

CRANFIELD UNIVERSITY

SUNAN CHUMALEE

ROBUST GAIN-SCHEDULED H_∞ CONTROL
FOR UNMANNED AERIAL VEHICLES

SCHOOL OF ENGINEERING

PhD THESIS

Supervisor: Dr James F. Widborne
June 2010

This page intentionally contains only this sentence.

CRANFIELD UNIVERSITY

SCHOOL OF ENGINEERING

PhD THESIS

SUNAN CHUMALEE

Robust gain-scheduled H_∞ control
for unmanned aerial vehicles

Supervisor:

Dr James F. Widborne

June 2010

©Cranfield University 2010. All rights reserved. No part of this publication may be reproduced without the written permission of the copyright owner.

This page intentionally contains only this sentence.

Abstract

This thesis considers the problem of the design of robust gain-scheduled flight controllers for conventional fixed-wing unmanned aerial vehicles (UAVs). The design approaches employ a linear parameter-varying (LPV) control technique, that is based on the principle of the gain-scheduled output feedback H_∞ control, because a conventional gain-scheduling technique is both expensive and time-consuming for many UAV applications. In addition, importantly, an LPV controller can guarantee the stability, robustness and performance properties of the closed-loop system across the full or defined flight envelope. A flight control application problem for conventional fixed-wing UAVs is considered in this thesis. This is an autopilot design (i.e. speed-hold, altitude-hold, and heading-hold) that is used to demonstrate the impacts of the proposed scheme in robustness and performance improvement of the flight controller design over a fuller range of flight conditions.

The LPV flight controllers are synthesized using single quadratic (SQLF) or parameter-dependent (PDLF) Lyapunov functions where the synthesis problems involve solving the linear matrix inequality (LMI) constraints that can be efficiently solved using standard software. To synthesize an LPV autopilot of a Jindivik UAV, the longitudinal and lateral LPV models are required in which they are derived from a six degree-of-freedom (6-DOF) nonlinear model of the vehicle using Jacobian linearization. However, the derived LPV models are nonlinearly dependent on the time-varying parameters, i.e. speed and altitude. To obtain a finite number of LMIs and avoid the gridding parameter technique, the Tensor-Product (TP) model transformation is applied to transform the nonlinearly parameter-dependent LPV model into a TP-type convex polytopic model form. Hence, the gain-scheduled output feedback H_∞ control technique can be applied to the resulting TP convex polytopic model using the single quadratic Lyapunov functions.

The parameter-dependent Lyapunov functions is also used to synthesize another LPV controller that is less conservative than the SQLF-based LPV controller. However, using the parameter-dependent Lyapunov functions involves solving an infinite number of LMIs for which a number of convexifying techniques exist, based on an affine LPV model, for obtaining a finite number of LMIs. In this thesis, an affine LPV model is converted from the nonlinearly parameter-dependent LPV model using a minimum least-squares method. In addition, an alternative approach for obtaining a finite number of LMIs is proposed, by simple manipulations on the bounded real

lemma inequality, a symmetric matrix polytope inequality form is obtained. Hence, the LMIs need only be evaluated at all vertices. A technique to construct the intermediate controller variables as an affine matrix-valued function in the polytopic coordinates of the scheduled parameter is also proposed.

The time-varying real parametric uncertainties are included in the system state-space model matrices of an affine LPV model as a linear fractional transformation (LFT) form in order to improve robustness of the designed LPV controllers in the presence of mismatch uncertainties between the nonlinearly parameter-dependent LPV model and the affine LPV model. Hence, a new class of LPV models is obtained called an uncertain affine LPV model which is less conservative than the existing parameter-dependent linear fractional transformation model (LPV/LFT). New algorithms of robust stability analysis and gain-scheduled controller synthesis for this uncertain affine LPV model using single quadratic and parameter-dependent Lyapunov functions are proposed. The analysis and synthesis conditions are represented in the form of a finite number of LMIs. Moreover, the proposed method is applied to synthesize a lateral autopilot, i.e. heading-hold, for a bounded flight envelope of the Jindivik UAV. The simulation results on a full 6-DOF Jindivik nonlinear model are presented to show the effectiveness of the approach.

Acknowledgements

Since the 1st of July, 2007 (that was a sunny July day and was also the first day of my Ph.D. research), it has been three years with fulfilling a lots of adventure, challenge, enjoyment, happiness and great experience. I have been particularly fortunate during my research to have had the opportunity to collaborate with a broad range of School of Engineering and students. I would like to take this opportunity to acknowledge the help and support of a few.

First and foremost, I would like to express my deep gratitude to my research advisor, Dr. James F. Whidborne, who motivate me with his enthusiasm, guide me with valuable suggestions, and support me throughout the research project. He helped me to expand my knowledge through self-learning and provided me the opportunity to work in this exciting field. His keen insights and clear guidance gave me great encouragement to carry out this research and made this thesis possible. I am also grateful to my thesis review panel, Dr James Whidborne (supervisor), Dr Alastair Cooke (appropriate academic panel member selected by the supervisor) and Dr Peter Sherar (research degree committee panel member), for their valuable comments and suggestions.

My research and study would not have been possible without financial support from the Royal Thai Air Force. There are a number of people, family and friends that I am grateful and deserve to be a part of my success. I would first like to thank my parents, Chiang Chumalee, and, Laor Chumalee, and my sister, Porntip Chumalee, for their ever-present support and encouragement during my study. Special thanks to Dr. Stephen Carnduff for his helpful advice on my first journal article. I would also like to thank all my colleagues in the Dynamics, Simulation and Control Group, especially Naveed Rahman, Naseem Akhtar, Stuart Andrews, Peter Thomas, Deborah Saban, Rick Drury, Pierre-Daniel Jameson, Mudassir Lone and Ken Lai. They are always open for discussions on my research struggles. Their friendship and support also sustained me through many challenging occasions, and helped me to move forward.

I also would like to give special thanks to the Thai community in Cranfield; they are supportive throughout my study. My time here at Cranfield has been memorable and valuable. I have enjoyed every aspect of this country. Once again, I would like to thank, everybody who helped me finish this research and made this Cranfield experience a most pleasurable one.

Last but not the least, I would like to thank my wife, Parichat, and my kids, Sirada and Sirasa, for their patience, understanding and support during the last three years. Their love was worth far more than any degree.

Sunan Chumalee
June, 2010

Contents

Contents	v
List of Figures	xi
List of Tables	xv
Notation	xvii
List of Acronyms	xxi
1 Introduction	1
1.1 Control Design Considerations	2
1.1.1 Aircraft Mathematical Nonlinear Model	2
1.1.2 Model Uncertainties, Disturbances, and Sensor Noises	2
1.1.3 Model Nonlinearities	3
1.1.4 Sensor Limitations	3
1.1.5 Actuator Limitations	3
1.2 Conventional Gain Scheduling	4
1.3 Aims & Objectives	4
1.4 Publications	5
1.5 Thesis Outline	6
2 Preliminaries	9
2.1 Introduction	9

2.2	Mathematical Preliminaries	10
2.2.1	Vector, Matrix, Signal, and System Norms	10
2.2.2	Singular Value Decomposition	12
2.2.3	Linear Fractional Transformation	13
2.2.4	Linear Matrix Inequalities	14
2.2.5	The S -Procedure	15
2.2.6	Useful Tools	16
2.3	Basic LPV Models	17
2.4	Literature Review	18
3	LPV Systems Theory	21
3.1	Methods for Deriving LPV models	21
3.1.1	Jacobian Linearization	21
3.1.2	State Transformation	24
3.1.3	Function Substitution	25
3.1.4	Summary of Three Derivation Methods	26
3.2	LPV Models for Controller Synthesis	27
3.2.1	Grid LPV Model	27
3.2.2	Affine LPV Model	27
3.2.3	TP Convex Polytopic Model	28
3.2.4	LPV/LFT Models	29
3.3	Stability Analysis of LPV Systems	31
3.3.1	Robustness Analysis using SQLF	32
3.3.2	Robustness Analysis using Small Gain Theorem	33
3.4	Controller Synthesis for LPV Systems	35
3.4.1	Gain-Scheduled Controller Design using SQLF	36
3.4.2	Gain-Scheduled Controller Design via LFT	39
4	Numerical Example	45

4.1	Jacobian Approach	46
4.2	State Transformation Approach	50
4.3	Function Substitution Approach	54
4.4	Mismatch Uncertainty	56
4.5	Pole Placement Approach	57
4.6	Conclusion	61
5	Longitudinal LPV Autopilot Design: A TP Approach	63
5.1	Jacobian-Based Longitudinal LPV Model	63
5.2	Longitudinal TP Convex Polytopic Model	69
5.3	Gain-Scheduled H_∞ Autopilot Design	72
5.4	Nonlinear Simulation Results	75
5.5	Conclusion	85
6	Longitudinal LPV Autopilot Design: A PDLF Approach	87
6.1	Stability Analysis using PDLF	88
6.2	Controller Synthesis using PDLF	90
6.3	Numerical Example	96
6.4	Longitudinal Affine LPV Model	96
6.5	Gain-Scheduled H_∞ Autopilot Design	99
6.6	Nonlinear Simulation Results	99
6.7	Conclusion	101
7	Robust Lateral LPV Autopilot Design	111
7.1	Jacobian-Based Lateral LPV Model	112
7.2	Stability Analysis of Uncertain Affine LPV Systems	115
7.2.1	Robustness Analysis using SQLF	118
7.2.2	Robustness Analysis using PDLF	121
7.3	Controller Synthesis for Uncertain Affine LPV Systems	122

7.3.1	Gain-Scheduled Controller Design using SQLF	123
7.3.2	Gain-Scheduled Controller Design using PDLF	127
7.4	Numerical Example	132
7.5	Lateral Uncertain Affine LPV Model	136
7.6	Robust Gain-Scheduled H_∞ Autopilot Design	138
7.7	Nonlinear Simulation Results	141
7.8	Conclusion	142
8	Conclusions	151
8.1	Conclusions & Discussions	151
8.2	Main Contributions	153
8.3	Further Work	154
A	Aircraft Nonlinear Model	157
A.1	Reference Frames & Sign Conventions	157
A.2	Aircraft Equation of Motion	160
A.3	Jindivik Nonlinear Mathematical Model	164
A.3.1	Aerodynamic Force and Moment Models	164
A.3.2	Thrust Model	171
A.3.3	Sensor Model	172
A.3.4	Actuator Model	174
A.4	Open-Loop Characteristics of the Jindivik Nonlinear Model	175
B	UAV aerodynamic model identification from a racetrack manoeuvre	181
B.1	Introduction	181
B.2	Flight System Configuration	183
B.2.1	Flight Control Computer	183
B.2.2	Avionic Instrumentation	183
B.2.3	Radio Telemetry	184

B.2.4	Racetrack manoeuvre	185
B.3	Aircraft Parameter Estimation	188
B.3.1	Model Postulation	188
B.3.2	Flight Data Post-Processing	189
B.3.3	Equation-error Method	192
B.3.4	Results	195
B.4	Design, HIL simulation and flight test	198
B.4.1	PID Autopilot Design	198
B.4.2	Hardware-In-the-Loop (HIL) Simulation	198
B.4.3	Flight Test	199
B.5	Conclusions	199
C	Explicit Controller Formulas for PDLF-based Gain-Scheduled H_∞ Synthesis	205
	Bibliography	209

This page intentionally contains only this sentence.

List of Figures

1.1	The Jindivik UAV	7
2.1	Block diagrams structures of linear fractional transformation	13
3.1	Block diagrams structure of an LPV/LFT model (3.52)	34
3.2	Block diagrams structures of LPV/LFT closed-loop systems	43
4.1	Open-loop dynamic step response of the system (4.1) at t=0	47
4.2	LPV system: H_∞ mixed S/KS synthesis problem	48
4.3	Singular value of S and KS over $\forall \theta \in \Theta$ (with frozen θ)	51
4.4	H_∞ norm of W_1S and W_2KS over $\forall \theta \in \Theta$ (with frozen θ)	52
4.5	Nonlinear step response from -3 to 0 of the Jacobian-based LPV controller (4.11)	53
4.6	Nonlinear step response from -3.16 to 0 of function substitution-based and state transformation-based LPV controllers	54
4.7	Nonlinear step response from -3.16 to 0 of new Jacobian-based LPV controller with the original nonlinear plant	58
4.8	Block diagram structure of the gain-scheduled pole placement controller	59
4.9	Pole placement controller design in MATLAB Simulink environment .	61
4.10	Nonlinear step response from -3.16 to 0 with the original nonlinear plant	62
4.11	Control input to the original nonlinear plant	62
5.1	A comparison between the simulated thrust and the estimated thrust	67
5.2	X_q and α_{trim} are nonlinearly dependent on speed and altitude	70

5.3	The open-loop characteristic of transfer function $\frac{u(s)}{r_{pm}(s)}$ of the Jindivik longitudinal nonlinearly parameter-dependent LPV model	71
5.4	The <i>cno</i> type convex weighting functions in one-dimensional parameter, i.e. $w_{n,j}(p_n(t))$, of the longitudinal TP polytopic model	73
5.5	The <i>cno</i> type convex weighting functions in two-dimensional parameters, i.e. $w_a(p(t))$, of the longitudinal TP polytopic model	73
5.6	The weighted open-loop interconnection for the longitudinal TP convex polytopic plant model	75
5.7	Singular value of S and KS over $\forall \theta \in \Theta$ (with frozen θ)	76
5.8	H_∞ norm of W_1S and W_2KS over $\forall \theta \in \Theta$ (with frozen θ)	77
5.9	The transient performance of H_∞ gain-scheduling autopilot is validated with the Jindivik nonlinear dynamic model about one condition inside the flight envelope, i.e. speed = 506 ft/s and altitude = 10,000 ft	81
5.10	The stability and robustness properties of the closed-loop system were achieved over the defined flight envelope	85
6.1	The transient performance of H_∞ gain-scheduling autopilot is validated with the Jindivik nonlinear dynamic model about one condition inside the flight envelope, i.e. speed = 506 ft/s and altitude = 10,000 ft	105
6.2	The desired performance and robustness objectives are achieved across the defined flight envelope	109
7.1	The open-loop characteristic of transfer function $\frac{v(s)}{\delta_r(s)}$ of the Jindivik lateral LPV model	116
7.2	Structure comparisons of uncertain affine LPV and uncertain LPV/LFT [9] models.	119
7.3	Structure comparisons of uncertain affine LPV and LPV/LFT [9] closed-loop systems.	134
7.4	Nonlinear step response from -3 to 0 of the LPV controller with the original nonlinear plant.	136
7.5	The variation of a nonlinear Y_{δ_r} and an affine Y_{δ_r} with speed and altitude	139
7.6	The variation of $\delta_{Y_{\delta_r}}$ and $\tilde{\delta}_{Y_{\delta_r}}$ (normalize) with speed and altitude . .	140

7.7	The weighted open-loop interconnection for the lateral uncertain affine LPV plant model.	141
7.8	The transient performance of H_∞ gain-scheduling autopilot is validated with the Jindivik nonlinear dynamic model about one condition inside the flight envelope, i.e. speed = 464 ft/s and altitude = 7,5000 ft.	146
7.9	The rate one turn of H_∞ gain-scheduling autopilot is validated with the Jindivik nonlinear dynamic model about one condition inside the flight envelope, i.e. speed = 506 ft/s and altitude = 10,000 ft.	150
A.1	Reference axes & sign conventions [58]	158
A.2	The Jindivik nonlinear model with its autopilot in MATLAB Simulink environment [41]	166
A.3	Aircraft open-loop dynamic responses to 0.1 degree elevator step . . .	176
A.4	Aircraft open-loop dynamic responses to 500 RPM engine speed step	177
A.5	Aircraft open-loop dynamic responses to 0.1 degree aileron step . . .	178
A.6	Aircraft open-loop dynamic responses to 0.1 degree rudder step . . .	179
B.1	RTAF aerial target	183
B.2	SWSDC flight controller	184
B.3	Flight system configuration	186
B.4	Flight data (racetrack pattern) for identification	187
B.5	Static thrust measurement result	189
B.6	Power spectral densities	191
B.7	Data compatibility analysis	193
B.8	Estimation of aerodynamic coefficients	196
B.9	Validation of aerodynamic coefficients	197
B.10	Structure design of the autopilot	200
B.11	Real-time hardware-in-the-loop simulation environment	201
B.12	Validation of the PID autopilot through flight test	202

This page intentionally contains only this sentence.

List of Tables

3.1	Comparison between three methods for deriving LPV models	26
5.1	Simulation data of Jindivik's engine	66
5.2	Jindivik's engine thrust coefficients	67
5.3	Stability and control derivative data (longitudinal mode) of Jindivik nonlinear model	68
5.4	Jindivik's longitudinal aerodynamic coefficients	68
6.1	Numerical comparisons of LPV synthesis techniques; an $(X(\theta), Y(\theta))$ case	97
6.2	Performance γ comparison for different cases of $(X(u, h), Y(u, h))$. .	100
6.3	Computational time of $\hat{A}_k(\theta)$, $\hat{B}_k(\theta)$, $\hat{C}_k(\theta)$ and $D_k(\theta)$	101
7.1	Stability and control derivative data (lateral mode) of Jindivik nonlinear model	114
7.2	Jindivik's lateral aerodynamic coefficients	114
7.3	Performance γ comparison for different cases of $(X(u, h), Y(u, h))$. .	141
A.1	The Jindivik UAV specifications	165
A.2	Mode characteristics of the determined longitudinal and lateral dynamics	180
B.1	RTAF aerial target specifications	184
B.2	Estimated instrumentation error parameters	192
B.3	Parameter correlation coefficient matrix (coefficients of drag, lift, and pitching moment)	195

B.4	Parameter correlation coefficient matrix (coefficients of sideforce, rolling, and yawing moments)	196
B.5	Parameter correlation coefficient matrix (engine speed coefficient) . .	196
B.6	The Details of Vehicle Instrumentation	203
B.7	The Identification Results	203

Notation

\mathbb{N} , \mathbb{R} , and \mathbb{C}	sets of natural, real, and complex numbers
$\mathbb{C}^{m \times n}$	set of $m \times n$ complex matrix
\mathfrak{X}	set of all possible decisions in an optimization problem
\mathfrak{s}	set of feasible decisions: $\mathfrak{s} \subset \mathfrak{X}$
\square	end of proof
\in	belong to
\forall	for all
\subset	subset
$:=$	defined as
(a, b)	$\{x \in R : a < x < b\}$
$[a, b]$	$\{x \in R : a \leq x \leq b\}$
I_n	$n \times n$ identity matrix
I_i	index upper bound
$[a_{ij}]$	a matrix with a_{ij} as its i th row and j th column element
$\text{diag}(a_1, \dots, a_n)$	an $n \times n$ diagonal matrix with a_i as its i -th diagonal element
A^T and A^*	transpose and complex conjugate of a matrix A
A^{-1}	inverse of a matrix A
A^+	Moore-Penrose pseudo-inverse of a matrix A
$\lambda(A)$	eigenvalue of a matrix A
$\bar{\sigma}(A)$ and $\underline{\sigma}(A)$	the largest and the smallest singular values of a matrix A
$\ A\ $	spectral norm of a matrix A : $\ A\ = \bar{\sigma}(A)$
L_2^m	set of m -dimensional vector valued, measurable, square integrable functions with norm $\ \cdot\ _2$ [23]
$\tilde{\delta}$	normalized time-varying real parametric uncertainties
γ	closed-loop quadratic H_∞ performance [10]
ω_n	natural frequency
θ	time-varying parameters
$\hat{\theta}$	structured block-diagonal real uncertainty
$\tilde{\theta}$	normalized time-varying parameters
ξ	damping ratio
$\sup\{\mathbb{F}\}$	the supremum or least upper bound of a set \mathbb{F}
$\inf\{\mathbb{F}\}$	the infimum or greatest lower bound of a set \mathbb{F}
$\mathcal{G}(t) * u(t)$	convolution integral of $g(t)$ and $u(t)$, $t \in R$
$\mathcal{F}_l(M, Q)$	lower LFT
$\mathcal{F}_u(M, Q)$	upper LFT

Chapters 5-B only

C_D, C_L	aerodynamic drag and lift coefficients in stability axes
C_{D_i}	induced drag coefficient
$C_{D_{C_L > C_{L_{crit}}}}$	drag coefficient due to the lift coefficient being greater than the critical lift coefficient
C_{D_M}	drag coefficient due to Mach number
$C_{D_{Re}}$	Reynolds number dependent profile drag coefficient
C_{D_s}	drag coefficient increment due to the deflection of spoilers
$C_{D_{uc}}$	drag coefficient increment due to the extension of undercarriage
C_{D_W}, C_{Y_W}	aerodynamic drag and side force coefficients in wind axes
$C_{D_{Z_0}}$	Reynolds number constant profile drag coefficient
$C_{L_{crit}}$	critical lift coefficient
C_{L_s}	lift coefficient increment due to the deflection of spoilers
C_{L_T}	tailplane lift coefficient
$C_{L_{wb}}$	wing-body lift coefficient
C_X, C_Y, C_Z	aerodynamic force coefficients
$C_{X_{wb}}, C_{Z_{wb}}$	wing-body combination coefficient of axial and normal forces
C_l, C_m, C_n	aerodynamic moment coefficients
$C_{m_{1/4}}$	quarter chord pitching moment coefficient
D_{int}	engine intake drag
I_x, I_y, I_z, I_{xz}	mass moments of inertia, slug-ft ²
K_{ff1}	flexibility factor applied to the rigid wing-body combination lift curve slope
K_{ff2}	flexible factor applied to the rigid body rate of change of downwash with respect to angle of attack
L, M, N	rolling, pitching, and yawing moments
$L_{aero}, M_{aero}, N_{aero}$	aerodynamic moments
M	Mach number
N_{nd}	non dimensional engine speed
P	atmospheric pressure
$P_{int_{ratio}}$	intake pressure ratio
P_{ratio}	pressure ratio
P_{rec}	engine pressure recovery
Re	Reynolds number
S	wing surface area, ft ²
S_T	tailplane surface area, ft ²
T	engine thrust, lb _f or outside air temperature, kelvin
$T_G, T_{G_{nd}}$	gross and non dimensional gross thrusts
T_{wl}	height of the tailplane from the waterline
V	airframe velocity, ft/s
$V_{\delta_{aR}}$	aileron reversal speed, ft/s
X, Y, Z	axial, side, normal forces
$X_{aero}, Y_{aero}, Z_{aero}$	aerodynamic forces
a	speed of sound
a_{1wb}	wing-body combination lift curve slope of the flexible aircraft

a_{1T}	tailplane lift curve slope
a_{2T}	elevator effectiveness
a_x, a_y, a_z	body axis translational axial, lateral, and normal accelerations, ft ² /s
b, \bar{c}	wingspan and mean aerodynamic chord, ft
$c_{1/4_{wl}}$	height of the quarter chord from the waterline
g	acceleration due to gravity, assume constant = 32.17 ft/s ²
h_0	aerodynamic centre position as a percentage of the mean aerodynamic chord
h_{cg}	centre of gravity position as a percentage of the mean aerodynamic chord
h_{wl}	height of the centre of gravity from the waterline
k_{TG}	gross thrust factor
$k_{P_{rec}}$	intake pressure recovery factor
k_h	altitude correction factor
$k_{\dot{m}}$	mass flow coefficient
l_T	tail moment arm, ft
lag_T	tailplane angle of attack lag angle due to downwash
m	aircraft mass, slug
\dot{m}	engine mass flow
p, q, r	body axis roll, pitch, and yaw angular rates, rad/s
\bar{q}	dynamic pressure, lb _f /ft ²
rpm, $N1$	engine speed
u, v, w	body axis translational axial, lateral, and normal velocities, ft/s
x_E, y_E, h	longitude, latitude, and altitude, ft
α, β	angle of attack and sideslip, rad
α_R, α_T	tailplane rigging and angle of attack angle, rad
α_w	wing setting angle, rad
α_{wb}	wing-body combination angle of attack, rad
β_{pg}	Prandtl-Glauert factor
β_{pg1}	Prandtl-Glauert factor at the critical mach number
$\delta_a, \delta_e, \delta_f, \delta_r, \delta_{th}$	aileron, elevator, flap, rudder, and throttle deflections, rad
η	dynamic viscosity
ϕ, ψ	roll, pitch, and yaw angles, rad
θ	pitch angle (rad) or unknown parameters vector
ρ	air density, slug/ft ³
ρ_0	sea level air density, = 0.002377 slug/ft ³
ε	downwash angle, rad

Subscripts

0, trim	bias and trim values
$p, q, r, v, \alpha, \dot{\alpha}$	derivative with respect to indicated quantity
$\beta, \delta_a, \delta_e, \delta_r$	

This page intentionally contains only this sentence.

List of Acronyms

6-DOF	Six Degree-of-Freedom
ACE	Asynchronous Communication Element
ADC	Analog-to-Digital Converter
CAS	Control Augmentation System
CFD	Computational Fluid Dynamics
DCM	Direction Cosine Matrix
GPS	Global Positioning System
HIL	Hardware-In-the-Loop
HOSVD	Higher Order Singular Value Decomposition
ISP	In-System Programmable
LFT	Linear Fractional Transformation
LFR	Linear Fractional Representation
LMI	Linear Matrix Inequality
LPV	Linear Parameter-Varying
LQE	Linear Quadratic Estimator
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
LTl	Linear Time-Invariant
NDI	Nonlinear Dynamic Inversion
PDLF	Parameter-Dependent Lyapunov Function
PID	Proportional, Integral and Derivative control
PSD	Power Spectral Density
PWM	Pulse-Width Modulation
RFI	Radio Frequency Interference
RTAF	Royal Thai Air Force
SAS	Stability Augmentation System
SDRE	State Dependent Riccati Equation
SLPV	Switching Linear Parameter-Varying
SQLF	Single Quadratic Lyapunov Function
SVD	Singular Value Decomposition
SWSDC	Science and Weapon System Development Center
TP	Tensor-Product
UAV	Unmanned Aerial Vehicle

This page intentionally contains only this sentence.

Chapter 1

Introduction

Unmanned Aerial Vehicles (UAVs) could be defined as power-driven aircraft, other than model aircraft, that are designed to fly without a human operator on board. Typically, UAVs are used primarily to avoid putting persons at risk, or for missions where the task is better suited to a machine, e.g. D^3 missions - “Dirty, Dull, Dangerous”. To date, UAVs have been used in both military and civilian applications. Some military type applications include reconnaissance, surveillance (air, land, maritime), border patrol, and drug interdiction. In addition, some civilian type applications include surveillance, hydro-line inspections, water resources management, flood damage, and city mapping. UAVs have shown potential as being strong effective platforms for supporting both military and civilian applications.

The increasing requirements on the capabilities of UAVs means that there is considerable ongoing research. Advanced control methodologies, such as optimal control, robust control, nonlinear control, intelligent control, etc., that guarantee the stability, robustness and performance properties of the closed-loop system are some of the very interesting active research areas for UAVs. Traditionally, automatic flight control systems can be categorised according to their function as; (i) Stability Augmentation System (SAS) is designed in order to improve flying characteristics (damping ratio and natural frequencies) of an aircraft to achieve an acceptable requirement level of flying qualities standards, e.g. MIL-F-8785C, (ii) Control Augmentation System (CAS) is additionally designed to provide a specific type of response to the pilot’s input, e.g. pitch rate commands, (iii) Autopilot system is a pilot relief task that is fully automatic control systems, e.g. airspeed hold, altitude hold, heading hold, and turn coordination.

Since the 1950’s, classical controller design techniques such as root-locus, Bode and Nyquist plots, and frequency response analysis have been successfully and popularly implemented in the automatic flight control systems for the aerospace industry. These design techniques are well understood, clearly visible, highly structured system, and highly amenable to implementation. However, classical control techniques are limited when the controllers for multivariable systems with high internal coupling are to be designed [21]. More detail of the limitations of classical control

techniques are described in the next section, where common control design considerations of flight control systems are also outlined. Sections 1.2 and 1.3 give the motivations and aim & objectives, respectively. The publications and thesis outline are briefly summarized in sections 1.4 and 1.5, respectively.

1.1 Control Design Considerations

1.1.1 Aircraft Mathematical Nonlinear Model

Before a flight controller can be designed, an aircraft mathematical nonlinear model has to be determined. Traditionally, the model parameters are determined using wind tunnel tests by measuring the aerodynamic forces and moments introduced on an aircraft. Recently, computational fluid dynamics (CFD) methods are becoming important. Furthermore, the aircraft moments of inertia are calculated and the aircraft engine model is determined from experimental data. However, these standard processes are both expensive and time-consuming and may not be affordable or practicable for many UAV applications [31, 32, 36].

1.1.2 Model Uncertainties, Disturbances, and Sensor Noises

In practice, most flight control design techniques require a linearized model of an aircraft's dynamics about some trim condition. An aircraft linear model is typically either derived from a six degree-of-freedom (6-DOF) nonlinear model [33, 68] or determined from experimental measurement by parameter identification methods [28, 36, 53, 57, 78]. Having an accurate linear model, a flight controller can be successfully designed. However, the linear model will never be an entirely accurate representation of aircraft flight because the true aircraft dynamic model parameters are not exactly known and the aircraft dynamics are non-linear. Often this is primarily because there are characteristics which cannot easily be modelled; this is especially the case under extremely aggressive manoeuvring of aircraft where highly nonlinear flight regimes yield unsteady and nonlinear aerodynamic effects which are hard and complicated to model [52]. In addition, there are unmodelled dynamics and/or inaccuracies in the model parameters hence there will always be modelling inaccuracies in aircraft dynamic models [52]. Moreover, there will always be (usually uncertain) external disturbance and sensor noises influencing the dynamic behavior of the aircraft. The model uncertainties, disturbances, and sensor noises are another practical problem for flight control design.

1.1.3 Model Nonlinearities

In general, most real plants are nonlinear; this is especially the case for aircraft where their dynamic characteristics vary, following some time-varying parameters. Chumalee and Whidborne [33] showed that, when an aircraft is about a wings level and constant altitude and airspeed flight condition, its dynamic characteristics are nonlinearly dependent on attitude and velocity. In addition, for some UAV missions such as targeting, deception, electronic warfare and offensive operations, UAVs are often required to operate over a full flight envelope at various attitudes and velocities in which the model nonlinearities are another common problem for flight control design.

1.1.4 Sensor Limitations

An automatic flight control system requires some data feedbacks in order to stabilize the aircraft and achieve asymptotic tracking of a known reference trajectory. Practically, these data feedbacks are passed to the automatic flight controller via flight sensors. Therefore, the sensor limitations, e.g. maximum measurement value or maximum update rate, are another actual problem for flight control design. For example, suppose the Crossbow VG400CD is assumed to be used in a flight control system for measuring attitude roll and pitch angles, roll, pitch, and yaw rates, and X, Y, and Z body axis accelerations. In addition, the pitch and roll rates are assumed to be data feedbacks for closed-loop control. Since the maximum roll, pitch, and yaw rates of the Crossbow VG400CD are $100^\circ/\text{s}$, the automatic flight controller has to be designed not to manoeuvre the aircraft with pitch and roll rates that are greater than $100^\circ/\text{s}$.

1.1.5 Actuator Limitations

An automatic flight control system controls an aircraft via actuators, hence actuator saturation (position and rate) and actuator time lag present another common problem for flight control design. For example, suppose the Futaba S9206 (high-torque airplane servo) is assumed to be used as the actuators in a flight control system. Since the time lag of Futaba S9206 is 0.02 s, the automatic flight controller has to be designed not to control the aircraft with a frequency that is greater than 8 Hz (-3 dB bandwidth). In addition, the ratio of elevator deflection to Futaba S9206 deflection is assumed to be one to three (1:3). Since the speed of Futaba S9206 is $315^\circ/\text{s}$, the automatic flight controller has to be also designed not to control the aircraft with an excessive gain that make elevator deflection's speed greater than $105^\circ/\text{s}$.

1.2 Conventional Gain Scheduling

A common approach in industry to handle the nonlinear property of the aircraft is by means of gain-scheduling. The conventional approach is to design a local linear time invariant (LTI) controller for each member of a set of operating conditions that cover the whole of the flight envelope. As the operating conditions change, a global controller of the closed-loop system is determined on-line and in flight by interpolating the gain values of each local LTI controller (that are within the varied operating conditions) according to the current value of the scheduling parameters. Experimentally, this design approach has been successfully and popularly implemented in many engineering applications (e.g. submarines, engines, aircraft, etc.) in order to cover the entire operating range of system plants but, theoretically, it comes with no guarantees on the robustness, performance, or even nominal stability of the overall gain scheduled design [90].

The cost-effective development of many UAV applications is very significant but the conventional gain-scheduling technique is both expensive and time-consuming. In addition it cannot guarantee the stability, robustness and performance properties of the closed-loop system [90]. Hence this design approach is less suitable for a UAV application. Based on these shortcomings of the conventional gain-scheduling technique, two challenging motivation problems of this thesis are how to design a single robust gain-scheduled flight controller for conventional fixed-wing UAVs for which (i) the designed controller can operate in a fuller range of flight conditions and (ii) the designed controller can guarantee the stability and robustness properties of the closed-loop system.

1.3 Aims & Objectives

An advanced robust gain-scheduling technique, namely linear parameter-varying (LPV) control [91] that is based on the principle of the H_∞ control [47, 104], can be used to handle uncertainties and nonlinearities of a nonlinear plant model. Importantly, an LPV controller theoretically guarantees stability, robustness, and performance properties of the closed-loop system [8, 23, 97]. Therefore, the motivation problems listed in the previous section can be solved using an LPV control technique with the mixed-sensitivity criterion [20, 33, 47]. This will provide a controller with good command following (i.e. small tracking error), good disturbance attenuation, low sensitivity to measurement noise, reasonably small control efforts, and that is robustly stable to additive plant perturbations. To design a single robust gain-scheduled flight controller using an LPV control technique for which the designed controller satisfies the two motivation problems is the main aim of this thesis.

Moreover, an interesting flight control application problem for conventional fixed-wing UAVs is considered in this thesis. This is an autopilot design (i.e. speed-hold, altitude-hold, and heading-hold) and is also a main objective of this thesis in order

to demonstrate the impacts of the proposed scheme in robustness and performance improvement of the flight controller design over a fuller range of flight conditions. The effectiveness of the proposed methods in designing an LPV autopilot is verified and validated through a 6-DOF nonlinear model of the Jindivik UAV, shown in Figure 1.1, that has been developed by Fitzgerald [41] in the MATLAB Simulink environment.

1.4 Publications

The publications of this thesis are the following:

Conference papers

- S. Chumalee and J. F. Whidborne. Pole Placement Controller Design for Linear Parameter Varying Plants. *Proceedings of the UKACC International Conference on Control 2008*, Manchester, UK, September 2008.
- S. Chumalee and J. F. Whidborne. Experimental Development of an UAV Nonlinear Dynamic Model. *Proceedings of the 24th Bristol International Unmanned Air Vehicle Systems (UAVS) Conference*, Bristol, UK, March 2009.
- S. Chumalee and J. F. Whidborne. LPV Autopilot Design of a Jindivik UAV. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Chicago, Illinois, Aug. 2009.
- S. Chumalee and J. F. Whidborne. Identification and Control of RTAF Aerial Target. *Proceedings of European Control Conference 2009*, Budapest, Hungary, Aug. 2009.

Journal articles

- S. Chumalee and J. F. Whidborne. UAV Aerodynamic Model Identification from a Racetrack Manoeuvre. *J. Aerospace Engineering, Proc.IMEchE Vol. 224 Part G*, pages 831–842, 2010.
- S. Chumalee and J. F. Whidborne. Gain-scheduled H_∞ Autopilot Design via Parameter-Dependent Lyapunov Functions. *Journal of Guidance, Control, and Dynamics*, 2010. (submitted).
- S. Chumalee and J. F. Whidborne. Robust Flight Control for Uncertain Affine Linear Parameter-Varying Models. *International Journal of Control*, 2010. (submitted).

1.5 Thesis Outline

This thesis is organized as follows:

Chapter 2 provides a brief overview of research in LPV control techniques. The relevant mathematical background is also briefly summarized.

Chapter 3 outlines theory involved in LPV systems which includes (i) methods for deriving LPV models, i.e. Jacobian linearization, state transformation and function substitution, (ii) LPV models, i.e. grid LPV model, affine LPV model and TP convex polytopic model, (iii) parameter-dependent linear fractional transformation models (LPV/LFT), (iv) stability analysis of LPV systems, i.e. Lyapunov-based stability analysis and small gain theorem, and (v) controller synthesis for LPV Systems, i.e. bounded real lemma, gain-scheduled H_∞ control and gain scheduling via LFT.

Chapter 4 illustrates the implementation of LPV systems theory for the nonlinear control problem via a simple numerical example [61] that is known to cause difficulties for LPV controllers. The parameters variation of the example [61] is cancelled using the gain-scheduled pole placement state feedback. For the example, the approach yields reliable closed-loop stability and good closed-loop transient performance of the system because it makes the nonlinear plant appear to be an LTI plant, hence well-developed LTI tools can be applied.

In chapter 5, a longitudinal nonlinearly parameter-dependent LPV model is derived from a 6-DOF nonlinear dynamic model using Jacobian linearization. To synthesize an LPV controller with a finite number of LMIs and avoid the gridding technique, the TP model transformation is employed in order to transform a given nonlinearly parameter-dependent LPV model into a TP convex polytopic model form. Having determined the longitudinal TP polytopic model, the H_∞ gain-scheduling control that is proposed by Apkarian et al. [10] can immediately be applied to the resulting TP polytopic model to yield an LPV autopilot that guarantees the stability and robustness properties of the closed-loop system.

In chapter 6, the longitudinal nonlinearly parameter-dependent LPV model is converted into an affine LPV model using the minimum least-squares method [58]. Based on this longitudinal affine LPV model, another LPV controller is synthesized with a finite number of LMIs using a new parameter-dependent Lyapunov functions approach. An existing PDFL approach, that is based on a multi-convexity method [11], is given in Appendix C for which an improvement of the parameter-dependent Lyapunov-based stability and performance analysis from the proposed method can be compared with those from a multi-convexity approach [11].

In chapter 7, a lateral nonlinearly parameter-dependent LPV model is derived from a 6-DOF nonlinear dynamic model using Jacobian Linearization. A lateral affine LPV model is obtained using the minimum least-squares method [58] in a similar approach to Chapter 6. The time-varying real parametric uncertainties are included in the system state-space model matrices in an LFT form in order to guarantee closed-loop stability and improve transient performance in presence of the mismatch



Figure 1.1: The Jindivik UAV

uncertainties between the lateral nonlinearly parameter-dependent LPV model and the affine LPV model. Based on the proposed uncertain affine LPV model, a robust lateral LPV controller is synthesized with a finite number of LMIs using parameter-dependent Lyapunov functions.

Chapter 8 summarizes the discussions and main contributions of the thesis, and suggestions for future work directions are given.

Appendix A briefly summarizes an aircraft mathematical model; this is especially applicable for the Jindivik nonlinear model [41] from which the mathematical modelling of aerodynamic forces and moments, thrust, sensors, and actuators are presented. In addition, two major lateral and longitudinal modes of the vehicle about wings level and constant altitude 10,000 ft and airspeed 506.3 ft/s straight flight condition are also presented.

In appendix B, an ordinary piloted manoeuvre and off-trim condition flight data (racetrack manoeuvre) of a Royal Thai Air Force (RTAF) aerial target was studied and identified in order to estimate the aerodynamic coefficients of the vehicle. Only two flight tests had to be undertaken. The first flight test was done to record flight data by controlling the aerial target manually. The second flight test was done to validate the proportional, integral and derivative (PID) autopilot. As shown by the flight test results of the PID autopilot, the identified 6-DOF non-linear model was sufficiently reliable and accurate for the design of a satisfactory control system. This appendix demonstrates the usefulness of system identification techniques for UAV control system design and development in a cost-effective manner.

