotation (m)

a : Acceleration (m/s²)

F_c : Centrifugal force (N)

M_c : Centrifugal torque (Nm).

A2.2 Moment of inertia

Assume the shield of the main rotor as a hoop with radius r_{ms} which its centre is connected to a point on the z-axis by the main beam of the TRMS with the length l_m . Suppose the angle between the beam and the x-y plane is α_v . The moment of inertia of the main shield about the z-axis can be found as,

$$I_z = \int r^2 dm \quad (\text{A2.6})$$

where, in the case of the main shield $r^2 = (r_{ms} \sin \theta)^2 + (l_m - r_{ms} \cos \theta)^2 \cos^2 \alpha_v$ and $dm = \frac{m_{ms}}{2\pi} d\theta$. Substituting into (A2.6) results in,

$$I_{1z} = \frac{m_{ms}}{2\pi} \int_0^{2\pi} \left[(r_{ms} \sin \theta)^2 + (l_m - r_{ms} \cos \theta)^2 \cos^2 \alpha_v \right] d\theta = m_{ms} r_{ms}^2 + m_{ms} l_m^2 \cos^2 \alpha_v \quad (\text{A2.7})$$

Similarly, the shield of the tail rotor can be assumed as a hoop with radius r_{ts} as shown in Figure A2.2.

The moment of inertia about the z-axis can be expressed as,

$$I_{2z} = \frac{m_{ts}}{2\pi} \int_0^{2\pi} \left[(l_t - r_{ts} \cos \theta) \cos \alpha_v - r_{ts} \sin \theta \sin \alpha_v \right]^2 d\theta = \frac{m_{ts}}{2} r_{ts}^2 + m_{ts} l_t^2 \cos^2 \alpha_v \quad (\text{A2.8})$$

Note that it is more convenient to find the mentioned moment of inertia using the so-called parallel axis theorem. The other parts of the moment of inertia about the z-axis are straightforward and they are not presented for the sake of brevity.

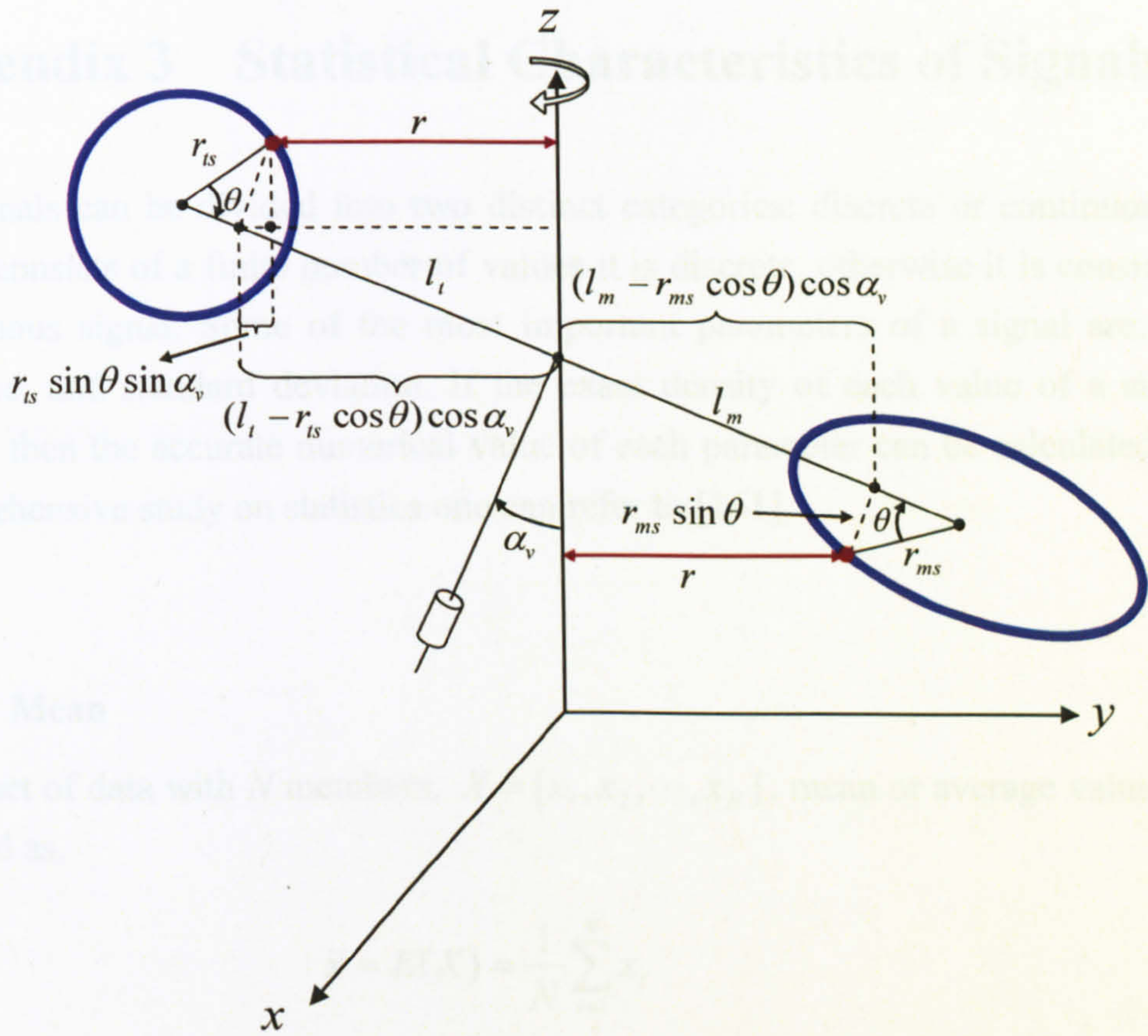


Figure A2.2: The partial representation of the system

E : Expectation operator

\bar{x} : mean value

N : Number of data

x_i : i -th member of data set.

It is noted that the more general form of the expectation for a signal to which members have different probabilities can be expressed as

$$E(x) = \sum_{i=1}^N (\text{value of } x_i) (\text{probability of } x_i) \quad (A2.2)$$

For a signal that its members have the same probability, the probability of each member is $1/N$ and equation (A2.2) leads to (A2.1).

A2.2: Variance

For a set of data with N members, $X = \{x_1, x_2, \dots, x_N\}$, variance can be calculated using

Appendix 3 Statistical Characteristics of Signals

All signals can be divided into two distinct categories: discrete or continuous. If a signal consists of a finite number of values it is discrete, otherwise it is considered a continuous signal. Some of the most important parameters of a signal are: mean, variance, and standard deviation. If the exact density of each value of a signal is known then the accurate numerical value of each parameter can be calculated. For a comprehensive study on statistics one can refer to [161].

A3.1 Mean

For a set of data with N members, $X = [x_1, x_2, \dots, x_N]$, mean or average value can be defined as,

$$\bar{x} = E(X) = \frac{1}{N} \sum_{i=1}^N x_i \quad (\text{A3.1})$$

where,

E : Expectation operator

\bar{x} : mean value

N : Number of data

x_i : i -th member of data set.

It is noted that the more general form of the expectation for a signal in which members have different probabilities can be expressed as,

$$E(X) = \sum_{\text{all } x} (\text{value of } x) (\text{probability of } x) \quad (\text{A3.2})$$

For a signal that its members have the same probability, the probability of each member is $1/N$ and equation (A3.2) leads to (A3.1).

A3.2 Variance

For a set of data with N members, $X = [x_1, x_2, \dots, x_N]$, variance can be calculated using,

$$\sigma^2 = E(X - \bar{x})^2 = \begin{cases} \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 & \text{for entire population } (n = N) \\ \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 & \text{for a sample of population } (n < N) \end{cases} \quad (\text{A3.3})$$

A3.3 Standard deviation

Standard deviation is the square root of variance and can be found as,

$$\sigma = \sqrt{E(X - \bar{x})^2} = \begin{cases} \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} & \text{if } n = N \\ \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} & \text{if } n < N \end{cases} \quad (\text{A3.4})$$

A3.4 Covariance

For a set of data with N members, $X = [x_1, x_2, \dots, x_N]$ the covariance with lag τ can be defined as,

$$R_x(\tau) = \frac{\sum_{i=1}^{N-\tau} (x_i - \bar{x})(x_{i+\tau} - \bar{x})}{N} \quad (\text{A3.5})$$

The covariance between two sets of data with the same length, $X = [x_1, x_2, \dots, x_N]$ and $Y = [y_1, y_2, \dots, y_N]$ can be defined as,

$$R_{xy}(\tau) = \frac{\sum_{i=1}^{N-\tau} (x_i - \bar{x})(y_{i+\tau} - \bar{y})}{N} \quad (\text{A3.6})$$

A3.5 Auto-correlation and cross-correlation

For a set of data with N members, $X = [x_1, x_2, \dots, x_N]$ the auto-correlation can be defined as,

$$r_{xx}(\tau) = \frac{\sum_{i=1}^{N-\tau} (x_i - \bar{x})(x_{i+\tau} - \bar{x})}{\sum_{i=1}^N (x_i - \bar{x})^2} \quad (\text{A3.7})$$

The auto-correlation of lag zero $\tau = 0$ is always equal to one. The cross-correlation between two sets of data with the same length, $X = [x_1, x_2, \dots, x_N]$ and $Y = [y_1, y_2, \dots, y_N]$, is equal to the covariance of two sets over the product of their standard deviations and can be defined as,

$$r_{xy}(\tau) = \begin{cases} \frac{\sum_{i=1}^{N-\tau} (x_i - \bar{x})(y_{i+\tau} - \bar{y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2 \sum_{i=1}^N (y_i - \bar{y})^2}} & \tau = 0, 1, \dots, N-1 \\ \frac{\sum_{i=1-\tau}^N (x_i - \bar{x})(y_{i+\tau} - \bar{y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2 \sum_{i=1}^N (y_i - \bar{y})^2}} & \tau = -N+1, -N+2, \dots, -1 \end{cases} \quad (\text{A3.8})$$

A3.6 Probability density function

The general form of a probability density function, $f(x)$, for a signal with average μ and standard deviation σ can be defined as the following Gaussian function (see Figure A3.1),

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (\text{A3.9})$$

The standard normal distribution that is a normal distribution with mean zero and standard deviation one is found by substitution $Z = (X - \mu)/\sigma$ in (A3.9) as,

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2} \quad (\text{A3.10})$$

The probability of having x between the interval $[a, b]$ can be expressed as,

$$P(a \leq x \leq b) = \int_a^b f(x) dx \quad (\text{A3.11})$$

That is equal to the probability of having z between the interval $[(a-\mu)/\sigma, (b-\mu)/\sigma]$ as follows,

$$P\left(\frac{a-\mu}{\sigma} \leq z \leq \frac{b-\mu}{\sigma}\right) = \int_{\hat{a}}^{\hat{b}} f(z) dz = \frac{1}{\sqrt{2\pi}} \int_{\hat{a}}^{\hat{b}} e^{-\frac{1}{2}z^2} dz \quad (\text{A3.12})$$

where $\hat{a} = \frac{a-\mu}{\sigma}$ and $\hat{b} = \frac{b-\mu}{\sigma}$. As the integral $\frac{1}{\sqrt{2\pi}} \int_{\hat{a}}^{\hat{b}} e^{-\frac{1}{2}z^2} dz$ cannot be solved analytically, therefore numerical solution and forming a table is an ordinary way to overcome this problem.

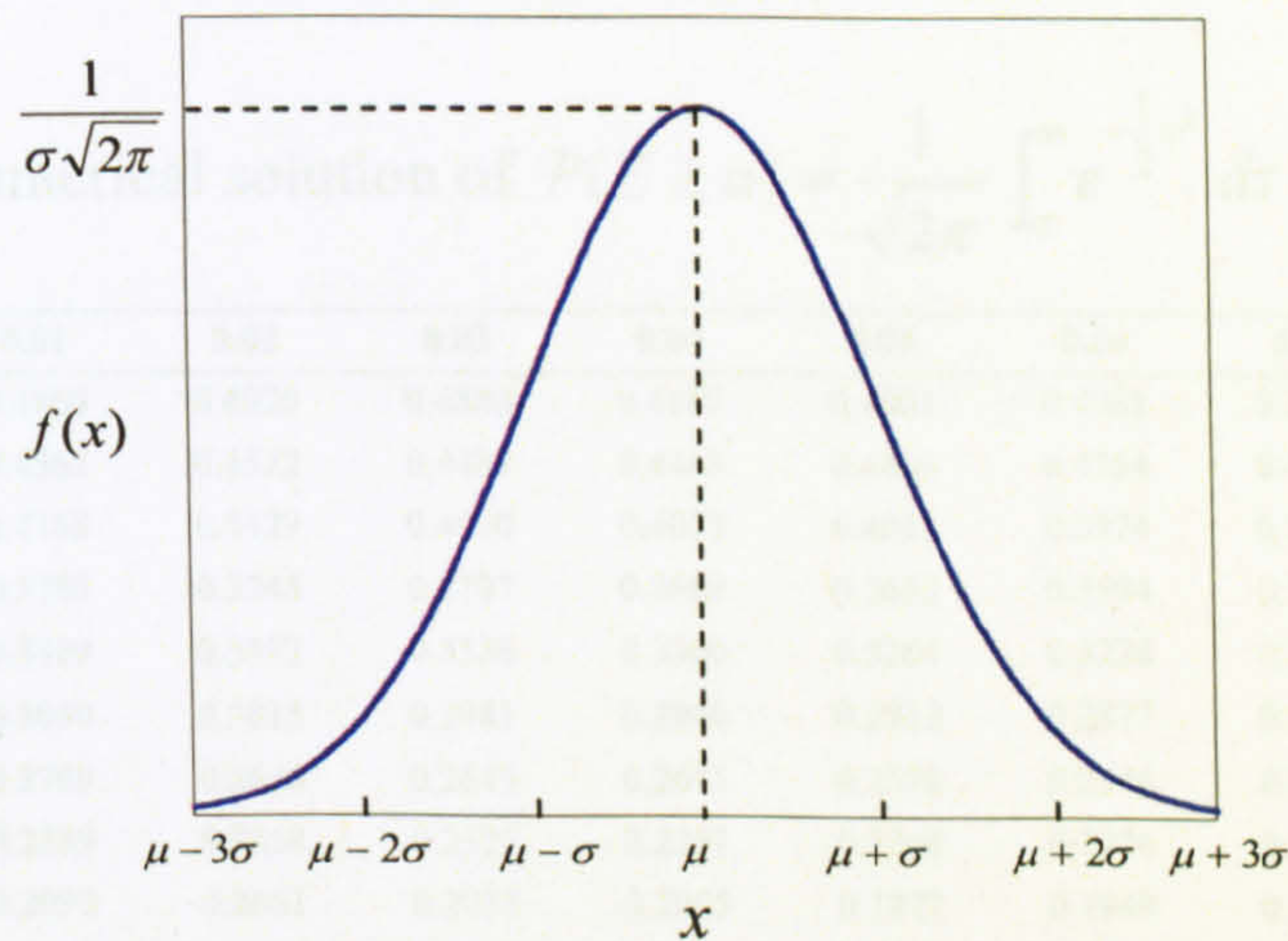


Figure A3.1: A probability density function

A3.7 Confidence interval on the mean

Assume a $1-\alpha$ confidence interval for μ is desired. In other words, there is an interval like $[a, b]$ in which,

$$P(a \leq \mu \leq b) = 1 - \alpha \quad (\text{A3.13})$$

In order to construct a confidence interval for μ according to the measured value of mean, \bar{x} , one can write,

$$P\left(\bar{x} - \frac{\sigma}{\sqrt{N}} z_{\alpha/2} \leq \mu \leq \bar{x} + \frac{\sigma}{\sqrt{N}} z_{\alpha/2}\right) = 1 - \alpha \quad (\text{A3.14})$$

where N is the number of measured values and $z_{\alpha/2}$ can be defined as,

$$P(Z \geq z_{\alpha/2}) = \frac{\alpha}{2} \quad (\text{A3.15})$$

As pointed out before, $z_{\alpha/2}$ is found using the numerical solution of integral (A3.12)

as listed in Table A3.1. Therefore, the 95% confidence interval refers to the fact that,

$$1 - \alpha = 0.95 \Rightarrow \alpha = 0.05 \Rightarrow \alpha/2 = 0.025$$

According to Table A3.1, the value of $z_{\alpha/2} = z_{0.025}$ is equal to 1.96, and the 95% confidence interval for a signal with zero mean and standard deviation of one, based on (A3.14), is found as,

$$0 - 1.96 \frac{1}{\sqrt{N}} \leq \mu \leq 0 + 1.96 \frac{1}{\sqrt{N}} \Rightarrow -\frac{1.96}{\sqrt{N}} \leq \mu \leq \frac{1.96}{\sqrt{N}}$$