Appendix C briefly summarizes an existing parameter-dependent Lyapunov functions approach for synthesizing a PDLF-based LPV controller using a multi-convexity method [11]. In addition, the controller is constructed using an explicit controller formulas [43].

This page intentionally contains only this sentence.

Chapter 2

Preliminaries

2.1 Introduction

Advanced control methodologies can be roughly classified according to their objective nature as: (i) optimal control is developed to achieve certain optimal performance (i.e. minimizing a quadratic cost function), (ii) robust control is developed to handle system plants subject to uncertainties, disturbances, and noise measurements with high performances, (iii) nonlinear control is developed to handle nonlinear systems with high performances, and (iv) intelligent control is developed to handle systems with unknown dynamic models of system plants. Although advanced control methodologies have shown potential in the field of improving robustness, better performance, de-coupling control and simplifying the design process, some of them do not yet have the maturity required for industrial application [67]. From an industrial point of view, the desirable features of a very good flight controller are simplicity, transparency, quality, accuracy, reliability, generality and implementability [67].

The optimal and robust control methods are fairly well-studied, however, their performances would be conservative when UAVs have to operate in a nonlinear flight regimes with a wide range of attitudes and velocities. Nonlinear control methods may be suitable for designing a flight controller in this thesis but they are generally very complex. The intelligent control would not be also suitable because it is hard to guarantee the robustness and performance properties of the closed-loop system in presence of model uncertainties and nonlinearities, disturbances, and noises. In addition, intelligent control typically requires a lot of training data and it is actually not very transparent. Moreover, an intelligent control can be quite hard to implement due to its complexity of the developed control algorithm.

In this thesis, an LPV control approach [91] is selected for designing a flight controller because it can handle uncertainties and nonlinearities of a nonlinear plant model. Importantly, an LPV controller can guarantee stability and robustness properties of the closed-loop system [8, 23, 97].

2.2 Mathematical Preliminaries

This section briefly summarizes useful mathematical background that will be used throughout this thesis.

2.2.1 Vector, Matrix, Signal, and System Norms

The material in this sub-section is essentially taken from Zhou et al. [104] and Gu et al. [47]. Let a vector $x = [x_1, \dots, x_n]^T \in \mathbb{C}^n$, then a real valued function $\|x\|_p$ is the vector p -norm of x defined as

$$\|x\|_p := \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}, \text{ for } 1 \leq p \leq \infty \quad (2.1)$$

$$\|x\|_1 := \sum_{i=1}^n |x_i|, \text{ for } p = 1 \quad (2.2)$$

$$\|x\|_2 := \sqrt{\sum_{i=1}^n |x_i|^2}, \text{ for } p = 2 \quad (2.3)$$

$$\|x\|_\infty := \max_{1 \leq i \leq n} |x_i|, \text{ for } p = \infty \quad (2.4)$$

A norm of a vector is a measure of the vector length. When $p = 2$, $\|x\|_2$ is the Euclidean norm of a vector x (or the Euclidean distance of a vector x from the origin). It shall be denoted by $\|x\| := \|x\|_2$.

Let a matrix $A = [a_{ij}] \in \mathbb{C}^{m \times n}$ and a vector $x \in \mathbb{C}^n$, the matrix norm induced by a vector p -norm is defined by

$$\|A\|_p := \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}, \text{ for } 1 \leq p \leq \infty \quad (2.5)$$

$$\|A\|_1 := \max_{1 \leq j \leq n} \sum_{i=1}^m |a_{ij}|, \text{ for } p = 1, \text{ column sum} \quad (2.6)$$

$$\|A\|_2 := \sqrt{\lambda_{\max}(A^*A)}, \text{ for } p = 2 \quad (2.7)$$

$$\|A\|_\infty := \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}|, \text{ for } p = \infty, \text{ row sum} \quad (2.8)$$

When $p = 2$, $\|A\|_2$ is the spectral norm (i.e. largest singular value) of a matrix A . It shall be denoted by $\|A\| := \|A\|_2$.

The p -norm of a vector signal $x(t) = [x_1(t), \dots, x_n(t)]^T$, $t \in \mathbb{R}$, is defined by

$$\|x\|_p := \left(\int_{-\infty}^{\infty} \sum_{i=1}^n |x_i(t)|^p dt \right)^{1/p}, \text{ for } 1 \leq p \leq \infty \quad (2.9)$$

$$\|x\|_1 := \int_{-\infty}^{\infty} \sum_{i=1}^n |x_i(t)| dt, \text{ for } p = 1 \quad (2.10)$$

$$\|x\|_2 := \sqrt{\int_{-\infty}^{\infty} \sum_{i=1}^n x_i^2(t) dt}, \text{ for } p = 2 \quad (2.11)$$

$$\|x\|_{\infty} := \sup_{t \in \mathbb{R}} \|x(t)\|_{\infty}, \text{ for } p = \infty \quad (2.12)$$

Note that, given a set of real numbers \mathbb{F} , $\sup(\mathbb{F})$ is the supremum or least upper bound of a set \mathbb{F} is defined to be the smallest real number that is greater than or equal to every number in \mathbb{F} while $\inf(\mathbb{F})$ is the infimum or greatest lower bound of a subset \mathbb{F} is defined to be the biggest real number that is smaller than or equal to every number in \mathbb{F} .

Let $\hat{x}(j\omega)$ be the Fourier transform of $x(t)$ where ω is the real frequency variable in radians per unit time, the frequency domain 2-norm is defined by

$$\|\hat{x}\|_2 := \sqrt{\frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{x}^*(j\omega) \hat{x}(j\omega) d\omega} \quad (2.13)$$

Note that, by Parseval's identity, $\|x\|_2 = \|\hat{x}\|_2$. That is

$$\sqrt{\int_{-\infty}^{\infty} x^2(t) dt} = \sqrt{\frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{x}^*(j\omega) \hat{x}(j\omega) d\omega} \quad (2.14)$$

The normed spaces, consisting of a vector signal $x(t) = [x_1(t), \dots, x_n(t)]^T$, $t \in \mathbb{R}$ with finite norm, is defined by

$$L_p^n(R) := \left\{ x(t) : \|x\|_p = \left(\int_{-\infty}^{\infty} \sum_{i=1}^n |x_i(t)|^p dt \right)^{1/p} < \infty \right\}, \text{ for } 1 \leq p \leq \infty \quad (2.15)$$

$$L_2^n(R) := \left\{ x(t) : \|x\|_2 = \sqrt{\int_{-\infty}^{\infty} \sum_{i=1}^n |x_i(t)|^2 dt} < \infty \right\}, \text{ for } p = 2 \quad (2.16)$$

For a stable LTI system $\mathcal{G} : L_2^m(\mathbb{R}) \rightarrow L_2^n(\mathbb{R})$, the H_{∞} norm of $G(s)$ is given by

$$\|G\|_{\infty} = \sup_{\omega \in \mathbb{R}} \bar{\sigma}\{G(j\omega)\} \quad (2.17)$$

where $G(s)$ or $G(j\omega) \in \mathbb{C}^{n \times m}$ is the transfer function matrix of \mathcal{G} and $\bar{\sigma}(\cdot)$ is defined in the next sub-section. In addition, let $u(t) \in L_2^m$ and $y(t) \in L_2^n$ be the input

and output vectors of system \mathcal{G} respectively, the induced L_2 -norm or L_2 -gain of the system \mathcal{G} is given by

$$\|\mathcal{G}\|_{i2} = \sup_{u \in L_2^m} \frac{\|\mathcal{G} * u\|_2}{\|u\|_2} = \sup_{u \in L_2^m} \frac{\|y\|_2}{\|u\|_2} \quad (2.18)$$

where $\mathcal{G}(t) * u(t)$ denotes the convolution integral of $\mathcal{G}(t)$ and $u(t)$. That is

$$y(t) = \mathcal{G}(t) * u(t) = \int_0^t g(t - \tau)u(\tau)d\tau \quad (2.19)$$

Note that, for LTI systems, the H_∞ norm is equal to the induced L_2 -norm or L_2 -gain (i.e. the maximal gain of the system) [37, page 75]. That is

$$\|G\|_\infty = \sup_{\omega \in \mathbb{R}} \bar{\sigma}\{G(j\omega)\} = \sup_{u \in L_2^m} \frac{\|y\|_2}{\|u\|_2} \quad (2.20)$$

2.2.2 Singular Value Decomposition

A singular value decomposition (SVD) is a very useful tool to measure the size of a matrix for which the corresponding singular vectors are good indications of strong/weak input/output directions.

Lemma 2.2.1. (*Singular Values and Eigenvalues [37]*) Given a complex matrix $A \in \mathbb{C}^{m \times n}$, the set of singular values of A is denoted by $\{\sigma_i(A)\}$ which equals the k largest square roots of the eigenvalues λ of A^*A where $k = \min\{m, n\}$. That is

$$\sigma_i(A) = \sqrt{\lambda(A^*A)}, \quad i = 1, 2, \dots, k \quad (2.21)$$

Normally, the singular values are ordered as $\sigma_i \geq \sigma_{i+1}$

Hence:

$$\bar{\sigma}(A) = \sigma_1(A) = \sup_{x \in \mathbb{C}^n} \frac{\|Ax\|_2}{\|x\|_2} = \|A\| \quad (2.22)$$

$$\underline{\sigma}(A) = \sigma_k(A) = \inf_{x \in \mathbb{C}^n} \frac{\|Ax\|_2}{\|x\|_2} \quad (2.23)$$

Theorem 2.2.2. (*Singular Value Decomposition [104, Theorem 2.11]*) Given a complex matrix $A \in \mathbb{C}^{m \times n}$, there exist two unitary matrices

$$U = \begin{bmatrix} u_1 & u_2 & \dots & u_m \end{bmatrix} \in \mathbb{C}^{m \times m}$$

$$V = \begin{bmatrix} v_1 & v_2 & \dots & v_m \end{bmatrix} \in \mathbb{C}^{m \times m}$$

such that

$$A = U\Sigma V^* \quad (2.24)$$

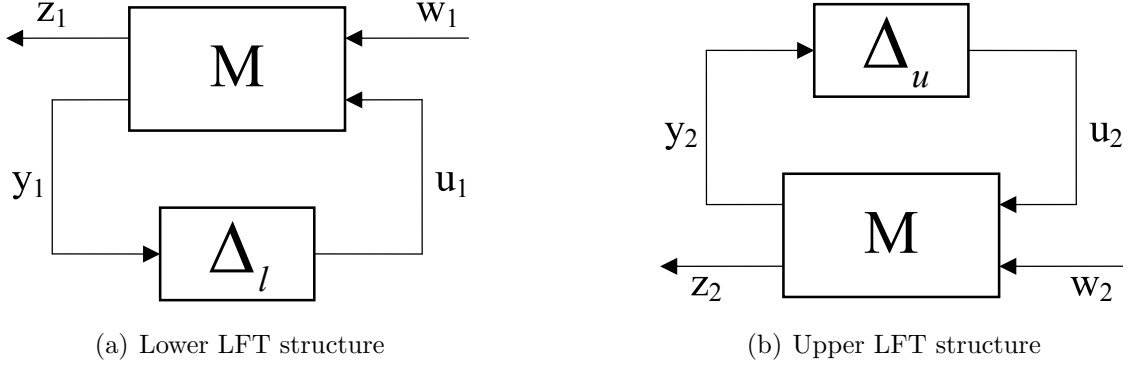


Figure 2.1: Block diagrams structures of linear fractional transformation

where

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{m \times n}$$

$$\Sigma_1 = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_k \end{bmatrix} \in \mathbb{R}^{k \times k}$$

and $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k \geq 0, k = \min\{m, n\}$.

2.2.3 Linear Fractional Transformation

Linear Fractional Transformations (LFTs) are a powerful and flexible approach to represent uncertainty in matrices and systems.

Definition 2.2.3. [104] Let M be a complex matrix partitioned as

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \in \mathbb{C}^{(p_1+p_2) \times (q_1+q_2)}$$

A lower LFT with respect to $\Delta_l \in \mathbb{C}^{q_2 \times p_2}$, shown in Figure 2.1-a, that a mapping $\mathcal{F}_l(M, \Delta_l) : \mathbb{C}^{q_2 \times p_2} \rightarrow \mathbb{C}^{p_1 \times q_1}$ is defined by

$$\mathcal{F}_l(M, \Delta_l) := M_{11} + M_{12}\Delta_l(I - M_{22}\Delta_l)^{-1}M_{21} \quad (2.25)$$

where the inverse $(I - M_{22}\Delta_l)^{-1}$ exists. An upper LFT with respect to $\Delta_u \in \mathbb{C}^{q_1 \times p_1}$, shown in Figure 2.1-b, that a mapping $\mathcal{F}_u(M, \Delta_u) : \mathbb{C}^{q_1 \times p_1} \rightarrow \mathbb{C}^{p_2 \times q_2}$ is defined by

$$\mathcal{F}_u(M, \Delta_u) := M_{22} + M_{21}\Delta_u(I - M_{11}\Delta_u)^{-1}M_{12} \quad (2.26)$$

where the inverse $(I - M_{11}\Delta_u)^{-1}$ exists.

2.2.4 Linear Matrix Inequalities

The following definitions are required.

Definition 2.2.4. (Convex sets [89]) A set \mathfrak{s} in a linear vector space is said to be convex if $x_1, x_2 \in \mathfrak{s}$ then $\{x := \alpha x_1 + (1 - \alpha)x_2\} \in \mathfrak{s}$ for all $\alpha \in (0, 1)$

Definition 2.2.5. (Convex combinations [89]) Let \mathfrak{s} be a subset of a vector space. The point

$$x := \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n = \sum_{i=1}^n \alpha_i x_i \quad (2.27)$$

is called a convex combination of $x_1, \dots, x_n \in \mathfrak{s}$ if $\alpha_i \geq 0$ for $i = 1, \dots, n$ and $\sum_{i=1}^n \alpha_i = 1$

Definition 2.2.6. (Convex hull [89]) The convex hull $Co\{\mathfrak{s}\}$ of any subset $\mathfrak{s} \subset \mathfrak{X}$ is the intersection of all convex sets containing \mathfrak{s} . If \mathfrak{s} consists of a finite number of elements, then these elements are referred to as the vertices of $Co\{\mathfrak{s}\}$.

Definition 2.2.7. (Affine functions [89]) A function $f : \mathfrak{s} \rightarrow \mathfrak{T}$ is affine if

$$f(\alpha x_1 + (1 - \alpha)x_2) = \alpha f(x_1) + (1 - \alpha)f(x_2) \quad (2.28)$$

for all $x_1, x_2 \in \mathfrak{s}$ and $\alpha \in \mathbb{R}$

Definition 2.2.8. [10] A matrix polytope is defined as the convex hull of a finite number of matrix vertices N_i with the same dimensions.

$$Co\{N_1, N_2, \dots, N_r\} := \left\{ \sum_{i=1}^r \alpha_i N_i : \alpha_i \geq 0, \sum_{i=1}^r \alpha_i = 1 \right\} \quad (2.29)$$

Definition 2.2.9. [34] Given a matrix $M \in \mathbb{R}^{p \times p}$, (i) M is a negative definite symmetric matrix, i.e. $M = M^T < 0$, if $X^T M X < 0$ for all nonzero vector $X \in \mathbb{R}^p$. (ii) M is a positive definite symmetric matrix, i.e. $M = M^T > 0$, if $X^T M X > 0$ for all nonzero vector $X \in \mathbb{R}^p$.

A linear matrix inequality is an affine function [89] mapping $F : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times m}$ that is expressed of the form

$$\begin{aligned} F(x) &= F_0 + x_1 F_1 + x_2 F_2 + \dots + x_n F_n \\ &= F_0 + \sum_{i=1}^n x_i F_i < 0 \end{aligned} \quad (2.30)$$

where $x = [x_1, \dots, x_n]^T \in \mathbb{R}^n$ is a vector of n real numbers called the decision variables. $F_0, \dots, F_n \in \mathbb{R}^{m \times m}$ are real symmetric matrices, i.e. $F_j = F_j^T$, for $j = 0, \dots, n$. $F(x)$ is a negative definite symmetric matrix hence, by Definition 2.2.9, $u^T F(x) u < 0$ for all nonzero vector $u \in \mathbb{R}^m$. This is equivalent to the condition that all eigenvalues $\lambda(F(x))$ are negative or equivalently, the maximal eigenvalue $\lambda_{\max}(F(x)) < 0$.

Normally the variable x in (2.30), which we are interested in, is composed of one or more matrices whose columns have been stacked as a vector [49]. That is, $F(x) = F(X_1, X_2, \dots, X_p)$ where $X_i \in \mathbb{R}^{q_i \times r_i}$ is a matrix, $\sum_{i=1}^p q_i \times r_i = n$, and the columns of all the matrix variables are stacked up to form a single vector variable. Hence, (2.30) can be modified further as [49]:

$$F(X_1, X_2, \dots, X_p) = F_0 + G_1 X_1 H_1 + G_2 X_2 H_2 \dots + G_p X_p H_p \quad (2.31)$$

$$= F_0 + \sum_{i=1}^p G_i X_i H_i < 0 \quad (2.32)$$

where $F_0, G_i \in \mathbb{R}^{m \times q_i}$, $H_i \in \mathbb{R}^{r_i \times m}$ are given matrices and the X_i are the matrix variables which we seek.

Corollary 2.2.10. *Let $\mathfrak{s} := \{x | F(x) < 0\}$ be the set of feasible solutions to the LMI $F(x) < 0$ then \mathfrak{s} is convex.*

Proof. We show that if $x_1, x_2 \in \mathfrak{s}$ then $\{x := \alpha x_1 + (1 - \alpha)x_2\} \in \mathfrak{s}$ for all $\alpha \in (0, 1)$. By Definition 2.2.7, $F(x) = F(\alpha x_1 + (1 - \alpha)x_2) = \alpha F(x_1) + (1 - \alpha)F(x_2)$. In addition, $\alpha > 0$, $(1 - \alpha) > 0$, $F(x_1) < 0$, and $F(x_2) < 0$, it is obvious that $\alpha F(x_1) + (1 - \alpha)F(x_2) < 0$ for all $\alpha \in (0, 1)$, therefore we have $\{x := \alpha x_1 + (1 - \alpha)x_2\} \in \mathfrak{s}$ for all $\alpha \in (0, 1)$ \square

2.2.5 The S-Procedure

The LPV control constraint often come with some quadratic inequalities (or quadratic function) being negative whenever some other quadratic inequalities are all negative. The several quadratic inequalities can be combined into one single inequality (generally with some conservatism) using the S -procedure [25]. Let F_0, \dots, F_p be quadratic functions of the variable $x \in \mathbb{R}^n$:

$$F_i(x) := x^T T_i x + 2u_i^T x + v, \quad i = 0, \dots, p,$$

where $T_i = T_i^T$. Consider

$$F_0(x) \geq 0 \text{ for all } x \text{ such that } F_j(x) \geq 0, \quad j = 1, \dots, p. \quad (2.33)$$

It is obvious that if there exist $\tau_1 \geq 0, \dots, \tau_p \geq 0$ such that for all x ,

$$F_0(x) - \sum_{i=1}^p \tau_i F_i(x) \geq 0 \quad (2.34)$$

then (2.33) is satisfied.

2.2.6 Useful Tools

Lemma 2.2.11. (*Congruence Transformation [51, page 399]*) Given a negative definite symmetric matrix $M = M^T \in \mathbb{R}^{p \times p}$, $M < 0$ then for another real matrix $T \in \mathbb{R}^{p \times q}$ such that $\text{rank}(T^T M T) = \text{rank}(T) = q$, the following inequality holds:

$$T^T M T < 0 \quad (2.35)$$

Note that, based on $\text{rank}(T) = q$, we have $q \in [1, p] \subset \mathbb{N}$.

Lemma 2.2.12. (*Schur complement [44]*) Let a partitioned symmetric matrix

$$N = \begin{bmatrix} P & M \\ M^T & Q \end{bmatrix} \in \mathbb{R}^{p \times p},$$

N is a negative definite symmetric matrix if and only if

$$\begin{aligned} Q &< 0 \\ P - M Q^{-1} M^T &< 0 \end{aligned} \quad (2.36)$$

where $P - M Q^{-1} M^T$ is the Schur complement of Q .

Proof. [89] Let a non-singular matrix $T \in \mathbb{R}^{p \times p}$ and $u = Tv$ where

$$T = \begin{bmatrix} I & 0 \\ -Q^{-1} M^T & I \end{bmatrix}, \quad (2.37)$$

We have $u^T N u < 0$ for all nonzero vector $u \in \mathbb{R}^p$ for which this is equivalent to $v^T T^T N T v < 0$ for all nonzero vector $v \in \mathbb{R}^p$. Computing $T^T N T$:

$$T^T N T = \begin{bmatrix} P - M Q^{-1} M^T & 0 \\ 0 & Q \end{bmatrix} \quad (2.38)$$

Obviously, $v^T T^T N T v < 0$ if and only if (2.36) is satisfied. \square

Lemma 2.2.13. [82] Given a pair of positive definite symmetric matrices $(X, Y) \in \mathbb{R}^{p \times p}$. Then there exists matrices $X_2, Y_2 \in \mathbb{R}^{p \times m}$ and $X_3, Y_3 \in \mathbb{R}^{m \times m}$, where m is a positive integer, such that $X_3 = X_3^T$,

$$\begin{bmatrix} X & X_2 \\ X_2^T & X_3 \end{bmatrix} > 0, \text{ and } \begin{bmatrix} X & X_2 \\ X_2^T & X_3 \end{bmatrix}^{-1} = \begin{bmatrix} Y & Y_2 \\ Y_2^T & Y_3 \end{bmatrix} \quad (2.39)$$

if and only if $X - Y^{-1} \geq 0$, and $\text{rank}(X - Y^{-1}) \leq m$.

Lemma 2.2.14. (*Projection lemma [44]*) Given an inequality problem of the form

$$\Psi + Q^T K^T P + P^T K Q < 0 \quad (2.40)$$

where $\Psi \in \mathbb{R}^{m \times m}$ is a symmetric matrix, Q and P are matrices with column dimension m . Let Q_\perp and P_\perp be any matrices whose columns form bases of the null spaces of Q and P respectively; the above problem is solvable for a matrix K of compatible dimensions if and only if

$$Q_\perp^T \Psi Q_\perp < 0, \quad P_\perp^T \Psi P_\perp < 0 \quad (2.41)$$

Lemma 2.2.15. (*Finsler's Lemma [49]*) For some real number $\sigma \in \mathbb{R}$, (2.40) is equivalent to two inequalities

$$\Psi + \sigma Q^T Q < 0 \quad (2.42)$$

$$\Psi + \sigma P^T P < 0 \quad (2.43)$$

2.3 Basic LPV Models

In nature, most real plants are nonlinear. A general nonlinear model can be written in the form.

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) \\ y(t) &= g(x(t), u(t)) \end{aligned} \quad (2.44)$$

where $t \in \mathbb{R}$ is the time, $x(t) = [x_1(t), \dots, x_p(t)]^T \in \mathbb{R}^p$ is the state vector, $u(t) = [u_1(t), \dots, u_m(t)]^T \in \mathbb{R}^m$ is the control input vector, $y(t) = [y_1(t), \dots, y_q(t)]^T \in \mathbb{R}^q$ is the measurement output vector, $f(\cdot)$ and $g(\cdot)$ are continuous mapping functions: $\mathbb{R}^p \times \mathbb{R}^m \rightarrow \mathbb{R}^p$, and $\mathbb{R}^p \times \mathbb{R}^m \rightarrow \mathbb{R}^q$, respectively. Having linearized (2.44) using Jacobian method about one equilibrium point, an LTI model is obtained in which it can be written as a state-space system of the form (see sub-section 3.1.1):

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \quad (2.45)$$

where $A \in \mathbb{R}^{p \times p}$, $B \in \mathbb{R}^{p \times m}$, $C \in \mathbb{R}^{q \times p}$, and $D \in \mathbb{R}^{q \times m}$.

In general, most real plants have more than one equilibrium point. At each equilibrium point, some operating parameters could be selected as time-varying parameters. When the time-varying parameters vary slowly, (2.45) become an LPV model for which it can be written in the form (see sub-section 3.1.1):

$$\begin{aligned} \dot{x}(t) &= A(\theta(t))x(t) + B(\theta(t))u(t) \\ y(t) &= C(\theta(t))x(t) + D(\theta(t))u(t) \end{aligned} \quad (2.46)$$

where $A(\cdot)$, $B(\cdot)$, $C(\cdot)$, and $D(\cdot)$ are known functions of time-varying parameters, $\theta(t) = [\theta_1(t), \dots, \theta_n(t)]^T \in \mathbb{R}^n$, and are continuous mapping matrix functions: $\mathbb{R}^n \rightarrow \mathbb{R}^{p \times p}$, $\mathbb{R}^n \rightarrow \mathbb{R}^{p \times m}$, $\mathbb{R}^n \rightarrow \mathbb{R}^{q \times p}$, and $\mathbb{R}^n \rightarrow \mathbb{R}^{q \times m}$, respectively. The time variation of each of the parameters $\theta(t)$ is not known in advance, but can be measured in real-time and lies in some set bounded by known minimum and maximum possible values, i.e. $\theta_1(t) \in [\underline{\theta}_1, \bar{\theta}_1]$, $\theta_2(t) \in [\underline{\theta}_2, \bar{\theta}_2]$, \dots , $\theta_n(t) \in [\underline{\theta}_n, \bar{\theta}_n]$.

It can be seen that two interesting features of an LPV model are (i) an LPV model can represent nonlinear dynamic characteristics of an original nonlinear model better than an LTI model because it uses the time-varying parameters $\theta(t)$ to capture the dynamic characteristics of the original nonlinear model and (ii) an LPV model is still a linear system, whose state-space descriptions are functions of time-varying parameters $\theta(t)$. Hence the single quadratic or parameter-dependent Lyapunov functions can be used to prove the stability of LPV models.

2.4 Literature Review

Comprehensive overviews of research in gain-scheduled control techniques can be found in [87] and [62]. Although most real plants are nonlinear, they can often be modelled as an LPV plant model [13, 39, 80] or characterized as a linear fractional transformation (LFT) [9, 29, 81, 98] or as a linear fractional representation (LFR) [86, 96] for gain-scheduled control synthesis and analysis purpose.

An LFT (or LFR) gain-scheduled controller is often synthesized via the scaled small-gain theorem [81] or scaled bounded real lemma [9, 96]. Two advantages of the parameter-dependent plant having LFT (or LFR) parameter dependency are (i) the existence of an LFT (or LFR) gain-scheduled controller that is fully characterized by a finite number of LMIs [96, 98], and (ii) its favourable LFT (or LFR) structure offers obvious advantages in reducing computational burden and ease of controller implementation [9, 83]. However, in the LFT (or LFR) formulation, the variations of parameters are allowed to be complex, thus conservatism is introduced when the scheduled parameters are real [10, 60, 83].

An LPV plant model was first introduced by Shamma and Athans [91] whereby its dynamic characteristics vary, following some time-varying parameters whose values are unknown *a priori* but can be measured in real-time and lie in some set bounded by known minimum and maximum possible values. An algebraic manipulation method, e.g. Jacobian linearization [39, 66, 80], state transformation [13, 92], or function substitution [94], etc., is normally used to derive an LPV model from the original nonlinear model. Moreover, in the literature, there are several different varieties of LPV models, e.g. the grid LPV model [39, 66, 100, 101], the affine LPV model [8, 7, 10] (or polytopic LPV model), the tensor-product (TP) convex polytopic model [15, 16, 18], etc., these have been introduced for the analysis and gain-scheduled control synthesis which is usually based on single quadratic Lyapunov function [10, 23] or parameter-dependent Lyapunov functions (e.g. parameter-dependent [8, 39, 100, 101], affine parameter-dependent [45], piecewise-affine parameter-dependent [63, 64], blending parameter-dependent [94], multiple parameter-dependent Lyapunov functions [65, 66], etc.).

A grid LPV model was introduced by Becker and Packard [23], whereby system state-space model matrices are functions of the scheduled parameters at all grid points over the entire parameter spaces and an affine LPV model, introduced by Apkarian et al. [10], the system matrices are known functions and depend affinely on the parameters that vary in a polytope of vertices. The TP convex polytopic model has been recently proposed by Baranyi [15] for transforming a given LPV model, whose system matrices are nonlinearly dependent on the parameters, into a convex polytopic model. It uses a higher order singular value decomposition in order to decompose a given N -dimensional tensor into a full orthonormal system in a special ordering of higher order singular values which express the rank properties of the given LPV model for each element of the parameter vector in the L_2 -norm [15, 16, 18]. Hence, the TP-type convex polytopic model is obtained, where the parameter-dependent weighting functions of the LTI vertice components of the polytopic model

are one-dimensional functions of the elements of the parameter vector [15, 16, 18].

Using single quadratic Lyapunov functions, for both the affine LPV models [10] and the TP convex polytopic models [33] cases, a finite number of LMIs need only be evaluated at all vertices while, for the grid LPV models [99] case, an infinite number of LMIs have to be evaluated at all points over the entire parameter space in order to determine a pair of positive definite symmetric matrices (X, Y) . However, in practice, the symmetric matrices (X, Y) can be determined from a finite number of LMIs by gridding the entire parameter space with a non-dense set of grid points. Having determined the symmetric matrices (X, Y) , a more dense grid points set can be tested with these determined symmetric matrices (X, Y) to check whether the LMIs are satisfied [100, 101]. If not, this process is repeated with a denser grid until the symmetric matrices (X, Y) , that satisfy the LMIs for all points over the entire parameters space, are obtained [64, 100, 101]. Hence, the result of heuristic gridding technique is unreliable and the analysis result is dependent on choosing the gridding points [96]. In addition, for a grid LPV model case, the resulting gain-scheduled controller has high computational on-line complexity at the gain-scheduling level [99] while, for the other two cases, the gain-scheduled controller is constructed as an affine matrix-valued function in the polytopic coordinates of the scheduled parameter [10, 33].

In general, the single quadratic Lyapunov function is more conservative than the parameter-dependent Lyapunov function when the parameters are time-invariant or slowly varying [45]. Moreover, when the parameters have a large variation, the piecewise-affine parameter-dependent [63, 64], blending parameter-dependent [94], and multiple parameter-dependent Lyapunov functions [65, 66] are less conservative than the parameter-dependent Lyapunov function. This is because an LPV model with a large parameter variation can be modeled as a switching linear parameter-varying (SLPV) system that can be discontinuous along the switching surface by dividing the entire parameters spaces into parameters subsets that are small variation regions. Hence, solving LMIs with the parameters subsets, the performance measure (γ) can be improved. Moreover, the sufficient conditions to guarantee the stability of the SLPV systems in terms of the dwell time and the average dwell time have been provided in [102]. However, using parameter-dependent [8, 39, 100, 101], blending parameter-dependent [94], and multiple parameter-dependent Lyapunov functions [65, 66], an infinite number of LMIs have to be evaluated at all points over the entire parameters space. Moreover, the resulting gain-scheduled controller requires more complex on-line computations at the gain-scheduling level.

This page intentionally contains only this sentence.

Chapter 3

LPV Systems Theory

An LPV control technique is motivated by the shortcomings of the conventional gain-scheduling technique. Since Shamma and Athans [91] introduced LPV plant models, the LPV control approach has developed quite rapidly. There are three common approaches that are normally used to derive an LPV model from an original nonlinear model. These are (i) Jacobian linearization [39, 66, 80], (ii) state transformation [13, 92], and (iii) function substitution [94]. Section 3.1 gives a brief overview of these three approaches. Details of three types of LPV models and details of parameter-dependent linear fractional transformation models (LPV/LFT) are also provided in sections 3.2. In addition, the theory involved in stability analysis (i.e. Lyapunov-based stability analysis and small gain theorem) and controller synthesis for LPV Systems (i.e. bounded real lemma, gain-scheduled H_∞ control and gain scheduling via LFT) are presented in sections 3.3 and 3.4, respectively.

3.1 Methods for Deriving LPV models

An LPV model is of great importance for synthesizing an LPV controller. The following sub-sections summarize the common methods for deriving a reliable LPV model from an original nonlinear model.

3.1.1 Jacobian Linearization

Jacobian linearization is a well known method which is usually used to linearize an nonlinear ordinary differential equation about a specific operating condition, called an equilibrium point (or trim point).

Consider an equilibrium point at t_i in the time space of a general nonlinear model of the form (2.44), the equilibrium state vector x_{trim} , equilibrium input vector u_{trim} ,

and equilibrium output vector y_{trim} can be defined as shown below,

$$\begin{aligned} x_{\text{trim}} &= x(t_i) \\ u_{\text{trim}} &= u(t_i) \\ y_{\text{trim}} &= g(x_{\text{trim}}, u_{\text{trim}}) \end{aligned} \quad (3.1)$$

where $f(x_{\text{trim}}, u_{\text{trim}}) = 0$. Consider a small perturbation about $(x_{\text{trim}}, u_{\text{trim}}, y_{\text{trim}})$. The deviation variables can be defined as shown below, in order to measure the difference.

$$\begin{aligned} \delta x(t) &= x(t) - x_{\text{trim}} \\ \delta u(t) &= u(t) - u_{\text{trim}} \end{aligned} \quad (3.2)$$

Substitute $x(t) = x_{\text{trim}} + \delta x(t)$ and $u(t) = u_{\text{trim}} + \delta u(t)$ into (2.44), we get

$$\begin{aligned} \dot{x}(t) &= f(x_{\text{trim}} + \delta x(t), u_{\text{trim}} + \delta u(t)) \\ y(t) &= g(x_{\text{trim}} + \delta x(t), u_{\text{trim}} + \delta u(t)) \end{aligned} \quad (3.3)$$

Applying the Taylor expansion to (3.3) gives

$$\begin{aligned} f(x_{\text{trim}} + \delta x(t), u_{\text{trim}} + \delta u(t)) &= f(x_{\text{trim}}, u_{\text{trim}}) + \delta x(t) \left. \frac{\partial f(x(t), u(t))}{\partial x(t)} \right|_{\substack{x(t)=x_{\text{trim}} \\ u(t)=u_{\text{trim}}}} \\ &\quad + \delta u(t) \left. \frac{\partial f(x(t), u(t))}{\partial u(t)} \right|_{\substack{x(t)=x_{\text{trim}} \\ u(t)=u_{\text{trim}}}} + \frac{1}{2!} \left\{ \delta x^2(t) \frac{\partial^2 f(x(t), u(t))}{\partial x(t)^2} \right. \\ &\quad + 2\delta x(t) \frac{\partial f(x(t), u(t))}{\partial x(t)} \delta u(t) \frac{\partial f(x(t), u(t))}{\partial u(t)} \\ &\quad \left. + \delta u^2(t) \frac{\partial^2 f(x(t), u(t))}{\partial u(t)^2} \right\} \Bigg|_{\substack{x(t)=x_{\text{trim}} \\ u(t)=u_{\text{trim}}}} + \dots \\ g(x_{\text{trim}} + \delta x(t), u_{\text{trim}} + \delta u(t)) &= g(x_{\text{trim}}, u_{\text{trim}}) + \delta x(t) \left. \frac{\partial g(x(t), u(t))}{\partial x(t)} \right|_{\substack{x(t)=x_{\text{trim}} \\ u(t)=u_{\text{trim}}}} \\ &\quad + \delta u(t) \left. \frac{\partial g(x(t), u(t))}{\partial u(t)} \right|_{\substack{x(t)=x_{\text{trim}} \\ u(t)=u_{\text{trim}}}} + \frac{1}{2!} \left\{ \delta x^2(t) \frac{\partial^2 g(x(t), u(t))}{\partial x(t)^2} \right. \\ &\quad + 2\delta x(t) \frac{\partial g(x(t), u(t))}{\partial x(t)} \delta u(t) \frac{\partial g(x(t), u(t))}{\partial u(t)} \\ &\quad \left. + \delta u^2(t) \frac{\partial^2 g(x(t), u(t))}{\partial u(t)^2} \right\} \Bigg|_{\substack{x(t)=x_{\text{trim}} \\ u(t)=u_{\text{trim}}}} + \dots \end{aligned} \quad (3.4)$$

By neglecting second and higher order terms, approximations of (3.4) are

$$\begin{aligned}
 f(x_{\text{trim}} + \delta x(t), u_{\text{trim}} + \delta u(t)) &= f(x_{\text{trim}}, u_{\text{trim}}) + \delta x(t) \left. \frac{\partial f(x(t), u(t))}{\partial x(t)} \right|_{\substack{x(t)=x_{\text{trim}} \\ u(t)=u_{\text{trim}}}} \\
 &\quad + \delta u(t) \left. \frac{\partial f(x(t), u(t))}{\partial u(t)} \right|_{\substack{x(t)=x_{\text{trim}} \\ u(t)=u_{\text{trim}}}} \\
 g(x_{\text{trim}} + \delta x(t), u_{\text{trim}} + \delta u(t)) &= g(x_{\text{trim}}, u_{\text{trim}}) + \delta x(t) \left. \frac{\partial g(x(t), u(t))}{\partial x(t)} \right|_{\substack{x(t)=x_{\text{trim}} \\ u(t)=u_{\text{trim}}}} \\
 &\quad + \delta u(t) \left. \frac{\partial g(x(t), u(t))}{\partial u(t)} \right|_{\substack{x(t)=x_{\text{trim}} \\ u(t)=u_{\text{trim}}}} \quad (3.5)
 \end{aligned}$$

Equation (3.5) can be rewritten as

$$\begin{aligned}
 \dot{x}(t) &= \delta x(t) \left. \frac{\partial f(x(t), u(t))}{\partial x(t)} \right|_{\substack{x(t)=x_{\text{trim}} \\ u(t)=u_{\text{trim}}}} + \delta u(t) \left. \frac{\partial f(x(t), u(t))}{\partial u(t)} \right|_{\substack{x(t)=x_{\text{trim}} \\ u(t)=u_{\text{trim}}}} \\
 y(t) - y_{\text{trim}} &= \delta x(t) \left. \frac{\partial g(x(t), u(t))}{\partial x(t)} \right|_{\substack{x(t)=x_{\text{trim}} \\ u(t)=u_{\text{trim}}}} + \delta u(t) \left. \frac{\partial g(x(t), u(t))}{\partial u(t)} \right|_{\substack{x(t)=x_{\text{trim}} \\ u(t)=u_{\text{trim}}}} \quad (3.6)
 \end{aligned}$$

Finally, (3.6) can be rewritten in a standard state-space of the form (LTI model).

$$\begin{aligned}
 \delta \dot{x}(t) &= A \delta x(t) + B \delta u(t) \\
 \delta y(t) &= C \delta x(t) + D \delta u(t) \quad (3.7)
 \end{aligned}$$

where $\delta \dot{x}(t) = \dot{x}(t)$, $\delta x(t) = x(t) - x_{\text{trim}}$, $\delta u(t) = u(t) - u_{\text{trim}}$, $\delta y(t) = y(t) - y_{\text{trim}}$, $(x_{\text{trim}}, u_{\text{trim}}, y_{\text{trim}})$ are at an equilibrium point, and

$$\begin{aligned}
 A &= \left. \frac{\partial f(x(t), u(t))}{\partial x(t)} \right|_{\substack{x(t)=x_{\text{trim}} \\ u(t)=u_{\text{trim}}}} & B &= \left. \frac{\partial f(x(t), u(t))}{\partial u(t)} \right|_{\substack{x(t)=x_{\text{trim}} \\ u(t)=u_{\text{trim}}}} \\
 C &= \left. \frac{\partial g(x(t), u(t))}{\partial x(t)} \right|_{\substack{x(t)=x_{\text{trim}} \\ u(t)=u_{\text{trim}}}} & D &= \left. \frac{\partial g(x(t), u(t))}{\partial u(t)} \right|_{\substack{x(t)=x_{\text{trim}} \\ u(t)=u_{\text{trim}}}}
 \end{aligned}$$

It is noted that Jacobian linearization gives a single LTI model about a given trim point (a small perturbation condition). However, most actual plants have more than one trim point. The set of trim points typically varies, following the operating condition. At each trim point, some operating parameters could be selected as time-varying parameters, e.g. Mach number, altitude, dynamic pressure, or angle of attack in case of aircraft. When the time-varying parameters are fixed, (3.7) is an LTI model. However, when the time-varying parameters vary slowly over the entire parameters space, (3.7) becomes an LPV model.