Table A3.1: Numerical solution of $P(Z \geq a) = \frac{1}{\sqrt{2\pi}} \int_a^{\infty} e^{-\frac{1}{2}z^2} dz$ with respect to a

α	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0	0.5000	0.4960	0.4920	0.4880	0.4840	0.4801	0.4761	0.4721	0.4681	0.4641
0.1	0.4602	0.4562	0.4522	0.4483	0.4443	0.4404	0.4364	0.4325	0.4286	0.4247
0.2	0.4207	0.4168	0.4129	0.4090	0.4052	0.4013	0.3974	0.3936	0.3897	0.3859
0.3	0.3821	0.3783	0.3745	0.3707	0.3669	0.3632	0.3594	0.3557	0.3520	0.3483
0.4	0.3446	0.3409	0.3372	0.3336	0.3300	0.3264	0.3228	0.3192	0.3156	0.3121
0.5	0.3085	0.3050	0.3015	0.2981	0.2946	0.2912	0.2877	0.2843	0.2810	0.2776
0.6	0.2743	0.2709	0.2676	0.2643	0.2611	0.2578	0.2546	0.2514	0.2483	0.2451
0.7	0.2420	0.2389	0.2358	0.2327	0.2297	0.2266	0.2236	0.2207	0.2177	0.2148
0.8	0.2119	0.2090	0.2061	0.2033	0.2005	0.1977	0.1949	0.1922	0.1894	0.1867
0.9	0.1841	0.1814	0.1788	0.1762	0.1736	0.1711	0.1685	0.1660	0.1635	0.1611
1.0	0.1587	0.1562	0.1539	0.1515	0.1492	0.1469	0.1446	0.1423	0.1401	0.1379
1.1	0.1357	0.1335	0.1314	0.1292	0.1271	0.1251	0.1230	0.1210	0.1190	0.1170
1.2	0.1151	0.1131	0.1112	0.1093	0.1075	0.1057	0.1038	0.1020	0.1003	0.0985
1.3	0.0968	0.0951	0.0934	0.0918	0.0901	0.0885	0.0869	0.0853	0.0838	0.0823
1.4	0.0808	0.0793	0.0778	0.0764	0.0749	0.0735	0.0721	0.0708	0.0694	0.0681
1.5	0.0668	0.0655	0.0643	0.0630	0.0618	0.0606	0.0594	0.0582	0.0571	0.0559
1.6	0.0548	0.0537	0.0526	0.0516	0.0505	0.0495	0.0485	0.0475	0.0465	0.0455
1.7	0.0446	0.0436	0.0427	0.0418	0.0409	0.0401	0.0392	0.0384	0.0375	0.0367
1.8	0.0359	0.0351	0.0344	0.0336	0.0329	0.0322	0.0314	0.0307	0.0301	0.0294
1.9	0.0287	0.0281	0.0274	0.0268	0.0262	0.0256	0.0250	0.0244	0.0239	0.0233
2.0	0.0227	0.0222	0.0217	0.0212	0.0207	0.0202	0.0197	0.0192	0.0188	0.0183
2.1	0.0179	0.0174	0.0170	0.0166	0.0162	0.0158	0.0154	0.0150	0.0146	0.0143
2.2	0.0139	0.0136	0.0132	0.0129	0.0125	0.0122	0.0119	0.0116	0.0113	0.0110
2.3	0.0107	0.0104	0.0102	0.0099	0.0096	0.0094	0.0091	0.0089	0.0087	0.0084
2.4	0.0082	0.0080	0.0078	0.0075	0.0073	0.0071	0.0069	0.0068	0.0066	0.0064
2.5	0.0062	0.0060	0.0059	0.0057	0.0055	0.0054	0.0052	0.0051	0.0049	0.0048
2.6	0.0047	0.0045	0.0044	0.0043	0.0041	0.0040	0.0039	0.0038	0.0037	0.0036
2.7	0.0035	0.0034	0.0033	0.0032	0.0031	0.0030	0.0029	0.0028	0.0027	0.0026
2.8	0.0026	0.0025	0.0024	0.0023	0.0023	0.0022	0.0021	0.0021	0.0020	0.0019
2.9	0.0019	0.0018	0.0018	0.0017	0.0016	0.0016	0.0015	0.0015	0.0014	0.0014
3.0	0.0013	0.0013	0.0013	0.0012	0.0012	0.0011	0.0011	0.0011	0.0010	0.0010
3.1	0.0010	0.0009	0.0009	0.0009	0.0008	0.0008	0.0008	0.0008	0.0007	0.0007
3.2	0.0007	0.0007	0.0006	0.0006	0.0006	0.0006	0.0006	0.0005	0.0005	0.0005
3.3	0.0005	0.0005	0.0005	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0003
3.4	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0002
3.5	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
3.6	0.0002	0.0002	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
3.7	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
3.8	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
3.9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Appendix 4 Kalman Filter

A4.1 EKF

The conventional Kalman filter is widely used for linear systems; however, most real processes are nonlinear. Therefore, EKF has been introduced to observe state variables of nonlinear systems. Assume a nonlinear state-space plant-model

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) + \mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{h}(\mathbf{x}(k)) + \mathbf{v}(k) \end{aligned} \quad (\text{A4.1})$$

where $\mathbf{w}(k)$ and $\mathbf{v}(k)$ are white noise with covariance matrices $\mathbf{Q}(k)$ and $\mathbf{R}(k)$ respectively. Optimum state variables estimation can be obtained using the following equations iteratively,

Prediction stage

$$\hat{\mathbf{x}}(k|k-1) = \mathbf{f}(\hat{\mathbf{x}}(k-1|k-1), \mathbf{u}(k)) \quad (\text{A4.2})$$

$$\mathbf{M}(k) = \mathbf{A}(k)\mathbf{P}(k-1)\mathbf{A}^T(k) + \mathbf{Q}(k) \quad (\text{A4.3})$$

where, $\hat{\mathbf{x}}(k|k-1)$ is the predicted state and $\mathbf{M}(k)$ is the predicted error covariance. Also matrix $\mathbf{A}(k)$ is defined using the following Jacobian,

$$\mathbf{A}(k) = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}(k-1|k-1), \mathbf{u}(k)} \quad (\text{A4.4})$$

Note that instead of the Jacobian-based linearisation, it is also possible to use a state-dependent system, and based on the previous sample time data, update the linear model.

Correction or update stage

$$\mathbf{L}'(k) = \mathbf{M}(k)\mathbf{C}^T(k) [\mathbf{C}(k)\mathbf{M}(k)\mathbf{C}^T(k) + \mathbf{R}(k)]^{-1} \quad (\text{A4.5})$$

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{L}'(k) [\mathbf{y}(k) - \mathbf{h}(\hat{\mathbf{x}}(k|k-1))] \quad (\text{A4.6})$$

$$\mathbf{P}(k) = (\mathbf{I} - \mathbf{L}'(k)\mathbf{C}(k))\mathbf{M}(k) \quad (\text{A4.7})$$

where, $\mathbf{L}'(k)$ is the Kalman gain, $\hat{\mathbf{x}}(k|k)$ is the updated state estimate, and $\mathbf{P}(k)$ is the update error covariance [162]. Matrix $\mathbf{C}(k)$ is defined as follows,

$$C(k) = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}(k|k-1)} \quad (\text{A4.8})$$

The flowchart of the extended Kalman state observer procedure is depicted in Figure A4.1.

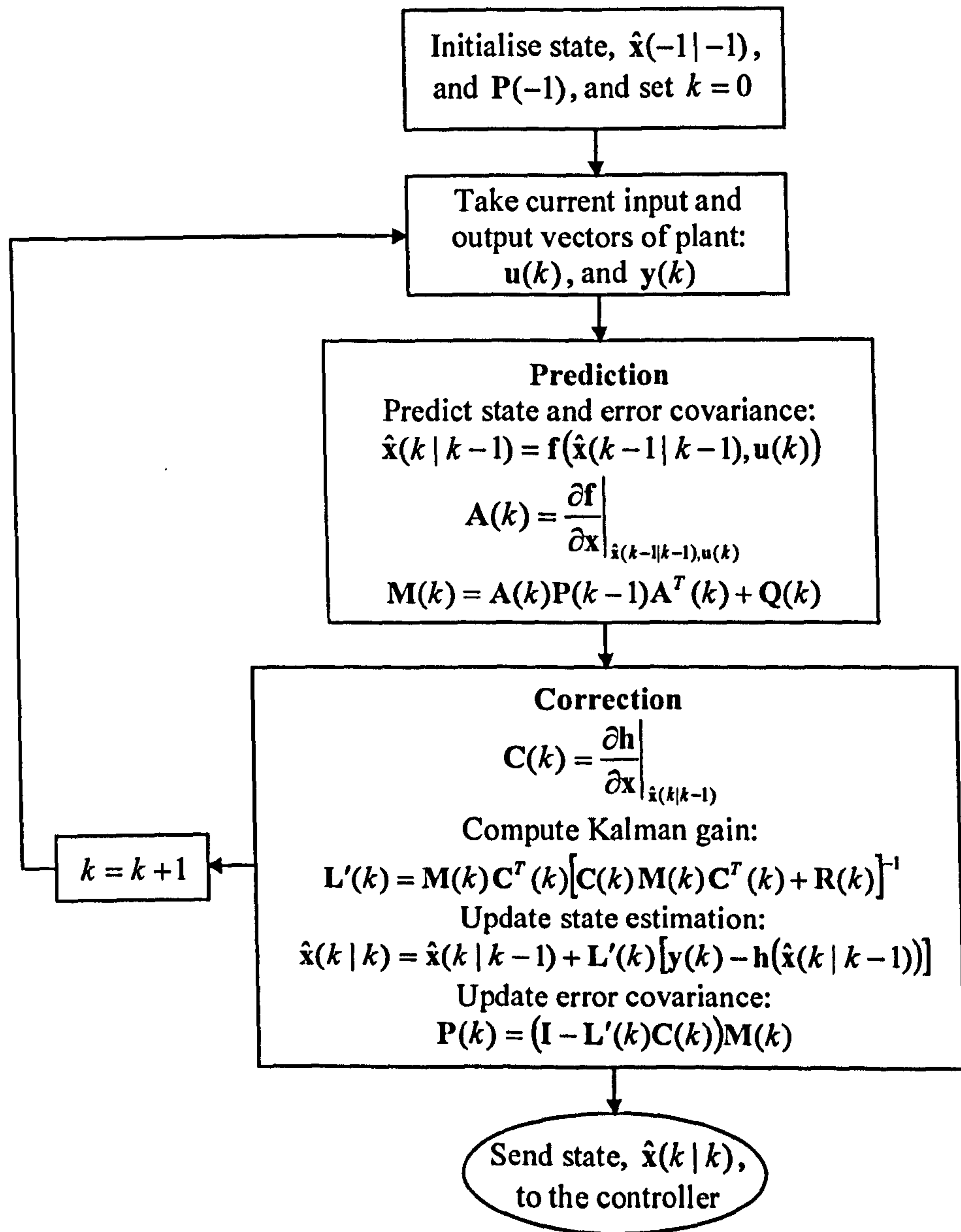


Figure A4.1: Flowchart of the extended Kalman observer procedure

A4.2 UKF

Prediction stage

The first step of a UKF is to augment the estimated state and covariance with the mean and covariance of the process noise, $\mathbf{w}(k)$ respectively.

$$\mathbf{x}_a(k-1|k-1) = \begin{bmatrix} \hat{\mathbf{x}}(k-1|k-1) \\ E[\mathbf{w}(k)] \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{a1}(k-1|k-1) \\ \mathbf{x}_{a2}(k-1|k-1) \end{bmatrix} \quad (\text{A4.9})$$

$$\mathbf{P}_a(k-1|k-1) = \begin{bmatrix} \mathbf{P}(k-1|k-1) & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}(k) \end{bmatrix} \quad (\text{A4.10})$$

According to these augmented state and covariance, a set of $2L+1$ sigma points is found as follows where L is the dimension of the augmented state,

$$\boldsymbol{\chi}_0(k-1|k-1) = \mathbf{x}_a(k-1|k-1) = \begin{bmatrix} \boldsymbol{\chi}_{01}(k-1|k-1) \\ \boldsymbol{\chi}_{02}(k-1|k-1) \end{bmatrix} \quad (\text{A4.11})$$

$$\begin{aligned} \boldsymbol{\chi}_i(k-1|k-1) &= \mathbf{x}_a(k-1|k-1) + \left(\sqrt{(L+\lambda)\mathbf{P}_a(k-1|k-1)} \right)_i \\ &= \begin{bmatrix} \boldsymbol{\chi}_{i1}(k-1|k-1) \\ \boldsymbol{\chi}_{i2}(k-1|k-1) \end{bmatrix} \quad i = 1, \dots, L \end{aligned} \quad (\text{A4.12})$$

$$\begin{aligned} \boldsymbol{\chi}_i(k-1|k-1) &= \mathbf{x}_a(k-1|k-1) - \left(\sqrt{(L+\lambda)\mathbf{P}_a(k-1|k-1)} \right)_{i-L} \\ &= \begin{bmatrix} \boldsymbol{\chi}_{i1}(k-1|k-1) \\ \boldsymbol{\chi}_{i2}(k-1|k-1) \end{bmatrix} \quad i = L+1, \dots, 2L \end{aligned} \quad (\text{A4.13})$$

where λ is a scaling parameter and $\left(\sqrt{(L+\lambda)\mathbf{P}_a(k-1|k-1)} \right)_i$ is the i -th column of the related matrix.

The sigma points should be propagated through the state functions,

$$\boldsymbol{\chi}_i(k|k-1) = \mathbf{f}(\boldsymbol{\chi}_{i1}(k-1|k-1), \mathbf{u}(k)) + \boldsymbol{\chi}_{i2}(k-1|k-1) \quad i = 0, \dots, 2L \quad (\text{A4.14})$$

The weighted sigma points are used to construct the predicted state and covariance,

$$\hat{\mathbf{x}}(k|k-1) = \sum_{i=0}^{2L} W_i^s \boldsymbol{\chi}_i(k|k-1) \quad (\text{A4.15})$$

$$\mathbf{P}(k|k-1) = \sum_{i=0}^{2L} W_i^c [\boldsymbol{\chi}_i(k|k-1) - \hat{\mathbf{x}}(k|k-1)][\boldsymbol{\chi}_i(k|k-1) - \hat{\mathbf{x}}(k|k-1)]^T \quad (\text{A4.16})$$

where the weights can be found as,

$$\begin{aligned} W_0^s &= \frac{\lambda}{L+\lambda} \\ W_0^c &= \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) \end{aligned}$$

$$W_i^s = W_i^c = \frac{1}{2(L + \lambda)}$$

and the scaling parameter can be calculated as $\lambda = \alpha^2(L + \kappa) - L$. Typical values of α , β and κ are 0.001, 2 and 0 respectively [130].