3.1.2 State Transformation

Shamma and Cloutier [92] proposed the state transformation method to derive a quasilinear parameter varying (quasi-LPV) of a missile nonlinear model. An LPV model is said to be a quasi-LPV model if its state vector can be partitioned into a scheduling state vector and a non-scheduling state vector.

In many applications (e.g. turbofan engine [13], missile [92], etc.), the nonlinear model of (2.44) can be rewritten as:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} A_{11}(x_1(t)) & A_{12}(x_1(t)) \\ A_{21}(x_1(t)) & A_{22}(x_1(t)) \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} B_1(x_1(t)) \\ B_2(x_1(t)) \end{bmatrix} u(t) + \begin{bmatrix} K_1(x_1(t)) \\ K_2(x_1(t)) \end{bmatrix} \quad (3.8)$$

where $x_1(t) = [x_{1_1}(t), \dots, x_{1_{p_1}}(t)]^T \in \mathbb{R}^{p_1}$ is the scheduling state vector and is also the time-varying parameters, $x_2(t) = [x_{2_1}(t), \dots, x_{2_{p_2}}(t)]^T \in \mathbb{R}^{p_2}$ is the non-scheduling state vector where $p = p_1 + p_2$. $A_{11}(\cdot)$, $A_{12}(\cdot)$, $A_{21}(\cdot)$, $A_{22}(\cdot)$, $B_1(\cdot)$, $B_2(\cdot)$, $K_1(\cdot)$, and $K_2(\cdot)$ are continuous mapping functions: $\mathbb{R}^{p_1} \rightarrow \mathbb{R}^{p_1 \times p_1}$, $\mathbb{R}^{p_1} \rightarrow \mathbb{R}^{p_1 \times p_2}$, $\mathbb{R}^{p_1} \rightarrow \mathbb{R}^{p_2 \times p_1}$, $\mathbb{R}^{p_1} \rightarrow \mathbb{R}^{p_2 \times p_2}$, $\mathbb{R}^{p_1} \rightarrow \mathbb{R}^{p_1 \times m}$, $\mathbb{R}^{p_1} \rightarrow \mathbb{R}^{p_2 \times m}$, $\mathbb{R}^{p_1} \rightarrow \mathbb{R}^{p_1}$, and $\mathbb{R}^{p_1} \rightarrow \mathbb{R}^{p_2}$, respectively.

Suppose there exist continuously differentiable functions $x_{2\text{eq}}(x_1(t))$ and $u_{\text{eq}}(x_1(t))$ that are continuous mapping functions: $\mathbb{R}^{p_1} \rightarrow \mathbb{R}^{p_2}$ and $\mathbb{R}^{p_1} \rightarrow \mathbb{R}^m$, respectively, such that for every $x_1(t)$,

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} A_{11}(x_1(t)) & A_{12}(x_1(t)) \\ A_{21}(x_1(t)) & A_{22}(x_1(t)) \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_{2\text{eq}}(x_1(t)) \end{bmatrix} + \begin{bmatrix} B_1(x_1(t)) \\ B_2(x_1(t)) \end{bmatrix} u_{\text{eq}}(x_1(t)) + \begin{bmatrix} K_1(x_1(t)) \\ K_2(x_1(t)) \end{bmatrix} \quad (3.9)$$

Subtracting (3.9) from (3.8) obtains

$$\begin{aligned} \dot{x}_1(t) &= A_{12}(x_1(t))[x_2(t) - x_{2\text{eq}}(x_1(t))] + B_1(x_1(t))[u(t) - u_{\text{eq}}(x_1(t))] \\ \dot{x}_2(t) &= A_{22}(x_1(t))[x_2(t) - x_{2\text{eq}}(x_1(t))] + B_2(x_1(t))[u(t) - u_{\text{eq}}(x_1(t))] \end{aligned} \quad (3.10)$$

Differentiating $x_{2\text{eq}}(x_1(t))$ with respect to time t gives

$$\begin{aligned} \frac{d}{dt}x_{2\text{eq}}(x_1(t)) &= \dot{x}_{2\text{eq}}(x_1(t)) = \frac{\partial x_{2\text{eq}}(x_1(t))}{\partial x_1(t)} \dot{x}_1(t) \\ &= \frac{\partial x_{2\text{eq}}(x_1(t))}{\partial x_1(t)} \{A_{12}(x_1(t))[x_2(t) - x_{2\text{eq}}(x_1(t))] + B_1(x_1(t))[u(t) - u_{\text{eq}}(x_1(t))]\} \end{aligned} \quad (3.11)$$

Subtracting (3.11) from (3.10) obtains

$$\begin{aligned} \dot{x}_2(t) - \dot{x}_{2\text{eq}}(x_1(t)) &= [A_{22}(x_1(t)) - \frac{\partial x_{2\text{eq}}(x_1(t))}{\partial x_1(t)} A_{12}(x_1(t))][x_2(t) - x_{2\text{eq}}(x_1(t))] \\ &\quad + [B_2(x_1(t)) - \frac{\partial x_{2\text{eq}}(x_1(t))}{\partial x_1(t)} B_1(x_1(t))][u(t) - u_{\text{eq}}(x_1(t))] \end{aligned} \quad (3.12)$$

Hence, (3.10) can be rewritten as a quasi-LPV model in a state-space equation of the form

$$\begin{bmatrix} \dot{\xi}_1(t) \\ \dot{\xi}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & A_{12}(\xi_1(t)) \\ 0 & \tilde{A}_{22}(\xi_1(t)) \end{bmatrix} \begin{bmatrix} \xi_1(t) \\ \xi_2(t) \end{bmatrix} + \begin{bmatrix} B_1(\xi_1(t)) \\ \tilde{B}_2(\xi_1(t)) \end{bmatrix} v(t) \quad (3.13)$$

where $\xi_1(t) = x_1(t)$, $\xi_2(t) = x_2(t) - x_{2\text{eq}}(x_1(t))$, $v(t) = u(t) - u_{\text{eq}}(x_1(t))$ and

$$\begin{aligned} \tilde{A}_{22}(\xi_1(t)) &= A_{22}(\xi_1(t)) - \frac{\partial x_{2\text{eq}}(\xi_1(t))}{\partial \xi_1(t)} A_{12}(\xi_1(t)) \\ \tilde{B}_2(\xi_1(t)) &= B_2(\xi_1(t)) - \frac{\partial x_{2\text{eq}}(\xi_1(t))}{\partial \xi_1(t)} B_1(\xi_1(t)) \end{aligned}$$

3.1.3 Function Substitution

Tan et al. [95] have introduced this method to linearize a missile nonlinear model around a single equilibrium point. Unlike Jacobian linearization, the principle of this method is to select a suitable single equilibrium point such that the original nonlinear model can be rewritten in a quasi-LPV model form.

Suppose a nonlinear model in (2.44) can be rewritten as (3.8). Let $x_{1r} = [x_{1r_1}, \dots, x_{1r_{p_1}}]^T \in \mathbb{R}^{p_1}$, $x_{2r} = [x_{2r_1}, \dots, x_{2r_{p_2}}]^T \in \mathbb{R}^{p_2}$, and $u_r = [u_{r_1}, \dots, u_{r_m}]^T \in \mathbb{R}^m$ be a selected equilibrium point. Define $x_1(t) = \tilde{x}_1(t) + x_{1r}$, $x_2(t) = \tilde{x}_2(t) + x_{2r}$, and $u(t) = \tilde{u}(t) + u_r$. Substituting into (3.8), we have

$$\begin{bmatrix} \dot{\tilde{x}}_1(t) \\ \dot{\tilde{x}}_2(t) \end{bmatrix} = \begin{bmatrix} A_{11}(x_1(t)) & A_{12}(x_1(t)) \\ A_{21}(x_1(t)) & A_{22}(x_1(t)) \end{bmatrix} \begin{bmatrix} \tilde{x}_1(t) \\ \tilde{x}_2(t) \end{bmatrix} + \begin{bmatrix} B_1(x_1(t)) \\ B_2(x_1(t)) \end{bmatrix} \tilde{u}(t) + \begin{bmatrix} F_1(x_1(t), x_{1r}, x_{2r}, u_r) \\ F_2(x_1(t), x_{1r}, x_{2r}, u_r) \end{bmatrix} \quad (3.14)$$

where $F_1(\cdot)$ and $F_2(\cdot)$ are continuous mapping functions: $\mathbb{R}^{p_1} \times \mathbb{R}^{p_1} \times \mathbb{R}^{p_2} \times \mathbb{R}^m \rightarrow \mathbb{R}^{p_1}$ and $\mathbb{R}^{p_1} \times \mathbb{R}^{p_1} \times \mathbb{R}^{p_2} \times \mathbb{R}^m \rightarrow \mathbb{R}^{p_2}$, respectively, and

$$\begin{bmatrix} F_1(x_1(t), x_{1r}, x_{2r}, u_r) \\ F_2(x_1(t), x_{1r}, x_{2r}, u_r) \end{bmatrix} = \begin{bmatrix} A_{11}(x_1(t)) & A_{12}(x_1(t)) \\ A_{21}(x_1(t)) & A_{22}(x_1(t)) \end{bmatrix} \begin{bmatrix} x_{1r} \\ x_{2r} \end{bmatrix} + \begin{bmatrix} B_1(x_1(t)) \\ B_2(x_1(t)) \end{bmatrix} u_r + \begin{bmatrix} K_1(x_1(t)) \\ K_2(x_1(t)) \end{bmatrix} \quad (3.15)$$

Given a suitable single equilibrium point (x_{1r}, x_{2r}, u_r) , the goal of this method is to decompose functions $F_1(\cdot)$ and $F_2(\cdot)$ into functions $E_1(\cdot)$ and $E_2(\cdot)$ that are linear in $\tilde{x}_1(t)$ and are continuous mapping functions: $\mathbb{R}^{p_1} \rightarrow \mathbb{R}^{p_1 \times p_1}$ and $\mathbb{R}^{p_1} \rightarrow \mathbb{R}^{p_2 \times p_1}$ such that for all $x_1(t)$,

$$\begin{bmatrix} E_1(x_1(t)) & 0 \\ E_2(x_1(t)) & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_1(t) \\ \tilde{x}_2(t) \end{bmatrix} \approx \begin{bmatrix} F_1(x_1(t), x_{1r}, x_{2r}, u_r) \\ F_2(x_1(t), x_{1r}, x_{2r}, u_r) \end{bmatrix} \quad (3.16)$$

The decomposition of functions $E_1(\cdot)$ and $E_2(\cdot)$ can be posed as an optimization problem, for example, as a minimum least-squares problem [58] that minimizes the sum of squared differences between the true functions $(F_1(\cdot), F_2(\cdot))$ and the approximate functions $(E_1(\cdot), E_2(\cdot))$. That is to minimize $\epsilon^T \epsilon / 2$ where

$$\epsilon = \begin{bmatrix} F_1(x_1(t), x_{1r}, x_{2r}, u_r) - E_1(x_1(t)) \tilde{x}_1(t) \\ F_2(x_1(t), x_{1r}, x_{2r}, u_r) - E_2(x_1(t)) \tilde{x}_1(t) \end{bmatrix} \quad (3.17)$$

Table 3.1: Comparison between three methods for deriving LPV models

Method	Jacobian linearization
Advantages	(i) Applicable to a general class of nonlinear models. (ii) Simple structure model.
Disadvantages	(i) Only accurately represent the original nonlinear dynamics about the neighborhood of a set of equilibrium points. (ii) The time-varying parameters must vary slowly.
Method	State Transformation
Advantages	(i) Exactly represent the original nonlinear dynamics.
Disadvantages	(i) Applicable to the only special form of nonlinear models (ii) Complex structure model.
Method	Function Substitution
Advantages	(i) Can represent the original nonlinear dynamics over a non-trim region.
Disadvantages	(i) Applicable to the only special form of nonlinear models (ii) How to find a suitable equilibrium point is still not established. (iii) Complex structure model.

For example, Shin et al. [94] proposed the function substitution method to derive a quasi-LPV model of the F-16 aircraft that can cover the aircraft non-trim region.

Having determined functions $E_1(\cdot)$ and $E_2(\cdot)$ and substituted them back into (3.14), we have a quasi-LPV model as

$$\begin{bmatrix} \dot{\tilde{x}}_1(t) \\ \dot{\tilde{x}}_2(t) \end{bmatrix} = \begin{bmatrix} A_{11}(x_1(t)) + E_1(x_1(t)) & A_{12}(x_1(t)) \\ A_{21}(x_1(t)) + E_2(x_1(t)) & A_{22}(x_1(t)) \end{bmatrix} \begin{bmatrix} \tilde{x}_1(t) \\ \tilde{x}_2(t) \end{bmatrix} + \begin{bmatrix} B_1(x_1(t)) \\ B_2(x_1(t)) \end{bmatrix} \tilde{u}(t) \quad (3.18)$$

3.1.4 Summary of Three Derivation Methods

A summary of the advantage and disadvantage of these three methods, i.e. Jacobian linearization, state transformation, and function substitution, is presented in Table 3.1. It can be seen from Table 3.1 that both state transformation-based and function substitution-based LPV models can represent nonlinear dynamic characteristics of the original nonlinear model better than Jacobian linearization-based LPV model. However, the state transformation approach assumes that there exist equilibrium function for the non-scheduling states, $x_{2eq}(x_1(t))$, and the control input, $u_{eq}(x_1(t))$. Unfortunately, this is not always the case. In addition, the function substitution approach lacks a theoretical method of obtaining a suitable equilibrium point.

3.2 LPV Models for Controller Synthesis

Having derived a parameter-dependent model from an original nonlinear model, the resulting model can often be grouped into three LPV model types for gain-scheduled control analysis and synthesis purposes. The following subsections outline these three model types that include the grid LPV model [39, 66, 100, 101], the affine LPV model [7, 8, 10] (or polytopic LPV model), and the tensor-product (TP) convex polytopic model [15, 16, 18].

3.2.1 Grid LPV Model

A resulting model, that is derived from an original nonlinear model using an algebraic manipulation method, is often nonlinearly dependent on the time-varying parameters in which an infinite number of LMIs is obtained when an LPV controller is synthesized based on this nonlinearly parameter-dependent LPV model. To obtain a finite number of LMIs, Becker and Packard [23] have introduced a grid LPV model which can be written as a state-space system of the form (2.46) where the system matrix, $S : \mathbb{R}^n \rightarrow \mathbb{R}^{(p+q) \times (p+m)}$, is a function of the scheduled parameters at all grid points over the entire parameter spaces and can be written as

$$S(\theta(t)) = \begin{bmatrix} A(\theta(t)) & B(\theta(t)) \\ C(\theta(t)) & D(\theta(t)) \end{bmatrix} \quad (3.19)$$

3.2.2 Affine LPV Model

An alternative LPV model that yields a finite number of LMIs when synthesizing an LPV controller is an affine LPV model that has been introduced by Apkarian et al. [10]. The affine LPV model can also be written as a state-space system of the form (2.46), but where the system matrix, $S(\theta(t))$, is assumed to depend affinely on the time-varying parameters. That is

$$S(\theta(t)) = S_0 + \theta_1(t)S_1 + \cdots + \theta_n(t)S_n \quad (3.20)$$

where $\theta(t)$ lies in a polytope Θ , $\theta(t) \in \Theta$, $\Theta = [\underline{\theta}_1, \bar{\theta}_1] \times [\underline{\theta}_2, \bar{\theta}_2] \times \cdots \times [\underline{\theta}_n, \bar{\theta}_n]$, n is the total number of $\theta(t)$, and

$$S_i = \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}, \quad i = 0, \dots, n \quad (3.21)$$

and S_0, S_1, \dots, S_n are known fixed matrices. The system matrix, $S(\theta(t))$, can also be written as a convex combination of the matrix vertices (see definition 2.2.8) as

$$\begin{aligned} S(\theta(t)) &= Co \left\{ \hat{S}_1, \hat{S}_2, \dots, \hat{S}_r \right\} = \alpha_1 \hat{S}_1 + \alpha_2 \hat{S}_2 + \cdots + \alpha_r \hat{S}_r \\ &= \sum_{j=1}^r \alpha_j \hat{S}_j, \quad \hat{S}_j = \begin{bmatrix} \hat{A}_j & \hat{B}_j \\ \hat{C}_j & \hat{D}_j \end{bmatrix} \end{aligned} \quad (3.22)$$

where $r = 2^n$ is the total number of vertices, \hat{S}_j are the LTI system matrices at each vertex, $\alpha_j \in [0, 1]$, and $\sum_{j=1}^r \alpha_j = 1$. S_i , $i = 0, \dots, n$ map to \hat{S}_j , $j = 1, \dots, r$ as

$$\begin{bmatrix} \hat{S}_1 \\ \hat{S}_2 \\ \hat{S}_3 \\ \vdots \\ \hat{S}_r \end{bmatrix} = \begin{bmatrix} 1 & \underline{\theta}_1 & \underline{\theta}_2 & \dots & \underline{\theta}_{n-1} & \underline{\theta}_n \\ 1 & \underline{\theta}_1 & \underline{\theta}_2 & \dots & \underline{\theta}_{n-1} & \underline{\theta}_n \\ 1 & \underline{\theta}_1 & \underline{\theta}_2 & \dots & \underline{\theta}_{n-1} & \underline{\theta}_n \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \bar{\theta}_1 & \bar{\theta}_2 & \dots & \bar{\theta}_{n-1} & \bar{\theta}_n \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ \vdots \\ S_n \end{bmatrix} \quad (3.23)$$

Following Pellanda et al. [83, Algorithm 3.1], in order to compute α_i , we first compute the normalized co-ordinates

$$\alpha_{\theta_i} = \frac{\bar{\theta}_i - \theta_i(t)}{\bar{\theta}_i - \underline{\theta}_i}, \quad i = 1, \dots, n \quad (3.24)$$

Then, for each vertex Θ_j , $j = 1, \dots, r$, the corresponding polytopic co-ordinates are calculated by

$$\alpha_j = \prod_{i=1}^n \tilde{\alpha}_{\theta_i}, \quad \tilde{\alpha}_{\theta_i} = \begin{cases} \alpha_{\theta_i}, & \text{if } \underline{\theta}_i \text{ is a co-ordinate of } \Theta_j; \\ 1 - \alpha_{\theta_i}, & \text{if } \bar{\theta}_i \text{ is a co-ordinate of } \Theta_j. \end{cases} \quad (3.25)$$

3.2.3 TP Convex Polytopic Model

Baranyi [15] has introduced a TP model transformation to transform a given parameter-dependent model to a convex polytopic model. The resulting model is called a TP polytopic model that also yields a finite number of LMIs when synthesizing an LPV controller. The TP model transformation is an automatically executable numerical method and has three key steps. The first step is the discretization of the given system matrix over a huge number of points. The discretized points are defined by a dense hyper-rectangular grid of the parameters. The second step extracts the LTI vertex systems from the discretized systems using a higher order singular value decomposition (HOSVD) to decompose a given n -dimensional tensor into a full orthonormal system in a special ordering of higher order singular values which express the rank properties of the given parameter-dependent model for each element of the parameter vector in the L_2 -norm. The third step defines the continuous weighting functions to the LTI vertex systems.

This sub-section only outlines a brief overview of the TP polytopic model; for further details refer to [15, 16, 18]. Although the TP polytopic model cannot be written in an affine combination because it is not a type of affine LPV model, following [15, 16, 18], the TP polytopic model can also be written in a convex combination of the matrix vertices as

$$\begin{pmatrix} \dot{x}(t) \\ y(t) \end{pmatrix} = \sum_{a=1}^R w_a(\theta(t)) S_a \begin{pmatrix} x(t) \\ u(t) \end{pmatrix} \quad (3.26)$$

where $R = I_1 \times I_2 \times \dots \times I_n = \prod_n I_n$ is the total number of vertices, n is the total number of the parameters vector, I_i , $i = 1, \dots, n$, is the index upper bounds of the

weighting functions used in the i -th dimension of the parameter vector, $\theta(t) \in \Theta$, and

$$\begin{aligned} w_a(\theta(t)) &= w_{1,j_1}(\theta_1(t)) \times w_{2,j_2}(\theta_2(t)) \times \cdots \times w_{n,j_n}(\theta_n(t)) \\ &= \prod_n w_{i,j}(\theta_i(t)), \quad j = 1, \dots, I_i, \quad i = 1, \dots, n \end{aligned} \quad (3.27)$$

$$S_a = S_{i_1 i_2 \dots i_n} = \begin{pmatrix} A_a & B_a \\ C_a & D_a \end{pmatrix} \in \mathbb{R}^{(p+q) \times (p+m)} \quad (3.28)$$

$w_a : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuous weighting function, $w_{i,j}(\theta_i(t))$ is the j -th one variable weighting function defined on the i -th dimension of Θ , and $\theta_i(t)$ is the i -th element of the parameter vector $\theta(t)$, S_a is an LTI vertex systems, $a =$ ordering $(i_1 i_2 \dots i_n)$.

The goal of the TP model transformation is to determine the LTI vertex systems, S_a , and the weighting functions, $w_{i,j}(\theta_i(t))$, such that the system matrix, $S(\theta(t))$, in (3.19) is given for any grid points over the entire parameter spaces and can be expressed as the combination of the vertex system matrices, S_a , and the weighting functions, $w_a(\theta(t))$, which are nonlinearly dependent on the scheduled parameters,

$$S(\theta(t)) \approx \sum_{a=1}^R w_a(\theta(t)) S_a, \quad \forall \theta(t) \in \Theta \quad (3.29)$$

$$\left\| S(\theta(t)) - \sum_{a=1}^R w_a(\theta(t)) S_a \right\| \leq \epsilon \quad (3.30)$$

Here, ϵ symbolizes the approximation error. Furthermore, the convex combination of the LTI vertex systems is ensured by the condition,

$$\forall i \in [1, n], j \in [1, I_i], \theta_i(t) : w_{i,j}(\theta_i(t)) \in [0, 1] \quad (3.31)$$

$$\forall i \in [1, n], \theta_i(t) : \sum_{j=1}^{I_i} w_{i,j}(\theta_i(t)) = 1 \quad (3.32)$$

$$\forall a \in [1, R], \theta(t) : w_a(\theta(t)) \in [0, 1] \quad (3.33)$$

$$\forall \theta(t) : \sum_{a=1}^R w_a(\theta(t)) = 1 \quad (3.34)$$

Hence, $S(\theta(t))$ is within the convex hull of the LTI vertex systems S_a for $\forall \theta(t) \in \Theta$.

For convenience, in the following sections, we will henceforth often drop the dependence on t .

3.2.4 LPV/LFT Models

An alternative approach for gain-scheduled control analysis and synthesis is to characterize a nonlinear model as a parameter-dependent linear fractional transformation

model (LPV/LFT) from which the LFT gain-scheduled controller is often synthesized via the scaled small-gain theorem [81] or scaled bounded real lemma [9, 96].

Assume the system (2.46) depends affinely on the scheduled parameters θ (see [81] for other cases). We have

$$\begin{aligned} A(\theta) &= A_0 + \theta_1 A_1 + \theta_2 A_2 + \cdots + \theta_n A_n \\ B(\theta) &= B_0 + \theta_1 B_1 + \theta_2 B_2 + \cdots + \theta_n B_n \\ C(\theta) &= C_0 + \theta_1 C_1 + \theta_2 C_2 + \cdots + \theta_n C_n \\ D(\theta) &= D_0 + \theta_1 D_1 + \theta_2 D_2 + \cdots + \theta_n D_n \end{aligned} \quad (3.35)$$

Define normalized time-varying parameters $\tilde{\theta}_i \in [-1, 1]$, $i = 1, \dots, n$ as [83]

$$\tilde{\theta}_i = \frac{\theta_i - T_i}{S_i} \quad (3.36)$$

where

$$T_i = \frac{\bar{\theta}_i + \underline{\theta}_i}{2} \quad (3.37)$$

$$S_i = \frac{\bar{\theta}_i - \underline{\theta}_i}{2} \quad (3.38)$$

Substitute (3.36) into (3.35), we get

$$\begin{aligned} A(\theta) &= [A_0 + T_1 A_1 + \cdots + T_n A_n] + \tilde{\theta}_1 (S_1 A_1) + \cdots + \tilde{\theta}_n (S_n A_n) \\ B(\theta) &= [B_0 + T_1 B_1 + \cdots + T_n B_n] + \tilde{\theta}_1 (S_1 B_1) + \cdots + \tilde{\theta}_n (S_n B_n) \\ C(\theta) &= [C_0 + T_1 C_1 + \cdots + T_n C_n] + \tilde{\theta}_1 (S_1 C_1) + \cdots + \tilde{\theta}_n (S_n C_n) \\ D(\theta) &= [D_0 + T_1 D_1 + \cdots + T_n D_n] + \tilde{\theta}_1 (S_1 D_1) + \cdots + \tilde{\theta}_n (S_n D_n) \end{aligned} \quad (3.39)$$

Based on the LFT technique [104], the scheduled parameters θ in (2.46) can be separated from the system state-space model matrices (3.39) as

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ z_{\theta_1} \\ z_{\theta_2} \\ \vdots \\ z_{\theta_n} \\ y \end{bmatrix} &= \begin{bmatrix} A & B_{\theta_1} & B_{\theta_2} & \cdots & B_{\theta_n} & B \\ C_{\theta_1} & D_{\theta\theta_{11}} & D_{\theta\theta_{12}} & \cdots & D_{\theta\theta_{1n}} & D_{\theta 1_1} \\ C_{\theta_2} & D_{\theta\theta_{21}} & D_{\theta\theta_{22}} & \cdots & D_{\theta\theta_{2n}} & D_{\theta 1_2} \\ \vdots & & & & & \\ C_{\theta_n} & D_{\theta\theta_{n1}} & D_{\theta\theta_{n2}} & \cdots & D_{\theta\theta_{nn}} & D_{\theta 1_n} \\ C & D_{1\theta_1} & D_{1\theta_2} & \cdots & D_{1\theta_n} & D \end{bmatrix} \begin{bmatrix} x \\ w_{\theta_1} \\ w_{\theta_2} \\ \vdots \\ w_{\theta_n} \\ u \end{bmatrix} \\ \begin{bmatrix} w_{\theta_1} \\ w_{\theta_2} \\ \vdots \\ w_{\theta_n} \end{bmatrix} &= \begin{bmatrix} \tilde{\theta}_1 I_{s_1} & 0 & \cdots & 0 \\ 0 & \tilde{\theta}_2 I_{s_2} & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & \tilde{\theta}_n I_{s_n} \end{bmatrix} \begin{bmatrix} z_{\theta_1} \\ z_{\theta_2} \\ \vdots \\ z_{\theta_n} \end{bmatrix} \end{aligned} \quad (3.40)$$

where $w_{\theta_i}, z_{\theta_i} \in \mathbb{R}^{s_i}$ and

$$\begin{aligned}
 A &= A_0 + \sum_{i=1}^n T_i A_i \\
 B &= B_0 + \sum_{i=1}^n T_i B_i \\
 C &= C_0 + \sum_{i=1}^n T_i C_i \\
 D &= C_0 + \sum_{i=1}^n T_i D_i \\
 \begin{bmatrix} S_i A_i & S_i B_i \\ S_i C_i & S_i D_i \end{bmatrix} &= \begin{bmatrix} B_{\theta_i} \\ D_{1\theta_i} \end{bmatrix} \begin{bmatrix} C_{\theta_i} & D_{\theta 1_i} \end{bmatrix}
 \end{aligned} \tag{3.41}$$

Note that, $D_{\theta\theta_{ii}}$ is introduced in order that (3.40) is in a general state-space equation form. With notation

$$\begin{aligned}
 w_{\theta} &= [w_{\theta_1}^T \ w_{\theta_2}^T \ \cdots \ w_{\theta_n}^T]^T \\
 z_{\theta} &= [z_{\theta_1}^T \ z_{\theta_2}^T \ \cdots \ z_{\theta_n}^T]^T \\
 B_{\theta} &= [B_{\theta_1} \ B_{\theta_2} \ \cdots \ B_{\theta_n}] \\
 C_{\theta} &= [C_{\theta_1}^T \ C_{\theta_2}^T \ \cdots \ C_{\theta_n}^T]^T \\
 D_{\theta\theta} &= \begin{bmatrix} D_{\theta\theta_{11}} & D_{\theta\theta_{12}} & \cdots & D_{\theta\theta_{1n}} \\ D_{\theta\theta_{21}} & D_{\theta\theta_{22}} & \cdots & D_{\theta\theta_{2n}} \\ \vdots & & & \\ D_{\theta\theta_{n1}} & D_{\theta\theta_{n2}} & \cdots & D_{\theta\theta_{nn}} \end{bmatrix} \\
 D_{\theta 1} &= [D_{\theta 1_1}^T \ D_{\theta 1_2}^T \ \cdots \ D_{\theta 1_n}^T]^T \\
 D_{1\theta} &= [D_{1\theta_1} \ D_{1\theta_2} \ \cdots \ D_{1\theta_n}]
 \end{aligned} \tag{3.42}$$

Equation (3.40) can be rewritten as

$$\begin{aligned}
 \dot{x} &= Ax + B_{\theta} w_{\theta} + Bu \\
 z_{\theta} &= C_{\theta} x + D_{\theta\theta} w_{\theta} + D_{\theta 1} u \\
 y &= Cx + D_{1\theta} w_{\theta} + Du \\
 w_{\theta} &= \hat{\theta} z_{\theta}
 \end{aligned} \tag{3.43}$$

where $w_{\theta}, z_{\theta} \in \mathbb{R}^s$, $s = s_1 + s_2 + \cdots + s_n$, $\hat{\theta} = \text{diag}(\tilde{\theta}_1 I_{s_1}, \tilde{\theta}_2 I_{s_2}, \dots, \tilde{\theta}_n I_{s_n})$, and $\|\hat{\theta}\| \leq 1$. The system (3.43) is called a parameter-dependent linear fractional transformation model (LPV/LFT).

3.3 Stability Analysis of LPV Systems

Robust stability analysis is one of the most important issues in control-systems design because it is a useful tool for control engineers to validate and guarantee the

stability property of the closed-loop system in the presence of perturbations and uncertainty in the parameters of the system plant. The following lemma is required.

Lemma 3.3.1. [34] *Given a symmetric matrix polytope, $M(\theta(t)) \in \mathbb{R}^{p \times p}$, for which $M(\theta(t)) = \sum_{i=1}^m \alpha_i M_i$, where α_i is determined using (3.24) and (3.25), is a negative definite symmetric matrix for all possible parameter trajectories, $M(\theta(t)) < 0$, $\forall \theta \in \Theta$, if and only if $M_i < 0$, $i = 1, \dots, m$.*

Proof. Sufficiency: Since $\alpha_i \in [0, 1]$ for $i = 1, \dots, m$ and $\sum_{i=1}^m \alpha_i = 1$ then there is always at least one i such that $\alpha_i > 0$. Thus $M_i < 0$ for all i implies $\sum_{i=1}^m \alpha_i M_i < 0$ and hence $M(\theta) < 0$ for all $\theta \in \Theta$.

Necessity: From (3.24) and (3.25), for all j there exists a $\theta \in \Theta$ such that there is an $\alpha_j = 1$ and $\alpha_i = 0$ for $i = 1, \dots, m$, $i \neq j$. Hence for all j there exists a $\theta \in \Theta$ such that $M(\theta) = M_j$ and so it is necessary that $M_j < 0$ for all j .

□

3.3.1 Robustness Analysis using SQLF

Consider the state-trajectories of system (2.46) with the control input vector u identically zero.

$$\dot{x} = A(\theta)x \quad (3.44)$$

Definition 3.3.2. (Quadratic Stability [5, 19, 23, 26, 96]) *The system (3.44) is said to be quadratically stable if there exists a quadratic Lyapunov function $V(x) = x^T P x$ whose derivative is negative, $d/dt(V(x)) < 0$, along all state trajectories.*

Note that, $d/dt(V(x)) = x^T [A^T(\theta)P + PA(\theta)]x$. The above definition is equivalent to the following proposition.

Proposition 3.3.3. *The system (3.44) is quadratically stable whenever there exists a positive definite symmetric matrix $P \in \mathbb{R}^{p \times p}$ such that the following LMI conditions hold*

$$P > 0 \quad (3.45)$$

$$A(\theta)^T P + PA(\theta) < 0, \forall \theta \in \Theta \quad (3.46)$$

Obviously, an inequality (3.46) yields an infinite number of LMIs. However, in practice, a finite number of LMIs can be obtained by gridding the entire parameter space with non-dense set of grid points (a grid LPV model approach) [39, 66, 100, 101]. An alternative approach [33] is to transform the system (3.44), that is nonlinearly dependent on θ , into a TP convex polytopic model using the TP model transformation [15, 16, 18]. Hence, $A(\theta)$ can be written as a convex combination of the matrix vertices in a similar manner to $S(\theta)$ in (3.29) as

$$A(\theta) = w_1(\theta)A_1 + w_2(\theta)A_2 + \dots + w_R(\theta)A_R \quad (3.47)$$

where $w_i(\theta) \in [0, 1]$ and $\sum_{i=1}^R w_i(\theta) = 1$. Substituting (3.47) into (3.46), we get

$$w_1(\theta)[A_1^T P + P A_1] + w_2(\theta)[A_2^T P + P A_2] + \cdots + w_R(\theta)[A_R^T P + P A_R] < 0, \forall \theta \in \Theta \quad (3.48)$$

By Lemma 3.3.1, solving the above inequality for a positive definite symmetric matrix P need only be done at all vertices. Hence, we get the following proposition.

Proposition 3.3.4. *Assume the system (3.44) is a TP convex polytopic system, then the system (3.44) is quadratically stable whenever there exists a positive definite symmetric matrix P such that the following LMI conditions hold*

$$P > 0 \quad (3.49)$$

$$A_a^T P + P A_a < 0, \quad a = 1, 2, \dots, R \quad (3.50)$$

In addition, suppose the system (3.44) depends affinely on the scheduled parameters θ , $A(\theta)$ can be written as a convex combination of the matrix vertices in a similar manner to $S(\theta)$ in (3.22) as

$$A(\theta) = \alpha_1 \hat{A}_1 + \alpha_2 \hat{A}_2 + \cdots + \alpha_r \hat{A}_r \quad (3.51)$$

Then, Proposition 3.3.4 is also applicable to the affine LPV systems by replacing A_a , $a = 1, 2, \dots, R$ with \hat{A}_j , $j = 1, 2, \dots, r$ in (3.50).

3.3.2 Robustness Analysis using Small Gain Theorem

Assume the system (3.44) depends affinely on θ . Moreover, having separated θ from $A(\theta)$, the system (3.44) can be written in a similar manner to (3.43) as

$$\begin{aligned} \dot{x} &= Ax + B_\theta w_\theta \\ z_\theta &= C_\theta x + D_\theta w_\theta \\ w_\theta &= \hat{\theta} z_\theta \end{aligned} \quad (3.52)$$

where $w_\theta, z_\theta \in \mathbb{R}^s$, $s = s_1 + s_2 + \cdots + s_n$, $\hat{\theta} = \text{diag}(\tilde{\theta}_1 I_{s_1}, \tilde{\theta}_2 I_{s_2}, \dots, \tilde{\theta}_n I_{s_n})$, and $\|\hat{\theta}\| \leq 1$.

Consider the system (3.52), define a transfer function matrix $M(s) = D_\theta + C_\theta(sI - A)^{-1} B_\theta \in \mathbb{C}^{s \times s}$, by the Nyquist and small-gain theorem [104], the system (3.52) is quadratically stable if and only if $I - M(s)\hat{\theta}(s)$ and $I - \hat{\theta}(s)M(s)$ are nonsingular. This is equivalent to the following theorem.

Theorem 3.3.5. *(Small Gain Theorem, [104, Theorem 9.1]) Suppose $\hat{\theta}(s)$ and $M(s)$ are stable and let $\gamma > 0$. Then the interconnected system (3.52) shown in Figure 3.1 is well-posed and internally stable for all $\hat{\theta}(s) \in \mathbb{C}^{s \times s}$ with*

$$\|\hat{\theta}\|_\infty \leq 1/\gamma \quad \text{if and only if} \quad \|M\|_\infty < \gamma$$

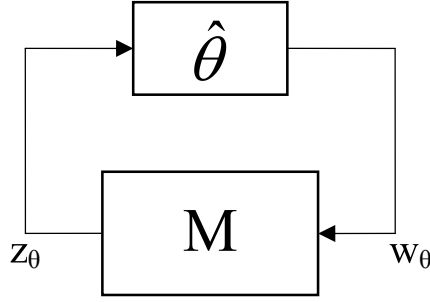


Figure 3.1: Block diagrams structure of an LPV/LFT model (3.52)

Note that $\hat{\theta}$ is a structured uncertainty. By Theorem 3.3.5, the necessary condition to make the system (3.52) quadratically stable is $\|M\|_{\infty} \leq \gamma$ that is equivalent to the induced L_2 -norm (or L_2 -gain) of the operator mapping the disturbance signal w_{θ} into the error signal z_{θ} of the system (3.52) is bounded by γ (i.e. $\|z_{\theta}\|_2 \leq \|w_{\theta}\|_2$ where, for this case, γ must be < 1). Based on single quadratic Lyapunov functions, $\|M\|_{\infty} \leq \gamma$ if and only if there exists $P = P^T$ such that

$$P > 0, \quad \frac{d}{dt}(x^T P x) + z_{\theta}^T z_{\theta} - \gamma^2 w_{\theta}^T w_{\theta} < 0, \quad \forall \theta \in \Theta \quad (3.53)$$

Inequality (3.53) leads to the well-known bounded real lemma[10] inequality

$$\begin{pmatrix} A^T P + P A & P B_{\theta} & C_{\theta}^T \\ B_{\theta}^T P & -\gamma I & D_{\theta}^T \\ C_{\theta} & D_{\theta} & -\gamma I \end{pmatrix} < 0 \quad (3.54)$$

The robust stability requirement is that $\gamma < 1$. However there generally exist an infinite number of the factor matrices pairs (B_{θ}, C_{θ}) in which only some factor matrices pair give $\gamma < 1$. Instead of searching for such a factor matrix pair manually, by introducing a scaling matrix $L^{1/2}$, we can select any factor matrix pair for which γ will always be < 1 if the system (3.52) is quadratically stable and the factor matrix pair can be determined using singular value decomposition, see sub-section 2.2.2. $L^{1/2}$ denotes the unique positive definite square root of $L \in L_{\theta}$. The set of L_{θ} is defined as

$$L_{\theta} = \left\{ L > 0 : L \hat{\theta} = \hat{\theta} L, \forall \theta \in \Theta \right\} \subset \mathbb{R}^{s \times s} \quad (3.55)$$

Therefore, (3.52) can be modified further to

$$\begin{aligned} \dot{x} &= A x + B_{\theta} L^{-\frac{1}{2}} \dot{w}_{\theta} \\ \dot{z}_{\theta} &= L^{\frac{1}{2}} C_{\theta} x + L^{\frac{1}{2}} D_{\theta} L^{-\frac{1}{2}} \dot{w}_{\theta} \\ \dot{w}_{\theta} &= \hat{\theta} \dot{z}_{\theta} \end{aligned} \quad (3.56)$$

where $\dot{z}_{\theta} = L^{1/2} z_{\theta}$ and $w_{\theta} = L^{-1/2} \dot{w}_{\theta}$. With parameters in (3.56), (3.54) becomes a scaled bounded real lemma [9], $\|L^{1/2} M(s) L^{-1/2}\|_{\infty} < \gamma$.

$$\begin{pmatrix} A^T P + P A & P B_{\theta} & C_{\theta}^T \\ B_{\theta}^T P & -\gamma L & D_{\theta}^T \\ C_{\theta} & D_{\theta} & -\gamma L^{-1} \end{pmatrix} < 0 \quad (3.57)$$

Rearranging (3.57) using the Schur complement (Lemma 2.2.12), we get the following proposition.