Update

Again, the predicted state and covariance are augmented but with the mean and covariance of the measurement noise, $\mathbf{v}(k)$.

$$\mathbf{x}_a(k | k - 1) = \begin{bmatrix} \hat{\mathbf{x}}(k | k - 1) \\ E[\mathbf{v}(k)] \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{a1}(k | k - 1) \\ \mathbf{x}_{a2}(k | k - 1) \end{bmatrix} \quad (\text{A4.17})$$

$$\mathbf{P}_a(k | k - 1) = \begin{bmatrix} \mathbf{P}(k | k - 1) & \mathbf{0} \\ \mathbf{0} & \mathbf{R}(k) \end{bmatrix} \quad (\text{A4.18})$$

Similarly, the new set of the sigma points is calculated as follows,

$$\chi_0(k | k - 1) = \mathbf{x}_a(k | k - 1) = \begin{bmatrix} \chi_{01}(k | k - 1) \\ \chi_{02}(k | k - 1) \end{bmatrix} \quad (\text{A4.19})$$

$$\begin{aligned} \chi_i(k | k - 1) &= \mathbf{x}_a(k | k - 1) + \left(\sqrt{(L + \lambda)\mathbf{P}_a(k | k - 1)} \right)_i \\ &= \begin{bmatrix} \chi_{i1}(k | k - 1) \\ \chi_{i2}(k | k - 1) \end{bmatrix} \quad i = 1, \dots, L \end{aligned} \quad (\text{A4.20})$$

$$\begin{aligned} \chi_i(k | k - 1) &= \mathbf{x}_a(k | k - 1) - \left(\sqrt{(L + \lambda)\mathbf{P}_a(k | k - 1)} \right)_{i-L} \\ &= \begin{bmatrix} \chi_{i1}(k | k - 1) \\ \chi_{i2}(k | k - 1) \end{bmatrix} \quad i = L + 1, \dots, 2L \end{aligned} \quad (\text{A4.21})$$

The sigma points are sent through the output functions

$$\gamma_i(k) = \mathbf{h}(\chi_{i1}(k | k - 1)) + \chi_{i2}(k | k - 1) \quad i = 0, \dots, 2L \quad (\text{A4.22})$$

The weighted sigma points are used to construct the predicted measurement and its covariance.

$$\hat{\mathbf{y}}(k) = \sum_{i=0}^{2L} W_i^s \gamma_i(k) \quad (\text{A4.23})$$

$$\mathbf{P}_{zz}(k) = \sum_{i=0}^{2L} W_i^c [\gamma_i(k) - \hat{\mathbf{y}}(k)] [\gamma_i(k) - \hat{\mathbf{y}}(k)]^T \quad (\text{A4.24})$$

The covariance between the state and measurement is as follows,

$$\mathbf{P}_{xz}(k) = \sum_{i=0}^{2L} W_i^c [\boldsymbol{\chi}_i(k|k-1) - \hat{\mathbf{x}}(k|k-1)] [\boldsymbol{\gamma}_i(k) - \hat{\mathbf{y}}(k)]^T \quad (\text{A4.25})$$

Now, the unscented Kalman gain can be obtained,

$$\mathbf{L}(k) = \mathbf{P}_{xz}(k) \mathbf{P}_{zz}^{-1}(k) \quad (\text{A4.26})$$

The state and covariance are updated using the following expressions,

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{L}(k) [\mathbf{y}(k) - \hat{\mathbf{y}}(k)] \quad (\text{A4.27})$$

$$\mathbf{P}(k|k) = \mathbf{P}(k|k-1) - \mathbf{L}(k) \mathbf{P}_{zz}(k) \mathbf{L}^T(k) \quad (\text{A4.28})$$

Appendix 5 Source Code of the Programmes

The source codes of all the programmes developed in this research are presented in this part. First, the programmes related to simulations are presented followed by the programmes used for practical implementation.

A5.1 Simulation Programmes

The following function, **TRMSpara.m**, contains the TRMS parameters and other initial values. These data are used for the Newtonian models and for controller design. The data for the Lagrangian model can be found using the following data; for the sake of brevity, the related m-file is not presented here.

```
function [lt, lm, Jv, g, A, B, C, E, D, F, H, Jtr, Rah, Lah, kthp, kthn, Btr, Jmr, ...
    Rav, Lav, ktvp, ktvn, Bmr, av0, kfhp, kfhn, kfvp, kfvn, koh, kov, kchp, ...
    kchn, km, kt, kg, k1, k2, Ts, Nc, Np, TC, kcfh, FRhi, FRho, kcfv, FRvi, FRvo, ...
    Cfi, Cfo, ICThi, ICTho, ICTvi, ICTvo]=TRMSpara;
% ===== TRMS parameters =====
lt = 0.282;          % Length of tail beam part (m)
lm = 0.246;          % Length of main beam part (m)
lb = 0.29;           % Length of counterweigh beam (m)
lcb = 0.276;         % Distance between counterweigh and joint (m)
rms = 0.155;         % Radius of main shield (m)
rts = 0.10;          % Radius of tail shield (m)
mtr = 0.2213;        % Mass of tail DC motor (kg)
mmr = 0.2357;        % Mass of main DC motor (kg)
mcb = 0.0688;        % Mass of counterweigh (kg)
mt = 0.0155;         % Mass of tail beam (kg)
mm = 0.0145;         % Mass of main beam (kg)
mb = 0.022;          % Mass of counterweigh beam (kg)
mts = 0.1193;        % Mass of tail shield (kg)
mms = 0.2187;        % Mass of main shield (kg)
% ===== Moment of inertia in vertical plane (kg.m^2) =====
Jv1 = mtr*lt^2;
Jv2 = mcb*lcb^2;
Jv3 = mmr*lm^2;
Jv4 = (mt*lt^2)/3;
Jv5 = (mm*lm^2)/3;
Jv6 = (mb*lb^2)/3;
Jv7 = mms/2*rms^2+mms*lm^2;
Jv8 = mts*rts^2+mts*lt^2;
Jv = Jv1+Jv2+Jv3+Jv4+Jv5+Jv6+Jv7+Jv8;
% ===== Jh=D*(cos(av))^2+E*(sin(av))^2+F =====
D = (mm/3+mmr+mms)*lm^2+(mt/3+mtr+mts)*lt^2;
E = mb/3*lb^2+mcb*lcb^2;
F = mms*rms^2+mts/2*rts^2;
% ===== Mv1=g((A-B)*cos(av)-C*sin(av)) =====
g = 9.81;
A = (mt/2+mtr+mts)*lt;
```

```

B = (mm/2+mmr+mms)*lm;
C = mb/2*lb+mcb*lcb;
% ===== Mv3=-OMEGAh^2*H*sin(av)*cos(av) =====
H = A*lt+B*lm-mb/2*lb^2-mcb*lcb^2;
% ===== Tail rotor parameters =====
Jtr = 3.1432e-5; % Moment of inertia of tail motor/load (kg.m^2)
Rah = 8; % Armature resistance (ohm)
Lah = 0.86e-3; % Armature inductance (H)
kthp = 0.5e-7; % TLh=kthp*sign(wh)*wh^2
kthn = 0.42e-7; % TLh=kthn*sign(wh)*wh^2
Btr = 2.3e-5; % Viscous friction coefficient of tail rotor
% ===== Main rotor parameters =====
Jmr = 2.1624e-4; % Moment of inertia of main motor/load (Kg.m^2)
Rav = 8; % Armature resistance (ohm)
Lav = 0.86e-3; % Armature inductance (H)
ktvp = 5.6e-7; % TLv=ktvp*sign(wv)*wv^2
ktvn = 5.1e-7; % TLv=ktvn*sign(wv)*wv^2
Bmr = 4.5e-5; % Viscous friction coefficient of main rotor
% ===== Propelling force coefficients =====
kfhp = 1.8377e-6; % Fh(wh)=kfhp*sign(wh)*wh^2
kfhn = 2.2040e-6; % Fh(wh)=kfhn*sign(wh)*wh^2
kfvp = 1.5691e-5; % Fv(wv)=kfvp*sign(wv)*wv^2
kfvn = 0.82682e-5; % Fv(wv)=kfvn*sign(wv)*wv^2
% ===== Friction coefficients =====
kvfh = 0.0047; % Horizontal viscous friction coefficient
kcfh = 3.96e-5; % Horizontal Coulomb friction coefficient
FRhi = [-1 -0.0001 -0.00005 0 0.00005 0.0001 1];
FRho = [-kvfh-kcfh -0.0001*kvfh-kcfh -0.00012*kvfh-kcfh 0 ...
0.00012*kvfh+kcfh 0.0001*kvfh+kcfh kvfh+kcfh];
kvfv = 0.00131; % Vertical viscous friction coefficient
kcfv = 1.5e-4; % Vertical Coulomb friction coefficient
FRvi = [-2 -0.0007 -0.0005 0 0.0005 0.0007 2];
FRvo = [-2*kvfv-kcfv -0.0007*kvfv-kcfv -0.0008*kvfv-kcfv 0 ...
0.0008*kvfv+kcfv 0.0007*kvfv+kcfv 2*kvfv+kcfv];
% ===== Coefficients of the horizontal cable force =====
kchp = 0.00854;
kchn = 0.00854;
CFi = [-4.2 -1 0 1.5];
CFo = [-3.5*kchn -kchn 0 1.5*kchp];
% ===== Mutual effect coefficients =====
km = 0.0002; % Effect of main rotor on the horizontal angle
kt = 0.000026; % Effect of tail rotor on the vertical angle
kg = 0.2; % Gyroscopic force coefficient
TC = 0.0202; % Torque constant (kav*phiv or kah*phih)
av0 = atan2((A-B),C); % Initial vertical angle
% ===== Interface circuit tables =====
% Horizontal
ICThi = [-2.5 -2.4 -2.3 -2.2 -2.1 -2 -1.9 -1.8 -1.7 -1.6 -1.5 ...
-1.4 -1.3 -1.2 -1.1 -1 -0.9 -0.8 -0.7 -0.6 -0.5 -0.4 -0.3 ...
-0.2 -0.1 -0.08 0 0.03 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 ...
1 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2 2.1 2.2 2.3 2.4 2.5];
ICTho = [-15.381 -14.844 -14.291 -13.744 -13.194 -12.648 -12.098 ...
-11.544 -10.986 -10.429 -9.871 -9.318 -8.754 -8.188 -7.616 ...
-7.055 -6.482 -5.9 -5.309 -4.706 -4.086 -3.443 -2.722 -1.848 ...
-0.789 0 0 1.271 2.24 3.021 3.695 4.331 4.935 5.531 6.115 ...
6.695 7.266 7.835 8.398 8.962 9.521 10.08 10.628 11.184 ...
11.737 12.287 12.84 13.396 13.944 14.493 15.041 15.595];
% Vertical
ICTvi = [-2.5 -2.4 -2.3 -2.2 -2.1 -2 -1.9 -1.8 -1.7 -1.6 -1.5 ...
-1.4 -1.3 -1.2 -1.1 -1 -0.9 -0.8 -0.7 -0.6 -0.5 -0.4 -0.3 ...
-0.2 -0.1 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 1.1 1.2 ...

```

```

1.3 1.4 1.5 1.6 1.7 1.8 1.9 2 2.1 2.2 2.3 2.4 2.5];
ICTvo = [-20.44 -20.02 -19.316 -18.525 -17.742 -16.977 -16.197 ...
-15.427 -14.656 -13.873 -13.093 -12.33 -11.554 -10.776 ...
-9.998 -9.205 -8.416 -7.625 -6.825 -6.019 -5.198 -4.35 ...
-3.46 -2.46 -1.13 0.35 1.8 3 3.8 4.8 5.621 6.425 7.222 ...
8.013 8.801 9.594 10.355 11.162 11.932 12.707 13.491 ...
14.275 15.064 15.874 16.689 17.5 18.324 19.064 19.69 ...
19.9 20.13];
k1 = 8.5; % Linearised vertical interface gain
k2 = 6.5; % Linearised horizontal interface gain
% ===== Controller parameters =====
Ts = 0.2; % Sample time
Np = 20; % Prediction horizon
Nc = 15; % Control horizon

```

PSD is calculated by the following m-file, **PSD.m**, using the MATLAB command *pwelch*. Note that before running the following m-file one needs to have access to the real and modelled data of the yaw and pitch angles in the MATLAB Command Window. The m-file assumes the number of data for the yaw and pitch angles in the case of the real TRMS are the same; also the number of data of both the angles of the model should be the same.

```

% PSD.m is used to calculate the Power Spectral Density (PSD)
% of the real and simulated TRMS using the so-called Welch's
% method. It receives the extracted data from the real TRMS
% and the developed model (ahm, ahr, avm and avr) and returns
% their PSD.
% Written By A. Rahideh on 08/05/2006 at QMUL.
Ts = 0.2; % Sample time
Fs = 1/Ts; % Sampling frequency
Tss = 0.001; % Data sampling rate
L = min(length(ahm),length(ahr)); % Number of data
ahm = ahm(1:Ts/Tss:L); % Yaw angle of model
ahr = ahr(1:Ts/Tss:L); % Yaw angle of real TRMS
[X1,F1] = pwelch(ahm,[],[],[],Fs); % PSD of model
[X2,F2] = pwelch(ahr,[],[],[],Fs); % PSD of real TRMS
avm = avm(1:Ts/Tss:L); % Pitch angle of model
avr = avr(1:Ts/Tss:L); % Pitch angle of real TRMS
[X3,F3] = pwelch(avm,[],[],[],Fs); % PSD of model
[X4,F4] = pwelch(avr,[],[],[],Fs); % PSD of real TRMS
figure(1)
semilogy(F2,X2,'k','linewidth',1);
hold on
semilogy(F1,X1,'b','linewidth',1);
xlabel('Frequency (Hz)');
ylabel('Magnitude');
legend('PSD of real TRMS','PSD of the model')
figure(2)
semilogy(F4,X4,'k','linewidth',1);
hold on
semilogy(F3,X3,'b','linewidth',1);
xlabel('Frequency (Hz)');

```

```

ylabel('Magnitude');
legend('PSD of real TRMS','PSD of the model')
MSEh = sum((ahr-ahm).^2)/L;           % Mean squared error of ah
MSEv = sum((avr-avm).^2)/L;           % Mean squared error of av

```

The codes related to the NN modelling are written in two separate files; the first one is a function, **NNDATA2D**, which forms and prepares the data for the training and test stages from several separate experimental data files, and the second file, **NN2D.m** deals with the training methods. Note that in order to be able to run these files one needs to have 25 .mat files named DATA1.mat to DATA25.mat. The first file is recalled from the second file which means to run the programme the second file only needs to be run. For the sake of brevity, only the codes for the 2DOF NN-based modelling is presented and for the case of the 1DOF vertical and horizontal the following programme needs to be slightly modified.

```

function [I,T,IT,TT]=NNDATA2D
% This function forms the data for training and test stages of a
% neural network from 25 sets of .mat data, i.e. DATA1 to DATA25.
% Each set of data (e.g. DATA1) contains a matrix 'Yout' with 500
% rows and 5 columns which are pitch angle, yaw angle, time,
% main motor voltage and tail motor voltage respectively.
% I :   input matrix for training the NN
% T :   target output matrix for training the NN
% IT:   input matrix for test the NN
% TT:   target output matrix for test the NN
% Written by A. Rahideh on 25-07-2006 at QMUL.
OI=2;           % Order (lag space) of input for NN
OO=3;           % Order (lag space) of output for NN
Ts=0.2;        % Sample time of the network
Tss=0.001;     % Sample time of real TRMS
NDS=22;        % Number of data sets for training
NDST=3;        % Number of data sets for test
% ===== Form the Train Data =====
clear I T
L=0;
for j=1:NDS
    load(['DATA' int2str(j)]);
    N=length(Yout);
    AV=Yout(1:Ts/Tss:N,1); % AlphaV: Pitch angle
    AH=Yout(1:Ts/Tss:N,2); % AlphaH: Yaw angle
    VV=Yout(1:Ts/Tss:N,4); % Vertical rotor voltage
    VH=Yout(1:Ts/Tss:N,5); % Horizontal rotor voltage
    % ==== Constructing the inputs of NN for training stage ====
    n=length(VV);
    I(L+1:L+n,1)=[zeros(1,1);VV(1:n-1)];
    for i=1:OI
        I(L+1:L+n,i+1)=[zeros(i+1,1);VV(1:n-i-1)];
    end
    for i=1:OO
        I(L+1:L+n,i+OI+1)=[zeros(i,1);AV(1:n-i)];
    end
end