Proposition 3.3.6. *The system (3.52) with $\|\hat{\theta}\| < 1/\gamma$ is quadratically stable along all possible parameter trajectories, $\forall \theta \in \Theta$, if and only if the following LMI condition hold for some positive definite symmetric and scaling matrices (P, L) :*

$$\begin{pmatrix} A^T P + P A + C_\theta^T L C_\theta & P B_\theta + C_\theta^T L D_\theta \\ B_\theta^T P + D_\theta^T L C_\theta & -\gamma^2 L + D_\theta^T L D_\theta \end{pmatrix} < 0 \quad (3.58)$$

Note that the minimization of γ can be achieved heuristically or by a simple grid search.

3.4 Controller Synthesis for LPV Systems

In the previous section, a sufficient condition to guarantee the stability property of the LPV closed-loop system has been presented in which the analysis conditions can be represented in the form of a finite number of LMIs. Next, we consider the problem of designing a gain-scheduled output feedback H_∞ control with guaranteed L_2 -gain performance for a class of affine LPV systems for which the proposed techniques in the previous section can be directly extended to synthesizing a gain-scheduled H_∞ controller. In addition, the proposed techniques in this section are also applicable to the TP Convex Polytopic LPV model and can be further modified for the grid LPV models. The material in this section is derived directly from [10, 8, 44] and [43].

Consider a given affine LPV plant model with state-space realization

$$\begin{aligned} \dot{x} &= A(\theta)x + B_1(\theta)w + B_2u \\ z &= C_1(\theta)x + D_{11}(\theta)w + D_{12}u \\ y &= C_2x + D_{21}w \end{aligned} \quad (3.59)$$

where $x \in \mathbb{R}^p$ is the state vector, $w \in \mathbb{R}^{m_1}$ is the generalized disturbance vector, $u \in \mathbb{R}^{m_2}$ is the control input vector, $z \in \mathbb{R}^{q_1}$ is the controlled variable or error vector, $y \in \mathbb{R}^{q_2}$ is the measurement output vector, $\theta \in \Theta$, and continuous mapping matrix functions $A : \mathbb{R}^n \rightarrow \mathbb{R}^{p \times p}$, $B_1 : \mathbb{R}^n \rightarrow \mathbb{R}^{p \times m_1}$, $C_1 : \mathbb{R}^n \rightarrow \mathbb{R}^{q_1 \times p}$ and $D_{11} : \mathbb{R}^n \rightarrow \mathbb{R}^{q_1 \times m_1}$.

The assumptions on the plant are as follows [10]: (i) $D_{22} = 0$, (ii) $(B_2, C_2, D_{12}, D_{21})$ are parameter-independent (constant) matrices, and (iii) the pairs $(A(\theta), B_2)$ and $(A(\theta), C_2,)$ are quadratically stabilizable and quadratically detectable over Θ respectively. If assumption (ii) is not satisfied, the computation for a problem solution requires solving an infinite number of LMI constraints, and is therefore not easily tractable [23]. However, the constant matrices restrictions can be overcome by pre-filtering of the control inputs u and/or post-filtering the measured outputs y ; for further details refer to [10]. A loop-shifting argument suffices to overcome the

$D_{22} = 0$ restriction, see [47, pages 44-45]. In addition, the quadratic stabilizability of $(A(\theta), B_2)$ over Θ in the assumption (iii) means the existence of a matrix $X > 0$ such that

$$\mathcal{N}^T[A^T(\theta)X + XA(\theta)]\mathcal{N} < 0, \forall \theta \in \Theta \quad (3.60)$$

where \mathcal{N} denote the null space of B_2^T [10]. Note that the $A(\cdot)$, $B_1(\cdot)$, $C_1(\cdot)$ and $D_{11}(\cdot)$ matrices can also be written as a convex combination of the matrix vertices in a similar manner to (3.51):

$$\begin{pmatrix} A(\theta) & B_1(\theta) & B_2 \\ C_1(\theta) & D_{11}(\theta) & D_{12} \\ C_2 & D_{21} & 0 \end{pmatrix} = \sum_{i=1}^r \alpha_i \begin{pmatrix} \hat{A}_i & \hat{B}_{1_i} & B_2 \\ \hat{C}_{1_i} & \hat{D}_{11_i} & D_{12} \\ C_2 & D_{21} & 0 \end{pmatrix} \quad (3.61)$$

Theorem 3.4.1. (Bounded real lemma [10, 44, 88]) *Given an LTI system $G(s)$ and a state-space realization $G(s) = D + C(sI - A)^{-1}B$. The following statements are equivalent:*

- (i) *A is stable and $\|G(s)\|_\infty < \gamma$*
- (ii) *The existence of a positive definite symmetric matrix P such that*

$$\begin{pmatrix} A^T P + P A & P B & C^T \\ B^T P & -\gamma I & D^T \\ C & D & -\gamma I \end{pmatrix} < 0$$

The bounded real lemma can be extended to LPV systems in conjunction with the notion of quadratic H_∞ performance [10].

Definition 3.4.2. (Quadratic H_∞ performance [10, 23]) *An LPV system of the form (2.46) has quadratic H_∞ performance γ if and only if the existence of a positive definite symmetric matrix P such that*

$$\begin{pmatrix} A(\theta)^T P + P A(\theta) & P B(\theta) & C^T(\theta) \\ B^T(\theta) P & -\gamma I & D^T(\theta) \\ C(\theta) & D(\theta) & -\gamma I \end{pmatrix} < 0$$

for all admissible parameter trajectories. Then, the system (2.46) is quadratically stable and ensures the induced L_2 -norm of the operator mapping the disturbance signal w into the controlled signal z is bounded by γ

3.4.1 Gain-Scheduled Controller Design using SQLF

The gain-scheduled output feedback H_∞ control problem using single quadratic Lyapunov functions is to compute a dynamic affine LPV controller, $K(\theta)$, with state-space equations

$$\begin{aligned} \dot{x}_k &= A_k(\theta)x_k + B_k(\theta)y \\ u &= C_k(\theta)x_k + D_k(\theta)y \end{aligned} \quad (3.62)$$

which stabilizes the closed-loop system, (3.59) and (3.62), and minimizes the closed-loop quadratic H_∞ performance (Definition 3.4.2)

$$\int_0^{t_1} z^T z dt \leq \gamma^2 \int_0^{t_1} w^T w dt, \quad \forall t_1 \geq 0 \quad (3.63)$$

along all possible parameter trajectories, $\forall \theta \in \Theta$. Note that A and A_k have the same dimensions, since we restrict ourselves to the full-order case. With the notation

$$K(\theta) = \begin{pmatrix} A_k(\theta) & B_k(\theta) \\ C_k(\theta) & D_k(\theta) \end{pmatrix} = \sum_{i=1}^r \alpha_i K_i \quad (3.64)$$

$$K_i = \begin{pmatrix} A_{k_i} & B_{k_i} \\ C_{k_i} & D_{k_i} \end{pmatrix}, \quad i = 1, 2, \dots, r \quad (3.65)$$

where r is the total number of vertices and α_i is determined using (3.24) and (3.25). The closed-loop system, (3.59) and (3.62), is described by the state-space equations

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{x}_k \end{bmatrix} &= A_{cl}(\theta) \begin{bmatrix} x \\ x_k \end{bmatrix} + B_{cl}(\theta) w \\ z &= C_{cl}(\theta) \begin{bmatrix} x \\ x_k \end{bmatrix} + D_{cl}(\theta) w \end{aligned} \quad (3.66)$$

where

$$\begin{aligned} A_{cl}(\theta) &= \begin{bmatrix} A(\theta) & 0 \\ 0 & 0_{p \times p} \end{bmatrix} + \mathcal{B}K(\theta)\mathcal{C} = \sum_{i=1}^r \alpha_i \hat{A}_{cl_i} \\ \hat{A}_{cl_i} &= \begin{bmatrix} \hat{A}_i & 0 \\ 0 & 0_{p \times p} \end{bmatrix} + \mathcal{B}K_i\mathcal{C} \\ B_{cl}(\theta) &= \begin{bmatrix} B_1(\theta) \\ 0 \end{bmatrix} + \mathcal{B}K(\theta)\mathcal{D}_{21} = \sum_{i=1}^r \alpha_i \hat{B}_{cl_i} \\ \hat{B}_{cl_i} &= \begin{bmatrix} \hat{B}_{1_i} \\ 0 \end{bmatrix} + \mathcal{B}K_i\mathcal{D}_{21} \\ C_{cl}(\theta) &= [C_1(\theta) \quad 0] + \mathcal{D}_{12}K(\theta)\mathcal{C} = \sum_{i=1}^r \alpha_i \hat{C}_{cl_i} \\ \hat{C}_{cl_i} &= [\hat{C}_{1_i} \quad 0] + \mathcal{D}_{12}K_i\mathcal{C} \\ D_{cl}(\theta) &= D_{11}(\theta) + \mathcal{D}_{12}K(\theta)\mathcal{D}_{21} = \sum_{i=1}^r \alpha_i \hat{D}_{cl_i} \\ \hat{D}_{cl_i} &= \hat{D}_{11_i} + \mathcal{D}_{12}K_i\mathcal{D}_{21} \end{aligned} \quad (3.67)$$

and

$$\begin{aligned} \mathcal{B} &= \begin{bmatrix} 0 & B_2 \\ I_p & 0 \end{bmatrix}, \quad \mathcal{C} = \begin{bmatrix} 0 & I_p \\ C_2 & 0 \end{bmatrix} \\ \mathcal{D}_{12} &= [0 \quad D_{12}], \quad \mathcal{D}_{21} = \begin{bmatrix} 0 \\ D_{21} \end{bmatrix} \end{aligned} \quad (3.68)$$

Based on the single quadratic Lyapunov functions $V(x) = x^T P x$, there is an LPV controller $K(\theta)$ of the form of (3.62) that stabilizes the closed-loop system, (3.59) and (3.62), and ensures the induced L_2 -norm of the operator mapping the disturbance signal w into the controlled signal z is bounded by γ along all possible parameter trajectories if and only if there exists $P = P^T$ such that [45]

$$P > 0, \quad \frac{d}{dt}(x^T P x) + z^T z - \gamma^2 w^T w < 0, \quad \forall \theta \in \Theta \quad (3.69)$$

Inequality (3.69) leads to the well-known bounded real lemma [10] inequality

$$\begin{pmatrix} A_{cl}^T(\theta)P + PA_{cl}(\theta) & PB_{cl}(\theta) & C_{cl}^T(\theta) \\ B_{cl}^T(\theta)P & -\gamma I & D_{cl}^T(\theta) \\ C_{cl}(\theta) & D_{cl}(\theta) & -\gamma I \end{pmatrix} < 0 \quad (3.70)$$

Substituting (3.67) in (3.70), we get

$$\sum_{i=1}^r \alpha_i \begin{pmatrix} \hat{A}_{cl_i}^T P + P \hat{A}_{cl_i} & P \hat{B}_{cl_i} & \hat{C}_{cl_i}^T \\ \hat{B}_{cl_i}^T P & -\gamma I & \hat{D}_{cl_i}^T \\ \hat{C}_{cl_i} & \hat{D}_{cl_i} & -\gamma I \end{pmatrix} < 0 \quad (3.71)$$

Inequality (3.71) can be also rewritten as (see [44])

$$\sum_{i=1}^r \alpha_i (\Psi_{cl_i} + \mathcal{Q}^T K_i^T \mathcal{P}_{cl} + \mathcal{P}_{cl}^T K_i \mathcal{Q}) < 0 \quad (3.72)$$

where

$$\Psi_{cl_i} = \begin{pmatrix} \begin{bmatrix} \hat{A}_i & 0 \\ 0 & 0_{p \times p} \end{bmatrix}^T P + P \begin{bmatrix} \hat{A}_i & 0 \\ 0 & 0_{p \times p} \end{bmatrix} & P \begin{bmatrix} \hat{B}_{1_i} \\ 0 \end{bmatrix} & [\hat{C}_{1_i} \ 0]^T \\ \begin{bmatrix} \hat{B}_{1_i} \\ 0 \end{bmatrix}^T P & -\gamma I & \hat{D}_{11_i}^T \\ [\hat{C}_{1_i} \ 0] & \hat{D}_{11_i} & -\gamma I \end{pmatrix} \quad (3.73)$$

$$\mathcal{Q} = [\mathcal{C}, \ \mathcal{D}_{21}, \ 0_{(p+q_2) \times q_1}] \quad (3.74)$$

$$\mathcal{P}_{cl} = [\mathcal{B}^T P, \ 0_{(p+m_2) \times m_1}, \ \mathcal{D}_{12}^T] \quad (3.75)$$

Having determined the quadratic Lyapunov variable $P \in \mathbb{R}^{2p \times 2p}$, the system matrix vertices K_i of the LPV controller $K(\theta)$ for each vertex Θ_i , $i = 1, \dots, r$, can be determined from (3.72) that is an LMI in K_i . By Lemma 3.3.1, the LMIs (3.72) need only be evaluated at all vertices. Alternatively, a more efficient explicit scheme for determining K_i is given in [43]. Knowing K_i , the controller system matrices $A_k(\theta), \dots, D_k(\theta)$ can be computed on-line in real-time using (3.64) with an instantaneous measurement value of θ .

To determine the quadratic Lyapunov variable P , we have to define a structure of P . Although the exact structure of P is still not certain, a typical structure of P is suggested in [10, 23, 43, 66, 96] and [101] for which, in this thesis, the structure of P is taken from [66, 96] and [101] as

$$P = \begin{bmatrix} X & -(X - Y^{-1}) \\ -(X - Y^{-1}) & X - Y^{-1} \end{bmatrix}, \quad P^{-1} = \begin{bmatrix} Y & Y \\ Y & (X - Y^{-1})^{-1} X Y \end{bmatrix} \quad (3.76)$$

where a pair of positive definite symmetric matrices $(X, Y) \in \mathbb{R}^{p \times p}$, $X - Y^{-1} \geq 0$, and $\text{rank}(X - Y^{-1}) \leq p$ [82]. By Lemma 2.2.14, LMIs (3.72) are solvable for K_i if and only if there exist a pair of positive definite symmetric matrices (X, Y) satisfying the following LMIs:

$$\sum_{i=1}^r \alpha_i \left(\begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \hat{A}_i^T X + X \hat{A}_i & X \hat{B}_{1i} & \hat{C}_{1i}^T \\ \hat{B}_{1i}^T X & -\gamma I & \hat{D}_{11i}^T \\ \hat{C}_{1i} & \hat{D}_{11i} & -\gamma I \end{pmatrix} \begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix} \right) < 0 \quad (3.77)$$

$$\sum_{i=1}^r \alpha_i \left(\begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \hat{A}_i Y + Y \hat{A}_i^T & Y \hat{C}_{1i}^T & \hat{B}_{1i} \\ \hat{C}_{1i} Y & -\gamma I & \hat{D}_{11i} \\ \hat{B}_{1i}^T & \hat{D}_{11i}^T & -\gamma I \end{pmatrix} \begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix} \right) < 0 \quad (3.78)$$

$$\begin{pmatrix} X & I \\ I & Y \end{pmatrix} > 0 \quad (3.79)$$

where N_X and N_Y denote bases of the null spaces of $[C_2, D_{21}]$ and $[B_2^T, D_{12}^T]$, respectively. Note that, (3.79) ensures $X, Y > 0$ and $X - Y^{-1} \geq 0$. By Lemma 3.3.1, (3.77)–(3.79) need only be evaluated at all vertices. Hence we get the following theorem.

Theorem 3.4.3. (Convex solvability conditions [10]) *There exists an LPV controller $K(\theta)$ guaranteeing the closed-loop system, (3.59) and (3.62), quadratic H_∞ performance γ along all possible parameter trajectories, $\forall \theta \in \Theta$, if and only if the following LMI conditions hold for some positive definite symmetric matrices (X, Y) , which further satisfy $\text{Rank}(X - Y^{-1}) \leq p$:*

$$\begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \hat{A}_i^T X + X \hat{A}_i & X \hat{B}_{1i} & \hat{C}_{1i}^T \\ \hat{B}_{1i}^T X & -\gamma I & \hat{D}_{11i}^T \\ \hat{C}_{1i} & \hat{D}_{11i} & -\gamma I \end{pmatrix} \begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix} < 0 \quad (3.80)$$

$$\begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \hat{A}_i Y + Y \hat{A}_i^T & Y \hat{C}_{1i}^T & \hat{B}_{1i} \\ \hat{C}_{1i} Y & -\gamma I & \hat{D}_{11i} \\ \hat{B}_{1i}^T & \hat{D}_{11i}^T & -\gamma I \end{pmatrix} \begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix} < 0 \quad (3.81)$$

$$\begin{pmatrix} X & I \\ I & Y \end{pmatrix} > 0 \quad (3.82)$$

where $i = 1, 2, \dots, r$

Note that, when the parameters θ are time-invariant or slowly varying, the conservatism of the above theorem can be reduced using parameter dependent Lyapunov functions in which the improved theorem is presented in section 6.2.

3.4.2 Gain-Scheduled Controller Design via LFT

Consider a given affine LPV plant model of the form (3.59) for which, based on the LFT technique [104], the scheduled parameters θ can be separated from the system

matrices in a similar manner to (3.43) as

$$\begin{aligned}
 \dot{x} &= Ax + B_\theta w_\theta + B_1 w + B_2 u \\
 z_\theta &= C_\theta x + D_{\theta\theta} w_\theta + D_{\theta 1} w + D_{\theta 2} u \\
 z &= C_1 x + D_{1\theta} w_\theta + D_{11} w + D_{12} u \\
 y &= C_2 x + D_{2\theta} w_\theta + D_{21} w \\
 w_\theta &= \hat{\theta} z_\theta
 \end{aligned} \tag{3.83}$$

where $w_\theta, z_\theta \in \mathbb{R}^s$, $s = s_1 + s_2 + \dots + s_n$, $\hat{\theta} = \text{diag}(\tilde{\theta}_1 I_{s_1}, \tilde{\theta}_2 I_{s_2}, \dots, \tilde{\theta}_n I_{s_n})$, and $\|\hat{\theta}\| \leq 1$. Note that, unlike the gain-scheduled controller design using SQLF case, the method in this sub-section is not applicable to a problem where the vectors w and z are not the same dimension, i.e. $m_1 \neq q_1$ [9, 24].

Consistently with (3.83), we seek an LPV controller such that (i) the closed-loop system is internally stable for all parameter trajectories, $\forall \theta \in \Theta$, and $\gamma^2 \hat{\theta}^T \hat{\theta} \leq 1$ and (ii) the induced L_2 -norm of the operator mapping the disturbance signal into the controlled signal is bounded by γ [9]. Note that, the controller is defined to have the same dependency on θ as the plant because it can use the available information of θ to adjust its dynamic to the current plant dynamic on-line in real-time. This LPV controller can be written as a state-space system of the form:

$$\begin{aligned}
 \begin{pmatrix} \dot{x}_k \\ u \\ \tilde{u} \end{pmatrix} &= \begin{pmatrix} A_k & \begin{bmatrix} B_{k_1} & B_{k_\theta} \end{bmatrix} \\ \begin{bmatrix} C_{k_1} \\ C_{k_\theta} \end{bmatrix} & \begin{bmatrix} D_{k_{11}} & D_{k_{1\theta}} \\ D_{k_{\theta 1}} & D_{k_{\theta\theta}} \end{bmatrix} \end{pmatrix} \begin{pmatrix} x_k \\ y \\ \tilde{y} \end{pmatrix} \\
 \tilde{y} &= \hat{\theta} \tilde{u}
 \end{aligned} \tag{3.84}$$

Note that A and A_k have the same dimensions. Actually, the above controller is given in a lower LFT with respect to $\hat{\theta}$, $\mathcal{F}_l(K, \hat{\theta})$, in which θ plays the role of scheduling variable and gives the rule for updating the controller state space matrices based on the measurements of θ . Equivalently, (3.84) can be further written as [9]

$$\begin{aligned}
 \dot{x}_k &= A_k(\theta) x_k + B_k(\theta) y \\
 u &= C_k(\theta) x_k + D_k(\theta) y
 \end{aligned} \tag{3.85}$$

where

$$\begin{aligned}
 A_k(\theta) &= A_k + B_{k_\theta} \Lambda_\theta C_{k_\theta} \\
 B_k(\theta) &= B_{k_1} + B_{k_\theta} \Lambda_\theta D_{k_{\theta 1}} \\
 C_k(\theta) &= C_{k_1} + D_{k_{1\theta}} \Lambda_\theta C_{k_\theta} \\
 D_k(\theta) &= D_{k_{11}} + D_{k_{1\theta}} \Lambda_\theta D_{k_{\theta 1}} \\
 \Lambda_\theta &= \hat{\theta} (I - D_{k_{\theta\theta}} \hat{\theta})^{-1}
 \end{aligned}$$

and assume the inverse $(I - D_{k_{\theta\theta}} \hat{\theta})^{-1}$ exists $\forall \theta \in \Theta$ with $\|\hat{\theta}\| \leq 1/\gamma$; for further details refer to [9]. To apply the small gain theorem for this problem, all parameter-dependent components $\hat{\theta}$, that enter both the plant (3.83) and the controller (3.84),

have to transfer into a single uncertainty block, shown in Figure 3.2. Introducing the augmented plant

$$\begin{pmatrix} \dot{x} \\ \tilde{z}_\theta \\ z_\theta \\ z \\ y \\ \tilde{y} \end{pmatrix} = \begin{pmatrix} A & \begin{bmatrix} 0 & B_\theta & B_1 \end{bmatrix} & \begin{bmatrix} B_2 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ C_\theta \\ C_1 \\ C_2 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & D_{\theta\theta} & D_{\theta 1} \\ 0 & D_{1\theta} & D_{11} \\ 0 & D_{2\theta} & D_{21} \\ I & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & I \\ D_{\theta 2} & 0 \\ D_{12} & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \end{pmatrix} \begin{pmatrix} x \\ \tilde{w}_\theta \\ w_\theta \\ w \\ u \\ \tilde{u} \end{pmatrix}$$

$$\begin{bmatrix} \tilde{w}_\theta \\ w_\theta \end{bmatrix} = \begin{bmatrix} \hat{\theta} & 0 \\ 0 & \hat{\theta} \end{bmatrix} \begin{bmatrix} \tilde{z}_\theta \\ z_\theta \end{bmatrix} \quad (3.86)$$

The closed-loop system, (3.86) and (3.84) shown in Figure 3.2, is described by the state-space equations

$$\begin{bmatrix} \dot{x}_{cl} \\ z_\Theta \\ z \end{bmatrix} = \begin{bmatrix} A_{cl} & B_{\theta_{cl}} L^{-\frac{1}{2}} & B_{1_{cl}} \\ L^{\frac{1}{2}} C_{\theta_{cl}} & L^{\frac{1}{2}} D_{\theta\theta_{cl}} L^{-\frac{1}{2}} & L^{\frac{1}{2}} D_{\theta 1_{cl}} \\ C_{1_{cl}} & D_{1\theta_{cl}} L^{-\frac{1}{2}} & D_{11_{cl}} \end{bmatrix} \begin{bmatrix} x_{cl} \\ w_\Theta \\ w \end{bmatrix}$$

$$w_\theta = \hat{\Theta} z_\theta \quad (3.87)$$

where

$$x_{cl} = \begin{bmatrix} x \\ x_k \end{bmatrix}, \quad z_\Theta = \begin{bmatrix} \tilde{z}_\theta \\ z_\theta \end{bmatrix}, \quad w_\Theta = \begin{bmatrix} \tilde{w}_\theta \\ w_\theta \end{bmatrix}, \quad \hat{\Theta} = \begin{bmatrix} \hat{\theta} & 0 \\ 0 & \hat{\theta} \end{bmatrix}, \quad L \in L_\Theta$$

L is a scaling matrix. The set of L_Θ is defined as

$$L_\Theta = \left\{ L = \begin{bmatrix} L_1 & L_2 \\ L_2^T & L_3 \end{bmatrix} > 0 : L\hat{\Theta} = \hat{\Theta}L, \forall \theta \in \Theta \right\} \subset \mathbb{R}^{2s \times 2s} \quad (3.88)$$

and

$$\begin{aligned} A_{cl} &= \begin{bmatrix} A & 0 \\ 0 & 0_{p \times p} \end{bmatrix} + \mathcal{B}K\mathcal{C}, \quad K = \begin{pmatrix} A_k & \begin{bmatrix} B_{k_1} & B_{k_\theta} \end{bmatrix} \\ \begin{bmatrix} C_{k_1} \\ C_{k_\theta} \end{bmatrix} & \begin{bmatrix} D_{k_{11}} & D_{k_{1\theta}} \\ D_{k_{\theta 1}} & D_{k_{\theta\theta}} \end{bmatrix} \end{pmatrix} \\ B_{\theta_{cl}} &= \begin{bmatrix} 0 & B_\theta \\ 0 & 0 \end{bmatrix} + \mathcal{B}K\mathcal{D}_{\theta 21}, \quad B_{1_{cl}} = \begin{bmatrix} B_1 \\ 0 \end{bmatrix} + \mathcal{B}K\mathcal{D}_{121} \\ C_{\theta_{cl}} &= \begin{bmatrix} 0 & 0 \\ C_\theta & 0 \end{bmatrix} + \mathcal{D}_{\theta 12}K\mathcal{C}, \quad C_{1_{cl}} = \begin{bmatrix} C_1 & 0 \end{bmatrix} + \mathcal{D}_{112}K\mathcal{C} \\ D_{\theta\theta_{cl}} &= \begin{bmatrix} 0 & 0 \\ 0 & D_{\theta\theta} \end{bmatrix} + \mathcal{D}_{\theta 12}K\mathcal{D}_{\theta 21}, \quad D_{\theta 1_{cl}} = \begin{bmatrix} 0 \\ D_{\theta 1} \end{bmatrix} + \mathcal{D}_{\theta 12}K\mathcal{D}_{121} \\ D_{1\theta_{cl}} &= \begin{bmatrix} 0 & D_{1\theta} \end{bmatrix} + \mathcal{D}_{112}K\mathcal{D}_{\theta 21}, \quad D_{11_{cl}} = D_{11} + \mathcal{D}_{112}K\mathcal{D}_{121} \end{aligned} \quad (3.89)$$

with

$$\mathcal{C} = \begin{bmatrix} 0 & I_p \\ C_2 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathcal{D}_{\theta 21} = \begin{bmatrix} 0 & 0 \\ 0 & D_{2\theta} \\ I & 0 \end{bmatrix}, \quad \mathcal{D}_{121} = \begin{bmatrix} 0 \\ D_{21} \\ 0 \end{bmatrix},$$

$$\mathcal{B} = \begin{bmatrix} 0 & B_2 & 0 \\ I_p & 0 & 0 \end{bmatrix}, \quad \mathcal{D}_{\theta 12} = \begin{bmatrix} 0 & 0 & I \\ 0 & D_{\theta 2} & 0 \end{bmatrix}, \quad \mathcal{D}_{112} = \begin{bmatrix} 0 & D_{12} & 0 \end{bmatrix}$$

Based on the single quadratic Lyapunov functions $V(x) = x^T P x$, there is a controller K , (3.84), that stabilizes the closed-loop system, (3.86) and (3.84), and ensures the induced L_2 -norm of the operator mapping the disturbance signal into the controlled signal is bounded by γ along all possible parameter trajectories if and only if there exists $P = P^T$ such that [45]

$$P > 0, \quad \frac{d}{dt}(x^T P x) + (z_\Theta^T z_\Theta + z^T z) - \gamma^2(w_\Theta^T w_\Theta + w^T w) < 0, \quad \forall \theta \in \Theta \quad (3.90)$$

Inequality (3.90) leads to the scaled bounded real lemma [9] inequality

$$\begin{pmatrix} A_{cl}^T P + P A_{cl} & P B_{\theta_{cl}} & P B_{1_{cl}} & C_{\theta_{cl}}^T & C_{1_{cl}}^T \\ B_{\theta_{cl}}^T P & -\gamma L & 0 & D_{\theta\theta_{cl}}^T & D_{1\theta_{cl}}^T \\ B_{1_{cl}}^T P & 0 & -\gamma I & D_{\theta 1_{cl}}^T & D_{11_{cl}}^T \\ C_{\theta_{cl}} & D_{\theta\theta_{cl}} & D_{\theta 1_{cl}} & -\gamma L^{-1} & 0 \\ C_{1_{cl}} & D_{1\theta_{cl}} & D_{11_{cl}} & 0 & -\gamma I \end{pmatrix} < 0 \quad (3.91)$$

Inequality (3.91) can be also rewritten as (see [44])

$$\Psi_{cl} + \mathcal{Q}^T K^T \mathcal{P}_{cl} + \mathcal{P}_{cl}^T K \mathcal{Q} < 0 \quad (3.92)$$

where

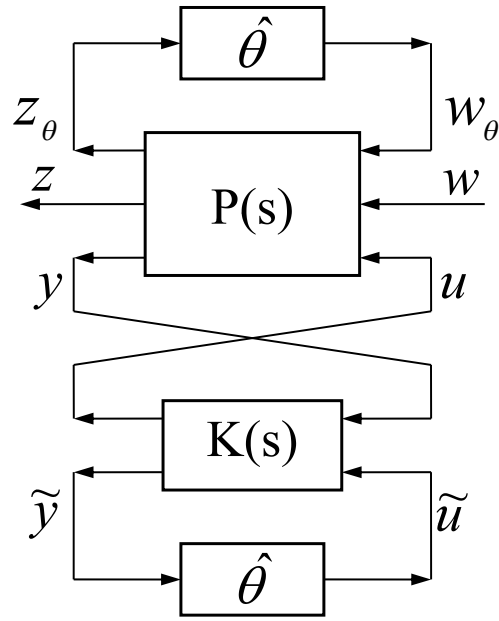
$$\Psi_{cl} = \begin{pmatrix} \begin{bmatrix} A & 0 \\ 0 & 0_{p \times p} \end{bmatrix}^T P + P \begin{bmatrix} A & 0 \\ 0 & 0_{p \times p} \end{bmatrix} & P \begin{bmatrix} 0 & B_\theta \\ 0 & 0 \end{bmatrix} & P \begin{bmatrix} B_1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ C_\theta & 0 \end{bmatrix}^T & [C_1 \ 0]^T \\ \begin{bmatrix} 0 & B_\theta \\ 0 & 0 \end{bmatrix}^T P & -\gamma L & 0 & \begin{bmatrix} 0 & 0 \\ 0 & D_{\theta\theta} \end{bmatrix}^T & [0 \ D_{1\theta}]^T \\ \begin{bmatrix} B_1 \\ 0 \end{bmatrix}^T P & 0 & -\gamma I & \begin{bmatrix} 0 \\ D_{\theta 1} \end{bmatrix}^T & D_{11}^T \\ \begin{bmatrix} 0 & 0 \\ C_\theta & 0 \\ C_1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & D_{\theta\theta} \\ 0 & D_{1\theta} \end{bmatrix} & \begin{bmatrix} 0 \\ D_{\theta 1} \\ D_{11} \end{bmatrix} & -\gamma L^{-1} & 0 \\ & & & 0 & -\gamma I \end{pmatrix} \quad (3.93)$$

$$\mathcal{Q} = [\mathcal{C}, \ \mathcal{D}_{\delta_{21}}, \ \mathcal{D}_{1_{21}}, \ 0_{(p+q_2) \times (q_1+s)}] \quad (3.94)$$

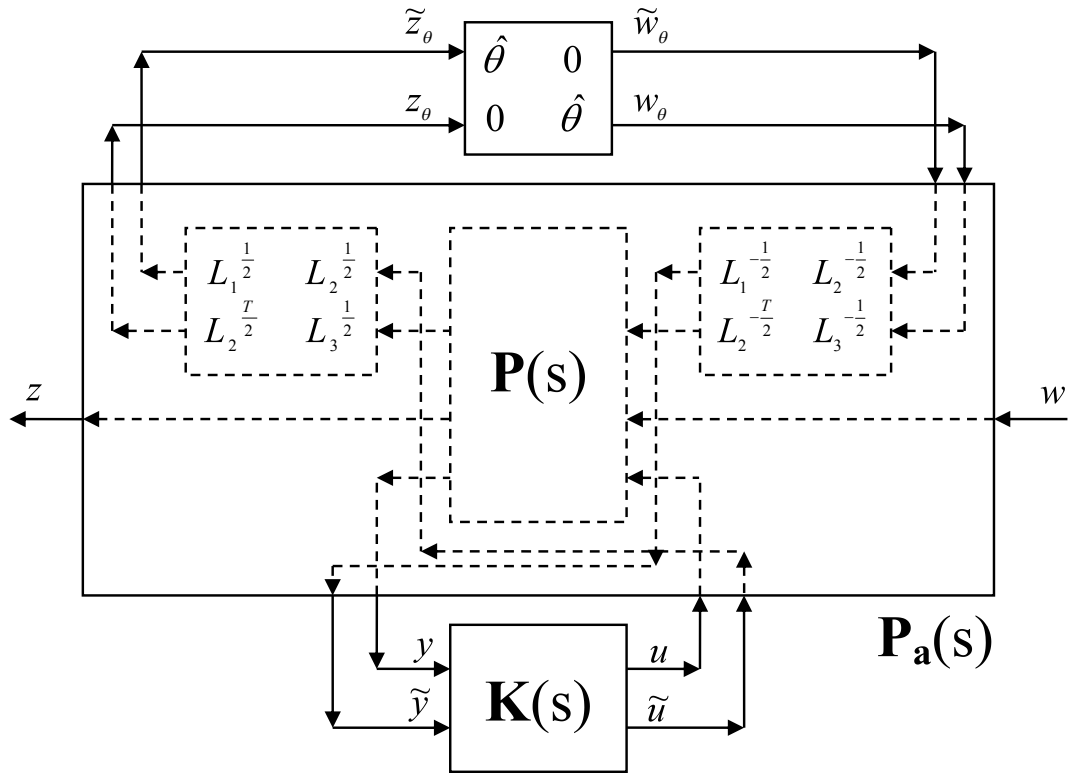
$$\mathcal{P}_{cl} = [\mathcal{B}^T P, \ 0_{(p+m_2) \times (m_1+s)}, \ \mathcal{D}_{\delta_{12}}^T, \ \mathcal{D}_{1_{12}}^T] \quad (3.95)$$

Having determined the quadratic Lyapunov variable $P \in \mathbb{R}^{2p \times 2p}$ and the scaling matrix $L \in L_\Theta$, the LPV controller K can be determined from (3.92) that is an LMI in K . Knowing $A_k, B_{k_1}, \dots, D_{k_{\theta\theta}}$, the controller system matrix $A_k(\theta), \dots, D_k(\theta)$ can be computed on-line in real-time using (3.85) with an instantaneous measurement value of θ . To determine the quadratic Lyapunov variable P and the scaling matrix L , we have to define a structure of P and L . The structure of P is defined as (3.76). Although the exact structure of L is still not certain, a typical structure of L is suggested in [9, 24] and [81] for which, in this thesis, the structure of L follows the structure of P which is taken from [66, 96] and [101] as

$$L = \begin{bmatrix} L_3 - J_3^{-1} & -(L_3 - J_3^{-1}) \\ -(L_3 - J_3^{-1}) & L_3 \end{bmatrix}, \quad L^{-1} = \begin{bmatrix} (L_3 - J_3^{-1})^{-1} L_3 J_3 & J_3 \\ J_3 & J_3 \end{bmatrix} \quad (3.96)$$



(a) LPV/LFT control structure



(b) Augmented structure

Figure 3.2: Block diagrams structures of LPV/LFT closed-loop systems

Obviously, based on (3.96), $L_3, J_3 \in L_\theta$ is a necessary condition to make $L \in L_\Theta$. In addition, $L_3 - J_3^{-1} \geq 0$, and $\text{Rank}(L_3 - J_3^{-1}) \leq s$ [82]. By Lemma 2.2.14, the following theorem is obtained.

Theorem 3.4.4. *There exists an LPV controller K guaranteeing the closed-loop system, (3.86) and (3.84), quadratic H_∞ performance γ along all possible parameter trajectories, $\forall \theta \in \Theta$ with $\|\hat{\theta}\| \leq 1/\gamma$, if and only if the following LMI conditions hold for some positive definite symmetric matrices $(X, Y) \in \mathbb{R}^{p \times p}$ and $(\tilde{L}_3, \tilde{J}_3) \in L_\theta$, which further satisfy $\text{Rank}(X - Y^{-1}) \leq p$ and $\text{Rank}(\tilde{L}_3 - \tilde{J}_3^{-1}) \leq s$.*

$$\begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} A^T X + X A & X B_\theta & X B_1 & C_\theta^T & C_1^T \\ B_\theta^T X & -\tilde{L}_3 & 0 & D_{\theta\theta}^T & D_{1\theta}^T \\ B_1^T X & 0 & -\gamma I & D_{\theta 1}^T & D_{11}^T \\ C_\theta & D_{\theta\theta} & D_{\theta 1} & -\tilde{J}_3 & 0 \\ C_1 & D_{1\theta} & D_{11} & 0 & -\gamma I \end{pmatrix} \begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix} < 0 \quad (3.97)$$

$$\begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} A Y + Y A^T & Y C_\theta^T & Y C_1^T & B_\theta & B_1 \\ C_\theta Y & -\tilde{J}_3 & 0 & D_{\theta\theta} & D_{\theta 1} \\ C_1 Y & 0 & -\gamma I & D_{1\theta} & D_{11} \\ B_\theta^T & D_{\theta\theta}^T & D_{1\theta}^T & -\tilde{L}_3 & 0 \\ B_1^T & D_{\theta 1}^T & D_{11}^T & 0 & -\gamma I \end{pmatrix} \begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix} < 0 \quad (3.98)$$

$$\begin{pmatrix} X & I \\ I & Y \end{pmatrix} > 0 \quad (3.99)$$

$$\begin{pmatrix} \tilde{L}_3 & \gamma \\ \gamma & \tilde{J}_3 \end{pmatrix} > 0 \quad (3.100)$$

where \tilde{L}_3 and \tilde{J}_3 are equal to γL_3 and γJ_3 respectively, N_X and N_Y denote bases of the null spaces of $[C_2, D_{2\theta}, D_{21}, 0]$ and $[B_2^T, D_{\theta 2}^T, D_{12}^T, 0]$ respectively. Note that, (3.99) and (3.100) ensure $X, Y > 0$, $X - Y^{-1} \geq 0$ and $L_3, J_3 > 0$, $L_3 - J_3^{-1} \geq 0$, respectively.