```

```

end
I(L+1:L+n,OI+OO+2)=[zeros(1,1);VH(1:n-1)];
for i=1:OI
    I(L+1:L+n,i+OI+OO+2)=[zeros(i+1,1);VH(1:n-i-1)];
end
for i=1:OO
    I(L+1:L+n,i+2*OI+OO+2)=[zeros(i,1);AH(1:n-i)];
end
% ===== Constructing the outputs of NN for training stage =====
T(L+1:L+n,1)=AV;
T(L+1:L+n,2)=AH;
L=length(T);
end
% ===== Form the test Data =====
clear IT TT
L=0;
for j=NDS+1:NDS+NDST
    load(['DATA' int2str(j)]);
    N=length(Yout);
    AV=Yout(1:Ts/Tss:N,1); % AlphaV: Pitch angle
    AH=Yout(1:Ts/Tss:N,2); % Alpha H: Yaw angle
    VV=Yout(1:Ts/Tss:N,4); % Vertical rotor voltage
    VH=Yout(1:Ts/Tss:N,5); % Horizontal rotor voltage
    % ===== Constructing the inputs of NN for test stage =====
    n=length(VV);
    IT(L+1:L+n,1)=[zeros(1,1);VV(1:n-1)];
    for i=1:OI
        IT(L+1:L+n,i+1)=[zeros(i+1,1);VV(1:n-i-1)];
    end
    for i=1:OO
        IT(L+1:L+n,i+OI+1)=[zeros(i,1);AV(1:n-i)];
    end
    IT(L+1:L+n,OI+OO+2)=[zeros(1,1);VH(1:n-1)];
    for i=1:OI
        IT(L+1:L+n,i+OI+OO+2)=[zeros(i+1,1);VH(1:n-i-1)];
    end
    for i=1:OO
        IT(L+1:L+n,i+2*OI+OO+2)=[zeros(i,1);AH(1:n-i)];
    end
    % ===== Constructing the outputs of NN for test stage =====
    TT(L+1:L+n,1)=AV;
    TT(L+1:L+n,2)=AH;
    L=length(TT);
end
end

```

As mentioned earlier the above function is recalled in the following m-file to train the NN-based on the selected training approach.

```

% This programme models the TRMS in terms of its 2DOF using
% an MLP neural network and 13 training methods
% Written by A. Rahideh on 2-08-2006 at QMUL.
disp('
                                The List of Training Methods')
disp('=====')
disp('= 1 Levenberg-Marquardt
                                =')

```

```

disp('= 2 Gradient descent with momentum and adaptive rate      =')
disp('= 3 BFGS quasi-Newton                                       =')
disp('= 4 Bayesian regularisation                                   =')
disp('= 5 Conjugate gradient with Powell-Beale restarts           =')
disp('= 6 Conjugate gradient with Fletcher-Reeves                   =')
disp('= 7 Conjugate gradient with Polak-Ribiere updates            =')
disp('= 8 Gradient descent                                           =')
disp('= 9 Gradient descent with adaptive learning rate              =')
disp('= 10 Gradient descent with momentum                           =')
disp('= 11 One step secant                                           =')
disp('= 12 Resilient                                                 =')
disp('= 13 Scaled conjugate gradient                                 =')
disp('=====')
tota=input('Choose a training algorithms by its code (1 to 13):');
% ===== Load data of the TRMS =====
[IN,TARGET,INT,TARGETT]=NNDATA2D;
minIN=min(IN)';
maxIN=max(IN)';
AF={'logsig' 'logsig' 'purelin'};
% ===== Type of Training Algorithm Selection =====
switch tota
case 1
    disp('Levenberg-Marquardt selected')
    net = newff([minIN maxIN],[3 5 2],AF,'trainlm');
    net.trainParam.epochs = 500;
    net.trainParam.goal = 0;
    net.trainParam.show = 100;
    net.trainParam.min_grad = 1e-12;
    net.trainParam.max_fail=5;
    net.trainParam.mem_reduc=1;
    net.trainParam.mu=0.0001;
    net.trainParam.mu_dec=0.3;
    net.trainParam.mu_inc=3;
    net.trainParam.mu_max=1e+10;
    net.trainParam.time=Inf;
case 2
    disp('Gradient descent with momentum and adap. rate selected')
    net = newff([minIN maxIN],[3 5 2],AF,'traingdx');
    net.trainParam.epochs = 500;
    net.trainParam.goal = 0;
    net.trainParam.lr = .001;
    net.trainParam.lr_dec = 0.7;
    net.trainParam.lr_inc = 1.1;
    net.trainParam.max_fail = 5;
    net.trainParam.max_perf_inc = 1.04;
    net.trainParam.mc = 0.9;
    net.trainParam.min_grad = 1.0e-6;
    net.trainParam.show = 100;
    net.trainParam.time = inf;
case 3
    disp('BFGS quasi-Newton selected')
    net = newff([minIN maxIN],[3 5 2],AF,'trainbfg');
    net.trainParam.epochs = 500;
    net.trainParam.show = 100;
    net.trainParam.goal = 0;
    net.trainParam.time=Inf;
    net.trainParam.min_grad = 1e-12 ;
    net.trainParam.max_fail=5;
    net.trainParam.searchFcn='srchbac';
    net.trainParam.scale_tol=20;
    net.trainParam.alpha=0.001;

```

```

net.trainParam.beta=0.1;
net.trainParam.delta=0.01;
net.trainParam.gama=0.1;
net.trainParam.low_lim=0.1;
net.trainParam.up_lim=0.5;
net.trainParam.maxstep=100;
net.trainParam.minstep= 1.0000e-006;
net.trainParam.bmax=26;
net.trainParam.mem_reduc=1;
net.trainParam.mu=0.0001;
net.trainParam.mu_dec=0.3;
net.trainParam.mu_inc=3;
net.trainParam.mu_max=1e+10;
case 4
disp('Bayesian regularisation selected')
net = newff([minIN maxIN],[3 5 2],AF,'trainbr');
net.trainParam.epochs = 500;
net.trainParam.show = 100;
net.trainParam.goal = 0;
net.trainParam.time=Inf;
net.trainParam.min_grad = 1e-12;
net.trainParam.max_fail=5;
net.trainParam.mem_reduc=1;
net.trainParam.mu=0.0001;
net.trainParam.mu_dec=0.3;
net.trainParam.mu_inc=3;
net.trainParam.mu_max=1e+10;
case 5
disp('Conjugate gradient with Powell-Beale restarts selected')
net = newff([minIN maxIN],[3 5 2],AF,'traincgb');
net.trainParam.epochs = 500;
net.trainParam.show = 100;
net.trainParam.goal = 0;
net.trainParam.time=Inf;
net.trainParam.min_grad = 1e-12;
net.trainParam.max_fail=5;
net.trainParam.searchFcn='srchcha';
net.trainParam.scale_tol=20;
net.trainParam.alpha=0.001;
net.trainParam.beta=0.1;
net.trainParam.delta=0.01;
net.trainParam.gama=0.1;
net.trainParam.low_lim=0.1;
net.trainParam.up_lim=0.5;
net.trainParam.maxstep=100;
net.trainParam.minstep=1e-6;
net.trainParam.bmax=26;
net.trainParam.mem_reduc=1;
net.trainParam.mu=0.0001;
net.trainParam.mu_dec=0.3;
net.trainParam.mu_inc=3;
net.trainParam.mu_max=1e+10;
case 6
disp('Conjugate gradient with Fletcher-Reeves selected')
net = newff([minIN maxIN],[3 5 2],AF,'traincgf');
net.trainParam.epochs = 500;
net.trainParam.show = 100;
net.trainParam.goal = 0;
net.trainParam.time=Inf;
net.trainParam.min_grad = 1e-12 ;
net.trainParam.max_fail=5;

```

```

net.trainParam.searchFcn='srchcha';
net.trainParam.scale_tol=20;
net.trainParam.alpha=0.001;
net.trainParam.beta=0.1;
net.trainParam.delta=0.01;
net.trainParam.gama=0.1;
net.trainParam.low_lim=0.1;
net.trainParam.up_lim=0.5;
net.trainParam.maxstep=100;
net.trainParam.minstep=1e-6;
net.trainParam.bmax=26;
case 7
disp('Conjugate gradient with Polak-Ribiere updates selected')
net = newff([minIN maxIN],[3 5 2],AF,'traingcp');
net.trainParam.epochs = 500;
net.trainParam.show = 100;
net.trainParam.goal = 0;
net.trainParam.time=Inf;
net.trainParam.min_grad = 1e-12;
net.trainParam.max_fail=5;
net.trainParam.searchFcn='srchcha';
net.trainParam.scale_tol=20;
net.trainParam.alpha=0.001;
net.trainParam.beta=0.1;
net.trainParam.delta=0.01;
net.trainParam.gama=0.1;
net.trainParam.low_lim=0.1;
net.trainParam.up_lim=0.5;
net.trainParam.maxstep=100;
net.trainParam.minstep=1e-6;
net.trainParam.bmax=26;
case 8
disp('Gradient descent selected')
net = newff([minIN maxIN],[3 5 2],AF,'traingd');
net.trainParam.epochs = 500;
net.trainParam.show = 100;
net.trainParam.goal = 0;
net.trainParam.lr = 0.1;
net.trainParam.max_fail = 5;
net.trainParam.min_grad = 1e-15;
net.trainParam.time=Inf;
case 9
disp('Gradient descent with adaptive rate selected')
net = newff([minIN maxIN],[3 5 2],AF,'traingda');
net.trainParam.epochs = 500;
net.trainParam.goal = 0;
net.trainParam.lr = 0.001;
net.trainParam.lr_dec = 0.7;
net.trainParam.lr_inc = 1.1;
net.trainParam.max_fail = 5;
net.trainParam.max_perf_inc = 1.04;
net.trainParam.min_grad = 1e-6;
net.trainParam.show = 100;
net.trainParam.time = inf;
case 10
disp('Gradient descent with momentum selected')
net = newff([minIN maxIN],[3 5 2],AF,'traingdm');
net.trainParam.epochs = 500;
net.trainParam.show = 100;
net.trainParam.goal = 0;
net.trainParam.lr = 0.4;

```



```

net.trainParam.mu=0.0001;
net.trainParam.max_fail = 5;
net.trainParam.min_grad = 1e-12;
net.trainParam.time=Inf;
case 11
disp('One step secant selected')
net = newff([minIN maxIN],[3 5 2],AF,'trainoss');
net.trainParam.epochs = 500;
net.trainParam.show = 100;
net.trainParam.goal = 0;
net.trainParam.time=Inf;
net.trainParam.min_grad = 1e-12;
net.trainParam.max_fail=5;
net.trainParam.searchFcn='srchbac';
net.trainParam.scale_tol=20;
net.trainParam.alpha=0.001;
net.trainParam.beta=0.1;
net.trainParam.delta=0.01;
net.trainParam.gama=0.1;
net.trainParam.low_lim=0.1;
net.trainParam.up_lim=0.5;
net.trainParam.maxstep=100;
net.trainParam.minstep= 1e-6;
net.trainParam.bmax=26;
case 12
disp('Resilient selected')
net = newff([minIN maxIN],[3 5 2],AF,'trainrp');
net.trainParam.epochs = 500;
net.trainParam.show = 100;
net.trainParam.goal = 0;
net.trainParam.time=Inf;
net.trainParam.min_grad=1e-6;
net.trainParam.max_fail=5;
net.trainParam.delt_inc=1.2;
net.trainParam.delt_dec=0.5;
net.trainParam.delta0=0.07;
net.trainParam.deltamax=50;
case 13
disp('Scaled conjugate gradient selected')
net = newff([minIN maxIN],[3 5 2],AF,'trainscg');
net.trainParam.epochs = 500;
net.trainParam.show = 100;
net.trainParam.goal = 0;
net.trainParam.time=Inf;
net.trainParam.min_grad=1e-6;
net.trainParam.max_fail=5;
net.trainParam.sigma=5e-5;
net.trainParam.lambda=5e-7;
otherwise
disp('Levenberg-Marquardt')
net = newff([minIN maxIN],[3 5 2],AF,'trainlm');
net.trainParam.epochs = 500;
net.trainParam.goal = 0;
net.trainParam.show = 100;
net.trainParam.min_grad = 1e-12;
net.trainParam.max_fail=5;
net.trainParam.mem_reduc=1;
net.trainParam.mu=0.0001;
net.trainParam.mu_dec=0.3;
net.trainParam.mu_inc=3;
net.trainParam.mu_max=1e+10;

```

```

net.trainParam.time=Inf;
end
% ===== Train the Network =====
net = train(net,IN',TARGET');
% ===== Plot the training results =====
OUTNN = sim(net,IN')';
figure(1)
plot(OUTNN(:,1),'b')
hold on
plot(TARGET(:,1),'k:')
title('Training')
ylabel('Pitch angle (rad)')
xlabel('sample')
legend('NN-based Model','Real TRMS')
figure(2)
plot(OUTNN(:,2),'b')
hold on
plot(TARGET(:,2),'k:')
title('Training')
ylabel('Yaw angle (rad)')
xlabel('sample')
legend('NN-based Model','Real TRMS')
% ===== Plot the test result =====
OUTNNT = sim(net,INT')';
t=linspace(0,length(OUTNNT)*0.2,length(OUTNNT));
figure(3)
plot(t,OUTNNT(:,1),'b','linewidth',1)
hold on
plot(t,TARGETT(:,1),'k','linewidth',1)
title('Test')
ylabel('Pitch angle (rad)')
xlabel('time (s)')
legend('NN-based Model','Real TRMS')
figure(4)
plot(t,OUTNNT(:,2),'b','linewidth',1)
hold on
plot(t,TARGETT(:,2),'k','linewidth',1)
title('Test')
ylabel('Yaw angle (rad)')
xlabel('time (s)')
legend('NN-based Model','Real TRMS')
W11=net.IW{1,1}; % Weight matrix between input and 1st hidden layer
W21=net.LW{2,1}; % Weight matrix between 1st and 2nd hidden layers
W32=net.LW{3,2}; % Weight matrix between 2nd hidden layer and output
b1=net.b{1}; % Bias vector of 1st hidden layer
b2=net.b{2}; % Bias vector of 2nd hidden layer
b3=net.b{3}; % Bias vector of output layer
MSEL=(sum((TARGET(:,1)-OUTNN(:,1)).^2))/length(TARGET);
fprintf('The pitch mean squared error of training is: %e\n',MSEL)
MSET=(sum((TARGETT(:,1)-OUTNNT(:,1)).^2))/length(TARGETT);
fprintf('The pitch mean squared error of test is: %e\n',MSET)
MSEL=(sum((TARGET(:,2)-OUTNN(:,2)).^2))/length(TARGET);
fprintf('The yaw mean squared error of training is: %e\n',MSEL)
MSET=(sum((TARGETT(:,2)-OUTNNT(:,2)).^2))/length(TARGETT);
fprintf('The yaw mean squared error of test is: %e\n',MSET)