Chapter 4

Numerical Example

This chapter aims to illustrate the implementation of LPV systems theory for the nonlinear control problem which include methods for deriving an LPV model from a nonlinear model, analysis and synthesis problems of gain-scheduled output feedback H_∞ controller design. To demonstrate the method, we explicitly consider the example of Leith and Leithead [61]. In this example it was shown that for an LPV model derived from the Jacobians, a common approach, with an LPV controller synthesized using the method of Apkarian et al. [10], is unstable when applied to the original nonlinear plant [30, 61].

Consider the nonlinear plant example taken from [61]

$$\begin{aligned}\dot{x}_1(t) &= -x_1(t) + r(t) \\ \dot{x}_2(t) &= x_1(t) - |x_2(t)|x_2(t) - 10 \\ y(t) &= x_2(t)\end{aligned}\tag{4.1}$$

where $t \in \mathbb{R}$ is time, both $x_1(t), x_2(t) \in \mathbb{R}$ are the state, $r(t), y(t) \in \mathbb{R}$ are the control input and the measurement output, respectively. The control requirement is to design an output-feedback controller which ensures a step response settling time of less than 2 seconds with zero steady-state error [61]. Step inputs with different amplitudes are applied to the system (4.1), at $t=0$ where an initial condition of the system is $x_1(0) = 0$ and $x_2(0) = -3.16$, in order to investigate the open-loop dynamic step response, shown in Figure 4.1. It can be seen that the system (4.1) is an open-loop stable system and its step response is similar to a first-order transfer function with a varying time constant (time lag) and a varying low frequency gain, following its output $y(t)$. Moreover, a set of equilibrium points of the system can be calculated by setting $\dot{x}_1 = \dot{x}_2 = 0$ in (4.1). Then, the results are

$$\begin{aligned}x_{1\text{trim}} = r_{\text{trim}} &= \begin{cases} x_{2\text{trim}}^2 + 10, & \text{if } x_{2\text{trim}} \geq 0; \\ 10 - x_{2\text{trim}}^2, & \text{if } x_{2\text{trim}} < 0. \end{cases} \\ y_{\text{trim}} &= x_{2\text{trim}}\end{aligned}\tag{4.2}$$

4.1 Jacobian Approach

Applying the Jacobian linearization method to (4.1) by using (3.7), we get

$$\begin{aligned}
A &= \begin{bmatrix} \frac{\partial(-x_1(t)+r(t))}{\partial x_1(t)} & \frac{\partial(-x_1(t)+r(t))}{\partial x_2(t)} \\ \frac{\partial(x_1(t)-|x_2(t)|x_2(t)-10)}{\partial x_1(t)} & \frac{\partial(x_1(t)-|x_2(t)|x_2(t)-10)}{\partial x_2(t)} \end{bmatrix}_{\substack{x_1(t)=x_{1\text{trim}} \\ x_2(t)=x_{2\text{trim}} \\ r(t)=r_{\text{trim}}}} \\
&= \begin{bmatrix} -1 & 0 \\ 1 & -2|x_{2\text{trim}}| \end{bmatrix} \\
B &= \begin{bmatrix} \frac{\partial(-x_1(t)+r(t))}{\partial r(t)} \\ \frac{\partial(x_1(t)-|x_2(t)|x_2(t)-10)}{\partial r(t)} \end{bmatrix}_{\substack{x_1(t)=x_{1\text{trim}} \\ x_2(t)=x_{2\text{trim}} \\ r(t)=r_{\text{trim}}}} \\
&= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\
C &= \begin{bmatrix} \frac{\partial x_2(t)}{\partial x_1(t)} & \frac{\partial x_2(t)}{\partial x_2(t)} \end{bmatrix}_{\substack{x_1(t)=x_{1\text{trim}} \\ x_2(t)=x_{2\text{trim}} \\ r(t)=r_{\text{trim}}}} \\
&= \begin{bmatrix} 0 & 1 \end{bmatrix} \\
D &= \begin{bmatrix} \frac{\partial x_2(t)}{\partial r(t)} \end{bmatrix}_{\substack{x_1(t)=x_{1\text{trim}} \\ x_2(t)=x_{2\text{trim}} \\ r(t)=r_{\text{trim}}}} \\
&= \begin{bmatrix} 0 \end{bmatrix}
\end{aligned} \tag{4.3}$$

where $(x_{1\text{trim}}, x_{2\text{trim}}, r_{\text{trim}}, \text{ and } y_{\text{trim}})$ is one point in a set of equilibrium points (4.2), therefore a Jacobian-based LTI model can be written

$$\begin{aligned}
\begin{bmatrix} \delta \dot{x}_1(t) \\ \delta \dot{x}_2(t) \end{bmatrix} &= \begin{bmatrix} -1 & 0 \\ 1 & -2|x_{2\text{trim}}| \end{bmatrix} \begin{bmatrix} \delta x_1(t) \\ \delta x_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \delta r(t) \\
\delta y(t) &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \delta x_1(t) \\ \delta x_2(t) \end{bmatrix}
\end{aligned} \tag{4.4}$$

where $\delta x_1(t) = x_1(t) - x_{1\text{trim}}$, $\delta x_2(t) = x_2(t) - x_{2\text{trim}}$, $\delta r(t) = r(t) - r_{\text{trim}}$, and $\delta y(t) = y(t) - y_{\text{trim}}$. Note that (4.2) shows that $x_{1\text{trim}}$, $x_{2\text{trim}}$ and r_{trim} are dependent on y_{trim} . With $x_{2\text{trim}}$ fixed, (4.4) is an LTI model. However, as $x_{2\text{trim}}$ varies slowly over the defined parameter space, (4.4) becomes an LPV model. That is

$$\begin{aligned}
\begin{bmatrix} \dot{n}_1 \\ \dot{n}_2 \end{bmatrix} &= \begin{bmatrix} -1 & 0 \\ 1 & -2\theta \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} r \\
y &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}
\end{aligned} \tag{4.5}$$

where $n_1(t) = x_1(t) - x_{1\text{trim}}$, $n_2(t) = x_2(t) - x_{2\text{trim}}$, and $\theta = |x_{2\text{trim}}| = |y(t)|$ is arbitrarily defined from 0 to 10. As a result of the Jacobian linearization method, the dynamic characteristics of the system (4.5) vary, following its output, $x_{2\text{trim}} = y_{\text{trim}}$ shown in (4.2), if the original plant (4.1) is operating about the neighborhood of a set of equilibrium points (4.2). However, $x_{2\text{trim}}$ is not equal to y_{trim} whenever the original plant is operating in a region faraway from its equilibrium points.

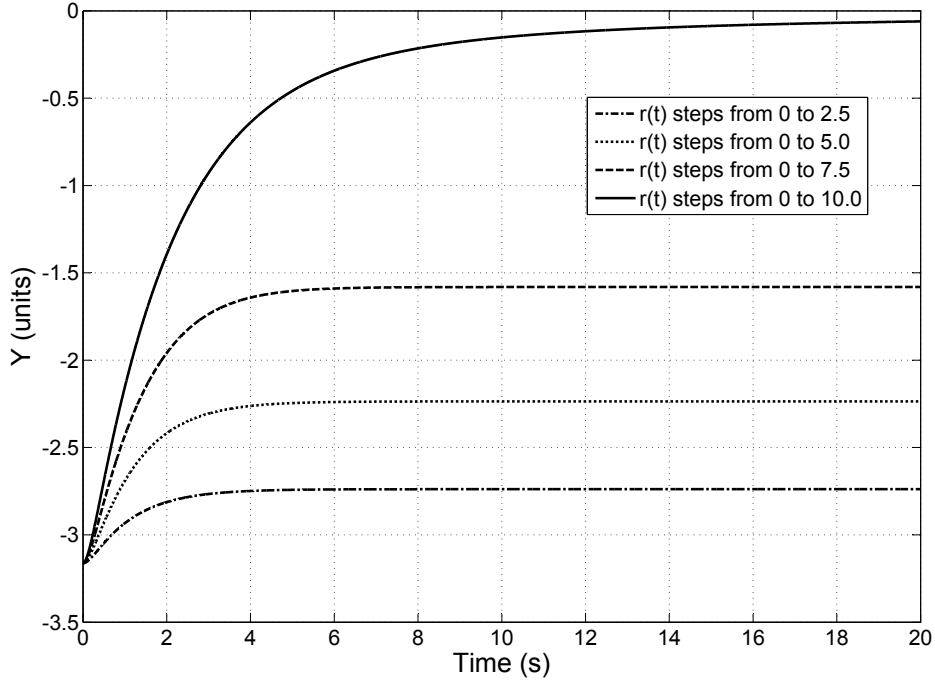


Figure 4.1: Open-loop dynamic step response of the system (4.1) at $t=0$

Remark 4.1.1. *An LPV controller, that is synthesized based on the Jacobian-based LPV model (4.5), adjusts its dynamics to the current plant dynamics using instantaneous measurement values of $y(t)$. But, whenever the original plant is not about an equilibrium condition, the true current dynamic of the original plant does not follow the value of $y(t)$. This means that the LPV controller will adjust its dynamic to the wrong plant dynamic. This is a usual problem for Jacobian linearization method.*

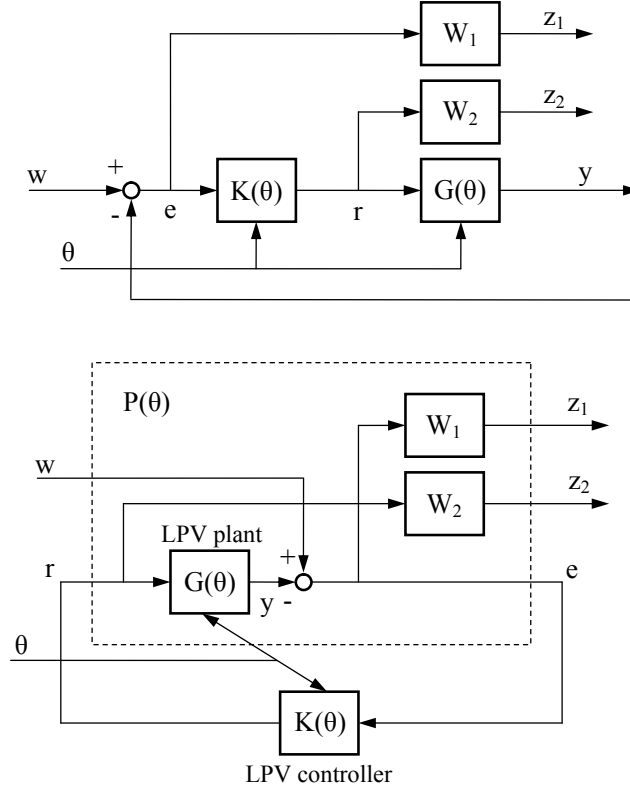
Having determined a Jacobian-based LPV model, a Jacobian-based LPV controller can be synthesized using the method of Apkarian et al. [10] with the criterion

$$\left\| \begin{array}{c} W_1 S \\ W_2 K S \end{array} \right\|_{\infty} < 1 \quad (4.6)$$

as shown in Figure 4.2, where the performance weighting functions W_1 and robustness weighting functions W_2 taken from [61] are

$$\begin{aligned} W_1(s) &= \frac{0.5}{s + 0.002} \\ W_2(s) &= \frac{0.02s}{s + 1000} \end{aligned} \quad (4.7)$$

Using MATLAB routines `ltisys`, we get W_1 and W_2 in the state-space equation of


 Figure 4.2: LPV system: H_∞ mixed S/KS synthesis problem

the form

$$\begin{aligned}
 \dot{x}_{w1}(t) &= -0.002x_{w1}(t) + 0.5e(t) \\
 z_1(t) &= x_{w1}(t) \\
 \dot{x}_{w2}(t) &= -1000x_{w2}(t) + 4r(t) \\
 z_2(t) &= -5x_{w2}(t) + 0.02r(t)
 \end{aligned} \tag{4.8}$$

From (4.5) and (4.8), in addition, $e(t) = w(t) - y(t)$ but $y(t) = n_2(t)$, hence $e(t) = w(t) - n_2(t)$. Finally, we get the minimal realization of the augmented plant $P(\theta)$ as

$$\begin{aligned}
 \dot{n}_1(t) &= -n_1(t) + r(t) \\
 \dot{n}_2(t) &= n_1(t) - 2\theta n_2(t) \\
 \dot{x}_{w1}(t) &= -0.002x_{w1}(t) + 0.5w(t) - 0.5n_2(t) \\
 \dot{x}_{w2}(t) &= -1000x_{w2}(t) + 4r(t) \\
 z_1(t) &= x_{w1}(t) \\
 z_2(t) &= -5x_{w2}(t) + 0.02r(t) \\
 e(t) &= w(t) - n_2(t)
 \end{aligned} \tag{4.9}$$

Equation (4.9) can be rewritten further as

$$\begin{pmatrix} \begin{bmatrix} \dot{n}_1 \\ \dot{n}_2 \\ \dot{x}_{w1} \\ \dot{x}_{w2} \\ z_1 \\ z_2 \\ e \end{bmatrix} \end{pmatrix} = \begin{pmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 \\ 1 & -2\theta & 0 & 0 \\ 0 & -0.5 & -0.002 & 0 \\ 0 & 0 & 0 & -1000 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0.5 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 0 \\ 4 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -5 \\ 0 & -1 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0.02 \\ 0 \end{bmatrix} \end{pmatrix} \begin{pmatrix} \begin{bmatrix} n_1 \\ n_2 \\ x_{w1} \\ x_{w2} \\ w \\ r \end{bmatrix} \end{pmatrix} \quad (4.10)$$

Note that, the augmented plant $P(\theta)$ (4.10) can also be determined using MATLAB routines `sysic`, `iconnect` or `sconnect`. Solving LMIs in Theorem 3.4.3 by using a MATLAB Robust Control Toolbox function [12], `mincx`, we get an LPV controller with a quadratic H_∞ performance $\gamma = 0.1211$ as shown below:

$$\begin{aligned} \dot{x}_k &= (\alpha_1 A_{k_1} + \alpha_2 A_{k_2})x_k + B_k(w - y) \\ r &= C_k x_k \end{aligned} \quad (4.11)$$

where

$$\begin{aligned} \alpha_1 &= \frac{10 - |y|}{10}, \quad \alpha_2 = \frac{|y|}{10} \\ A_{k_1} &= \begin{bmatrix} 11.4933e + 000 & 220.5506e + 000 & 93.1838e + 000 & 15.6255e + 003 \\ -4.1441e + 000 & -34.9702e + 000 & -26.0301e + 000 & 46.2968e + 003 \\ -578.7503e - 003 & -12.5446e + 000 & -10.2185e + 000 & 20.4856e + 003 \\ 1.0769e + 000 & 2.1290e + 000 & 16.1633e + 000 & -66.1258e + 003 \end{bmatrix} \\ A_{k_2} &= \begin{bmatrix} 11.4904e + 000 & 220.4540e + 000 & 93.4033e + 000 & 15.6255e + 003 \\ -4.2406e + 000 & -38.2060e + 000 & -18.6740e + 000 & 46.2966e + 003 \\ -359.2582e - 003 & -5.1885e + 000 & -26.9418e + 000 & 20.4861e + 003 \\ 684.0884e - 003 & -11.0360e + 000 & 46.0926e + 000 & -66.1267e + 003 \end{bmatrix} \\ B_k &= \begin{bmatrix} -2.4197e - 003 \\ -6.4566e - 003 \\ -26.6143e - 003 \\ -18.6908e + 000 \end{bmatrix} \\ C_k &= [-12.7061e + 000 \quad -222.1982e + 000 \quad -94.5584e + 000 \quad -12.8570e + 003] \end{aligned}$$

The matrices are different from those presented in [61]. Also note that the LPV controller presented in [61] is actually open-loop unstable. The controller presented above is open-loop stable for all $|y| \in [0, 10]$. In addition, the synthesizing scheme from Theorem 3.4.3 has been implemented in a MATLAB Robust Control Toolbox function as `hinfsgs` [12].

To confirm that the mixed-sensitivity criterion (4.6) is achieved, the singular values of the transfer matrices S and KS are computed over $\forall \theta \in \Theta$ (with frozen θ), and are shown in Figure 4.3. Obviously, the singular values of S and KS are shaped and bound by W_1 and W_2 respectively. In addition, Figure 4.4 shows that $\|[W_1 S, W_2 K S]^T\|_\infty < \gamma$. Hence, the mixed-sensitivity criterion (4.6) is satisfied. But, the simulation results that are presented in Figure 4.5 still show the closed-loop instability problem described in [61]. It can be seen that the closed-loop system is

stable when the LPV controller is applied to the LPV model for a step response that is a change in demand from -3 units to 0 units. However, when the same LPV controller is applied to the original nonlinear plant, the nonlinear closed-loop system appears to be unstable.

In order to investigate this closed-loop instability with more information, different LPV controller synthesis methods or different deriving LPV model techniques should be also employed. Unfortunately, the Theorem 3.4.4 can not be applied because of $q_1 \neq m_1$. Moreover, in order to derive a TP convex polytopic model of the system (4.1), the TP model transformation toolbox [17] has been used to apply a TP transformation to (4.4) by selecting $|x_{2\text{trim}}|$ as a scheduling parameter. It turns out that the resulting TP polytopic model is identical to the Jacobian-based LPV model in (4.5).

4.2 State Transformation Approach

According to (4.2), we can define $r_{eq}(x_2) = x_{1_{eq}}(x_2) = |x_2|x_2 + 10$. The equation (4.1) can be rewritten in the form of equation (3.8) as

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} -1 & 0 \\ 1 & -|x_2| \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} r + \begin{bmatrix} 0 \\ -10 \end{bmatrix} \\ y &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned} \quad (4.12)$$

Since $r_{eq}(x_2)$ and $x_{1_{eq}}(x_2)$ is a continuously differentiable function, such that for every x_2 , (4.12) is in equilibrium points $\dot{x} = 0$. We get

$$\begin{aligned} \begin{bmatrix} 0 \\ 0 \end{bmatrix} &= \begin{bmatrix} -1 & 0 \\ 1 & -|x_2| \end{bmatrix} \begin{bmatrix} x_{1_{eq}}(x_2) \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} r_{eq}(x_2) + \begin{bmatrix} 0 \\ -10 \end{bmatrix} \\ y_{eq}(x_2) &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1_{eq}}(x_2) \\ x_2 \end{bmatrix} \end{aligned} \quad (4.13)$$

Subtracting (4.13) from (4.12) obtains

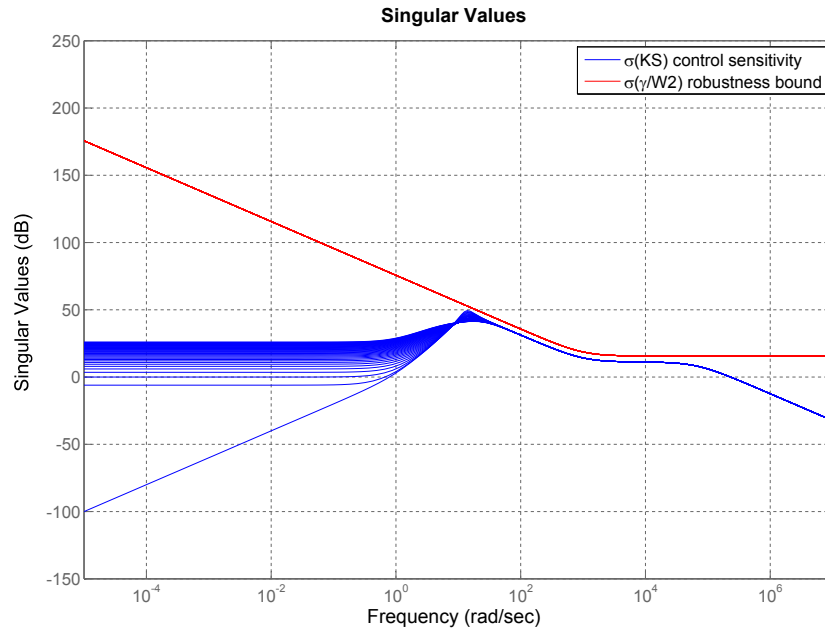
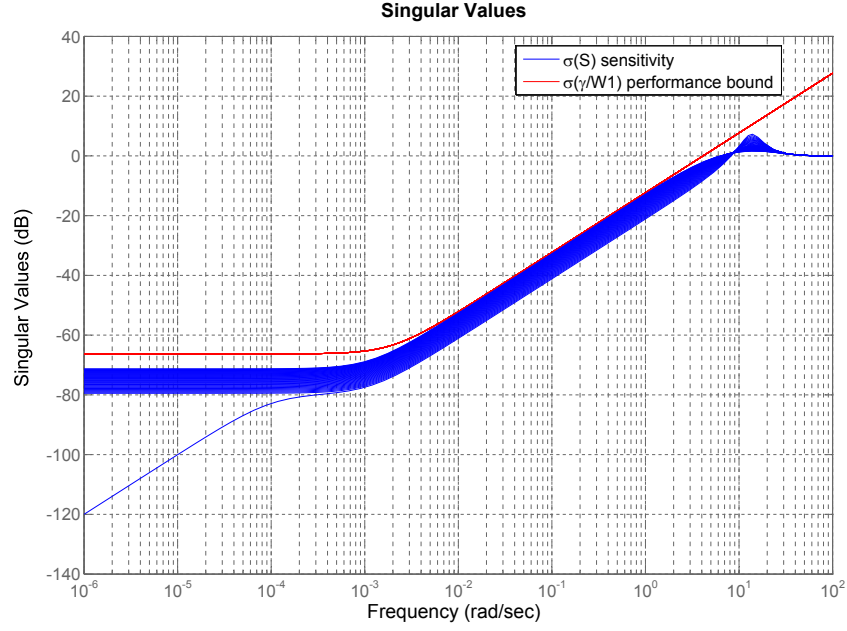
$$\begin{aligned} \dot{x}_1 &= -x_1 + x_{1_{eq}}(x_2) + r - r_{eq}(x_2) \\ \dot{x}_2 &= x_1 - x_{1_{eq}}(x_2) \\ y - y_{eq}(x_2) &= 0 \end{aligned} \quad (4.14)$$

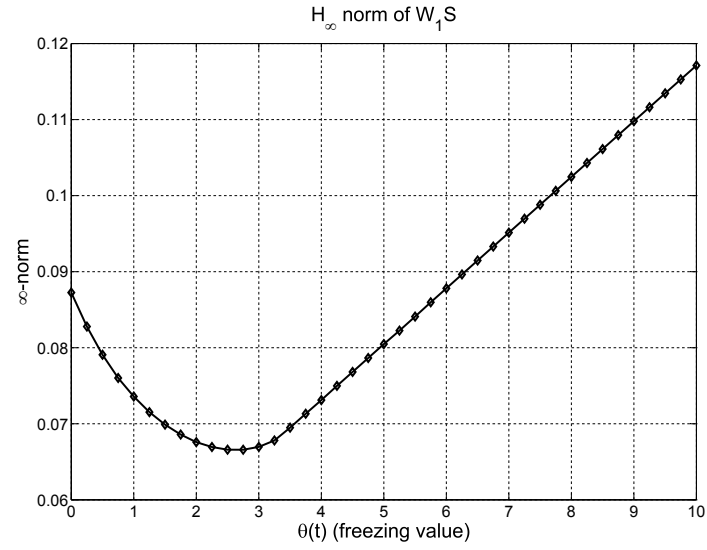
Then, $y = y_{eq}(x_2) = x_2$. Moreover, differentiating $x_{1_{eq}}(x_2) = |x_2|x_2 + 10$ with respect to t gives

$$\frac{d}{dt}x_{1_{eq}}(x_2) = \dot{x}_{1_{eq}}(x_2) = \frac{\partial(|x_2|x_2 + 10)}{\partial x_2} \frac{d}{dt}x_2 = 2|x_2|[x_1 - x_{1_{eq}}(x_2)] \quad (4.15)$$

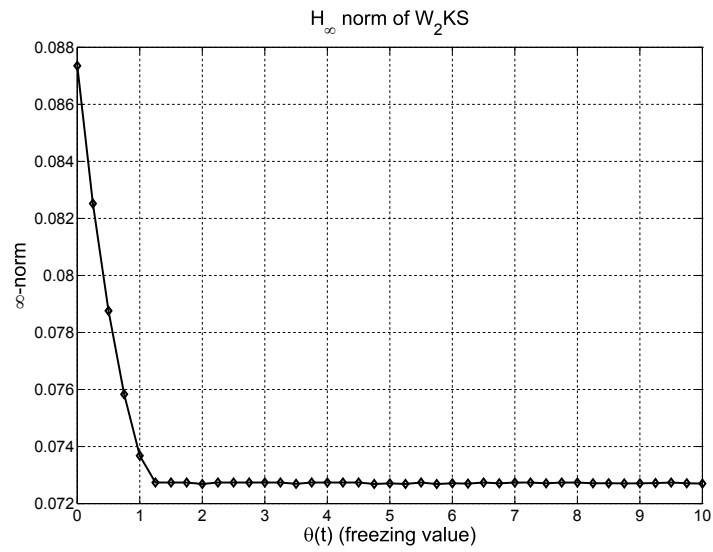
Subtracting (4.15) from (4.14) obtains

$$\dot{x}_1 - \dot{x}_{1_{eq}}(x_2) = [-1 - 2|x_2|][x_1 - x_{1_{eq}}(x_2)] + [r - r_{eq}(x_2)] \quad (4.16)$$


 Figure 4.3: Singular value of S and KS over $\forall \theta \in \Theta$ (with frozen θ)



(a) $\|W_1 S\|_\infty$



(b) $\|W_2 K S\|_\infty$

Figure 4.4: H_∞ norm of $W_1 S$ and $W_2 K S$ over $\forall \theta \in \Theta$ (with frozen θ)

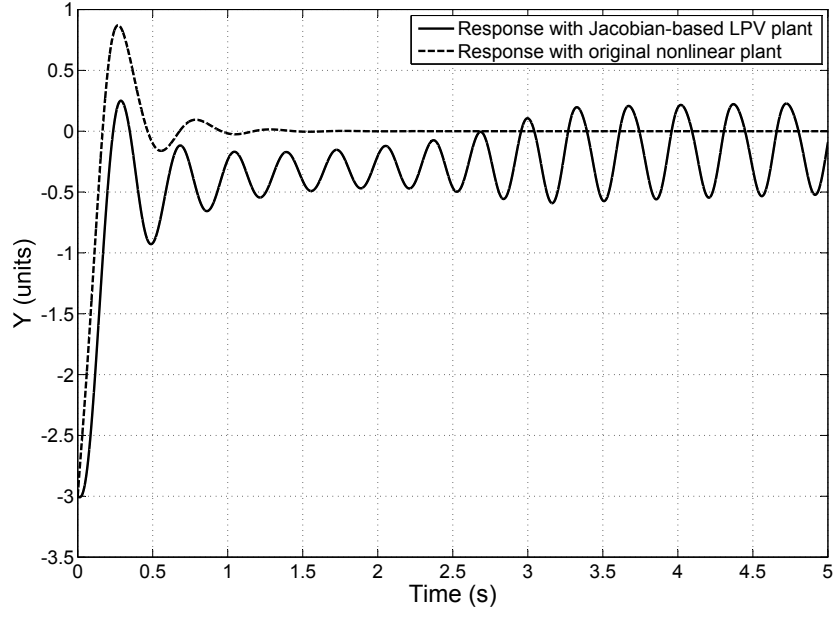


Figure 4.5: Nonlinear step response from -3 to 0 of the Jacobian-based LPV controller (4.11)

Hence, (4.14) can be rewritten as a state-space equation of the form

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 - \dot{x}_{1eq}(x_2) \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} -1 - 2|x_2| & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 - x_{1eq}(x_2) \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} (r - r_{eq}(x_2)) \\ y &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 - x_{1eq}(x_2) \\ x_2 \end{bmatrix} \end{aligned} \quad (4.17)$$

Subsequently, (4.17) can be rewritten as an LPV model of the form

$$\begin{aligned} \begin{bmatrix} \dot{n}_1 \\ \dot{n}_2 \end{bmatrix} &= \begin{bmatrix} -1 - 2|n_2| & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \end{aligned} \quad (4.18)$$

where $n_1 = x_1 - x_{1eq}(x_2) = x_1 - |x_2|x_2 - 10$, $n_2 = x_2$, and $u = r - r_{eq}(x_2) = r - |x_2|x_2 - 10$. The state transformation-based LPV plant presented above is identical to that presented in [93]. Applying `hinfgs` with the same weighting function, the state transformation-based LPV controller for this LPV model is obtained with $\gamma = 0.08863$. Figure 4.6 shows the simulation results of the state transformation-based LPV controller. The closed-loop instability does not occur.

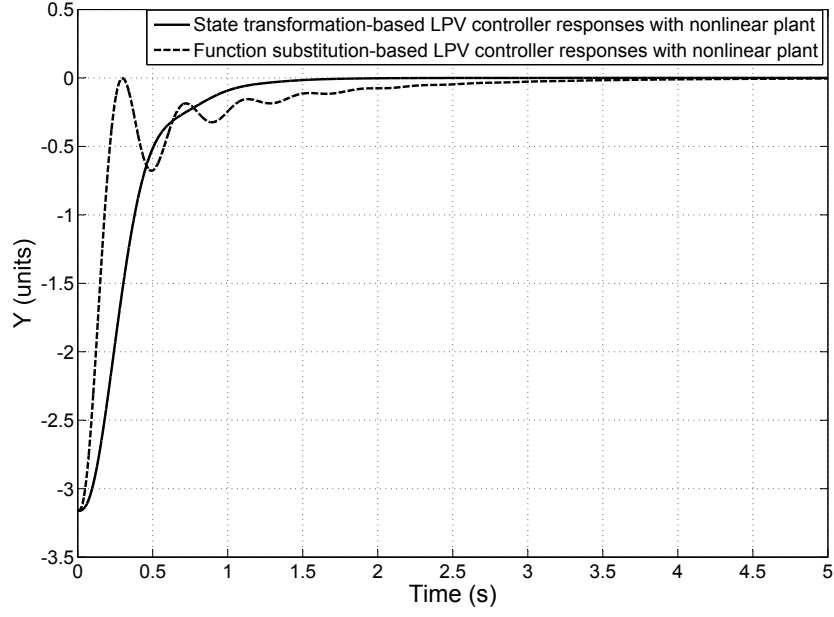


Figure 4.6: Nonlinear step response from -3.16 to 0 of function substitution-based and state transformation-based LPV controllers

4.3 Function Substitution Approach

Substitute $x_1 = \delta x_1 + x_{1r}$, $x_2 = \delta x_2 + x_{2r}$, $r = \delta r + r_r$, and $y = \delta y + y_r$ in equation (4.1), we get

$$\begin{aligned}
 \dot{x}_1 &= \frac{d}{dt}\delta x_1 + \frac{d}{dt}x_{1r} = \dot{\delta x}_1 + 0 \\
 &= -\delta x_1 + \delta r + [-x_{1r} + r_r] \\
 \dot{x}_2 &= \frac{d}{dt}\delta x_2 + \frac{d}{dt}x_{2r} = \dot{\delta x}_2 + 0 \\
 &= \delta x_1 + [x_{1r} - |\delta x_2 + x_{2r}|(\delta x_2 + x_{2r}) - 10] \\
 y &= \delta y + y_r = \delta x_2 + x_{2r}
 \end{aligned} \tag{4.19}$$

where $(x_{1r}, x_{2r}, r_r, \text{ and } y_r)$ is one trim point in a set of the equilibrium points. Selecting a trim point as $(x_{1r} = 10, x_{2r} = 0, r_r = 10, \text{ and } y_r = 0)$, equation (4.19) can be rearranged as an state-space equation of the form

$$\begin{aligned}
 \begin{bmatrix} \delta \dot{x}_1 \\ \delta \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} -1 & 0 \\ 1 & -|\delta x_2| \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \delta x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \delta r \\
 \delta y &= [0 \quad 1] \begin{bmatrix} \delta x_1 \\ \delta x_2 \end{bmatrix}
 \end{aligned} \tag{4.20}$$

Furthermore, (4.20) can be rewritten in an LPV model of the form

$$\begin{aligned} \begin{bmatrix} \dot{n}_1 \\ \dot{n}_2 \end{bmatrix} &= \begin{bmatrix} -1 & 0 \\ 1 & -|n_2| \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \end{aligned} \quad (4.21)$$

where $n_1 = \delta x_1$, $n_2 = \delta x_2$, and $u = r - 10$. The function substitution-based LPV plant presented above is identical to that presented in [93]. Using `hinfgs` with the same weighting function, the function substitution-based LPV controller for this LPV model is obtained with $\gamma = 0.09469$. Figure 4.6 shows the simulation results of the state transformation-based LPV controller. The closed-loop instability does not occur.

According to Figures 4.5 and 4.6, in this particular example, we make two assumptions that (i) function substitution and state transformation methods give an LPV plant model that more accurately represents the nonlinear plant than the Jacobian linearization method, and (ii) there is a mismatch uncertainty between the Jacobian-based LPV model and the original nonlinear model. However, both state transformation-based and function substitution-based LPV models are identical to the original nonlinear model as shown below.

First, we show that the state transformation-based LPV model is identical to the original nonlinear model. By substituting

$$\begin{aligned} r(t) &= u(t) + |x_2(t)|x_2(t) + 10 \\ x_1(t) &= n_1(t) + |x_2(t)|x_2(t) + 10 \\ x_2(t) &= n_2(t) \end{aligned} \quad (4.22)$$

in (4.1). A new nonlinear equation can be obtained in the form.

$$\begin{aligned} \dot{n}_1(t) &= -(1 + 2|n_2(t)|)n_1(t) + u(t) \\ \dot{n}_2(t) &= n_1(t) \\ y(t) &= n_2(t) \end{aligned} \quad (4.23)$$

which can be rearranged as an LPV equation of the form (4.18). Next, we show that the function substitution-based LPV model is identical to the original nonlinear model. By substituting

$$\begin{aligned} r(t) &= u(t) + 10 \\ x_1(t) &= n_1(t) + 10 \\ x_2(t) &= n_2(t) \end{aligned} \quad (4.24)$$

in (4.1). Another nonlinear equation can be obtained in the form.

$$\begin{aligned} \dot{n}_1(t) &= -n_1(t) + u(t) \\ \dot{n}_2(t) &= n_1(t) - |n_2(t)|n_2(t) \\ y(t) &= n_2(t) \end{aligned} \quad (4.25)$$

which can be rearranged as an LPV equation of the form (4.21).

4.4 Mismatch Uncertainty

Consider the transfer function of LPV plant models (2.46) over $\forall \theta \in \Theta$ (with frozen θ) that is given by [10]

$$G(s, \theta) = D(\theta) + C(\theta)[sI - A(\theta)]^{-1}B(\theta) \quad (4.26)$$

Substituting the matrices $A(\cdot), \dots, D(\cdot)$ of the Jacobian-based LPV model (4.5) into (4.26), the transfer function can be determined as

$$G(s, \theta) = \frac{1}{(s+1)(s+2\theta)} \quad (4.27)$$

where, according to (4.5), $\theta = |n_2|$ varying from 0 to 10. This transfer function has two poles; one pole fixes at -1, the other pole varies from -20 to 0. The location of the varying pole of the Jacobian-based LPV model is equal to -2θ , but the true location of the varying pole of the original nonlinear plant is not equal to -2θ whenever this nonlinear plant is not in an equilibrium condition. As $y(t) = n_2(t)$ moves closer to 0, the mismatch uncertainty between the Jacobian-based LPV model and the original nonlinear model becomes more significant and makes the nonlinear closed-loop system unstable as shown below.

Consider the state-trajectories of the closed-loop system of the Jacobian-based LPV controller and LPV model, taken from [30] with disturbance w identically zero, is

$$\dot{x} = (\alpha_1 \hat{A}_1 + \alpha_2 \hat{A}_2)x \quad (4.28)$$

where $\alpha_1 = \frac{10-\theta}{10}$, $\alpha_2 = \frac{\theta}{10}$ and

$$\hat{A}_1 = \begin{pmatrix} -1 & 0 & 0 & 0 & 0.3681 & -210.94 & -118.22 & 7255 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.5 & -0.002 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1000 & 1.4724 & -843.75 & -472.88 & 29020 \\ 0 & -0.0019163 & 0 & 0 & -0.63435 & -210.86 & -118.14 & 7353.2 \\ 0 & 0.027254 & 0 & 0 & 0.086568 & -12.33 & -13.428 & -14430 \\ 0 & -10.534 & 0 & 0 & 0.47837 & -0.18076 & -9.3407 & -21057 \end{pmatrix}$$

$$\hat{A}_2 = \begin{pmatrix} -1 & 0 & 0 & 0 & 0.3681 & -210.94 & -118.22 & 7255 \\ 1 & -20 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.5 & -0.002 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1000 & 1.4724 & -843.75 & -472.88 & 29020 \\ 0 & -0.0019163 & 0 & 0 & -0.63435 & -210.86 & -118.14 & 7353.2 \\ 0 & 0.027254 & 0 & 0 & 0.086568 & -12.33 & -13.428 & -14430 \\ 0 & -10.534 & 0 & 0 & 0.47837 & -0.18076 & -9.3407 & -21057 \end{pmatrix}$$

As a result of Proposition 3.3.4, this closed-loop system is quadratically stable. Note that, the MATLAB Robust Control Toolbox [12] also provides a function `quadstab`

to test the quadratic stability for a class of affine LPV systems. However, when including the time-varying real parametric uncertainty to both \hat{A}_1 and \hat{A}_2 of (4.28) in a region close to the right-half s -plane

$$G(s, \theta) = \frac{1}{(s+1)(s+2\theta+0.4\delta)}, \quad \delta \in [-1, 1] \quad (4.29)$$

Equation (4.28) becomes

$$\dot{x} = \left(\alpha_1(\hat{A}_1 + \delta A_\delta) + \alpha_2(\hat{A}_2 + \delta A_\delta) \right) x \quad (4.30)$$

where

$$A_\delta = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (4.31)$$

As a result of **quadstab**, the closed-loop system (4.30) is quadratically unstable.

Having determined the reasons of the closed-loop instability for the LPV controller with the original nonlinear plant, the problem can be solved by simply increasing the conservativeness of the LPV plant model. That is, by setting a new range of the time-varying parameter to cover the uncertainty in the region close to the right-half s -plane. For example, setting $\theta = |n_2|$ to vary from -1 to 10, indicates that the varying pole can vary from -20 to 2 even though, in fact, it can only vary from -20 to 0. Using **hinfgs** with the same weighting function as previously but with the new range of θ , the new LPV controller is obtained with $\gamma = 0.1463$. The simulation results of the nonlinear closed-loop system with the new Jacobian-based LPV controller are presented in Figure 4.7. The closed-loop instability disappears but the transient performance is degraded because of setting a more conservative range of θ .