```

The model predictive controller consists of three functions; the first one is the main body of the programme, **Main.m**, in which two other functions, **TRMSlin.m** and **MyMPC.m** are recalled.

```

function [UX]=Main(IN)
[lt, lm, Jv, g, A, B, C, E, D, F, H, Jtr, Rah, Lah, kthp, kthn, Btr, Jmr, ...
  Rav, Lav, ktvp, ktvn, Bmr, av0, kfhp, kfhn, kfvp, kfvn, kvfh, ...
  kvfv, kchp, kchn, km, kt, kg, k1, k2, Ts, Nc, Np, TC, kcfh, FRhi, ...
  FRho, kcfv, FRvi, FRvo, CFi, CFo, ICThi, ICTho, ICTvi, ICTvo]=TRMSpara;
nx=6;
nu=2;
ny=2;
Q=[.05 0;0 1]; % Weight factor for error
R=[0.0125 0;0 0.0125]; % Weight factor for control effort
Cu=[-2.5 2.5;-2.5 2.5]; % Input plant constraints
Cdu=[-2 2;-2 2]; % Input change constraints
Cy=[-1.5 1.5;-1.5 1.5]; % Output plant constraints
al=[0 0;0 0]; % Coefficients of reference trajectories
ref=IN(1:2); % Reference signals: [ahref;avref]
Up=IN(3:4); % Input signals: [Uh;Uv]
Xp=IN(5:10); % State variables: [wh,Sh,ah,wv,Sv,av]
Ym=IN(11:12); % Model outputs: [ahm,avm]
d=(Xp([3 6])-Ym); % Output disturbance model
dx=zeros(6,1); % State disturbance
dU0=[IN(13:13+2*Nc-3);0;0]; % Previous input change
tsim=IN(11+2*Nc);
% ==== Discrete linearise state-space model parameters ====
AA=zeros(nx*Np,nx);
BB=zeros(nx*Np,nu);
CC=zeros(ny*Np,nx);
Upf=Up;
Xpf=Xp;
for i=1:Np
  [Ad,Bd,Cd,Dd]=TRMSlin(Xp,Up);
  AA(nx*(i-1)+1:nx*i,:)=Ad;
  BB(nx*(i-1)+1:nx*i,:)=Bd;
  CC(ny*(i-1)+1:ny*i,:)=Cd;
  Uhp=Up(1);
  Uvp=Up(2);
  % ==== Update the state variables at each stage =====
  whp=Xp(1);
  Shp=Xp(2);
  ahp=Xp(3);
  wvp=Xp(4);
  Svp=Xp(5);
  avp=Xp(6);
  if whp>=0
    whn=whp+Ts*(TC*(k2*Uhp-TC*whp)/Rah-Btr*whp-kthp*whp^2)/Jtr;
  else
    whn=whp+Ts*(TC*(k2*Uhp-TC*whp)/Rah-Btr*whp+kthn*whp^2)/Jtr;
  end
  if wvp>=0
    wvn=wvp+Ts*(TC*(k1*Uvp-TC*wvp)/Rav-Bmr*wvp-ktvp*wvp^2)/Jmr;
  else
    wvn=wvp+Ts*(TC*(k1*Uvp-TC*wvp)/Rav-Bmr*wvp+ktvn*wvp^2)/Jmr;
  end
  if whp>=0

```

```

Shn=Shp+Ts*(lt*kfhp*whp^2*cos(avp+av0)-kvfh*Shp-...
kchp*ahp)/(D*(cos(avp+av0))^2+E*(sin(avp+av0))^2+F)-...
Ts*kvfh*km*wvp*cos(avp+av0)/(D*(cos(avp+av0))^2+E*...
(sin(avp+av0))^2+F)^2;
else
Shn=Shp+Ts*(-lt*kfhn*whp^2*cos(avp+av0)-kvfh*Shp-...
kchp*ahp)/(D*(cos(avp+av0))^2+E*(sin(avp+av0))^2+F)-...
Ts*kvfh*km*wvp*cos(avp+av0)/(D*(cos(avp+av0))^2+E*...
(sin(avp+av0))^2+F)^2;
end
if wvp>=0
Svn=Svp+Ts*(lm*kfvp*wvp^2-kvfv*Svp-kvfv*kt*whp/Jv+...
g*((A-B)*cos(avp+av0)-C*sin(avp+av0)))/Jv;
else
Svn=Svp+Ts*(-lm*kfvn*wvp^2-kvfv*Svp-kvfv*kt*whp/Jv+...
g*((A-B)*cos(avp+av0)-C*sin(avp+av0)))/Jv;
end
ahn=ahp+Ts*Shp+Ts*km*wvp*cos(avp+av0)/...
(D*(cos(avp+av0))^2+E*(sin(avp+av0))^2+F);
avn=avp+Ts*Svp+Ts*kt*whp/Jv;
Xp=[whn;Shn;ahn;wvn;Svn;avn];
% ==== Update the inputs at each stage =====
if i<=Nc
Up=Up+dU0((i-1)*nu+1:i*nu,1);
end
end
Cz=Cd(1:2,:);
% ==== In the case of TRMS Bd and Cd are constant ====
[U,X,dUp]=MyMPC(AA,BB,Cz,Dd,Np,Nc,Q,R,ref,Cu,...
Cdu,Cy,Xpf,Upf,d,al,dx,dU0,tsim);
UX=[U;X;dUp];

```

The **TRMSlin.m** is the function for adaptive linearisation of the nonlinear model,

```

function [As,Bs,Cs,Ds]=TRMSlin(Xp,Up)
whp=Xp(1);
Shp=Xp(2);
ahp=Xp(3);
wvp=Xp(4);
Svp=Xp(5);
avp=Xp(6);
Uhp=Up(1);
Uvp=Up(2);
[lt,lm,Jv,g,A,B,C,E,D,F,H,Jtr,Rah,Lah,kthp,kthn,Btr,Jmr,...
Rav,Lav,ktvp,ktvn,Bmr,av0,kfhp,kfhn,kfvp,kfvn,kvfh,...
kvfv,kchp,kchn,km,kt,kg,k1,k2,Ts,Nc,Np,TC,kcfh,FRhi,...
FRho,kcfv,FRvi,FRvo,CFi,CFo,ICThi,ICTho,ICTvi,ICTvo]=TRMSpara;
% =====
%
% | wh(k+1) | | a11 a12 a13 a14 a15 a16 | | wh(k) | | b11 b12 |
% | Sh(k+1) | | a21 a22 a23 a24 a25 a26 | | Sh(k) | | b21 b22 |
% | ah(k+1) | | a31 a32 a33 a34 a35 a36 | | ah(k) | | b31 b32 | | Uh |
% | wv(k+1) | = | a41 a42 a43 a44 a45 a46 | | wv(k) | + | b41 b42 | | Uv |
% | Sv(k+1) | | a51 a52 a53 a54 a55 a56 | | Sv(k) | | b51 b52 | - -
% | av(k+1) | | a61 a62 a63 a64 a65 a66 | | av(k) | | b61 b62 |

```

```

% - - - - -
%
%
%
% | wh(k) |
% | Sh(k) |
% | ah(k) |
% | ah(k+1) | = | c11 c12 c13 c14 c15 c16 | | wv(k) | + | d11 d12 | | Uh |
% | av(k+1) | | c21 c22 c23 c24 c25 c26 | | Sv(k) | | d21 d22 | | Uv |
% - - - - - | av(k) | - - - - -
%
% =====
As=zeros(6,6);
Bs=zeros(6,2);
Cs=zeros(2,6);
Ds=zeros(2,2);
dx=zeros(6,1);
DEN=D*(cos(avp+av0))^2+E*(sin(avp+av0))^2+F;
if whp>=0
    As(1,1)=1-Ts*(Btr+kthp*whp+TC^2/Rah)/Jtr;
    As(2,1)=Ts*lt*kfhp*whp*cos(avp+av0)/DEN;
else
    As(1,1)=1-Ts*(Btr-kthn*whp+TC^2/Rah)/Jtr;
    As(2,1)=-Ts*lt*kfhn*whp*cos(avp+av0)/DEN;
end
As(2,2)=1-Ts*kvfh/DEN;
As(2,3)=-Ts*kchp/DEN;
As(2,4)=-Ts*kvfh*km*cos(avp+av0)/DEN^2;
As(3,2)=Ts;
As(3,3)=1;
As(3,4)=Ts*km*cos(avp+av0)/DEN;
if wvp>=0
    As(4,4)=1-Ts*(Bmr+ktvp*wvp+TC^2/Rav)/Jmr;
    As(5,4)=Ts*lm*kfvwvp/Jv;
else
    As(4,4)=1-Ts*(Bmr-ktvn*wvp+TC^2/Rav)/Jmr;
    As(5,4)=-Ts*lm*kfvn*wvp/Jv;
end
As(5,1)=-Ts*kvfv*kt/(Jv^2);
As(5,5)=1-Ts*kvfv/Jv;
if avp==0
    As(5,6)=Ts*g*((B-A)*sin(avp+av0)-C*cos(avp+av0))/Jv;
else
    As(5,6)=Ts*g*((A-B)*cos(avp+av0)-C*sin(avp+av0))/(Jv*avp);
end
As(6,1)=Ts*kt/Jv;
As(6,5)=Ts;
As(6,6)=1;
Bs(1,1)=Ts*TC*k2/(Jtr*Rah);
Bs(4,2)=Ts*TC*k1/(Jmr*Rav);
Cs(1,3)=1;
Cs(2,6)=1;

```

The formulation and optimisation of the MPC is carried out in a separate function entitled **MyMPC.m**. Note in the following function the MATLAB function **quadprog.m** is employed to solve the quadratic optimisation problem.

```

function [U,X,dUp]=MyMPC(AA,BB,C,D,Np,Nc,Q,R,ref,Cu,...
    Cdu,Cy,Xp,Up,d,al,dx,dU0,tsim)
% MyMPC solve a linear model predictive control according
% to a state-space model.
% [U,X,dUp]=MyMPC(AA,BB,C,D,Np,Nc,Q,R,ref,Cu,...
% Cdu,Cy,Xp,Up,d,al,dx,dU0,tsim)
% Inputs:
%     AA, BB, C, D: The state-space model parameters:
%                   X(k+1) = AA X(k) + BB U(k)
%                   Y(k)   = C X(k) + D U(k)
%     Np : Prediction horizon
%     Nc : Control horizon
%     Q  : Output weight matrix with dim. (ny,ny) & diagonal
%     R  : Input weight matrix with dim. (nu,nu) & diagonal
%     r  : Setpoint signal with dimension of (ny,1)
%     Cu : Input range constraint:
%         [min1 max1 ; min2 max2 ; ... ; min_nu max_nu]
%     Cdu : Change input range constraint:
%         [min1 max1 ; min2 max2 ; ... ; min_nu max_nu]
%     Cy : Output range constraint:
%         [min1 max1 ; min2 max2 ; ... ; min_ny max_ny]
%     Xp : Previous state variables X(k)
%     Up : Previous inputs U(k-1)
%     d  : Disturbances
%     al : vector of reference trajectories coefficients
%         w(k+i|k)=al*w(k+i-1|k)+(1-al)*r    w(k|k)=y(k)
% Outputs:
%     U  : Optimum input signal that enters to the process
%     X  : Current state variables
%     dUp : optimal change inputs during the control horizon
% Written By A. Rahideh on 5/09/2007 at QMUL
% =====
nx=size(AA,2);           % Number of state variables
nu=size(BB,2);           % Number of inputs
ny=size(C,1);            % Number of outputs
Ma=zeros(Np*nx,nx);
Mb=zeros(Np*nx,nu);
Mu=zeros(Np*nx,Nc*nu);
Mc=zeros(Np*ny,Np*nx);
Mref=zeros(Np*ny,1);
MQ=zeros(Np*ny,Np*ny);
MR=zeros(Nc*nu,Nc*nu);
Md=zeros(Np*ny,1);
Mdx=zeros(Np*nx,1);
Ymax=zeros(Np*ny,1);
Ymin=zeros(Np*ny,1);
Adu=zeros(2*Nc*nu,Nc*nu);
bdu=zeros(2*Nc*nu,1);
Au=zeros(2*Nc*nu,Nc*nu);
bu=zeros(2*Nc*nu,1);
for i=1:Np
    if i==1
        Ma(nx*(i-1)+1:nx*i , 1:nx)=AA(nx*(i-1)+1:nx*i , :);
        Mb(nx*(i-1)+1:nx*i , 1:nu)=BB(nx*(i-1)+1:nx*i , :);
        Mu(nx*(i-1)+1:nx*i , 1:nu)=BB(nx*(i-1)+1:nx*i , :);
        Mref(ny*(i-1)+1:ny*i , 1)=Xp([3 6])*0+(eye(ny)-al)*ref;
    else
        Ma(nx*(i-1)+1:nx*i , 1:nx)=AA(nx*(i-1)+1:nx*i , :)*...
            Ma(nx*(i-2)+1:nx*(i-1) , 1:nx);
        % === Forming Mba for the ith row of matrix Mb ===
    end
end

```

```

clear Mba
Mba=zeros(nx*(i-1),nx);
for j=1:i-1
    if j==1
        Mba(nx*(j-1)+1:nx*j,1:nx)=AA(nx*(i-j)+1:nx*...
            (i-j+1) , :);
    else
        Mba(nx*(j-1)+1:nx*j,1:nx)=Mba(nx*(j-2)+1:nx*...
            (j-1),1:nx)*AA(nx*(i-j)+1:nx*(i-j+1) , :);
    end
end
clear Mbi
Mbi(1:nx,1:nu)=BB(nx*(i-1)+1:nx*i , : );
for j=1:i-1
    Mbi(nx*j+1:nx*(j+1),1:nu)=Mbi(nx*(j-1)+1:nx*j,1:nu)+...
        Mba(nx*(j-1)+1:nx*j,1:nx)*BB(nx*(i-j-1)+1:nx*...
            (i-j) , :);
end
% === Forming the elements of Mb ===
Mb(nx*(i-1)+1:nx*i , 1:nu)=Mbi(nx*(i-1)+1:nx*i,1:nu);
% === Forming the elements of Mu ===
if i<=Nc
    for k=1:i
        Mu(nx*(i-1)+1:nx*i , nu*(k-1)+1:nu*k)=...
            Mbi(nx*(i-k)+1:nx*(i-k+1),1:nu);
    end
else
    for k=1:Nc
        Mu(nx*(i-1)+1:nx*i , nu*(k-1)+1:nu*k)=...
            Mbi(nx*(i-k)+1:nx*(i-k+1),1:nu);
    end
end
Mref(ny*(i-1)+1:ny*i , 1)=al*...
    Mref(ny*(i-2)+1:ny*(i-1) , 1)+(eye(ny)-al)*ref;
end
Mc(ny*(i-1)+1:ny*i , nx*(i-1)+1:nx*i)=C;
MQ(ny*(i-1)+1:ny*i , ny*(i-1)+1:ny*i)=Q;
Md(ny*(i-1)+1:ny*i , 1)=d;
Mdx(nx*(i-1)+1:nx*i , 1)=dx;
Ymax(ny*(i-1)+1:ny*i , 1)=Cy(:,2);
Ymin(ny*(i-1)+1:ny*i , 1)=Cy(:,1);
end
for i=1:Nc
    MR(nu*(i-1)+1:nu*i , nu*(i-1)+1:nu*i)=R;
    Adu(nu*(2*i-2)+1:nu*(2*i-1) , nu*(i-1)+1:nu*i)=eye(nu,nu);
    Adu(nu*(2*i-1)+1:nu*(2*i) , nu*(i-1)+1:nu*i)=-eye(nu,nu);
    bdu(nu*(2*i-2)+1:nu*(2*i-1) , 1)=Cdu(:,2);
    bdu(nu*(2*i-1)+1:nu*(2*i) , 1)=-Cdu(:,1);
    for j=1:i
        Au(nu*(2*i-2)+1:nu*(2*i-1) , nu*(j-1)+1:nu*j)=eye(nu,nu);
        Au(nu*(2*i-1)+1:nu*(2*i) , nu*(j-1)+1:nu*j)=-eye(nu,nu);
    end
    bu(nu*(2*i-2)+1:nu*(2*i-1) , 1)=Cu(:,2)-Up;
    bu(nu*(2*i-1)+1:nu*(2*i) , 1)=-Cu(:,1)+Up;
end
% Calculating the parameters of Quadratic Programming (H,G,c)
%      min    U'*H*U/2 + U'*G + c
%      subject to    At*U <= b & Aeq*U=beq
E=Mc*(Ma*Xp+Mb*Up+Mdx)+Md-Mref;
Mdu=Mc*Mu;
c=E'*MQ*E;

```

```

G=2*Mdu'*MQ*E;
H=2*(Mdu'*MQ*Mdu+MR);
% Arranging the constraint parameters (At, b)
Ay=[Mdu;-Mdu];
by=[Ymax-Mc*(Ma*Xp+Mb*Up);-Ymin+Mc*(Ma*Xp+Mb*Up)];
At=[Adu;Au;Ay];
b=[bdu;bu;by];
% ===== Equality constraints for stability =====
[lt, lm, Jv, gi, Ai, Bi, Ci, Ei, Di, Fi, Hi, Jtr, Rah, Lah, kthp, kthn, Btr, Jmr, ...
Rav, Lav, ktvp, ktvn, Bmr, av0, kfhp, kfhn, kfvp, kfvn, kvfh, ...
kvfv, kchp, kchn, km, kt, kg, k1, k2, Ts, Nc, Np, TC, kcfh, FRhi, ...
FRho, kcfv, FRvi, FRvo, CFi, CFo, ICThi, ICTho, ICTvi, ICTvo]=TRMSpara;
if ref(1)>=0
    FhNp=kchp*ref(1)/(lt*cos(ref(2)+av0));
    whNp=sqrt(abs(FhNp/kfhp));
else
    FhNp=kchn*ref(1)/(lt*cos(ref(2)+av0));
    whNp=-sqrt(abs(FhNp/kfhn));
end
SvNp=-kt*whNp/Jv;
FvNp=-gi*((Ai-Bi)*cos(ref(2)+av0)-Ci*sin(ref(2)+av0))/lm;
if FvNp>=0
    wvNp=sqrt(abs(FvNp/kfvp));
else
    wvNp=-sqrt(abs(FvNp/kfvn));
end
ShNp=-km*wvNp*cos(ref(2)+av0)/(Ei*(sin(ref(2)+av0))^2+...
Di*(cos(ref(2)+av0))^2+Fi);
xNp=[whNp;ShNp;ref(1);wvNp;SvNp;ref(2)];
Aeq=Mu(nx*(Np-1)+1:nx*Np,:);
beq=xNp-Ma(nx*(Np-1)+1:nx*Np,:)*Xp-Mb(nx*(Np-1)+1:nx*Np,:)*Up;
options = optimset('LargeScale','off');
[dU,FVAL,EXITFLAG] = quadprog(H,G,At,b,Aeq,beq,[],[],dU0,options);
Uint(1:nu,1)=Up;
Xint(1:nx,1)=Xp;
for i=1:Np
    if i<=Nc
        Uint(nu*i+1:nu*(i+1),1)=Uint(nu*(i-1)+1:nu*i,1)+...
            dU(nu*(i-1)+1:nu*i,1);
    else
        Uint(nu*i+1:nu*(i+1),1)=Uint(nu*(i-1)+1:nu*i,1);
    end
    Xint(nx*i+1:nx*(i+1),1)=AA(nx*(i-1)+1:nx*i,1:nx)*...
        Xint(nx*(i-1)+1:nx*i,1)+BB(nx*(i-1)+1:nx*i,1:nu)*...
        Uint(nu*i+1:nu*(i+1),1);
    ahint(i)=Xint(nx*i+3,1);
    avint(i)=Xint(nx*i+6,1);
end
U=dU(1:nu)+Up;
dUp=dU(nu+1:nu*Nc);
X=AA(1:nx,:)*Xp+BB(1:nx,:)*U;

```

The EKF based on the state-dependent state-space model is developed according to the following function.


```

function [OUT]=MyKalman(IN)
Uhp=IN(1); % Plant input (Manipulated variable)
Uvp=IN(2); % Plant input (Manipulated variable)
ahp=IN(3); % Plant output (controlled variable)
avp=IN(4); % Plant output (controlled variable)
whp=IN(5); % previous estimated state
Shp=IN(6); % previous estimated state
wvp=IN(8); % previous estimated state
Svp=IN(9); % previous estimated state
nx=6; % Number of state variables
nu=2; % Number of inputs
ny=2; % Number of outputs
Pkp=IN(11:10+nx*nx); % Vector form of P(k-1|k-1)
% === Change the vector form to the matrix form ===
for i=1:nx
    Pkp(:,i)=Pkpv([1:nx]+(i-1)*nx,1);
end
Q=eye(nx,nx)*0.01; % Matrix covariance of process noise
R=eye(ny,ny)*0.01; % Matrix covariance of measurement noise
[lt,lm,Jv,g,A,B,C,E,D,F,H,Jtr,Rah,Lah,kthp,kthn,Btr,Jmr,...
Rav,Lav,ktvp,ktvn,Bmr,av0,kfhp,kfhn,kfvp,kfvn,kvfh,...
kvfv,kchp,kchn,km,kt,kg,k1,k2,Ts,Nc,Np,TC,kcfh,FRhi,...
FRho,kcfv,FRvi,FRvo,CFi,CFo,ICThi,ICTho,ICTvi,ICTvo]=TRMSpara;
% ===== Nonlinear discrete state-space equations of the TRMS =====
if whp>=0
    whn=whp+Ts*(TC*(k2*Uhp-TC*whp)/Rah-Btr*whp-kthp*whp^2)/Jtr;
    Shn=Shp+Ts*(lt*kfhp*whp^2*cos(avp+av0)-kvfh*Shp-kchp*...
    ahp)/(D*(cos(avp+av0))^2+E*(sin(avp+av0))^2+F)-...
    Ts*kvfh*km*wvp*cos(avp+av0)/(D*(cos(avp+av0))^2+...
    E*(sin(avp+av0))^2+F)^2;
else
    whn=whp+Ts*(TC*(k2*Uhp-TC*whp)/Rah-Btr*whp+kthn*whp^2)/Jtr;
    Shn=Shp+Ts*(-lt*kfhn*whp^2*cos(avp+av0)-kvfh*Shp-kchp*...
    ahp)/(D*(cos(avp+av0))^2+E*(sin(avp+av0))^2+F)-...
    Ts*kvfh*km*wvp*cos(avp+av0)/(D*(cos(avp+av0))^2+...
    E*(sin(avp+av0))^2+F)^2;
end
if wvp>=0
    wvn=wvp+Ts*(TC*(k1*Uvp-TC*wvp)/Rav-Bmr*wvp-ktvp*wvp^2)/Jmr;
    Svn=Svp+Ts*(lm*kfvp*wvp^2-kvfv*Svp-kvfv*kt*whp/Jv+...
    g*((A-B)*cos(avp+av0)-C*sin(avp+av0)))/Jv;
else
    wvn=wvp+Ts*(TC*(k1*Uvp-TC*wvp)/Rav-Bmr*wvp+ktvn*wvp^2)/Jmr;
    Svn=Svp+Ts*(-lm*kfvn*wvp^2-kvfv*Svp-kvfv*kt*whp/Jv+...
    g*((A-B)*cos(avp+av0)-C*sin(avp+av0)))/Jv;
end
ahn=ahp+Ts*Shp+Ts*km*wvp*cos(avp+av0)/...
(D*(cos(avp+av0))^2+E*(sin(avp+av0))^2+F);
avn=avp+Ts*Svp+Ts*kt*whp/Jv;
% ===== Linearise state-space equations =====
As=zeros(nx,nx);
Bs=zeros(nx,nu);
Cs=zeros(ny,nx);
DEN=D*(cos(avp+av0))^2+E*(sin(avp+av0))^2+F;
if whp>=0
    As(1,1)=1-Ts*(Btr+kthp*whp+TC^2/Rah)/Jtr;
    As(2,1)=Ts*lt*kfhp*whp*cos(avp+av0)/DEN;
else
    As(1,1)=1-Ts*(Btr-kthn*whp+TC^2/Rah)/Jtr;
    As(2,1)=-Ts*lt*kfhn*whp*cos(avp+av0)/DEN;

```

```

end
As(2,2)=1-Ts*kvfh/DEN;
As(2,3)=-Ts*kchp/DEN;
As(2,4)=-Ts*kvfh*km*cos(avp+av0)/DEN^2;
As(3,2)=Ts;
As(3,3)=1;
As(3,4)=Ts*km*cos(avp+av0)/DEN;
if wvp>=0
    As(4,4)=1-Ts*(Bmr+ktvp*wvp+TC^2/Rav)/Jmr;
    As(5,4)=Ts*lm*kfvp*wvp/Jv;
else
    As(4,4)=1-Ts*(Bmr-ktvn*wvp+TC^2/Rav)/Jmr;
    As(5,4)=-Ts*lm*kfvn*wvp/Jv;
end
As(5,1)=-Ts*kvfv*kt/(Jv^2);
As(5,5)=1-Ts*kvfv/Jv;
if avp==0
    As(5,6)=Ts*g*((B-A)*sin(avp+av0)-C*cos(avp+av0))/Jv;
else
    As(5,6)=Ts*g*((A-B)*cos(avp+av0)-C*sin(avp+av0))/(Jv*avp);
end
As(6,1)=Ts*kt/Jv;
As(6,5)=Ts;
As(6,6)=1;
Bs(1,1)=Ts*TC*k2/(Jtr*Rah);
Bs(4,2)=Ts*TC*k1/(Jmr*Rav);
Cs(1,3)=1;
Cs(2,6)=1;
% ===== Predict stage of extended Kalman filter =====
Xk=[whn;Shn;ahn;wvn;Svn;avn];
Pk=As*Pkp*As'+Q;
% ===== Update stage of EKF =====
Lk=Pk*Cs'*inv(Cs*Pk*Cs'+R);
Xkn=Xk+Lk*([ahp;avp]-Cs*Xk);
Pkn=(eye(nx,nx)-Lk*Cs)*Pk;
% === Change the matrix Pkn into a vector form (Pknv) ===
Pknv=zeros(nx*nx,1);
for i=1:nx
    Pknv([1:nx]+(i-1)*nx,1)=Pkn(:,i);
end
OUT=[Xkn;Pknv];

```

The robust MPC source codes are as follows. As mentioned earlier the LMI-based optimisation problem has been solved using the LMI toolbox of MATLAB.

```

function [OUT]=RMPC(xk)
% This m-file takes the current state vector and finds
% the optimum plant input on the basis of robust model
% predictive control using linear matrix inequalities.
% By: A. Rahideh 10-12-2007
% xk=[wh;Sh;ah;wv;Sv;av]
[V,AA,B,C,ymax]=LinearModels(xk);
[Vo,A]=vertices(V,AA);
umax=[2.5 0;0 2.5];
L=size(A,3);

```

```

Q1=eye(6,6)*0.00004;
Q1(3,3)=30;
Q1(6,6)=60;
Q1s=sqrt(Q1);
R=[1 0;0 1]*0.0005;
Rs=sqrt(R);
% Initialise LMIs
setlmis([]);
% LMI variables
[Gamma,nG,sG]=lmivar(1,[1 0]);
[Q,nQ,sQ]=lmivar(1,[6 1]);
[Y,nY,sY]=lmivar(2,[2 6]);
[X,nX,sX]=lmivar(1,[2 1]);
% Z=lmivar(1,[2 1]);
[Xd,nXd,sXd]=lmivar(3,[sX(1,1) 0;0 sX(2,2)]);
% 1st LMI
lmiterm([-1 1 1 0],1);
lmiterm([-1 2 1 0],xk);
lmiterm([-1 2 2 Q],1,1);
% 2nd to (L+1)-th LMIs
for i=2:L+1
    lmiterm([-i 1 1 Q],1,1);
    lmiterm([-i 2 1 Q],A(:, :, i-1),1);
    lmiterm([-i 2 1 Y],B,1);
    lmiterm([-i 3 1 Q],Q1s,1);
    lmiterm([-i 4 1 Y],Rs,1);
    lmiterm([-i 2 2 Q],1,1);
    lmiterm([-i 3 3 Gamma],1,eye(6));
    lmiterm([-i 4 4 Gamma],1,eye(2));
end
lmiterm([- (L+2) 1 1 Q],1,1);
% input Constraints LMIs
lmiterm([- (L+3) 1 1 X],1,1);
lmiterm([- (L+3) 1 2 Y],1,1);
lmiterm([- (L+3) 2 2 Q],1,1);
lmiterm([(L+4) 1 1 Xd],1,1);
lmiterm([(L+4) 1 1 0],-(umax.^2));
% output Constraints LMIs
for i=L+5:2*L+4
    CA=C*A(:, :, i-(L+4));
    CB=C*B;
    lmiterm([-i 1 1 Q],1,1);
    lmiterm([-i 2 1 Q],CA,1);
    lmiterm([-i 2 1 Y],C*B,1);
    lmiterm([-i 2 2 0],ymax.^2);
end
% =====
LMIsys=getlmis;
n=decnbr(LMIsys);
c=zeros(n,1);
c(1,1)=1;
options=[1e-4,0,0,0,1];
[copt,xopt]=mincx(LMIsys,c,options);
if isempty(xopt)==1
    OUT=[0;0]
else
    Gopt=dec2mat(LMIsys,xopt,1)
    Qopt=dec2mat(LMIsys,xopt,2);
    Yopt=dec2mat(LMIsys,xopt,3);
    F=Yopt/Qopt;
    OUT=F*xk;

```

```
end
OUT=[OUT;Gopt];
```

```
function [Vvertix,AAvertix]=vertices(V,AA)
warning off
% This program receives several linear models and finds
% the vertices of the corresponding polytope.
% V is a Q*N matrix in which each column contains the
% variable entries of matrix A relating to a single
% linear model. Q is the number of variable entries
% of matrix A and N is the number of linear models.
% Vvertix is the selected linear models relating to
% vertices only.
% By: A. Rahideh 28-11-2007
[Q,N]=size(V);
N0=N;
i0=1;
for i=1:N
    if i0==1
        A=[V(:, [i0+1:N0]);ones(1,N0-1)];
    else
        A=[V(:, [1:i0-1,i0+1:N0]);ones(1,N0-1)];
    end
    b=[V(:,i0);1];
    X=lsqnonneg(A,b);
    e=A*X-b;
    tol=1e-4;
    if sum(abs(X))<1+tol & sum(abs(X))>1-tol & sum(abs(e))<tol
        N0=N0-1;
        V(:,i0)=[];
        AA(:, :, i0)=[];
    else
        i0=i0+1;
    end
end
Vvertix=V;
AAvertix=AA;
```