4.5 Pole Placement Approach

Pole placement with state feedback can be used to overcome the closed-loop instability problem without degrading the transient performance. In this approach, we restrict ourselves to special LPV plants of the form

$$\begin{aligned} \dot{x} &= A(\theta)x + Bu \\ A(\theta) &= A_0 + \theta_1 A_1 + \cdots + \theta_n A_n \\ y &= Cx \end{aligned} \quad (4.32)$$

where $A(\theta)$ is known functions and depends affinely on time-varying parameters, θ . Furthermore, in order to apply state feedback and state observer, this LPV

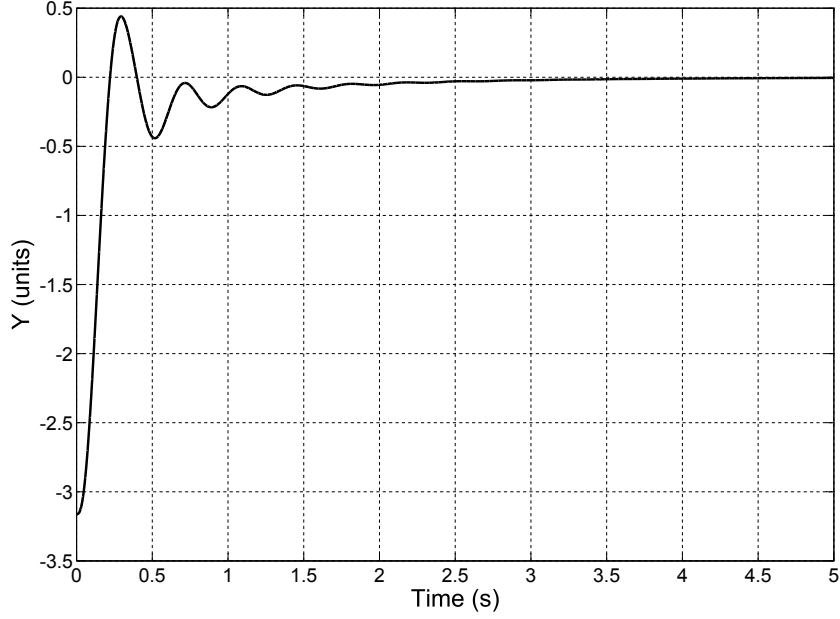


Figure 4.7: Nonlinear step response from -3.16 to 0 of new Jacobian-based LPV controller with the original nonlinear plant

plant is assumed to be state controllable and observable for all possible parameters trajectories θ . Figure 4.8 shows a block diagram structure of the gain-scheduled pole placement controller.

We apply state feedback $u = -K(\theta)x + n$. The state feedback gain, $K(\theta)$, is parameter-dependent and can be a nonlinear function of θ . Substituting u in (4.32), the state feedback closed-loop system becomes

$$\begin{aligned}\dot{x} &= \tilde{A}(\theta)x + Bn \\ \tilde{A}(\theta) &= [A(\theta) - BK(\theta)] \\ y &= Cx\end{aligned}\tag{4.33}$$

where n is the new input of the state feedback closed-loop system. By determining the state feedback gain, it is possible to achieve any closed-loop eigenvalue assignment. However for the example, the states cannot be measured. Hence, a state observer is used to estimate state values. A general state observer can be constructed using observer feedback gain $K_e(\theta)$ which is parameter-dependent and can be a nonlinear function of θ . The state observer closed-loop system is given by

$$\dot{x}_e = [A(\theta) - K_e(\theta)C]x_e + Bn + K_e(\theta)Cx\tag{4.34}$$

Subtracting (4.34) from (4.32), we obtain

$$\dot{x} - \dot{x}_e = [A(\theta) - K_e(\theta)C](x - x_e)\tag{4.35}$$

To demonstrate the method, we consider the Jacobian-based LPV model (4.5) taken

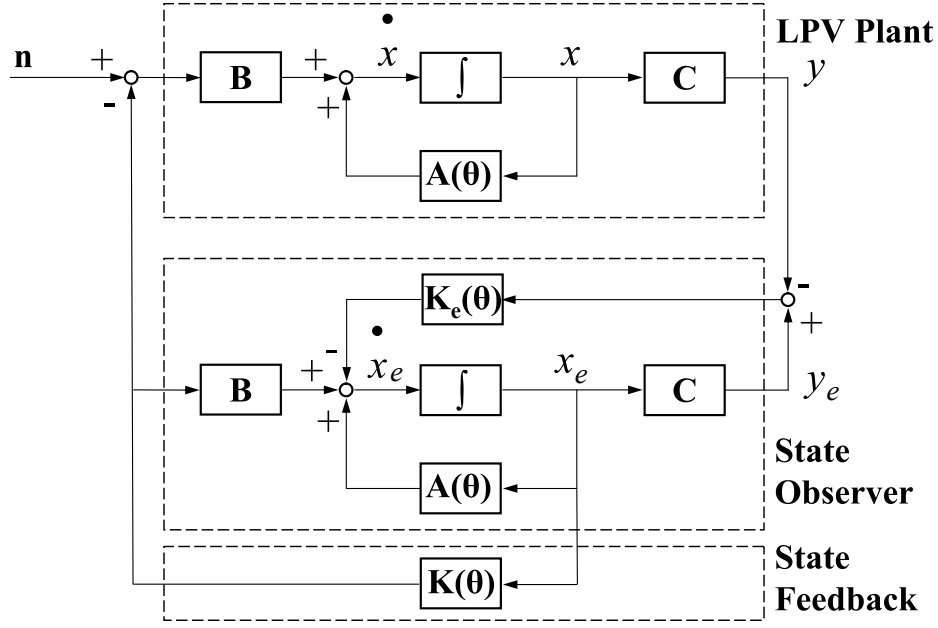


Figure 4.8: Block diagram structure of the gain-scheduled pole placement controller

from Leith and Leithead [61]. We also select the state feedback closed-loop system to have a realistic closed loop characteristic, i.e. natural frequency, $\omega_n = 10$ rad/s and damping ratio, $\xi = 0.707$, in order to prevent actuator saturation. Hence, the desired characteristic equation can be written as

$$\lambda^2 + (2 \times 0.707 \times 10)\lambda + 10^2 = 0 \quad (4.36)$$

The state feedback gain $K(\theta)$ can be determined by solving

$$\left| \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} - \left\{ \begin{bmatrix} -1 & 0 \\ 1 & -2\theta \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix} [k_1 \ k_2] \right\} \right| = 0$$

$$\lambda^2 + (2\theta + k_1 + 1)\lambda + (2\theta[1 + k_1] + k_2) = 0 \quad (4.37)$$

Equating coefficients of the polynomial yields the state feedback gain as

$$k_1 = 13.14 - 2\theta, \ k_2 = 4\theta^2 - 28.28\theta + 100 \quad (4.38)$$

Having determined the state feedback gain $K(\theta)$, the observer feedback gain $K_e(\theta)$ can be determined by solving

$$\left| \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} - \left\{ \begin{bmatrix} -1 & 0 \\ 1 & -2\theta \end{bmatrix} - \begin{bmatrix} k_{e1} \\ k_{e2} \end{bmatrix} [0 \ 1] \right\} \right| = 0$$

$$\lambda^2 + (2\theta + k_{e2} + 1)\lambda + (2\theta + k_{e1} + k_{e2}) = 0 \quad (4.39)$$

The dynamics of the state observer must be faster than the system being controlled. Therefore we select the state observer to have a suitable characteristic, i.e. $\omega_n = 40$

rad/s and $\xi = 0.707$, in order to avoid amplifying the noise of the controlled output. Then, the desired characteristic equation can be written as

$$\lambda^2 + (2 \times 0.707 \times 40)\lambda + 40^2 = 0 \quad (4.40)$$

Equating the coefficients of the polynomial yields the observer feedback gain as

$$k_{e1} = 1543.44, \quad k_{e2} = 55.56 - 2\theta \quad (4.41)$$

Substituting the matrices $A(\cdot)$, $C(\cdot)$ and $K_e(\theta)$ of the state observer in (4.35) yields a state-space form as

$$\begin{bmatrix} \dot{n}_1 - \dot{n}_{e1} \\ \dot{n}_2 - \dot{n}_{e2} \end{bmatrix} = \begin{bmatrix} -1 & -1543.44 \\ 1 & -55.56 \end{bmatrix} \begin{bmatrix} n_1 - n_{e1} \\ n_2 - n_{e2} \end{bmatrix} \quad (4.42)$$

The state observer has two poles at $-28.3 \pm 28.3i$. Since both poles are in the left-half s -plane, the state observer is stable. Having applied the state observer and the state feedback to the Jacobian-based LPV model, the state feedback closed-loop system can be determined by substituting the matrices $A(\cdot), \dots, C(\cdot)$ and $K(\theta)$ in (4.33) yielding the state-space form as

$$\begin{aligned} \begin{bmatrix} \dot{n}_1 \\ \dot{n}_2 \end{bmatrix} &= \begin{bmatrix} 2\theta - 14.14 & -4\theta^2 + 28.28\theta - 100 \\ 1 & -2\theta \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} n \\ y &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \end{aligned} \quad (4.43)$$

where n is the new input of state feedback closed-loop system. Substituting the matrices $A(\cdot), \dots, C(\cdot)$ of the state feedback closed-loop system into (4.26) gives the transfer function

$$G(s) = \frac{1}{s^2 + 14.14s + 100} \quad (4.44)$$

The state feedback closed-loop system has two constant poles at $-7.07 \pm 7.07i$. Both poles are in the left-half s -plane therefore the state feedback closed-loop system is stable.

According to the state feedback closed-loop transfer function, this approach shows that the parameters variation of the special LPV plants can be cancelled. In order to obtain performance from the system, an additional linear time invariant (LTI) controller can be applied as an outer loop as shown in Figure 4.9. For this particular example, an H_∞ -mixed-sensitivity controller is used. Using MATLAB Robust Control Toolbox function [12], `mixsyn`, with the same weighting function as previously, the following H_∞ controller is obtained with $\gamma = 0.0761$ for the state feedback closed-loop system as

$$\begin{aligned} \dot{x}_k &= A_k x_k + B_k (y_{ref} - y) \\ n &= C_k x_k \end{aligned} \quad (4.45)$$

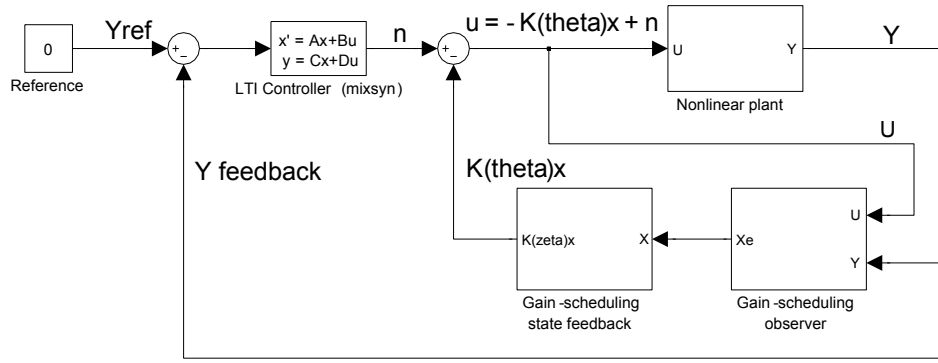


Figure 4.9: Pole placement controller design in MATLAB Simulink environment

where

$$A_k = \begin{bmatrix} -0.002 & 0 & 0 & 0 \\ 1355.3 & -61.186 & -14.911 & -38.782 \\ 84.705 & 58.676 & -15.072 & -14.924 \\ 0 & 0 & 8 & -2.662e-016 \end{bmatrix}$$

$$B_k = \begin{bmatrix} 0.079057 \\ 0 \\ -1.296e-016 \\ -8.4181e-017 \end{bmatrix}, \quad C_k = [2142.9 \quad 1484.4 \quad -23.577 \quad -61.319]$$

The simulation results of the nonlinear closed-loop system are presented in Figures 4.10 and 4.11. The closed-loop instability problem is solved with good transient performance of the system.

4.6 Conclusion

The example from Leith and Leithead [61] is very interesting. The closed-loop instability of the LPV controller with the original nonlinear model occurs because the mismatch uncertainty between the Jacobian-based LPV model and the original nonlinear model is in a region close to the right-half s -plane. In addition, for this particular example, both function substitution and state transformation methods give an LPV plant model that more accurately represents the nonlinear plant than the Jacobian linearization method. In this chapter, a design method for cancelling the parameters variation of the example of Leith and Leithead [61] by pole-placement state feedback is proposed. For the example, the approach yields reliable closed-loop stability and good closed-loop transient performance of the system because it makes the nonlinear plant appear to be an LTI plant, hence well-developed LTI tools can be applied.

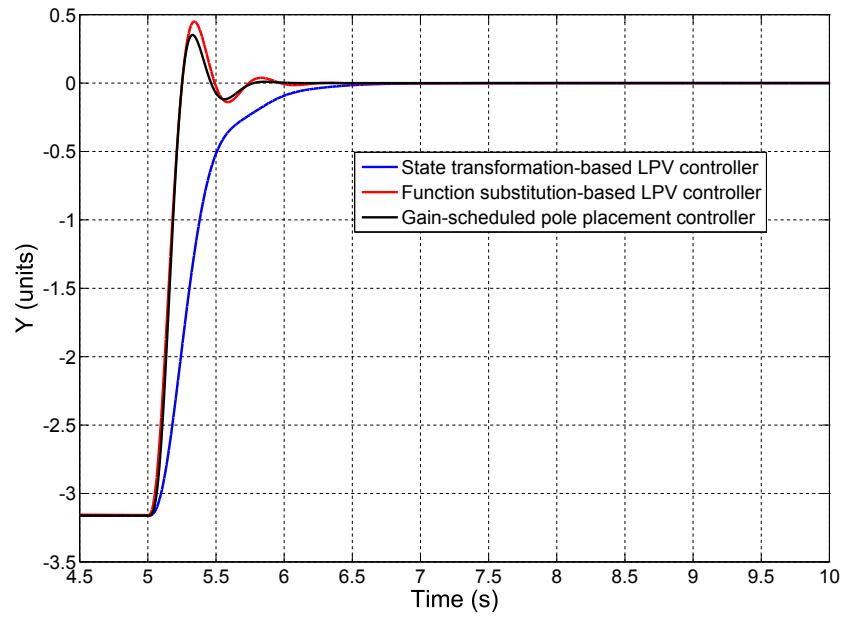


Figure 4.10: Nonlinear step response from -3.16 to 0 with the original nonlinear plant

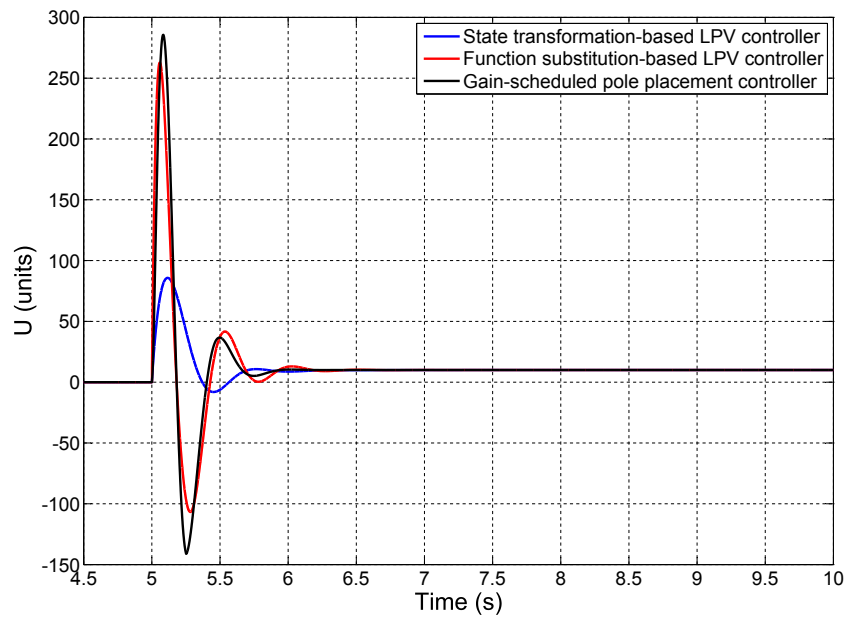


Figure 4.11: Control input to the original nonlinear plant

Chapter 5

Longitudinal LPV Autopilot Design: A TP Approach

This chapter describes a design of a longitudinal autopilot, i.e. speed-hold and altitude-hold, for the entire flight envelope of a Jindivik UAV using a linear parameter-varying (LPV) technique that is based on a gain-scheduled output feedback H_∞ control [10] for which an LPV model is required for control synthesis and analysis. Typically, LPV models are derived from original nonlinear equation models using an algebraic manipulation method (see section 3.1). However, the derived LPV models from those methods are often nonlinearly dependent on the time-varying parameters; this is especially the case for a longitudinal Jacobian-based LPV model that is nonlinearly dependent on speed and altitude.

A grid LPV model is usually used to synthesize a controller in the case of nonlinear parameter dependence, however, the result of heuristic gridding technique is unreliable and the analysis result is dependent on choosing the gridding points [96]. To synthesize an LPV autopilot with a finite number of LMIs and avoid the gridding technique, the TP model transformation is employed in order to transform a longitudinal nonlinearly parameter-dependent LPV model into a TP convex polytopic model form. Therefore, based on single quadratic Lyapunov functions (Theorem 3.4.3), the LMIs need only be evaluated at all vertices as shown in (3.80)–(3.82).

5.1 Jacobian-Based Longitudinal LPV Model

An LPV model is required for gain-scheduled H_∞ performance analysis and synthesis. Chapters 3 and 4 show that the state transformation-based LPV model more accurately represents the original nonlinear dynamics than the Jacobian-based LPV model that only accurately represents the nonlinear dynamics about the neighborhood of a set of equilibrium points. However, the state transformation approach requires sufficient data in order to derive a transformation-based LPV model from a nonlinear model. In addition, the transformation-based LPV model is a complex

structure model. Therefore, in this work, the Jacobian method is employed to derive a longitudinal LPV model from the standard 6-DOF equations of motion because it is applicable to a general class of nonlinear models and has simple structure model.

Consider the standard 6-DOF equations of motion for a conventional fixed wing aircraft, (A.10) and (A.11), where detail of the aircraft equations of motion is briefly summarized in section A.2. Although the exact form of aerodynamic coefficients structures for a general fixed wing aircraft are not certain, typical linear model structures are suggested in [55, 58] and [77]. The linear model structures, that are used in this thesis, from [55] are

$$C_X = C_{X_0} + C_{X_\alpha}\alpha + C_{X_q}\left(\frac{q\bar{c}}{2V}\right) + C_{X_{\delta_e}}\delta_e \quad (5.1)$$

$$C_Y = C_{Y_0} + C_{Y_\beta}\beta + C_{Y_p}\left(\frac{pb}{2V}\right) + C_{Y_r}\left(\frac{rb}{2V}\right) + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r \quad (5.2)$$

$$C_Z = C_{Z_0} + C_{Z_\alpha}\alpha + C_{Z_q}\left(\frac{q\bar{c}}{2V}\right) + C_{Z_{\delta_e}}\delta_e \quad (5.3)$$

$$C_l = C_{l_0} + C_{l_\beta}\beta + C_{l_p}\left(\frac{pb}{2V}\right) + C_{l_r}\left(\frac{rb}{2V}\right) + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r \quad (5.4)$$

$$C_m = C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}\left(\frac{q\bar{c}}{2V}\right) + C_{m_{\delta_e}}\delta_e \quad (5.5)$$

$$C_n = C_{n_0} + C_{n_\beta}\beta + C_{n_p}\left(\frac{pb}{2V}\right) + C_{n_r}\left(\frac{rb}{2V}\right) + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r \quad (5.6)$$

Equations (A.10)–(A.11) are 6-DOF dynamics in three-dimensional space, i.e. North-East-Down axis, which can often be simplified into two motions of 3-DOF dynamics in two-dimensional space, i.e. longitudinal and lateral motions. In this chapter, we only consider the longitudinal motion. In addition, we assume an aircraft is about a wings level and constant altitude and airspeed flight condition, and, assume it can manoeuvre only in North-Down plane, hence, all of the lateral states in (A.10)–(A.11) frozen and equal to zero, i.e. $v = p = r = \beta = \phi = \psi = y_E = 0$. Moreover, after substituting $\alpha = \tan^{-1}\left(\frac{w}{u}\right)$, $V = \sqrt{u^2 + w^2}$, and $\bar{q} = \frac{1}{2}\rho V^2$, we get equations of longitudinal motion as

$$\begin{aligned} \dot{u} = & \frac{\rho S}{2m} \left[C_{X_0} + C_{X_\alpha} \tan^{-1}\left(\frac{w}{u}\right) \right] u^2 + \frac{\rho S}{2m} \left[C_{X_0} + C_{X_\alpha} \tan^{-1}\left(\frac{w}{u}\right) \right] w^2 \\ & + \left[\left(\frac{\rho S \bar{c}}{4m} C_{X_q} \sqrt{u^2 + w^2} \right) - w \right] q - g \sin \theta + \frac{\rho S}{2m} (u^2 + w^2) C_{X_{\delta_e}} \delta_e + \frac{T}{m} \end{aligned} \quad (5.7)$$

$$\begin{aligned} \dot{w} = & \frac{\rho S}{2m} \left[C_{Z_0} + C_{Z_\alpha} \tan^{-1}\left(\frac{w}{u}\right) \right] u^2 + \frac{\rho S}{2m} \left[C_{Z_0} + C_{Z_\alpha} \tan^{-1}\left(\frac{w}{u}\right) \right] w^2 \\ & + \left[\left(\frac{\rho S \bar{c}}{4m} C_{Z_q} \sqrt{u^2 + w^2} \right) + u \right] q + g \cos \theta + \frac{\rho S}{2m} (u^2 + w^2) C_{Z_{\delta_e}} \delta_e \end{aligned} \quad (5.8)$$

$$\begin{aligned} \dot{q} = & \frac{\rho S \bar{c}}{2I_y} \left[C_{m_0} + C_{m_\alpha} \tan^{-1}\left(\frac{w}{u}\right) \right] u^2 + \frac{\rho S \bar{c}}{2I_y} \left[C_{m_0} + C_{m_\alpha} \tan^{-1}\left(\frac{w}{u}\right) \right] w^2 \\ & + \frac{\rho S \bar{c}}{4I_y} (\sqrt{u^2 + w^2}) C_{m_q} q + \frac{\rho S \bar{c}}{2I_y} (u^2 + w^2) C_{m_{\delta_e}} \delta_e \end{aligned} \quad (5.9)$$

$$\dot{\theta} = q \quad (5.10)$$

$$\dot{h} = (\sin \theta)u - (\cos \theta)w \quad (5.11)$$

In practice, the inertias I_x , I_y , and I_z of an UAV can be determined using a torsional pendulum experiment that is presented in [6, 36, 54] and [103] where the inertia I_{xz} is often neglected. In addition, the engine thrust T can be modelled and estimated using the technique that is presented in [54] and [55]. However, in this work, the thrust is modelled from data in Table 5.1, where the data are obtained by using MATLAB function, `trim`, to trim the Jindivik nonlinear model [41] about a wings level and constant altitude and airspeed flight condition. Based on *ad hoc* methods, a simple model of thrust, that fits the data in Table 5.1, is

$$T = C_{T_0} + C_{T_u} u + C_{T_{uh}} u \times h + C_{T_{\text{rpm}}} \text{rpm} + C_{T_{u\text{rpm}}} u \times \text{rpm} + C_{T_{h\text{rpm}}} h \times \text{rpm} + C_{T_{uh\text{rpm}}} u \times h \times \text{rpm} \quad (5.12)$$

where all thrust coefficients in the above equation are given in Table 5.2. Figure 5.1 shows a comparison of the simulated thrust, from Table 5.1, and the estimated thrust, from (5.12). Note that, the mathematical modelling of aerodynamic forces and moments, thrust, sensors, and actuators of the Jindivik nonlinear model [41] are given in section A.3. After (A.10)–(A.11) were linearized about a wings level and constant altitude and airspeed flight condition using Jacobian linearization method, we get a longitudinal LTI model as a state-space system of the form,

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} X_u & X_w & X_q & X_\theta & 0 \\ Z_u & Z_w & Z_q & Z_\theta & 0 \\ M_u & M_w & M_q & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ h_u & h_w & 0 & h_\theta & 0 \end{bmatrix} \begin{bmatrix} \tilde{u} \\ \tilde{w} \\ \tilde{q} \\ \tilde{\theta} \\ \tilde{h} \end{bmatrix} + \begin{bmatrix} X_{\delta_e} & X_{\delta_{\text{rpm}}} \\ Z_{\delta_e} & 0 \\ M_{\delta_e} & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{\delta_e} \\ \tilde{\delta_{\text{rpm}}} \end{bmatrix} \quad (5.13)$$

where $\tilde{u} = u - u_{\text{trim}}$, $\tilde{w} = w - w_{\text{trim}}$, $\tilde{q} = q$, $\tilde{\theta} = \theta - \theta_{\text{trim}}$, $\tilde{h} = h - h_{\text{trim}}$, $\tilde{\delta_e} = \delta_e - \delta_{e\text{trim}}$, $\tilde{\delta_{\text{rpm}}} = \text{rpm} - \text{rpm}_{\text{trim}}$, and

$$\begin{aligned} X_u &= \frac{\rho S}{m} (C_{X_0} + C_{X_\alpha} \alpha_{\text{trim}} + C_{X_{\delta_e}} \delta_{e\text{trim}}) u_{\text{trim}} - \frac{\rho S}{2m} C_{X_\alpha} w_{\text{trim}} \\ &\quad + \frac{1}{m} (C_{T_u} + C_{T_{uh}} h_{\text{trim}} + C_{T_{u\text{rpm}}} \text{rpm}_{\text{trim}} + C_{T_{uh\text{rpm}}} h_{\text{trim}} \text{rpm}_{\text{trim}}) \end{aligned} \quad (5.14)$$

$$X_w = \frac{\rho S}{m} (C_{X_0} + C_{X_\alpha} \alpha_{\text{trim}} + C_{X_{\delta_e}} \delta_{e\text{trim}}) w_{\text{trim}} + \frac{\rho S}{2m} C_{X_\alpha} u_{\text{trim}} \quad (5.15)$$

$$X_q = -w_{\text{trim}} + \frac{\rho S \bar{c}}{4m} C_{X_q} \sqrt{u_{\text{trim}}^2 + w_{\text{trim}}^2} \quad (5.16)$$

$$X_\theta = -g \cos \theta_{\text{trim}} \quad (5.17)$$

$$X_{\delta_e} = \frac{\rho S}{2m} C_{X_{\delta_e}} (u_{\text{trim}}^2 + w_{\text{trim}}^2) \quad (5.18)$$

$$X_{\delta_{\text{rpm}}} = \frac{1}{m} (C_{T_{\text{rpm}}} + C_{T_{u\text{rpm}}} u_{\text{trim}} + C_{T_{h\text{rpm}}} h_{\text{trim}} + C_{T_{uh\text{rpm}}} u_{\text{trim}} h_{\text{trim}}) \quad (5.19)$$

$$Z_u = \frac{\rho S}{m} (C_{Z_0} + C_{Z_\alpha} \alpha_{\text{trim}} + C_{Z_{\delta_e}} \delta_{e\text{trim}}) u_{\text{trim}} - \frac{\rho S}{2m} C_{Z_\alpha} w_{\text{trim}} \quad (5.20)$$

$$Z_w = \frac{\rho S}{m} (C_{Z_0} + C_{Z_\alpha} \alpha_{\text{trim}} + C_{Z_{\delta_e}} \delta_{e\text{trim}}) w_{\text{trim}} + \frac{\rho S}{2m} C_{Z_\alpha} u_{\text{trim}} \quad (5.21)$$

$$Z_q = u_{\text{trim}} + \frac{\rho S \bar{c}}{4m} C_{Z_q} \sqrt{u_{\text{trim}}^2 + w_{\text{trim}}^2} \quad (5.22)$$

$$Z_\theta = -g \sin \theta_{\text{trim}} \quad (5.23)$$

Table 5.1: Simulation data of Jindivik's engine*

Altitude (ft)	Speed (ft/s)	RPM	Thrust (lb _f)
2,000	337	8,543	294
2,000	443	9,140	378
2,000	548	9,992	533
2,000	654	10,814	717
2,000	759	11,500	927
6,000	337	8,675	290
6,000	432	9,126	342
6,000	527	9,828	453
6,000	622	10,556	597
6,000	717	11,214	759
10,000	337	8,900	293
10,000	421	9,130	311
10,000	506	9,672	386
10,000	590	10,296	495
10,000	675	10,894	618
14,000	337	9,255	303
14,000	411	9,294	298
14,000	485	9,579	340
14,000	559	10,038	408
14,000	632	10,571	500
18,000	337	9,602	331
18,000	400	9,408	293
18,000	464	9,532	306
18,000	542	9,853	348
18,000	590	10,247	403

* about wings level and constant altitude and air-speed flight condition

$$Z_{\delta_e} = \frac{\rho S}{2m} C_{Z_{\delta_e}} (u_{\text{trim}}^2 + w_{\text{trim}}^2) \quad (5.24)$$

$$M_u = \frac{\rho S \bar{c}}{I_y} (C_{m_0} + C_{m_\alpha} \alpha_{\text{trim}} + C_{m_{\delta_e}} \delta_{e\text{trim}}) u_{\text{trim}} - \frac{\rho S \bar{c}}{2I_y} C_{m_\alpha} w_{\text{trim}} \quad (5.25)$$

$$M_w = \frac{\rho S \bar{c}}{I_y} (C_{m_0} + C_{m_\alpha} \alpha_{\text{trim}} + C_{m_{\delta_e}} \delta_{e\text{trim}}) w_{\text{trim}} + \frac{\rho S \bar{c}}{2I_y} C_{m_\alpha} u_{\text{trim}} \quad (5.26)$$

$$M_q = \frac{\rho S \bar{c}^2}{4I_y} C_{m_q} \sqrt{u_{\text{trim}}^2 + w_{\text{trim}}^2} \quad (5.27)$$

$$M_{\delta_e} = \frac{\rho S \bar{c}}{2I_y} C_{m_{\delta_e}} (u_{\text{trim}}^2 + w_{\text{trim}}^2) \quad (5.28)$$

$$h_u = \sin \theta_{\text{trim}} \quad (5.29)$$

$$h_w = -\cos \theta_{\text{trim}} \quad (5.30)$$

$$h_\theta = u_{\text{trim}} \cos \theta_{\text{trim}} + w_{\text{trim}} \sin \theta_{\text{trim}} \quad (5.31)$$

Equations (5.14)–(5.31) are the stability and control derivatives (longitudinal

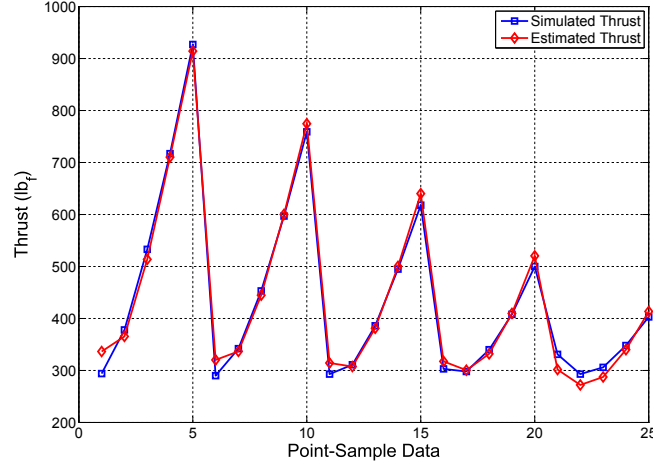


Figure 5.1: A comparison between the simulated thrust and the estimated thrust

Table 5.2: Jindivik's engine thrust coefficients

Coefficient	C_{T_0}	C_{T_u}	$C_{T_{rpm}}$	
Value	-278.86	-4.2915	0.12773	
Coefficient	$C_{T_{uh}}$	$C_{T_{urpm}}$	$C_{T_{hrpm}}$	$C_{T_{uhrpm}}$
Value	1.2676×10^{-4}	3.4457×10^{-4}	-3.6870×10^{-6}	-7.6625×10^{-9}

mode). The trim values in the above equations can be calculated by setting $\dot{u} = \dot{w} = \dot{q} = \dot{\theta} = \dot{h} = 0$ and $\rho = \rho_0 [1 - (6.876 \times 10^{-6})h]^{4.256}$ in (5.7)–(5.11). Moreover, since α is actually small for this flight condition, we assume $\tan \alpha \approx \alpha$ and $V \approx u$. Then, the results are

$$\alpha_{\text{trim}} = \frac{\frac{S}{2}u^2 \left(C_{Z_{\delta_e}} C_{m_0} - C_{Z_0} C_{m_{\delta_e}} \right) \left(\rho_0 [1 - (6.876 \times 10^{-6})h]^{4.256} \right) - mg C_{m_{\delta_e}}}{\frac{S}{2}u^2 \left(C_{Z_\alpha} C_{m_{\delta_e}} - C_{Z_{\delta_e}} C_{m_\alpha} \right) \left(\rho_0 [1 - (6.876 \times 10^{-6})h]^{4.256} \right)} \quad (5.32)$$

$$w_{\text{trim}} = u \tan \alpha_{\text{trim}} \quad (5.33)$$

$$\theta_{\text{trim}} = \alpha_{\text{trim}} \quad (5.34)$$

$$\delta_{e_{\text{trim}}} = \frac{-\left(C_{m_0} + C_{m_\alpha} \alpha_{\text{trim}} \right)}{C_{m_{\delta_e}}} \quad (5.35)$$

$$\text{rpm}_{\text{trim}} = \frac{mg \alpha_{\text{trim}} - \frac{S}{2}u^2 \left(C_{X_0} + C_{X_\alpha} \alpha_{\text{trim}} + C_{X_{\delta_e}} \delta_{e_{\text{trim}}} \right) \left(\rho_0 [1 - (6.876 \times 10^{-6})h]^{4.256} \right)}{\left(C_{T_{\text{rpm}}} + C_{T_{urpm}} u + C_{T_{hrpm}} h + C_{T_{uhrpm}} uh \right)} - \frac{\left(C_{T_0} + C_{T_u} u + C_{T_{uh}} uh \right)}{\left(C_{T_{\text{rpm}}} + C_{T_{urpm}} u + C_{T_{hrpm}} h + C_{T_{uhrpm}} uh \right)} \quad (5.36)$$

From (5.14)–(5.36), it can be seen that, the stability and control derivatives are

Table 5.3: Stability and control derivative data (longitudinal mode) of Jindivik nonlinear model*

u (ft/s)	464	464	464	497	497	497	531	531	531
h (ft)	9,900	10,500	11,100	9,900	10,500	11,100	9,900	10,500	11,100
α_{trim} (rad)	0.00655	0.00756	0.00860	0.00034	0.00124	0.00216	-0.00472	-0.00392	-0.00309
$\delta_{e\text{trim}}$ (rad)	0.0838	0.0832	0.0826	0.0872	0.0867	0.0861	0.0901	0.0896	0.0891
rpm_{trim}	9,445	9,460	9,467	9,630	9,631	9,632	9,818	9,813	9,808
X_u	-0.0127	-0.0125	-0.0128	-0.0141	-0.0138	-0.0135	-0.0146	-0.0143	-0.0139
X_w	0.0302	0.0323	0.0343	0.0142	0.0165	0.0187	0.0301	0.0319	0.0336
X_q	-3.1210	-3.5911	-4.0719	-0.2628	-0.7086	-1.1647	2.4346	2.0089	1.5736
X_θ	-32.1735	-32.1732	-32.1730	-32.1742	-32.1741	-32.1741	-32.1738	-32.1739	-32.1740
X_{δ_e}	-2.9187	-2.8380	-2.7585	-3.5278	-3.4351	-3.3438	-3.1418	-3.0558	-2.9710
$X_{\delta_{\text{rpm}}}$	0.00118	0.00116	0.00194	0.00207	0.00202	0.00198	0.00215	0.00211	0.00206
Z_u	-0.1191	-0.1178	-0.1166	-0.1175	-0.1161	-0.1148	-0.1163	-0.1149	-0.1134
Z_w	-1.4009	-1.3738	-1.3472	-1.4856	-1.4567	-1.4284	-1.5672	-1.5366	-1.5065
Z_q	462.88	462.90	462.92	496.56	496.59	496.61	530.23	530.26	530.29
Z_θ	-0.2107	-0.2435	-0.2770	-0.0110	-0.0399	-0.0696	0.1520	0.1261	0.0996
Z_{δ_e}	-40.7848	-40.0205	-39.2672	-46.7547	-45.8777	-45.0133	-53.1034	-52.1055	-51.1220
M_u	0.00017	0.00018	0.00020	0.00003	0.00005	0.00006	-0.00011	-0.00008	-0.00006
M_w	-0.0274	-0.0267	-0.0261	-0.0307	-0.0301	-0.0294	-0.0341	-0.0334	-0.0327
M_q	-0.7324	-0.7186	-0.7050	-0.7811	-0.7664	-0.7519	-0.8289	-0.8133	-0.7979
M_{δ_e}	-23.8180	-23.3709	-22.9303	-27.3123	-26.7993	-26.2936	-31.0254	-30.4422	-29.8674
h_u	0.0065	0.0075	0.0086	0.00034	0.0012	0.0021	-0.0047	-0.0039	-0.0030
h_w	-1	-1	-1	-1	-1	-1	-1	-1	-1
h_θ	464	464	464	497	497	497	531	531	531

* about wings level and constant altitude and airspeed flight condition

Table 5.4: Jindivik's longitudinal aerodynamic coefficients

Coefficient	C_{X_0}	C_{X_α}	C_{X_q}	$C_{X_{\delta_e}}$
Value	-0.0213	0.0898	-0.1355	-0.0202
Coefficient	C_{Z_0}	C_{Z_α}	C_{Z_q}	$C_{Z_{\delta_e}}$
Value	-0.1824	-4.8202	-2.1895	-0.3064
Coefficient	C_{m_0}	C_{m_α}	C_{m_q}	$C_{m_{\delta_e}}$
Value	0.0643	-0.4157	-5.2417	-0.7363

nonlinearly dependent on only speed and altitude. With the speed and altitude are fixed, (5.13) is a longitudinal LTI model. However, as the speed and altitude vary slowly over the entire flight envelope, (5.13) becomes a longitudinal nonlinearly parameter-dependent LPV model.

Moreover, these equations show that the accuracy of this LPV model depends on the accuracy of the information that provides the aerodynamic and thrust coefficients. Traditionally, the aerodynamic coefficients are often determined using wind tunnel tests by measuring the aerodynamic forces and moments introduced on the aircraft. However, the wind tunnel tests are expensive in terms of schedule and budget for UAV applications. System identification techniques are an alternative approach that can be used to estimate stability and control derivatives or aerodynamic coefficients from flight data, where the details of the method are presented in [28, 36, 53, 54, 57, 58, 74, 75] and [78].

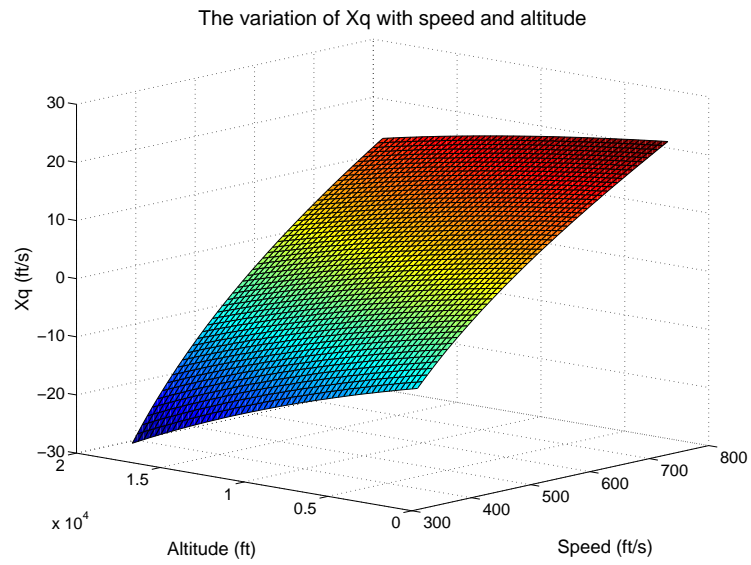
In this chapter, we use the MATLAB function, `linmod`, to emulate aircraft parameter identification techniques. Note that Appendix B presents the details of UAV aerodynamic model identification from a racetrack manoeuvre. Using functions `trim` and `linmod`, the stability and control derivative values, shown in Table 5.3, are obtained about one flight condition (speed = 497 ft/s and altitude = 10,500 ft). After substituting the data from Table 5.3 in (5.14)–(5.36), we can determine approximate aerodynamic coefficients of the Jindivik UAV as shown in Table 5.4 where the exact aerodynamic coefficients of the Jindivik nonlinear model [41] are given in sub-section A.3.1.