A5.2 Experimental Programmes

As mentioned before, the experimental programmes have been written using *Embedded-MATLAB-Functions* and then converted to *S-Functions* to make us able to use them in Real-Time Workshop of MATLAB version 6. The following files are the UKF and MPC programmes respectively.

```
function OUTU =UKF(INU)
Uhp=INU(1); % Plant input (Manipulated variable)
Uvp=INU(2); % Plant input (Manipulated variable)
ah=INU(3); % Plant output (controlled variable)
av=INU(4); % Plant output (controlled variable)
ahp=INU(5); % Previous Plant output
```

```

avp=INU(6); % Previous Plant output
whp=INU(7); % previous estimated state
Shp=INU(8); % previous estimated state
wvp=INU(10); % previous estimated state
Svp=INU(11); % previous estimated state
nx=6; % Number of state variables
nu=2; % Number of inputs
ny=2; % Number of outputs
Pkp=INU(13:12+nx*nx); % Vector form of P(k-1|k-1)
Pkp=zeros(nx,nx); % Matrix form of P(k-1|k-1)
for i=1:nx
    Pkp(:,i)=Pkpv([1:nx]+(i-1)*nx,1);
end
% ===== TRMS parameters =====
lts=2.82e-1; % Length of tail beam part
lm=2.46e-1; % Length of main beam part
Jv=5.26e-2; % Moment of inertia about horizontal axis
D=5.52877e-2;
E=5.85764e-3;
F=5.85077e-3;
g=9.81; % Gravitational acceleration
A=9.82347e-2;
B=1.135659e-1;
C=2.21788e-2;
H=4.94734e-2;
Jtr=3.1432e-5; % Moment of inertia for tail motor/load
Rah=8; % Armature resistance
Lah=8.6e-4; % Armature inductance
kthp=5e-8; % Th=kthp*sign(wh)*(wh)^2
kthn=4.41e-8; % Th=kthn*sign(wh)*(wh)^2
Btr=2.3e-5; % Viscous friction coefficient of tail rotor
Jmr=2.0098e-4; % Moment of inertia for main motor/load
Rav=8; % Armature resistance
Lav=8.6e-4; % Armature inductance
ktvp=5.6e-7; % Th=ktvp*sign(wv)*(wv)^2
ktvn=5.1e-7; % Th=ktvn*sign(wv)*(wv)^2
Bmr=2.97e-5; % Viscous friction coefficient of main rotor
av0=-6.048e-1; % Initial pitch angle
kfhp=2.1377e-6; % Fh(wh)=kfhp*sign(wh)*(wh)^2
kfhn=1.9056e-6; % Fh(wh)=kfhn*sign(wh)*(wh)^2
kfvp=1.9506e-5; % Fv(wv)=kfvp*sign(wv)*(wv)^2
kfvn=1.1012e-5; % Fv(wv)=kfvn*sign(wv)*(wv)^2
kvfh=4.91e-3; % Horizontal viscous friction
kvfv=5.48e-3; % Vertical viscous friction
kchp=5.60e-3; % Cable force coefficient
km=0.00023; % Effect of main rotor on horizontal angle
kt=0.000026; % Effect of tail rotor on vertical angle
TC=0.0202; % Torque constant
k1=8.5; % Interface vertical gain
k2=6.5; % Interface horizontal gain
Ts=0.2; % Sample time
% ===== Mean and covariance =====
Ew=[1 0.1 0.032 1 0.1 0.032]'; % Vector mean of Process noise
Ev=ones(ny,1)*0.032; % Vector mean of measurement noise
Q=eye(nx,nx)*0.4; % Matrix Covariance of Process noise
R=eye(ny,ny)*0.4; % Matrix Covariance of measurement noise
% ===== UKF parameters =====
al=0.001; % Alpha parameter of UKF
bet=2; % Beta parameter of UKF
kai=0; % Kai parameter of UKF
lam=al^2*(2*nx+kai)-2*nx; % Lambda parameter of UKF

```

```

% ===== Augmented state and covariance =====
Xkp=[whp;Shp;ahp;wvp;Svp;avp]; % x(k-1|k-1)
Xkpa=zeros(2*nx,1); % xa(k-1|k-1)
Xkpa(1:nx,1)=Xkp;
Xkpa(nx+1:2*nx)=Ew;
Pkp=zeros(2*nx,2*nx); % Pa(k-1|k-1)
Pkpa(1:nx,1:nx)=Pkp;
Pkpa(1+nx:2*nx,1+nx:2*nx)=Q;
% ===== Form the 4*nx+1 sigma points =====
Ksip=zeros(2*nx,4*nx+1); % Xi(k-1|k-1)
Ksip(:,1)=Xkpa;
% ===== Find the square roots of the matrix Pkpa by =====
% ===== diagonalisation using eigenvalues and eigenvectors =====
VPkp=zeros(nx,nx);
DPkp=zeros(nx,nx);
[VPkp,DPkp]=myeig(Pkp);
DPkpsqrt=sqrt(abs(DPkp));
YPkp=VPkp*DPkpsqrt*inv(VPkp);
YPkpa=[YPkp zeros(nx,nx);zeros(nx,nx) sqrt(Q)];
% =====
Pkpasqr=sqrt(2*nx+lam)*YPkpa;
for i=1:2*nx
    Ksip(:,i+1)=Xkpa+Pkpasqr(:,i);
    Ksip(:,i+2*nx+1)=Xkpa-Pkpasqr(:,i);
end
% ===== Sigma points Propagation through nonlinear =====
% ===== discrete state-space equations of the TRMS =====
Ksi=zeros(nx,4*nx+1);
for i=1:4*nx+1
    whp=Ksip(1,i);
    Shp=Ksip(2,i);
    ahp=Ksip(3,i);
    wvp=Ksip(4,i);
    Svp=Ksip(5,i);
    avp=Ksip(6,i);
    if whp>=0
        whn=whp+Ts*(TC*(k2*Uhp-TC*whp)/Rah-Btr*whp-kthp*whp^2)/Jtr;
        Shn=Shp+Ts*(lts*kfhp*whp^2*cos(avp+av0)-kvfh*Shp-kchp*...
            ahp)/(D*(cos(avp+av0))^2+E*(sin(avp+av0))^2+F)-...
            Ts*kvfh*km*wvp*cos(avp+av0)/(D*(cos(avp+av0))^2+...
            E*(sin(avp+av0))^2+F)^2;
    else
        whn=whp+Ts*(TC*(k2*Uhp-TC*whp)/Rah-Btr*whp+kthn*whp^2)/Jtr;
        Shn=Shp+Ts*(-lts*kfhn*whp^2*cos(avp+av0)-kvfh*Shp-kchp*...
            ahp)/(D*(cos(avp+av0))^2+E*(sin(avp+av0))^2+F)-...
            Ts*kvfh*km*wvp*cos(avp+av0)/(D*(cos(avp+av0))^2+...
            E*(sin(avp+av0))^2+F)^2;
    end
    if wvp>=0
        wvn=wvp+Ts*(TC*(k1*Uvp-TC*wvp)/Rav-Bmr*wvp-ktvp*wvp^2)/Jmr;
        Svn=Svp+Ts*(lm*kfvp*wvp^2-kvfv*Svp-kvfv*kt*whp/Jv+...
            g*((A-B)*cos(avp+av0)-C*sin(avp+av0)))/Jv;
    else
        wvn=wvp+Ts*(TC*(k1*Uvp-TC*wvp)/Rav-Bmr*wvp+ktvn*wvp^2)/Jmr;
        Svn=Svp+Ts*(-lm*kfvn*wvp^2-kvfv*Svp-kvfv*kt*whp/Jv+...
            g*((A-B)*cos(avp+av0)-C*sin(avp+av0)))/Jv;
    end
    ahn=ahp+Ts*Shp+Ts*km*wvp*cos(avp+av0)/(D*(cos(avp+av0))^2+...
        E*(sin(avp+av0))^2+F);
    avn=avp+Ts*Svp+Ts*kt*whp/Jv;
    Ksi(:,i)=[whn;Shn;ahn;wvn;Svn;avn]+Ksip(nx+1:2*nx,i); % Xi(k|k-1)
end

```

```

end
% === The weighted sigma points are used to construct ===
% === the predicted state and covariance =====
Ws0=lam/(2*nx+lam); % Weight of predicted state
Wc0=lam/(2*nx+lam)+(1-al^2+bet); % Weight of predicted covariance
Wi=1/(2*(2*nx+lam)); % Weights of both
Xk=Ws0*Ksi(:,1)+Wi*sum(Ksi(:,2:4*nx+1),2); % x(k|k-1)
Pk=zeros(nx,nx);
for i=1:4*nx
    Pk=Pk+Wi*((Ksi(:,i+1)-Xk)*(Ksi(:,i+1)-Xk)');
end
Pk=Pk+Wc0*(Ksi(:,1)-Xk)*(Ksi(:,1)-Xk)';
% ===== Update stage of EKF =====
lam=al^2*(nx+ny+kai)-nx-ny; % Lambda parameter of UKF
% ===== Augmented state and covariance =====
Xka=zeros(nx+ny,1); % xa(k|k-1)
Xka(1:nx,1)=Xk;
Xka(nx+1:nx+ny)=Ev;
Pka=zeros(nx+ny,nx+ny); % Pa(k|k-1)
Pka(1:nx,1:nx)=Pk;
Pka(1+nx:nx+ny,1+nx:nx+ny)=R;
% ===== Form the 4*nx+1 sigma points =====
Ksiy=zeros(nx+ny,2*(nx+ny)+1); % Xi(k|k-1)
Ksiy(:,1)=Xka;
% ===== Find the square roots of the matrix Pka by =====
% ===== diagonalisation using eigenvalues and eigenvectors =====
VPk=zeros(nx,nx);
DPk=zeros(nx,nx);
[VPk,DPk]=myeig(Pk);
DPksqrt=sqrt(abs(DPk));
YPk=VPk*DPksqrt*inv(VPk);
YPka=[YPk zeros(nx,ny);zeros(ny,nx) sqrt(R)];
% =====
Pkasqr=sqrt(nx+ny+lam)*YPka;
for i=1:(nx+ny)
    Ksiy(:,i+1)=Xka+Pkasqr(:,i);
    Ksiy(:,i+nx+ny+1)=Xka-Pkasqr(:,i);
end
Cs=[0 0 1 0 0 0;0 0 0 0 0 1];
gam=zeros(ny,2*(nx+ny)+1); % Gama(k)
for i=1:2*(nx+ny)+1
    gam(:,i)=Cs*Ksiy(1:nx,i)+Ksiy(nx+1:nx+ny,i);
end
% === The weighted sigma points are used to construct =====
% ===== the predicted state and covariance =====
Ws0=lam/(nx+ny+lam); % Weight of predicted state
Wc0=lam/(nx+ny+lam)+(1-al^2+bet); % Weight of predicted covariance
Wi=1/(2*(nx+ny+lam)); % Weights of both
yh=Ws0*gam(:,1)+Wi*sum(gam(:,2:2*(nx+ny)+1),2); % yhat(k)
Pzz=zeros(ny,ny);
Pxz=zeros(nx,ny);
for i=1:2*(nx+ny)
    Pzz=Pzz+Wi*(gam(:,i+1)-yh)*(gam(:,i+1)-yh)';
    Pxz=Pxz+Wi*(Ksiy(1:nx,i+1)-Xk)*(gam(:,i+1)-yh)';
end
Pzz=Pzz+Wc0*(gam(:,1)-yh)*(gam(:,1)-yh)';
Pxz=Pxz+Wc0*(Ksiy(1:nx,1)-Xk)*(gam(:,1)-yh)';
Lk=Pxz*inv(Pzz);
Xkn=Xk+Lk*([ah;av]-yh);
Pkn=Pk-Lk*Pzz*Lk';
% == Change the diagonal matrix Pkn into a vector form (Pknv) ==

```

```

Pknv=zeros(nx*nx,1);
for i=1:nx
    Pknv([1:nx]+(i-1)*nx,1)=Pkn(:,i);
end
OUTU=[Xkn;Pknv];

function [V,D]=myeig(A)
L=length(A);
Q=zeros(L,L);
V=eye(L,L);
for i=1:L^3
    [Q,R]=QRdec(A);
    A=R*Q;
    V=V*Q;
end
D=zeros(L,L);
for i=1:L
    D(i,i)=A(i,i);
end

function [Q,R]=QRdec(A)
L=length(A);
U=zeros(L,L);
Q=zeros(L,L);
if mynorm(A(:,1))~=0
    Q(:,1)=A(:,1)/mynorm(A(:,1));
end
for i=1:L
    for j=1:i-1
        if mynorm(Q(:,j))~=0
            U(:,i)=U(:,i)-Q(:,j)'*A(:,i)/(Q(:,j)'*Q(:,j))*Q(:,j);
        end
    end
    U(:,i)=U(:,i)+A(:,i);
    if mynorm(U(:,i))~=0
        Q(:,i)=U(:,i)/mynorm(U(:,i));
    end
end
R=Q'*A;

function [Y]=mynorm(X)
Y=sqrt(sum(X.^2));

```

The total source codes of the MPC programme are as follows. As pointed out earlier, the TRMS parameters obtained from the grey-box model have been used in the experimental part of the work. Since, MATLAB functions cannot be recalled inside *Embedded-MATLAB-Functions*, the quadratic optimisation problem solver of MATLAB has been modified to solve the quadratic optimisation problem of the proposed MPC algorithm.

```
function UX =Main(IN)
```



```

% ===== TRMS parameters =====
lts=2.82e-1; % Length of tail beam part
lm=2.46e-1; % Length of main beam part
Jv=5.26e-2; % Moment of inertia about horizontal axis
D=5.52877e-2;
E=5.85764e-3;
F=5.85077e-3;
g=9.81; % Gravitational acceleration
A=9.82347e-2;
B=1.135659e-1;
C=2.21788e-2;
H=4.94734e-2;
Jtr=3.1432e-5; % Moment of inertia for tail motor/load
Rah=8; % Armature resistance
Lah=8.6e-4; % Armature inductance
kthp=5e-8; % Th=kthp*sign(wh)*(wh)^2
kthn=4.41e-8; % Th=kthn*sign(wh)*(wh)^2
Btr=2.3e-5; % Viscous friction coefficient of tail rotor
Jmr=2.0098e-4; % Moment of inertia for main motor/load
Rav=8; % Armature resistance
Lav=8.6e-4; % Armature inductance
ktvp=5.6e-7; % Th=ktvp*sign(wv)*(wv)^2
ktvn=5.1e-7; % Th=ktvn*sign(wv)*(wv)^2
Bmr=2.97e-5; % Viscous friction coefficient of main rotor
av0=-6.048e-1; % Initial pitch angle
kfhp=2.1377e-6; % Fh(wh)=kfhp*sign(wh)*(wh)^2
kfhn=1.9056e-6; % Fh(wh)=kfhn*sign(wh)*(wh)^2
kfvp=1.9506e-5; % Fv(wv)=kfvp*sign(wv)*(wv)^2
kfvn=1.1012e-5; % Fv(wv)=kfvn*sign(wv)*(wv)^2
kvfh=4.91e-3; % Horizontal viscous friction
kvfv=5.47e-3; % Vertical viscous friction
kchp=5.60e-3; % Cable force coefficient
kchn=5.60e-3; % Cable force coefficient
km=0.00023; % Effect of main rotor on horizontal angle
kt=0.000026; % Effect of tail rotor on vertical angle
TC=0.0202; % Torque constant
k1=8.5; % Interface vertical gain
k2=6.5; % Interface horizontal gain
Ts=0.2; % Sample time
Np=20; % Prediction horizon
Nc=15; % Control horizon
nx=6;
nu=2;
ny=2;
Q=[1 0;0 3]; % Weight factor for error
R=[0.2 0;0 0.2]; % Weight factor for control effort
Cu=[-2.5 2.5;-2.5 2.5]; % Input plant constraints
Cdu=[-2.5 2.5;-2.5 2.5]; % Input change constraints
Cy=[-1.5 1.5;-1.5 1.5]; % Output plant constraints
al=[0 0;0 0]; % Coefficients of reference trajectories
ref=IN(1:2); % reference signals : [ahref;avref]
Up=IN(3:4); % Input signals : [Uh;Uv]
Xp=IN(5:10); % State variables : [wh,Sh,ah,wv,Sv,av]
Ym=IN(11:12); % Model outputs : [ahm,avm]
d=(Ym-Xp([3 6])); % Output disturbance model
dx=zeros(6,1); % State disturbance
dU0=[IN(13:13+2*Nc-3);0;0];
Cs=[0 0 1 0 0 0;0 0 0 0 0 1];
% ===== Discrete linearise state-space model parameters =====
AA=zeros(nx*Np,nx);
BB=zeros(nx*Np,nu);