Knowing the aerodynamic coefficients, the system matrices of the longitudinal nonlinearly parameter-dependent LPV model (5.13) at all points over the entire parameter spaces can be determined using (5.14)–(5.36). Figure 5.2 shows the determined value of X_q and α_{trim} for a calculation example. In addition, Figure 5.3 shows the variation of open-loop characteristic of this nonlinearly parameter-dependent LPV model, i.e. $u(s)/rpm(s)$, over an entire flight envelope. According to Figure 5.3, two poles of short period mode are open-loop stable with variation of the damping ratio (ξ) and natural frequency (ω_n) from 0.256 to 0.319 and 2.42 rad/s to 7.19 rad/s respectively, where the other two poles of Phugoid mode are open-loop unstable with variation of the damping ratio and natural frequency from -0.109 to -0.288 and 0.0449 rad/s to 0.129 rad/s respectively. Moreover, the system of (5.13) also has non-minimum phase zeros as shown in Figure 5.3.

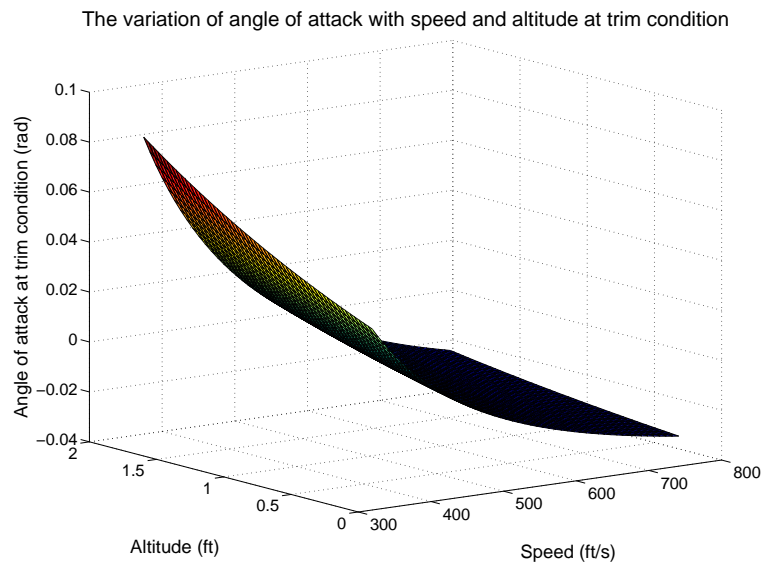
5.2 Longitudinal TP Convex Polytopic Model

Based on the system of (5.13), the speed and altitude are the only time-varying parameters. The entire flight envelope of the Jindivik UAV, taken from [41], is that the speed and altitude vary from 337.6 ft/s to 759.5 ft/s and 1,000 ft to 18,000 ft respectively. To synthesize an LPV controller for the system of (5.13) with a finite number of LMIs, the gridding technique that is presented in [39, 66, 99, 100] and [101] can be used. However, the result of heuristic gridding technique is unreliable and the analysis result is dependent on choosing the gridding points [96].

A TP model transformation is an alternative approach that can be used to obtain a finite number of LMIs for which the method transforms a given nonlinearly parameter-dependent LPV model (5.13) into a TP convex polytopic model. We applied the MATLAB Tensor Product Model Transformation Toolbox from [17] to determine the LTI vertex systems, S_a , and the weighting functions, $w_a(p(t))$, as shown in (3.26). The transformation space is defined as $\Omega = [337.6, 759.5] \times [1000, 18000]$ and let the density of the sampling grid be 500×100 . In addition, the weighting type of *cno* convex hull is used during the transformation in order to have a tight hull representation. A tensor of size $8 \times 6 \times 10 \times 7$ was received with the singular values in speed dimension as: 39198, 2554.5, 697.63, 49.012, 1.0486, 0.032623, 0.00050345, and 4.2483e-005. and in altitude dimension as: 39191, 2754.9, 103.23, 0.40544, 0.0018278, and 1.4039e-005.

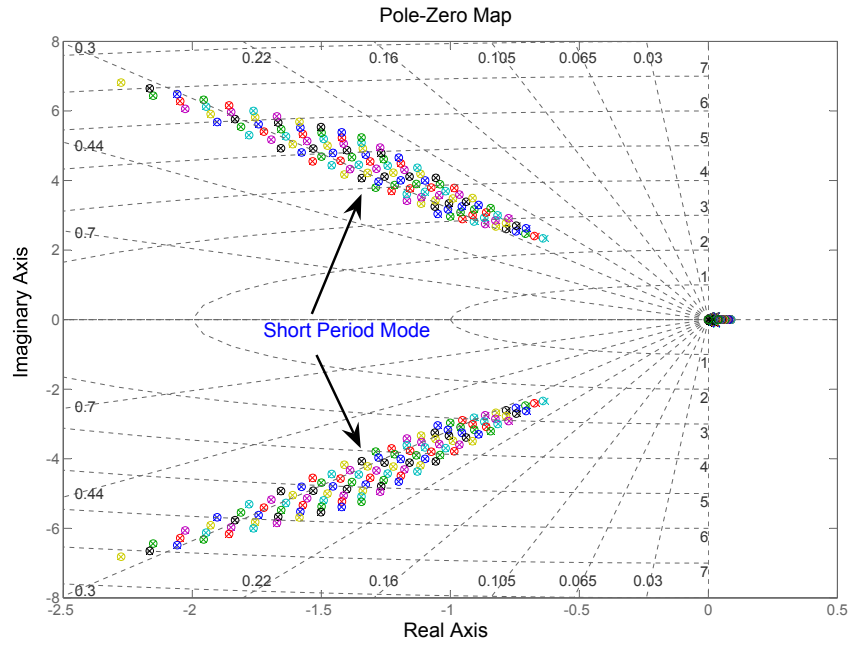


(a) X_q

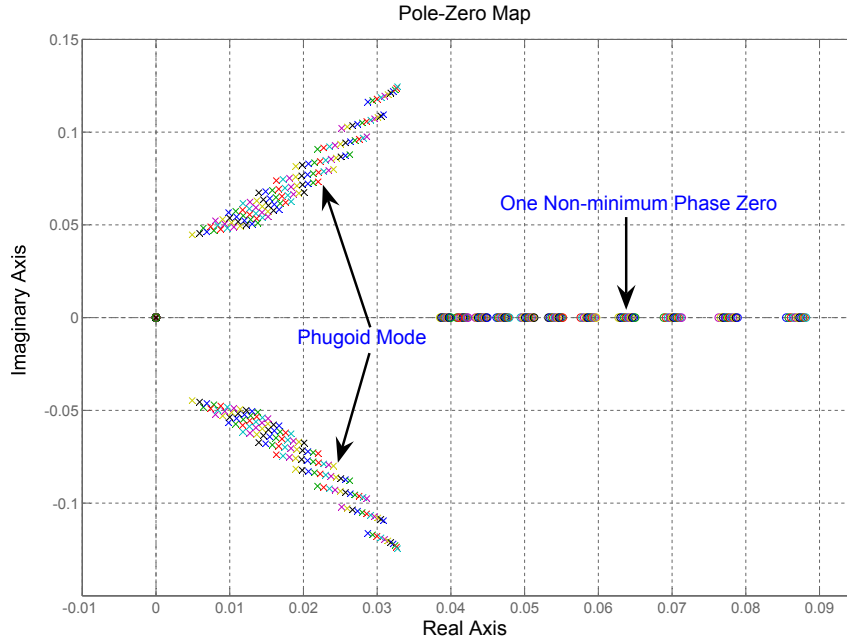


(b) α_{trim}

Figure 5.2: X_q and α_{trim} are nonlinearly dependent on speed and altitude



(a) Overview



(b) Zoom in

Figure 5.3: The open-loop characteristic of transfer function $\frac{u(s)}{rpm(s)}$ of the Jindivik longitudinal nonlinearly parameter-dependent LPV model

This means that the longitudinal nonlinearly parameter-dependent LPV model of the Jindivik UAV can exactly be given as a convex combination of $8 \times 6 = 48$ LTI vertex systems. However, in practice, a small number of controllers is preferred for implementation in real applications, therefore, we kept only the four and three largest singular values in speed and altitude dimension respectively. The number of LTI vertex systems was reduced to $5 \times 4 = 20$. Theoretically, the maximum error in L_2 matrix norm approximation is the sum of the discarded small singular values, thus, $1.0486 + 0.032623 + 0.00050345 + 4.2483e - 005 + 0.40544 + 0.0018278 + 1.4039e - 005 = 1.4891$.

However, we have compared the decomposed TP polytopic model with the original nonlinearly parameter-dependent LPV model, (5.13), over 2,000 test points of randomly selected parameter values, i.e. speed and altitude, in the ranges given by Ω . The maximum and mean error in the L_2 matrix norm, ϵ , was received as 0.0035904 and 0.0025777 respectively. Thus, the decomposed TP polytopic model can be reduced to a system of half the complexity while it is still accurate enough for real world experiments. Hence, the longitudinal TP polytopic model can be written as

$$\dot{x}(t) = \sum_{i=1}^5 \sum_{j=1}^4 w_{1,i}(u(t)) w_{2,j}(h(t)) (A_{i,j}x(t) + B_{i,j}u(t)) \quad (5.37)$$

where the weighting functions $w_{n,j}(p_n(t))$ are presented in Figure 5.4. Moreover, Figure 5.5 shows $w_1(p(t))$ and $w_2(p(t))$ as an example for determining $w_a(p(t))$. Some of the LTI system matrices, S_a , of this TP polytopic model are shown below

$$A_{1,1} = \begin{pmatrix} -0.0249 & 0.0226 & -9.9344 & -32.1718 & 0 \\ -0.1472 & -1.2368 & 336.4799 & -0.9429 & 0 \\ 0.0009 & -0.0258 & -0.6508 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0.0293 & -0.9999 & 0 & 337.6224 & 0 \end{pmatrix} \quad B_{1,1} = \begin{pmatrix} -1.7419 & 0.0021 \\ -26.4220 & 0 \\ -15.4316 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \quad (5.38)$$

$$A_{2,1} = \begin{pmatrix} -0.0279 & 0.0272 & -0.0268 & -32.1833 & 0 \\ -0.1322 & -1.4657 & 401.2744 & 0.3468 & 0 \\ 0.0001 & -0.0308 & -0.7761 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -0.0108 & -1.0003 & 0 & 402.4091 & 0 \end{pmatrix} \quad B_{2,1} = \begin{pmatrix} -1.7733 & 0.0023 \\ -26.8975 & 0 \\ -15.7093 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \quad (5.39)$$

$$A_{3,1} = \begin{pmatrix} -0.0389 & 0.0526 & 23.0197 & -32.1598 & 0 \\ -0.1656 & -2.7741 & 758.8454 & 0.9822 & 0 \\ -0.0017 & -0.0582 & -1.4677 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -0.0305 & -0.9996 & 0 & 761.7082 & 0 \end{pmatrix} \quad B_{3,1} = \begin{pmatrix} -8.8508 & 0.0034 \\ -134.2520 & 0 \\ -78.4090 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \quad (5.40)$$

$$\vdots$$

$$A_{5,4} = \begin{pmatrix} -0.0086 & 0.0134 & -26.0233 & -32.1294 & 0 \\ -0.1251 & -0.7550 & 307.4327 & -2.1679 & 0 \\ 0.0014 & -0.0156 & -0.3932 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0.0674 & -0.9986 & 0 & 308.6531 & 0 \end{pmatrix} \quad B_{5,4} = \begin{pmatrix} -0.7557 & 0.0013 \\ -11.4620 & 0 \\ -6.6943 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \quad (5.41)$$

5.3 Gain-Scheduled H_∞ Autopilot Design

In practice, the plant model is normally augmented with some weighting functions before we can apply the H_∞ control synthesis to compute an LPV controller. In

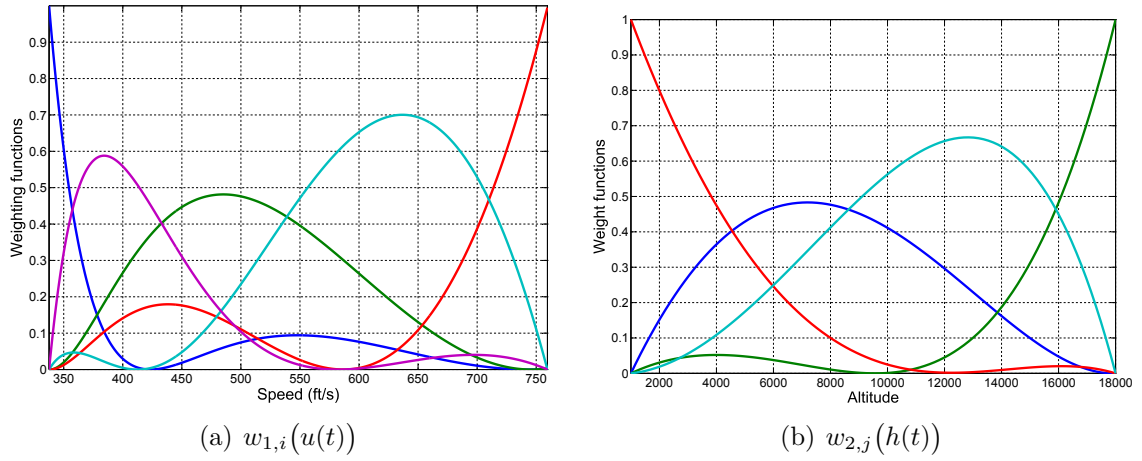


Figure 5.4: The *cno* type convex weighting functions in one-dimensional parameter, i.e. $w_{n,j}(p_n(t))$, of the longitudinal TP polytopic model

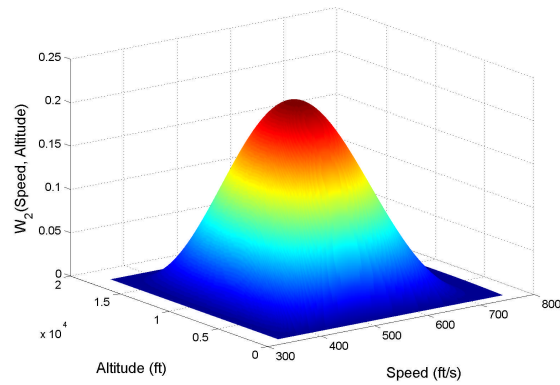
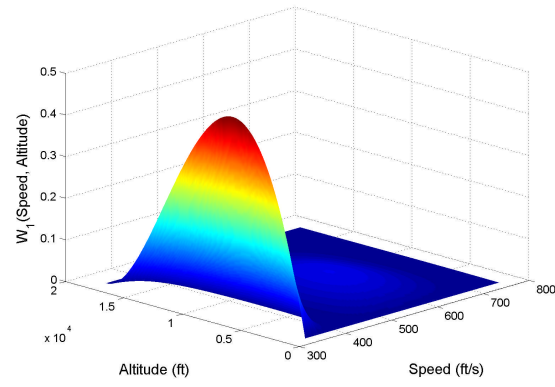


Figure 5.5: The *cno* type convex weighting functions in two-dimensional parameters, i.e. $w_a(p(t))$, of the longitudinal TP polytopic model

this chapter, we used the mixed-sensitivity criterion [20, 33, 47]

$$\left\| \begin{array}{c} W_1 S \\ W_2 K S \end{array} \right\|_\infty < 1 \quad (5.42)$$

The objective of this mixed-sensitivity function is to shape the sensitivity function S and control sensitivity function KS with performance weighting functions W_1 and robustness weighting functions W_2 respectively. Hence, we should get a controller that is good at command following (i.e. small tracking error), good at disturbance attenuation (i.e. attenuation of the effect of disturbance on output), low sensitivity to measurement noise, with reasonably small control efforts, and that is robustly stable to additive plant perturbations. Figure 5.6 shows the weighted open-loop interconnection for synthesis where

$$W_1(s) = \begin{pmatrix} \frac{0.5s+0.0664}{s+6.64 \times 10^{-5}} & 0 \\ 0 & \frac{0.5s+0.0664}{s+6.64 \times 10^{-5}} \end{pmatrix} \quad (5.43)$$

$$W_2(s) = \begin{pmatrix} \frac{100s+3.32}{0.001s+0.0664} & 0 \\ 0 & \frac{0.01s+3.32 \times 10^{-4}}{0.001s+0.0664 \times 10^{-5}} \end{pmatrix} \quad (5.44)$$

$$W_{pre-filter}(s) = \begin{pmatrix} \frac{500}{s+500} & 0 \\ 0 & \frac{1000}{s+1000} \end{pmatrix} \quad (5.45)$$

The purpose of $W_{pre-filter}$ is to make matrices B_2 and D_{12} of the plant model to be parameter-independent [10], hence, the gain-scheduled output feedback H_∞ controller design method of [10] can be used. In addition, the values of weighting functions W_1 and W_2 are hand-tuned until the desired objectives of performance and robustness of the closed-loop system are achieved.

Having augmented the longitudinal TP convex polytopic model, taken from (5.38)–(5.41), with weighting functions W_1 and W_2 , an LPV controller can be synthesized using the routine `hinfgs`. As a result of `hinfgs`, the LPV controller with $\gamma = 3.0395$ was obtained. Once the twenty LTI system matrix vertices of the LPV controller are obtained from the routine `hinfgs`, this LPV controller can be constructed by the combination of the system matrix vertices and weighting functions, $w_a(p(t))$, in the same fashion as the TP convex polytopic model, hence,

$$\begin{aligned} \dot{x}_k(t) &= \sum_{i=1}^5 \sum_{j=1}^4 w_{1,i}(u(t)) w_{2,j}(h(t)) \left(A_{k_{i,j}} x_k(t) + B_{k_{i,j}} \begin{bmatrix} u_{ref} - u(t) \\ w(t) \\ q(t) \\ \theta(t) \\ h_{ref} - h(t) \end{bmatrix} \right) \\ \begin{bmatrix} \delta_e(t) \\ rpm(t) \end{bmatrix} &= \sum_{i=1}^5 \sum_{j=1}^4 w_{1,i}(u(t)) w_{2,j}(h(t)) \left(C_{k_{i,j}} x_k(t) + D_{k_{i,j}} \begin{bmatrix} u_{ref} - u(t) \\ w(t) \\ q(t) \\ \theta(t) \\ h_{ref} - h(t) \end{bmatrix} \right) \end{aligned} \quad (5.46)$$

To confirm that the mixed-sensitivity criterion (5.42) is achieved, the singular values

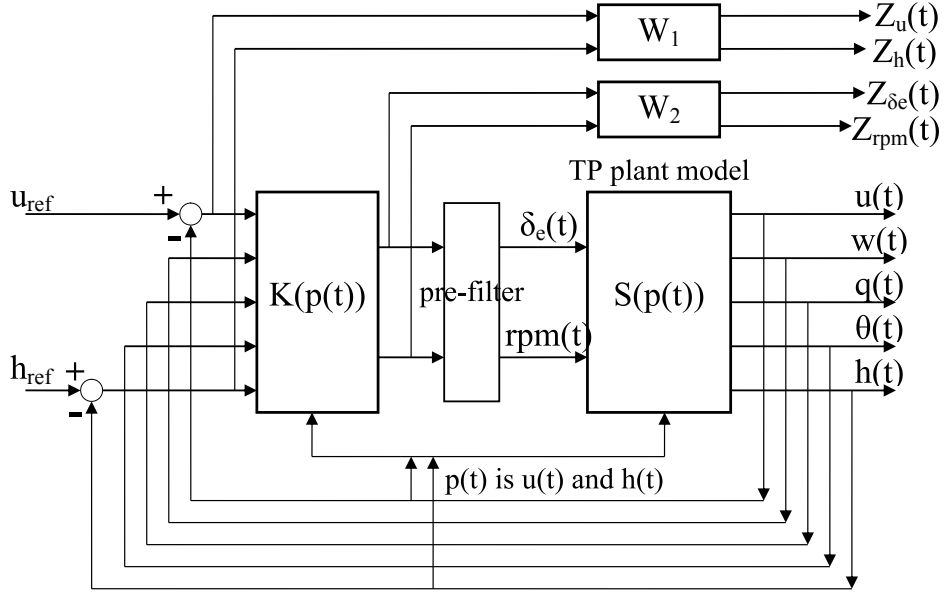


Figure 5.6: The weighted open-loop interconnection for the longitudinal TP convex polytopic plant model

of the transfer matrices S and KS are computed over all $\theta \in \Theta$ (with frozen values of θ), and are shown in Figure 5.7. Obviously, the singular values of S and KS are shaped and bound by W_1 and W_2 respectively. In addition, Figure 5.8 shows that $\|[W_1S, W_2KS]^T\|_\infty < \gamma$. Hence, the mixed-sensitivity criterion (5.42) is satisfied.

5.4 Nonlinear Simulation Results

The designed H_∞ gain-scheduling autopilot is validated with the Jindivik nonlinear model [41] in a MATLAB Simulink simulation. Note that all twenty weighting functions, $w_a(p(t))$, used in the simulation were constructed using two-dimensional look-up tables. In Figure 5.9, the transient response of the simulated vehicle for small demanded changes in speed and altitude are shown for one particular point in the flight envelope. Similar responses for other points in the flight envelope were obtained. Figure 5.10 shows a simulated flight that cover a wide range of the flight envelope. It demonstrates that the stability and robustness properties of the closed-loop system were achieved over the defined flight envelope.

Note that, there is an effect on regulating an altitude when the autopilot is tracking a speed demand as well as there is an effect on regulating a speed when the autopilot is tracking an altitude demand. This is because the longitudinal TP polytopic model is a quasi-LPV model where the scheduling parameters, speed and altitude, are also states of the system. This is a common problem for quasi-LPV models when synthesizing an LPV controller using single quadratic Lyapunov function since the parameter variation rate is as fast as the system states.

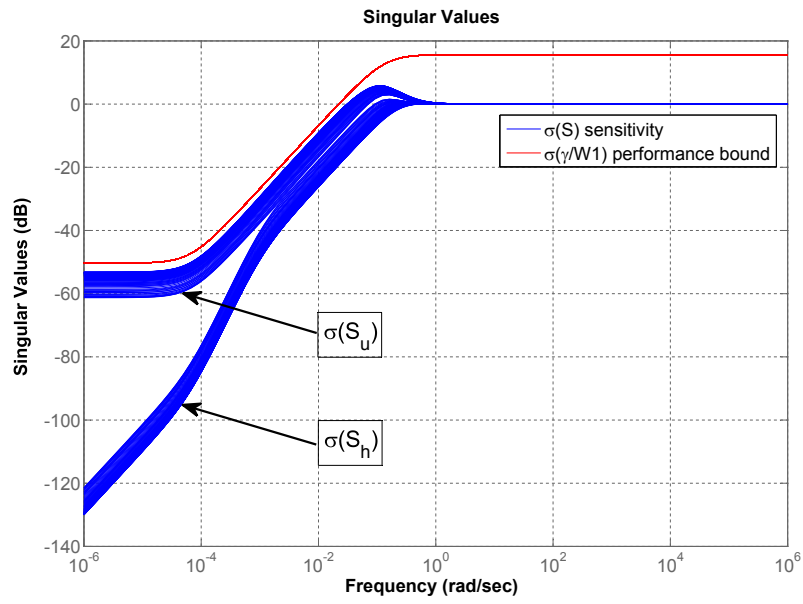
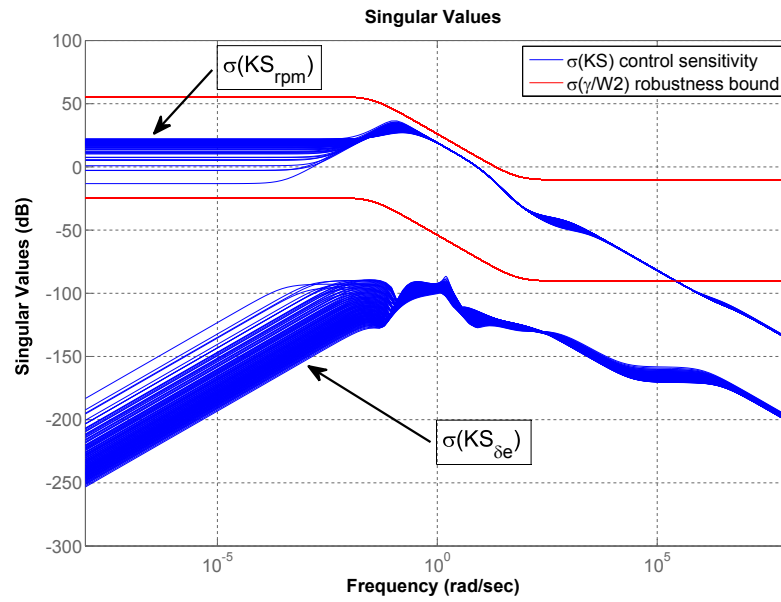
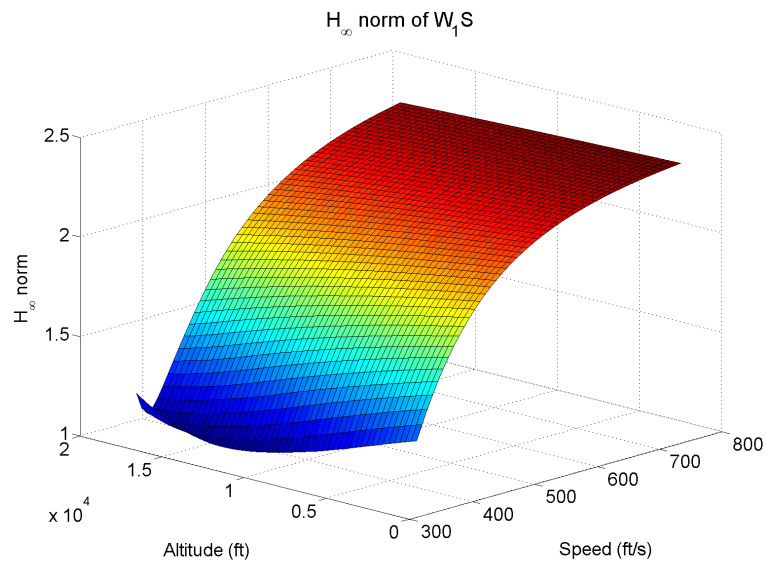
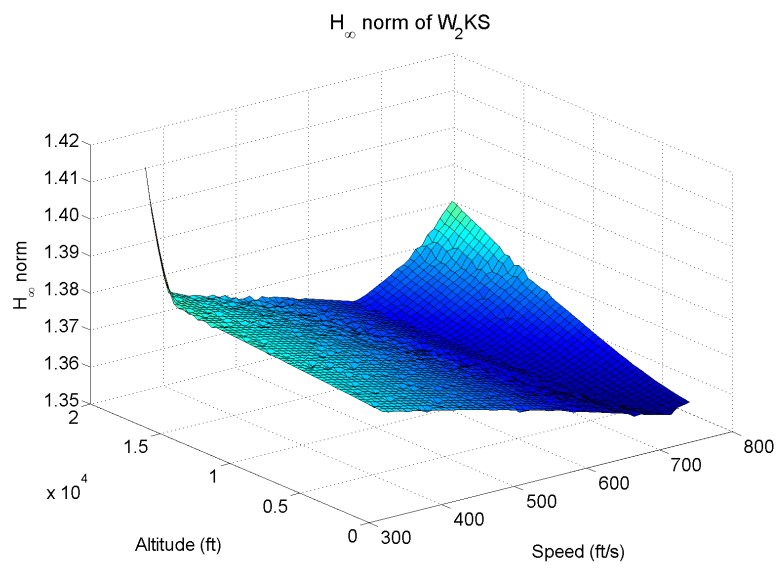
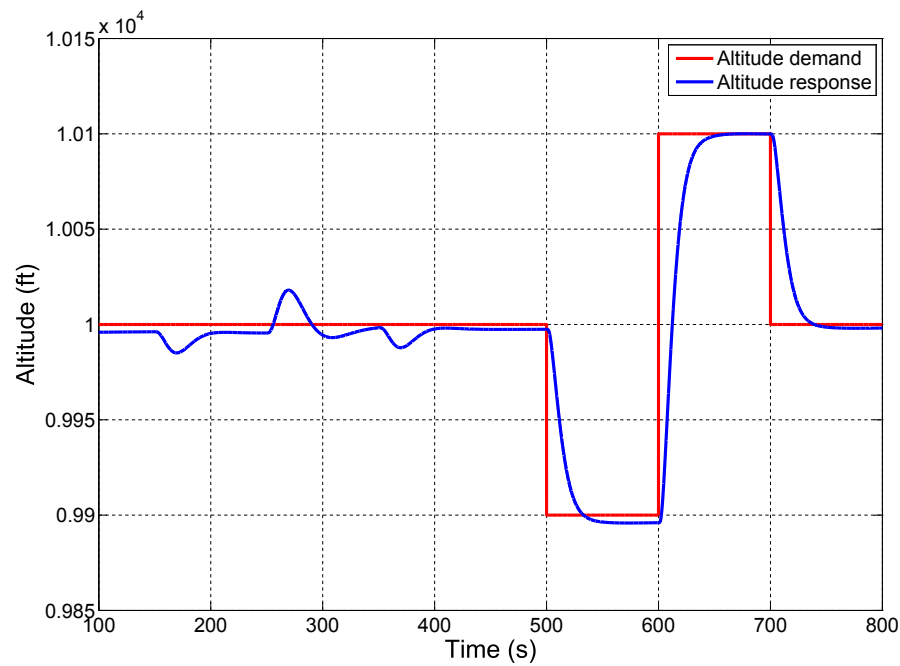
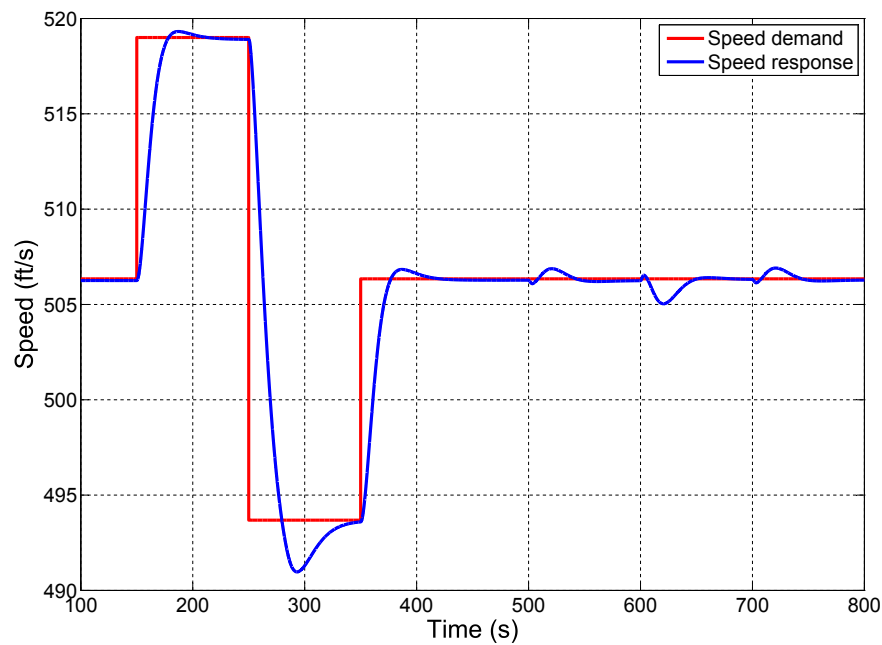

 (a) $\sigma(S)$

 (b) $\sigma(KS)$

 Figure 5.7: Singular value of S and KS over $\forall \theta \in \Theta$ (with frozen θ)

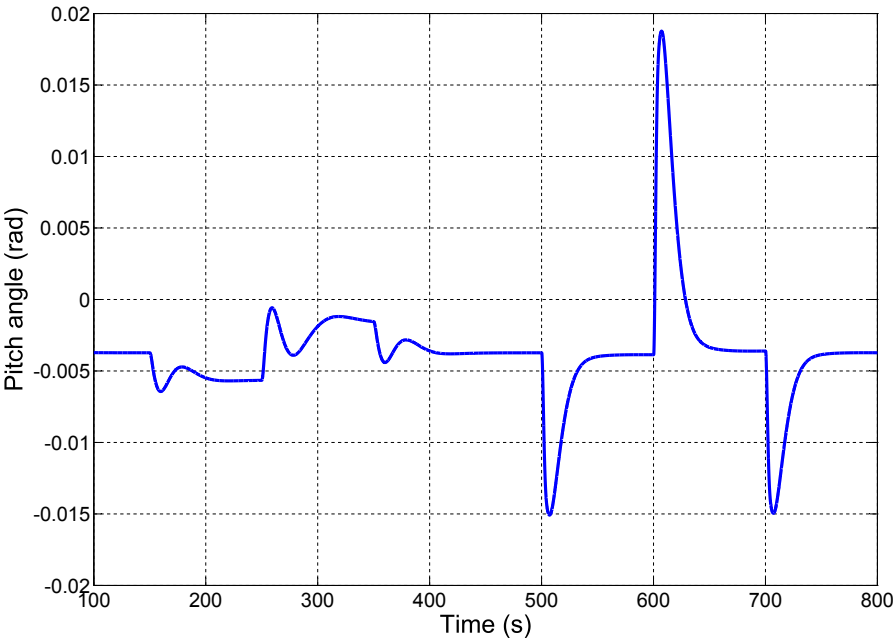
(a) $\|W_1 S\|_\infty$ (b) $\|W_2 K S\|_\infty$ Figure 5.8: H_∞ norm of $W_1 S$ and $W_2 K S$ over $\forall \theta \in \Theta$ (with frozen θ)



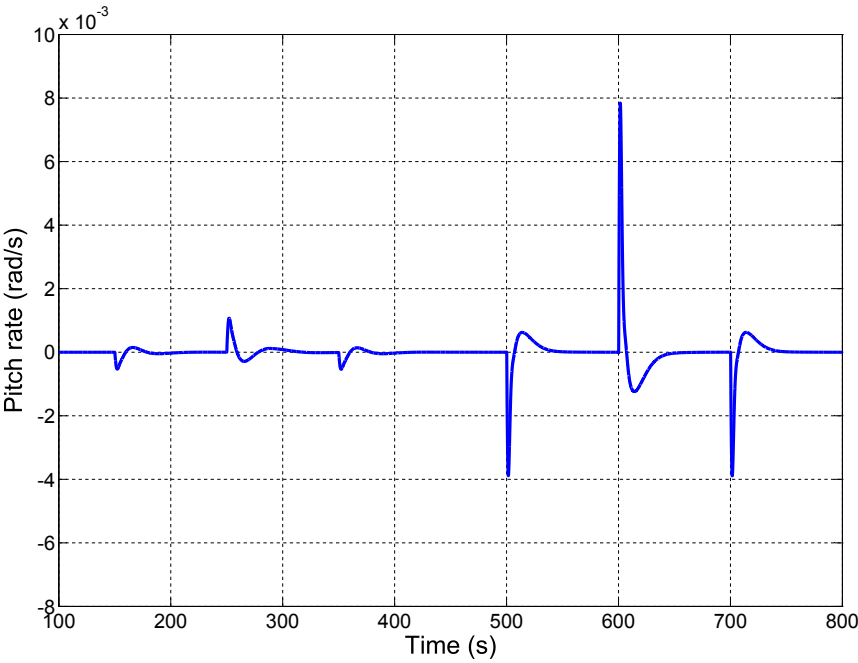
(a) Altitude (ft)



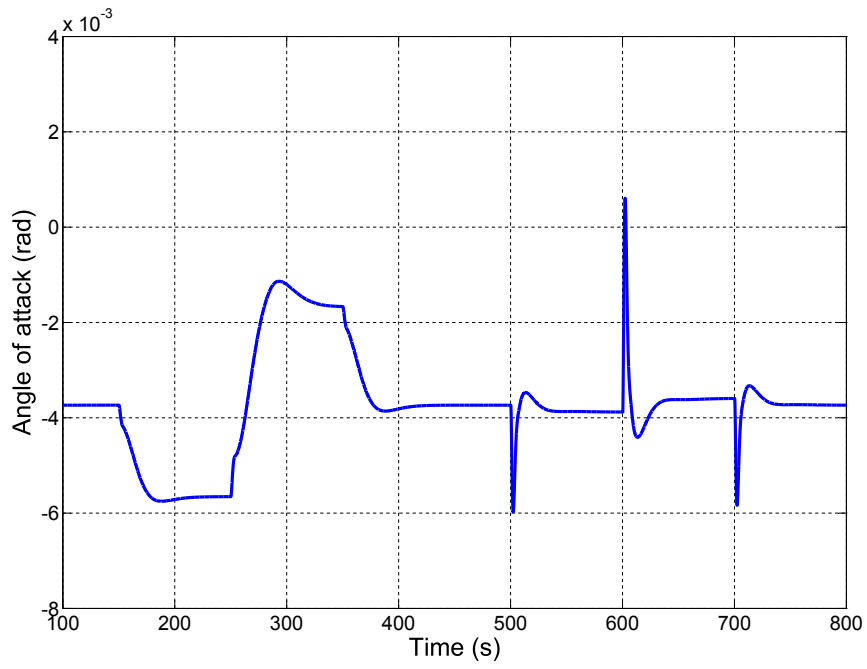
(b) Speed (ft/s)



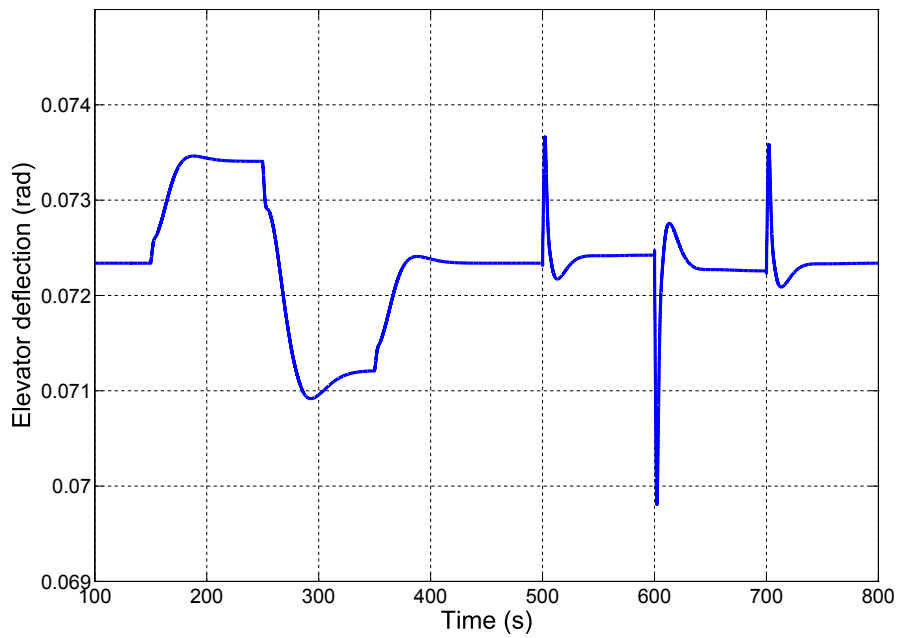
(c) Pitch angle (rad)