```

```

Upf=Up;
Xpf=Xp;
for i=1:Np
    % ===== TRMSlin Function =====
    whp=Xp(1);
    Shp=Xp(2);
    ahp=Xp(3);
    wvp=Xp(4);
    Svp=Xp(5);
    avp=Xp(6);
    Uhp=Up(1);
    Uvp=Up(2);
    As=zeros(6,6);
    Bs=zeros(6,2);
    Ds=zeros(2,2);
    dx=zeros(6,1);
    DEN=D*(cos(avp+av0))^2+E*(sin(avp+av0))^2+F;
    if whp>=0
        As(1,1)=1-Ts*(Btr+kthp*whp+TC^2/Rah)/Jtr;
        As(2,1)=Ts*lts*kfhp*whp*cos(avp+av0)/DEN;
    else
        As(1,1)=1-Ts*(Btr-kthn*whp+TC^2/Rah)/Jtr;
        As(2,1)=-Ts*lts*kfhn*whp*cos(avp+av0)/DEN;
    end
    As(2,2)=1-Ts*kvfh/DEN;
    if ahp>=0
        As(2,3)=-Ts*kchp/DEN;
    else
        As(2,3)=-Ts*kchn/DEN;
    end
    As(2,4)=-Ts*kvfh*km*cos(avp+av0)/DEN^2;
    As(3,2)=Ts;
    As(3,3)=1;
    As(3,4)=Ts*km*cos(avp+av0)/DEN;
    if wvp>=0
        As(4,4)=1-Ts*(Bmr+ktvp*wvp+TC^2/Rav)/Jmr;
        As(5,4)=Ts*(lm*kfvp*wvp)/Jv;
    else
        As(4,4)=1-Ts*(Bmr-ktvn*wvp+TC^2/Rav)/Jmr;
        As(5,4)=Ts*(-lm*kfvn*wvp)/Jv;
    end
    As(5,1)=-Ts*kvfv*kt/(Jv^2);
    As(5,5)=1-Ts*kvfv/Jv;
    if avp==0
        As(5,6)=Ts*g*((B-A)*sin(avp+av0)-C*cos(avp+av0))/Jv;
    else
        As(5,6)=Ts*g*((A-B)*cos(avp+av0)-C*sin(avp+av0))/(Jv*avp);
    end
    As(6,1)=Ts*kt/Jv;
    As(6,5)=Ts;
    As(6,6)=1;
    Bs(1,1)=Ts*TC*k2/(Jtr*Rah);
    Bs(4,2)=Ts*TC*k1/(Jmr*Rav);
    AA(nx*i+(1-nx:0),:)=As;
    BB(nx*i+(1-nx:0),:)=Bs;
    Uhp=Up(1);
    Uvp=Up(2);
    % ===== Update the state variables at each stage =====
    whp=Xp(1);
    Shp=Xp(2);
    ahp=Xp(3);

```

```

wvp=Xp(4);
Svp=Xp(5);
avp=Xp(6);
if whp>=0
    whn=whp+Ts*(TC*(k2*Uhp-TC*whp)/Rah-Btr*whp-kthp*whp^2)/Jtr;
else
    whn=whp+Ts*(TC*(k2*Uhp-TC*whp)/Rah-Btr*whp+kthn*whp^2)/Jtr;
end
if wvp>=0
    wvn=wvp+Ts*(TC*(k1*Uvp-TC*wvp)/Rav-Bmr*wvp-ktvp*wvp^2)/Jmr;
else
    wvn=wvp+Ts*(TC*(k1*Uvp-TC*wvp)/Rav-Bmr*wvp+ktvn*wvp^2)/Jmr;
end
if whp>=0
    Shn=Shp+Ts*(lts*kfhp*whp^2*cos(avp+av0)-kvfh*Shp-...
        kchp*ahp)/(D*(cos(avp+av0))^2+E*(sin(avp+av0))^2+F)-...
        Ts*kvfh*km*wvp*cos(avp+av0)/(D*(cos(avp+av0))^2+...
        E*(sin(avp+av0))^2+F)^2;
else
    Shn=Shp+Ts*(-lts*kfhn*whp^2*cos(avp+av0)-kvfh*Shp-...
        kchp*ahp)/(D*(cos(avp+av0))^2+E*(sin(avp+av0))^2+F)-...
        Ts*kvfh*km*wvp*cos(avp+av0)/(D*(cos(avp+av0))^2+...
        E*(sin(avp+av0))^2+F)^2;
end
if wvp>=0
    Svn=Svp+Ts*(lm*kfvp*wvp^2-kvfv*Svp-kvfv*kt*whp/Jv+...
        g*((A-B)*cos(avp+av0)-C*sin(avp+av0)))/Jv;
else
    Svn=Svp+Ts*(-lm*kfvn*wvp^2-kvfv*Svp-kvfv*kt*whp/Jv+...
        g*((A-B)*cos(avp+av0)-C*sin(avp+av0)))/Jv;
end
ahn=ahp+Ts*Shp+Ts*km*wvp*cos(avp+av0)/(D*(cos(avp+av0))^2+...
    E*(sin(avp+av0))^2+F);
avn=avp+Ts*Svp+Ts*kt*whp/Jv;
Xp=[whn;Shn;ahn;wvn;Svn;avn];
% ===== Update the inputs at each stage =====
if i<=Nc
    Up=Up+dU0(i*nu+(1-nu:0),1);
end
end
% ===== MyMPC1 Function =====
Up=Upf;
Xp=Xpf;
Ma=zeros(Np*nx,nx);
Mb=zeros(Np*nx,nu);
Mu=zeros(Np*nx,Nc*nu);
Mc=zeros(Np*ny,Np*nx);
Mref=zeros(Np*ny,1);
MQ=zeros(Np*ny,Np*ny);
MR=zeros(Nc*nu,Nc*nu);
Md=zeros(Np*ny,1);
Mdx=zeros(Np*nx,1);
Ymax=zeros(Np*ny,1);
Ymin=zeros(Np*ny,1);
Adu=zeros(2*Nc*nu,Nc*nu);
bdu=zeros(2*Nc*nu,1);
Au=zeros(2*Nc*nu,Nc*nu);
bu=zeros(2*Nc*nu,1);
for i=1:Np
    if i==1
        Ma(nx*i+(1-nx:0),1:nx)=AA(nx*i+(1-nx:0),:);
    end
end

```

```

Mb(nx*i+(1-nx:0) , 1:nu)=BB(nx*i+(1-nx:0) , :);
Mu(nx*i+(1-nx:0) , 1:nu)=BB(nx*i+(1-nx:0) , :);
Mref(ny*i+(1-ny:0) , 1)=Xp([3 6])*0+(eye(ny)-al)*ref;
else
Ma(nx*i+(1-nx:0) , 1:nx)=AA(nx*i+(1-nx:0) , :)*...
    Ma(nx*(i-1)+(1-nx:0) , 1:nx);
% ==== Forming Mba for the i-th row of matrix Mb ====
Mba=zeros(nx*(Np-1),nx);
for j=1:i-1
    if j==1
        Mba(nx*j+(1-nx:0),1:nx)=AA(nx*(i-j)+(1:nx) , :);
    else
        Mba(nx*j+(1-nx:0),1:nx)=Mba(nx*(j-1)+...
            (1-nx:0),1:nx)*AA(nx*(i-j)+(1:nx) , :);
    end
end
Mbi=zeros(nx*Np,nu);
Mbi(1:nx,1:nu)=BB(nx*i+(1-nx:0) , : );
for j=1:i-1
    Mbi(nx*j+(1:nx),1:nu)=Mbi(nx*j+(1-nx:0),1:nu)+...
        Mba(nx*j+(1-nx:0),1:nx)*BB(nx*(i-j)+(1-nx:0) , :);
end
% === Forming the elements of Mb ====
Mb(nx*i+(1-nx:0) , 1:nu)=Mbi(nx*i+(1-nx:0),1:nu);
% === Forming the elements of Mu ====
if i<=Nc
    for k=1:i
        Mu(nx*i+(1-nx:0) , nu*k+(1-nu:0))=...
            Mbi(nx*(i-k)+(1:nx),1:nu);
    end
else
    for k=1:Nc
        Mu(nx*i+(1-nx:0) , nu*k+(1-nu:0))=...
            Mbi(nx*(i-k)+(1:nx),1:nu);
    end
end
Mref(ny*i+(1-ny:0) , 1)=al*Mref(ny*(i-1)+(1-ny:0) , 1)+...
    (eye(ny)-al)*ref;
end
Mc(ny*i+(1-ny:0) , nx*i+(1-nx:0))=Cs;
MQ(ny*i+(1-ny:0) , ny*i+(1-ny:0))=Q;
Md(ny*i+(1-ny:0) , 1)=d;
Mdx(nx*i+(1-nx:0) , 1)=dx;
Ymax(ny*i+(1-ny:0) , 1)=Cy(:,2);
Ymin(ny*i+(1-ny:0) , 1)=Cy(:,1);
end
for i=1:Nc
MR(nu*i+(1-nu:0) , nu*i+(1-nu:0))=R;
Adu(nu*(2*i-1)+(1-nu:0) , nu*i+(1-nu:0))=eye(nu,nu);
Adu(nu*(2*i)+(1-nu:0) , nu*i+(1-nu:0))=-eye(nu,nu);
bdu(nu*(2*i-1)+(1-nu:0) , 1)=Cdu(:,2);
bdu(nu*(2*i)+(1-nu:0) , 1)=-Cdu(:,1);
for j=1:i
    Au(nu*(2*i-1)+(1-nu:0) , nu*j+(1-nu:0))=eye(nu,nu);
    Au(nu*(2*i)+(1-nu:0) , nu*j+(1-nu:0))=-eye(nu,nu);
end
bu(nu*(2*i-1)+(1-nu:0) , 1)=Cu(:,2)-Up;
bu(nu*(2*i)+(1-nu:0) , 1)=-Cu(:,1)+Up;
end
% Calculating the parameters of Quadratic Programming (H,G,c)
%      min    U'*H*U/2 + U'*G + c

```

```

%          subject to   At*U <= b & Aeq*U=beq
Eo=Mc*(Ma*Xp+Mb*Up+Mdx)+Md-Mref;
Mdu=Mc*Mu;
c=Eo'*MQ*Eo;
G=2*Mdu'*MQ*Eo;
Ho=2*(Mdu'*MQ*Mdu+MR);
% ===== Arranging the constraint parameters (At, b) =====
Ay=[Mdu;-Mdu];
by=[Ymax-Mc*(Ma*Xp+Mb*Up);-Ymin+Mc*(Ma*Xp+Mb*Up)];
At=[Adu;Au;Ay];
b=[bdu;bu;by];
% ===== Equality constraints =====
if ref(1)>=0
    FhNp=kchp*ref(1)/(lts*cos(ref(2)+av0));
    whNp=sqrt(abs(FhNp/kfhp));
else
    FhNp=kchn*ref(1)/(lts*cos(ref(2)+av0));
    whNp=-sqrt(abs(FhNp/kfhn));
end
SvNp=-kt*whNp/Jv;
FvNp=-g*((A-B)*cos(ref(2)+av0)-C*sin(ref(2)+av0))/lm;
if FvNp>=0
    wvNp=sqrt(abs(FvNp/kfvp));
else
    wvNp=-sqrt(abs(FvNp/kfvn));
end
ShNp=-km*wvNp*cos(ref(2)+av0)/(E*(sin(ref(2)+av0))^2+...
    D*(cos(ref(2)+av0))^2+F);
xNp=[whNp;ShNp;ref(1);wvNp;SvNp;ref(2)];
Aeq=Mu(nx*(Np-1)+1:nx*Np,:);
beq=xNp-Ma(nx*(Np-1)+1:nx*Np,:)*Xp-Mb(nx*(Np-1)+1:nx*Np,:)*Up;
% ===== Solve the Quadratic Program using active set method =====
[dU] = MyQP(Ho,G,At,b);
U=dU(1:nu)+Up;
dUp=dU(nu+1:nu*Nc);
X=AA(1:nx,:)*Xp+BB(1:nx,:)*U;
UX=[U;X;dUp];

function [xopt] = MyQP(H,f,A,b)
xmin=-2*ones(size(f));
mnu=length(f);
nc=length(b);
H=inv(H);
TAB=[-H H*A';A*H -A*H*A'];
bas=[-xmin-H*f;b+A*H*f];
ibi=-[1:mnu+nc]';
ili=-ibi;
basis=zeros(mnu+nc,1);
ib=zeros(mnu+nc,1);
il=zeros(mnu+nc,1);
maxiter=750;
[basis,ib,il,iter]=Mydantzgmp(TAB,bas,ibi,ili,maxiter);
xopt=zeros(mnu,1);
for j=1:mnu
    if il(j) <= 0
        xopt(j)=xmin(j);
    else
        xopt(j)=basis(il(j))+xmin(j);
    end
end
return

```

```

function [bas,ib,il,iter,tab]=Mydantzgmp(a,bv,ibo,ilo,maxiter);
tab=a;bas=bv;ib=ibo;il=ilo;
[m,n]=size(tab);
istand=0;
i=0;
ichk=0;
ichki=0;
iout=0;
for iter=1:maxiter
    ibgz=zeros(1,m);
    itnz=zeros(1,m);
    irgez=zeros(1,m);
    test=zeros(1,m);
    btest=zeros(1,m);
    rat=zeros(1,m);
    if istand == 0
        basmin=0;
        for j=1:m
            if il(j)>0 && bas(il(j))<basmin
                basmin=bas(il(j));
                i=j;
            end
        end
        if basmin >= 0
            return
        end
        iad=i ;
        ichk=il(i) ;
        ichki=i+n ;
        ic=-ib(iad) ;
    else
        iad=istand;
        ic=-il(istand-n);
    end
    for j=1:m
        if ib(j)>0
            ibgz(j)=j;
            test(j)=tab(ib(j),ic);
            btest(j)=bas(ib(j));
            if test(j)~=0
                rat(j)=btest(j)/test(j);
            end
        end
    end
    end
    rmin=-1;
    ir=0;
    k=0;
    for j=1:m
        if rat(j)>0 || (rat(j)==0 && test(j)>0)
            k=k+1;
            if k==1
                rmin=rat(j);
                i=j;
            elseif rat(j)<rmin
                rmin=rat(j);
                i=j;
            end
            end
            ir=ib(i);
            iout=i;
        end
    end
end

```

```

end
  if tab(ichk,ic) ~= 0
    rat1=bas(ichk)/tab(ichk,ic);
    if rat1 >= 0 && (ir == 0 || rat1 <= rmin)
      ir=ichk;
      iout=ichki;
    end
  end
end
if ir == 0
  iter=-iter;
  return
end
if iout <= n
  ib(iout)=-ic;
else
  il(iout-n)=-ic;
end,
if iad > n
  il(iad-n)=ir;
else
  ib(iad)=ir;
end
% ==== Start of procedure that switches basis variables ====
ipiv=1/tab(ir,ic);
tab(ir,:)=tab(ir,:)*ipiv;
temp=bas(ir)*ipiv;
bas=bas-temp*tab(:,ic);
bas(ir)=temp;
tab(ir,ic)=0;
tab=tab-tab(:,ic)*tab(ir,:);
tab(:,ic)=-ipiv*tab(:,ic);
tab(ir,ic)=ipiv;
% ==== end of switching procedure =====
istand=0;
for j=1:m
  if ib(j)>0 && il(j)>0
    istand=iout+n;
  end
end
end
return

```