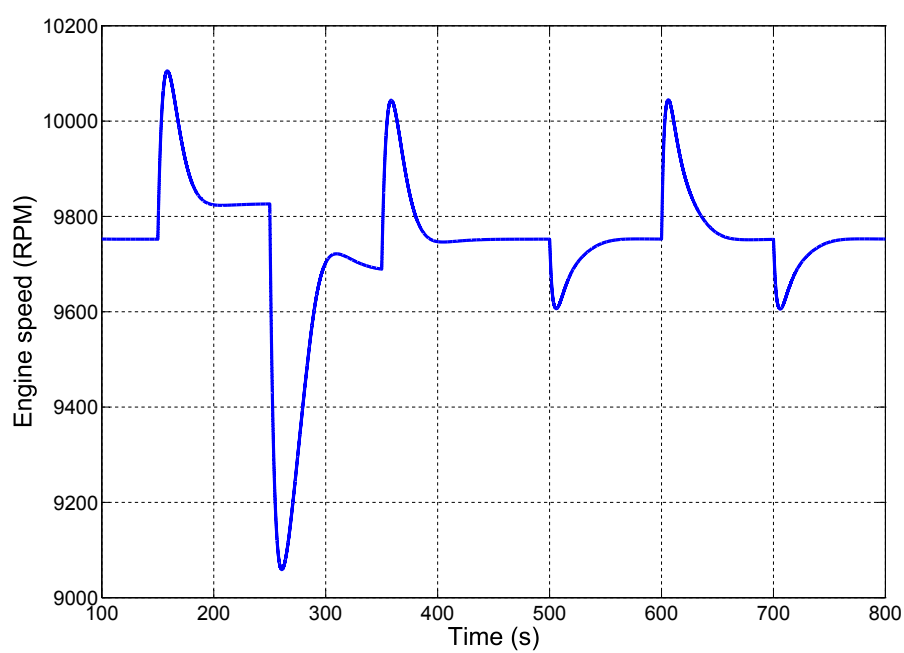
(d) Pitch rate (rad/s)



(e) Angle of Attack (rad)

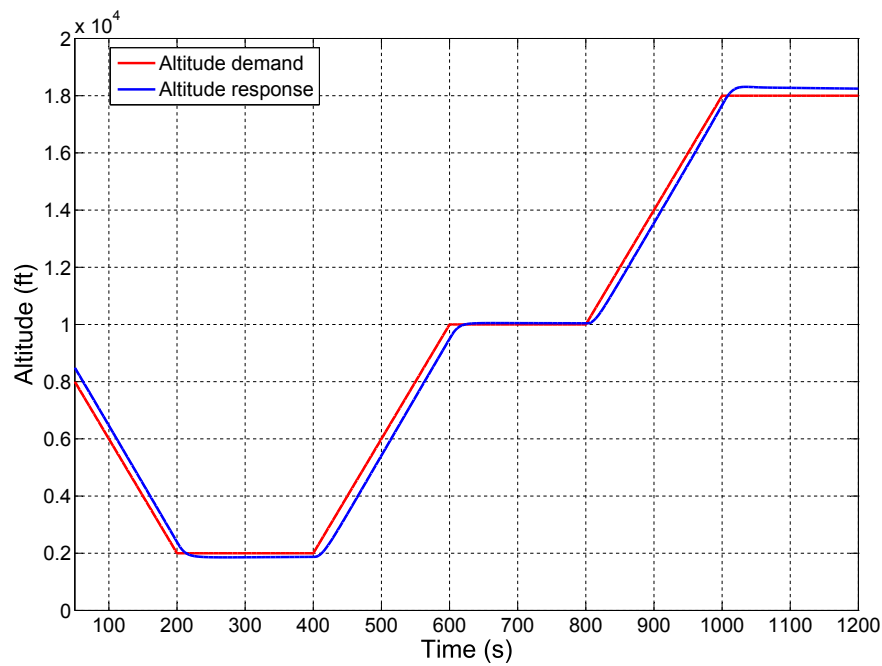


(f) Elevator deflection (rad)

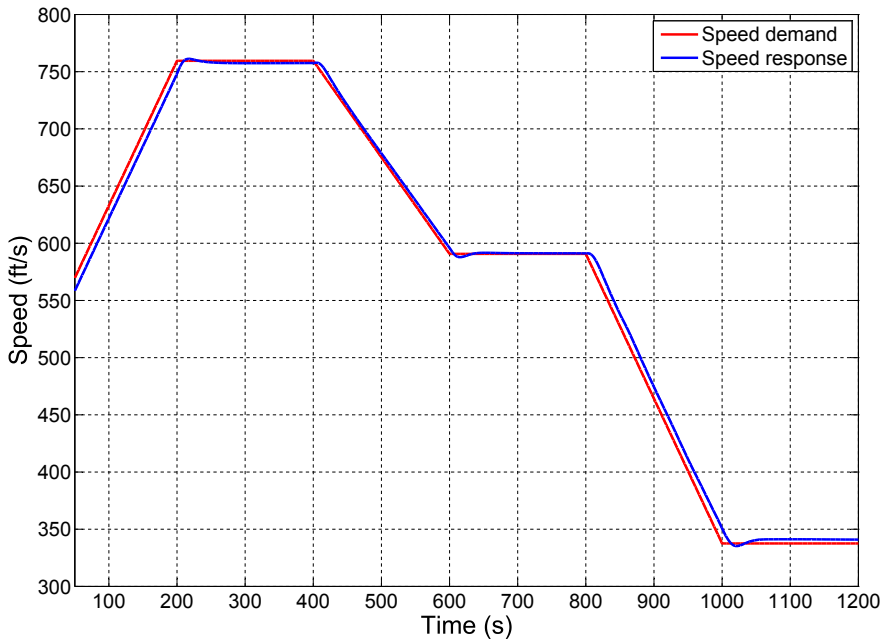


(g) Engine speed (RPM)

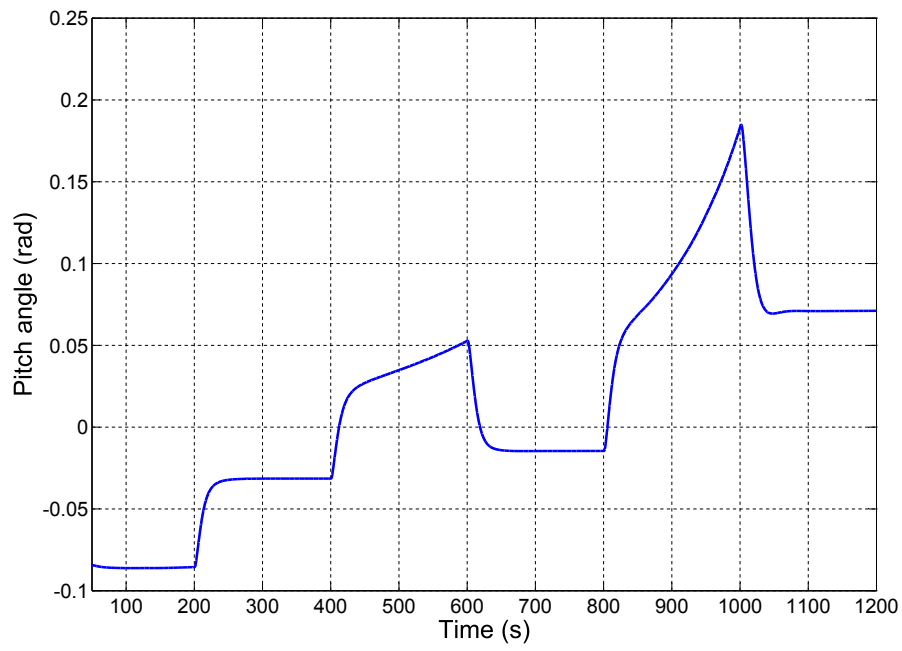
Figure 5.9: The transient performance of H_∞ gain-scheduling autopilot is validated with the Jindivik nonlinear dynamic model about one condition inside the flight envelope, i.e. speed = 506 ft/s and altitude = 10,000 ft



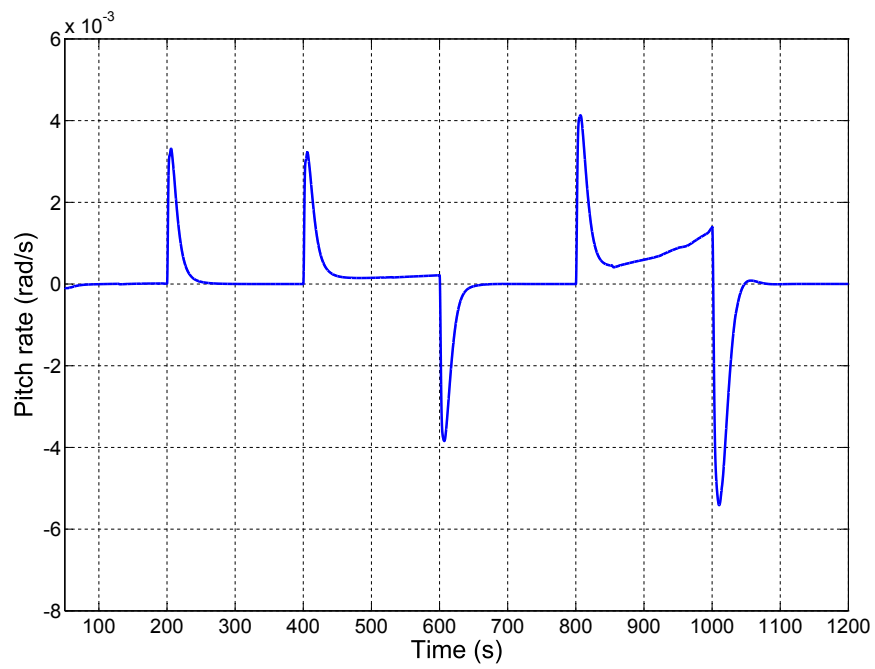
(a) Altitude (ft)



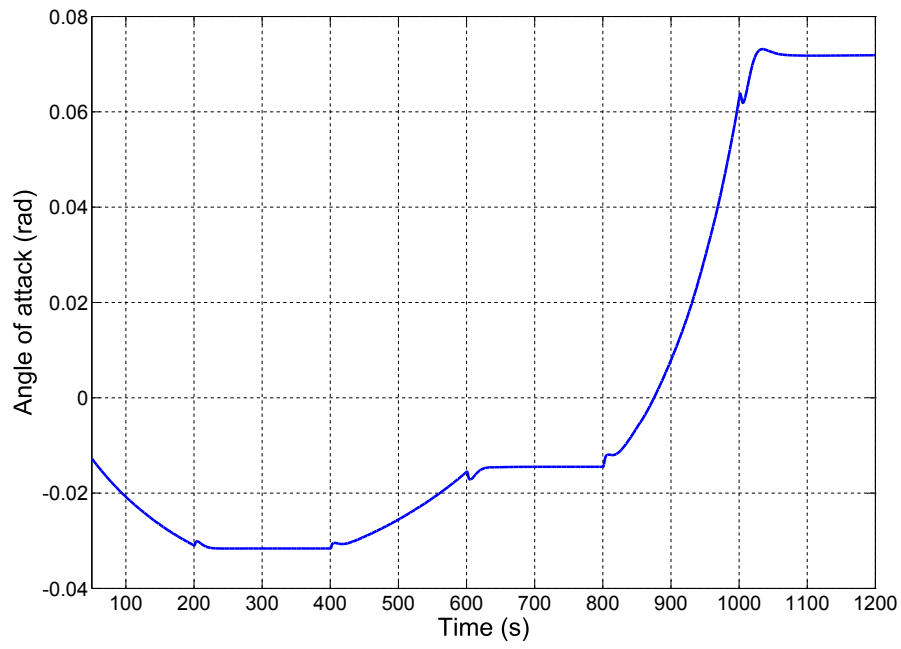
(b) Speed (ft/s)



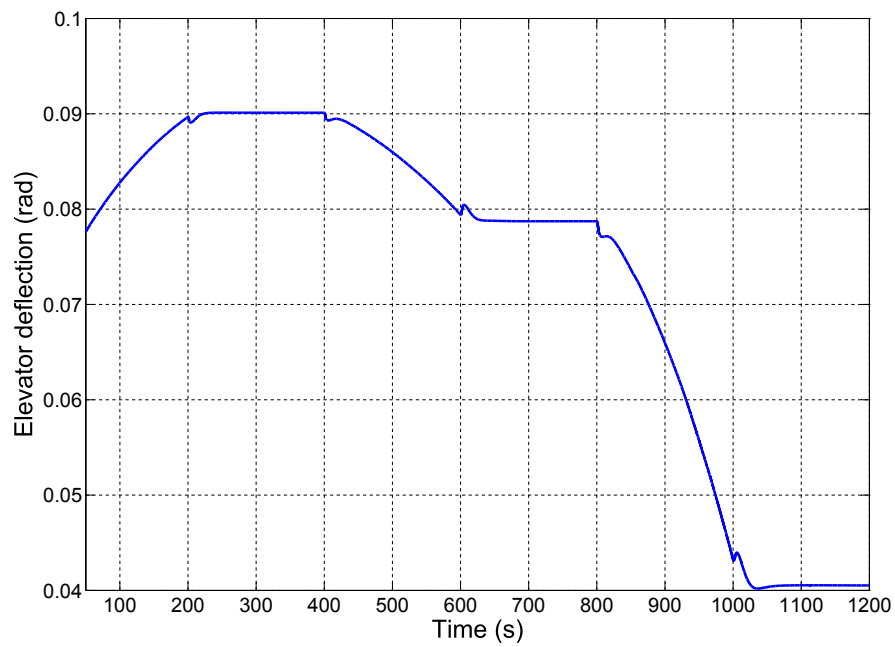
(c) Pitch angle (rad)



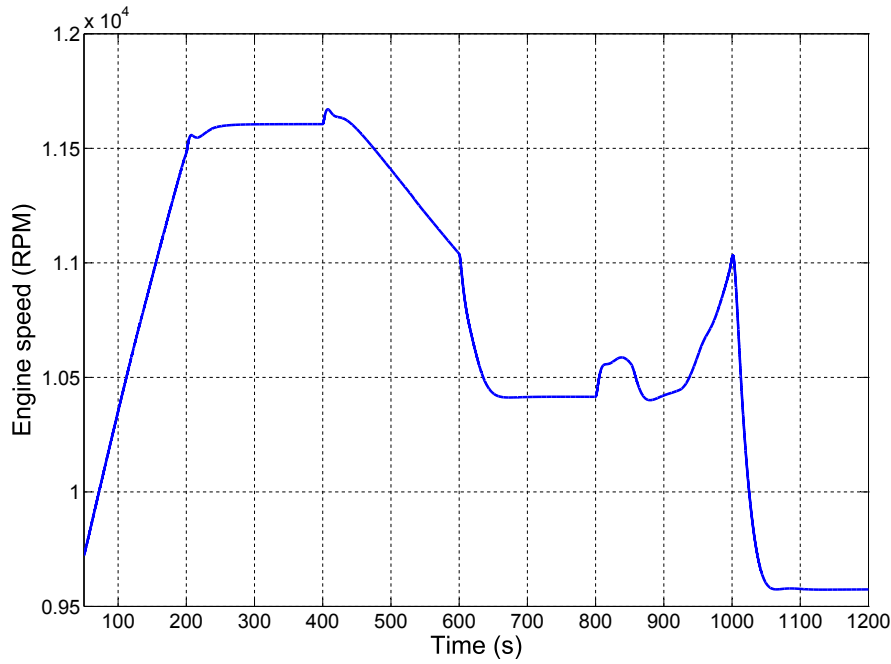
(d) Pitch rate (rad/s)



(e) Angle of Attack (rad)



(f) Elevator deflection (rad)



(g) Engine speed (RPM)

Figure 5.10: The stability and robustness properties of the closed-loop system were achieved over the defined flight envelope

5.5 Conclusion

A recently proposed technique, the tensor-product (TP) model transformation [18], is applied to generate a convex polytopic representation of a longitudinal nonlinearly parameter-dependent LPV model of the Jindivik UAV. The gain-scheduled output feedback H_∞ controller design method [10] was applied to the resulting TP convex polytopic model to yield a controller that guarantees the stability, robustness and performance properties of the closed-loop system over the whole grid. The method is relatively easy to apply owing to the availability of good computational tools [17] and [46]. The controller was tested with a full 6-DOF simulation of the vehicle. These results show that the stability and robustness properties of the closed-loop system were achieved over the defined flight envelope.

This page intentionally contains only this sentence.

Chapter 6

Longitudinal LPV Autopilot Design: A PDLF Approach

In general, a single quadratic Lyapunov function is more conservative than a parameter-dependent Lyapunov function when the parameters are time-invariant or slowly varying [45]. Hence, using the parameter-dependent Lyapunov function can reduce the conservatism of the designed LPV controller in the previous chapter and can also solve a problem of a quasi-LPV model (that the scheduling parameter is dependent on the system state) where both scheduling parameter and system state variations are at the same speed. However, to synthesize a gain-scheduled output feedback H_∞ controller for a class of affine (or polytopic) LPV plant model using the parameter-dependent Lyapunov function involves solving an infinite number of LMIs for which a number of convexifying techniques exist for obtaining a finite number of LMIs. In this chapter, an alternative approach for obtaining a finite number of LMIs is proposed, by simple manipulations on the bounded real lemma inequality, a symmetric matrix polytope inequality form is obtained. Hence, the LMIs need only be evaluated at all vertices.

A technique to construct the intermediate controller variables as an affine matrix-valued function in the polytopic coordinates of the scheduled parameter is also proposed. Computational results on a numerical example [61] using the proposed approach are compared with those from a multi-convexity approach [11] in order to demonstrate the impacts of the proposed method in the parameter-dependent Lyapunov-based stability and performance analysis. The proposed method is applied to synthesize a longitudinal autopilot for a full flight envelope of the Jindivik UAV. The resulting autopilot is tested for a bounded flight envelope with a full 6-DOF Jindivik nonlinear model and the simulation results are presented to show the effectiveness of the approach.

6.1 Stability Analysis using PDLF

The system of (3.44) is said to be parameter-dependent stable if there exists a continuously differentiable parameter-dependent Lyapunov function $V(x, \theta) = x^T P(\theta)x$ whose derivative is negative along all state trajectories. This is equivalent to the existence of a $P(\theta) = P(\theta)^T$ such that

$$P(\theta) > 0, A(\theta)^T P(\theta) + P(\theta)A(\theta) + \dot{P}(\theta) < 0, \forall (\theta, \dot{\theta}) \in \Theta \times \Phi \quad (6.1)$$

where the rate of variation $\dot{\theta}$ is well defined at all times and satisfies $\dot{\theta}_i \in [\underline{v}_i, \bar{v}_i]$ and $\dot{\theta}$ lies in a polytope Φ , $\dot{\theta} \in \Phi$, $\Phi = [\underline{v}_1, \bar{v}_1] \times [\underline{v}_2, \bar{v}_2] \times \dots \times [\underline{v}_n, \bar{v}_n]$, n is the total number of $\theta(t)$. Although an exact parameter-dependent function for a continuously differentiable parameter-dependent Lyapunov variable $P(\theta)$ is still not established, a basis parameter-dependent function for the parameter-dependent Lyapunov variable is suggested in [8, 100] and [101] and is to copy the plant's parameter-dependent function. Therefore, we can constrain the basis parameter-dependent function for the parameter-dependent Lyapunov variable to vary in an affine fashion.

$$P(\theta) = P_0 + \theta_1 P_1 + \dots + \theta_n P_n = \alpha_1 \hat{P}_1 + \alpha_2 \hat{P}_2 + \dots + \alpha_r \hat{P}_r \quad (6.2)$$

where $r = 2^n$, α_i is determined using (3.24) and (3.25) and

$$\begin{bmatrix} \hat{P}_1 \\ \hat{P}_2 \\ \hat{P}_3 \\ \vdots \\ \hat{P}_r \end{bmatrix} = \begin{bmatrix} 1 & \underline{\theta}_1 & \underline{\theta}_2 & \dots & \underline{\theta}_{n-1} & \underline{\theta}_n \\ 1 & \underline{\theta}_1 & \underline{\theta}_2 & \dots & \underline{\theta}_{n-1} & \bar{\theta}_n \\ 1 & \bar{\theta}_1 & \bar{\theta}_2 & \dots & \bar{\theta}_{n-1} & \underline{\theta}_n \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \bar{\theta}_1 & \bar{\theta}_2 & \dots & \bar{\theta}_{n-1} & \bar{\theta}_n \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_n \end{bmatrix} \quad (6.3)$$

Although the set of feasible solutions of inequality (6.1) is reduced by defining the parameter-dependent Lyapunov function, $P(\theta)$, as affine (as in (6.2)), this is a practical approach and makes inequality (6.1) tractable. This is a common. Differentiating (6.2) with respect to time gives

$$\dot{P}(\theta) = \dot{\theta}_1 P_1 + \dots + \dot{\theta}_n P_n = \beta_1 \tilde{P}_1 + \beta_2 \tilde{P}_2 + \dots + \beta_r \tilde{P}_r \quad (6.4)$$

where β_i can be determined in a similar manner to α_i using (3.24) and (3.25) and

$$\begin{bmatrix} \tilde{P}_1 \\ \tilde{P}_2 \\ \tilde{P}_3 \\ \vdots \\ \tilde{P}_r \end{bmatrix} = \begin{bmatrix} 0 & \underline{v}_1 & \underline{v}_2 & \dots & \underline{v}_{n-1} & \underline{v}_n \\ 0 & \underline{v}_1 & \underline{v}_2 & \dots & \underline{v}_{n-1} & \bar{v}_n \\ 0 & \bar{v}_1 & \bar{v}_2 & \dots & \bar{v}_{n-1} & \underline{v}_n \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \bar{v}_1 & \bar{v}_2 & \dots & \bar{v}_{n-1} & \bar{v}_n \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_n \end{bmatrix} \quad (6.5)$$

Substituting (3.51), (6.2) and (6.4) into (6.1), and recalling that $\sum_{i=1}^r \alpha_i = 1$ and $\sum_{i=1}^r \beta_i = 1$, we get

$$P(\theta) > 0, \sum_{i=1}^r \sum_{k=1}^r \alpha_i^2 \beta_k \left(\hat{A}_i^T \hat{P}_i + \hat{P}_i \hat{A}_i + \tilde{P}_k \right) + 2 \sum_{i=1}^{r-1} \sum_{j=i+1}^r \sum_{k=1}^r \alpha_i \alpha_j \beta_k \left(\frac{1}{2} \left(\hat{A}_i^T \hat{P}_j + \hat{P}_j \hat{A}_i + \hat{A}_j^T \hat{P}_i + \hat{P}_i \hat{A}_j + 2\tilde{P}_k \right) \right) < 0, \forall (\theta, \dot{\theta}) \in \Theta \times \Phi \quad (6.6)$$

As $\alpha_i^2 \beta_k \in [0, 1]$, $i, k = 1, \dots, r$, $2\alpha_i \alpha_j \beta_k \in [0, 0.5]$, $i = 1, \dots, r-1$, $j = i+1, \dots, r$, $k = 1, \dots, r$, and $\sum_{i=1}^r \sum_{k=1}^r \alpha_i^2 \beta_k + 2 \sum_{i=1}^{r-1} \sum_{j=i+1}^r \sum_{k=1}^r \alpha_i \alpha_j \beta_k = 1$, by Lemma 3.3.1 solving the above inequality for positive definite symmetric matrices P_i need only be done at all vertices. Hence we get the following proposition.

Proposition 6.1.1. *The system of (3.44) is parameter-dependent stable whenever there exist a positive definite symmetric matrix P_i , $i = 1, 2, \dots, r$, such that the following LMI conditions hold*

$$\hat{P}_i > 0 \quad (6.7)$$

$$\hat{A}_i^T \hat{P}_i + \hat{P}_i \hat{A}_i + \tilde{P}_k < 0 \quad (6.8)$$

$$\hat{A}_i^T \hat{P}_j + \hat{P}_j \hat{A}_i + \hat{A}_j^T \hat{P}_i + \hat{P}_i \hat{A}_j + 2\tilde{P}_k < 0 \quad (6.9)$$

for $i, k = 1, \dots, r$ and $1 \leq i < j \leq r$

Note that the numbers of LMIs for (6.7)–(6.9) are r , r^2 and $r^2(r-1)/2$, respectively. Therefore, the total number of LMIs to be solved is $r(r^2 + r + 2)/2$. Note also that there are other approaches that could be used to obtain a finite number of LMIs, e.g. multi-convexity [11, 45], S-procedure [40], gridding parameter space [39, 66, 100, 101], etc.

Proposition 6.1.2. *(Multi-convexity approach, [11, Proposition 5.1]) Assume $\dot{\theta} = 0$, The system of (3.44) is parameter-dependent stable whenever there exist a positive definite symmetric matrix P_i , $i = 1, 2, \dots, r$, and scalars λ_i , $i = 1, 2, \dots, r$, such that the following LMI conditions hold*

$$\hat{P}_k > 0 \quad (6.10)$$

$$\lambda_k \geq 0 \quad (6.11)$$

$$\hat{A}_k^T \hat{P}_k + \hat{P}_k \hat{A}_k < -\lambda_k I \quad (6.12)$$

$$\hat{A}_i^T \hat{P}_i + \hat{P}_i \hat{A}_i + \hat{A}_j^T \hat{P}_j + \hat{P}_j \hat{A}_j - (\hat{A}_i^T \hat{P}_j + \hat{P}_j \hat{A}_i + \hat{A}_j^T \hat{P}_i + \hat{P}_i \hat{A}_j) \geq -(\lambda_i + \lambda_j) I \quad (6.13)$$

for $k = 1, \dots, r$ and $1 \leq i < j \leq r$

Corollary 6.1.3. *Assume that $\dot{\theta} = 0$. If Proposition 6.1.2 is satisfied then Proposition 6.1.1 is satisfied.*

Proof. First, we show that if (6.12) is satisfied then (6.8) is satisfied. Let $\hat{A}_k^T \hat{P}_k + \hat{P}_k \hat{A}_k = M_k$ and $M_k + \lambda_k I < 0$ hence, for all nonzero vector $X \in R^p$, $X^T M_k X + \lambda_k X^T X < 0$ if and only if $X^T M_k X < -\lambda_k X^T X \leq 0$ since $\lambda_k \geq 0$ and $X^T X > 0$. This yields $X^T M_k X < 0$ or $\hat{A}_k^T \hat{P}_k + \hat{P}_k \hat{A}_k < 0$.

Next, we show that if (6.13) is satisfied then (6.9) is satisfied. Let $\hat{A}_i^T \hat{P}_i + \hat{P}_i \hat{A}_i + \hat{A}_j^T \hat{P}_j + \hat{P}_j \hat{A}_j = M$ and $\hat{A}_i^T \hat{P}_j + \hat{P}_j \hat{A}_i + \hat{A}_j^T \hat{P}_i + \hat{P}_i \hat{A}_j = M_{ij}$. We have $X^T M X - X^T M_{ij} X + (\lambda_i + \lambda_j) X^T X \geq 0$, for all nonzero vectors $X \in R^p$. From (6.12), we have $X^T M X + (\lambda_i + \lambda_j) X^T X < 0$, therefore $0 > X^T M X + (\lambda_i + \lambda_j) X^T X \geq X^T M_{ij} X$. This yields $X^T M_{ij} X < 0$ or $\hat{A}_i^T \hat{P}_j + \hat{P}_j \hat{A}_i + \hat{A}_j^T \hat{P}_i + \hat{P}_i \hat{A}_j < 0$ \square

The corollary above shows that Proposition 6.1.2 is a subset of Proposition 6.1.1 which is more general since $\hat{A}_k^T \hat{P}_k + \hat{P}_k \hat{A}_k$ and $\hat{A}_i^T \hat{P}_j + \hat{P}_j \hat{A}_i + \hat{A}_j^T \hat{P}_i + \hat{P}_i \hat{A}_j$ can be less than a negative definite symmetric matrix rather than they are just less than a diagonal negative definite matrix $-\lambda_k I$ and $-(\lambda_i + \lambda_j)I$, respectively. In addition, Proposition 6.1.1 shows that the determination of a negative definite symmetric matrix is not necessary, hence, comparing with Proposition 6.1.2, the number of LMIs, decision variables and the computational time are reduced while the achieved performance γ level is improved.

6.2 Controller Synthesis using PDLF

In the previous section, a sufficient condition to guarantee the stability property of the closed-loop system using the parameter-dependent Lyapunov function has been presented, where it is sufficient to evaluate the LMIs at all vertices. Next, we consider the problem of designing a gain-scheduled output feedback H_∞ control with guaranteed L_2 -gain performance for affine LPV systems using the parameter-dependent Lyapunov function for which the proposed technique in the previous section can be directly extended to synthesizing a gain-scheduled H_∞ controller. The material in this section is drawn from [8, 10, 43] and [44].

Consider a given affine LPV plant model with state-space realization of the form (3.59). The gain-scheduled output feedback H_∞ control problem using the parameter-dependent Lyapunov function is to compute a dynamic LPV controller, $K(\theta)$, with state-space equations

$$\begin{aligned} \dot{x}_k &= A_k(\theta, \dot{\theta})x_k + B_k(\theta)y \\ u &= C_k(\theta)x_k + D_k(\theta)y \end{aligned} \quad (6.14)$$

which stabilizes the closed-loop system, (3.59) and (6.14), and minimizes the closed-loop quadratic H_∞ performance, γ , ensures the induced L_2 -norm of the operator mapping the disturbance signal w into the controlled signal z is bounded by γ

$$\int_0^{t_1} z^T z dt \leq \gamma^2 \int_0^{t_1} w^T w dt, \quad \forall t_1 \geq 0 \quad (6.15)$$

along all possible parameter trajectories, $\forall(\theta, \dot{\theta}) \in \Theta \times \Phi$. Note that A and A_k have the same dimensions, since we restrict ourselves to the full-order case. The closed-loop system, (3.59) and (6.14), is described by the state-space equations

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{x}_k \end{bmatrix} &= A_{cl}(\theta) \begin{bmatrix} x \\ x_k \end{bmatrix} + B_{cl}(\theta)w \\ z &= C_{cl}(\theta) \begin{bmatrix} x \\ x_k \end{bmatrix} + D_{cl}(\theta)w \end{aligned} \quad (6.16)$$

where

$$\begin{aligned} A_{cl}(\theta, \dot{\theta}) &= \begin{bmatrix} A(\theta) + B_2 D_k(\theta) C_2 & B_2 C_k(\theta) \\ B_k(\theta) C_2 & A_k(\theta, \dot{\theta}) \end{bmatrix}, \quad B_{cl}(\theta) = \begin{bmatrix} B_1(\theta) + B_2 D_k(\theta) D_{21} \\ B_k(\theta) D_{21} \end{bmatrix} \\ C_{cl}(\theta) &= \begin{bmatrix} C_1(\theta) + D_{12} D_k(\theta) C_2 & D_{12} C_k(\theta) \end{bmatrix}, \quad D_{cl}(\theta) = D_{11}(\theta) + D_{12} D_k(\theta) D_{21} \end{aligned} \quad (6.17)$$

Based on the parameter-dependent Lyapunov function, $V(x, \theta) = x^T P(\theta) x$, there is an LPV controller $K(\theta)$ of the form of (6.14) that stabilizes the closed-loop system, (3.59) and (6.14), and ensures the induced L_2 -norm of the operator mapping the disturbance signal w into the controlled signal z is bounded by γ along all possible parameter trajectories if and only if there exists $P(\theta) = P^T(\theta)$ such that [45]

$$P(\theta) > 0, \quad \frac{d}{dt}(x^T P(\theta) x) + z^T z - \gamma^2 w^T w < 0, \quad \forall(\theta, \dot{\theta}) \in \Theta \times \Phi \quad (6.18)$$

Note that, unlike the single quadratic Lyapunov function case (sub-section 3.4.1), $P(\theta)$, $A_k(\theta, \dot{\theta})$, $B_k(\theta)$, \dots , $D_k(\theta)$ and $A_{cl}(\theta, \dot{\theta})$, $B_{cl}(\theta)$, \dots , $D_{cl}(\theta)$ do not depend affinely on the scheduled parameters θ . Inequality (6.18) leads to [45]

$$\begin{pmatrix} A_{cl}^T(\theta, \dot{\theta}) P(\theta) + P(\theta) A_{cl}(\theta, \dot{\theta}) + \dot{P}(\theta) & P(\theta) B_{cl}(\theta) & C_{cl}^T(\theta) \\ B_{cl}^T(\theta) P(\theta) & -\gamma^2 I & D_{cl}^T(\theta) \\ C_{cl}(\theta) & D_{cl}(\theta) & -\gamma^2 I \end{pmatrix} < 0 \quad (6.19)$$

Introducing intermediate controller variables, i.e. $\hat{A}_k(\theta)$, $\hat{B}_k(\theta)$, $\hat{C}_k(\theta)$, as [8, 43]

$$\begin{aligned} A_k(\theta, \dot{\theta}) &= N^{-1}(\theta) \left(X(\theta) \dot{Y}(\theta) + N(\theta) \dot{M}^T(\theta) + \hat{A}_k(\theta) - X(\theta) (A(\theta) - \right. \\ &\quad \left. B_2 D_k(\theta) C_2) Y(\theta) - \hat{B}_k(\theta) C_2 Y(\theta) - X(\theta) B_2 \hat{C}_k(\theta) \right) M^{-T}(\theta) \end{aligned} \quad (6.20)$$

$$B_k(\theta) = N^{-1}(\theta) \left(\hat{B}_k(\theta) - X(\theta) B_2 D_k(\theta) \right) \quad (6.21)$$

$$C_k(\theta) = \left(\hat{C}_k(\theta) - D_k(\theta) C_2 Y(\theta) \right) M^{-T}(\theta) \quad (6.22)$$

where $N(\theta) = -X(\theta) + Y^{-1}(\theta)$, $\dot{N}(\theta) = -\dot{X}(\theta) - Y^{-1}(\theta) \dot{Y}(\theta) Y^{-1}(\theta)$, $M(\theta) = Y(\theta)$ and $\dot{M}(\theta) = \dot{Y}(\theta)$. A pair of positive definite symmetric matrices $(X(\theta), Y(\theta))$ is taken from the structure of the parameter-dependent Lyapunov variable, $P(\theta)$,

which is defined as

$$\begin{aligned} P(\theta) &= \begin{bmatrix} X(\theta) & -(X(\theta) - Y^{-1}(\theta)) \\ -(X(\theta) - Y^{-1}(\theta)) & X(\theta) - Y^{-1}(\theta) \end{bmatrix} \\ &= \begin{bmatrix} I_p & X(\theta) \\ 0_{p \times p} & -(X(\theta) - Y^{-1}(\theta)) \end{bmatrix} \begin{bmatrix} Y(\theta) & I_p \\ Y(\theta) & 0_{p \times p} \end{bmatrix}^{-1} \end{aligned} \quad (6.23)$$

$$\dot{P}(\theta) = \begin{bmatrix} \dot{X}(\theta) & -\dot{X}(\theta) - Y^{-1}(\theta)\dot{Y}(\theta)Y^{-1}(\theta) \\ -\dot{X}(\theta) - Y^{-1}(\theta)\dot{Y}(\theta)Y^{-1}(\theta) & \dot{X}(\theta) + Y^{-1}(\theta)\dot{Y}(\theta)Y^{-1}(\theta) \end{bmatrix} \quad (6.24)$$

$$\begin{aligned} P(\theta)^{-1} &= \begin{bmatrix} Y(\theta) & Y(\theta) \\ Y(\theta) & (X(\theta) - Y^{-1}(\theta))^{-1}X(\theta)Y(\theta) \end{bmatrix} \\ &= \begin{bmatrix} Y(\theta) & I_p \\ Y(\theta) & 0_{p \times p} \end{bmatrix} \begin{bmatrix} I_p & X(\theta) \\ 0_{p \times p} & -(X(\theta) - Y^{-1}(\theta)) \end{bmatrix}^{-1} \end{aligned} \quad (6.25)$$

where the positive definite symmetric matrices $(X(\theta), Y(\theta)) \in R^{p \times p}$, $X(\theta) - Y^{-1}(\theta) \geq 0$, and $\text{rank}(X(\theta) - Y^{-1}(\theta)) \leq p$ [82]. Note that, (6.20)–(6.22) show that $A_k(\theta, \dot{\theta})$, $B_k(\theta)$ and $C_k(\theta)$ can not depend affinely on the scheduled parameters θ when the symmetric matrix X or Y is parameter-dependent. In this thesis, we propose the intermediate controller variables, i.e. $\hat{A}_k(\theta)$, $\hat{B}_k(\theta)$, $\hat{C}_k(\theta)$ and $D_k(\theta)$, and $(X(\theta), Y(\theta))$ to depend affinely on the parameters θ as

$$\hat{A}_k(\theta) = \hat{A}_{k_0} + \theta_1 \hat{A}_{k_1} + \cdots + \theta_n \hat{A}_{k_n} = \alpha_1 \tilde{A}_{k_1} + \alpha_2 \tilde{A}_{k_2} + \cdots + \alpha_r \tilde{A}_{k_r} \quad (6.26)$$

$$\hat{B}_k(\theta) = \hat{B}_{k_0} + \theta_1 \hat{B}_{k_1} + \cdots + \theta_n \hat{B}_{k_n} = \alpha_1 \tilde{B}_{k_1} + \alpha_2 \tilde{B}_{k_2} + \cdots + \alpha_r \tilde{B}_{k_r} \quad (6.27)$$

$$\hat{C}_k(\theta) = \hat{C}_{k_0} + \theta_1 \hat{C}_{k_1} + \cdots + \theta_n \hat{C}_{k_n} = \alpha_1 \tilde{C}_{k_1} + \alpha_2 \tilde{C}_{k_2} + \cdots + \alpha_r \tilde{C}_{k_r} \quad (6.28)$$

$$D_k(\theta) = D_{k_0} + \theta_1 D_{k_1} + \cdots + \theta_n D_{k_n} = \alpha_1 \tilde{D}_{k_1} + \alpha_2 \tilde{D}_{k_2} + \cdots + \alpha_r \tilde{D}_{k_r} \quad (6.29)$$

$$X(\theta) = X_0 + \theta_1 X_1 + \cdots + \theta_n X_n = \alpha_1 \hat{X}_1 + \alpha_2 \hat{X}_2 + \cdots + \alpha_r \hat{X}_r \quad (6.30)$$

$$Y(\theta) = Y_0 + \theta_1 Y_1 + \cdots + \theta_n Y_n = \alpha_1 \hat{Y}_1 + \alpha_2 \hat{Y}_2 + \cdots + \alpha_r \hat{Y}_r \quad (6.31)$$

$$\dot{X}(\theta) = \dot{\theta}_1 X_1 + \dot{\theta}_2 X_2 + \cdots + \dot{\theta}_n X_n = \beta_1 \tilde{X}_1 + \beta_2 \tilde{X}_2 + \cdots + \beta_r \tilde{X}_r \quad (6.32)$$

$$\dot{Y}(\theta) = \dot{\theta}_1 Y_1 + \dot{\theta}_2 Y_2 + \cdots + \dot{\theta}_n Y_n = \beta_1 \tilde{Y}_1 + \beta_2 \tilde{Y}_2 + \cdots + \beta_r \tilde{Y}_r \quad (6.33)$$

Note that \tilde{X}_j and \tilde{Y}_j , $j = 1, \dots, r$, map to X_i and Y_i , $i = 1, \dots, n$, respectively in a similar manner to (6.5). This proposed technique offers obvious advantages in reducing computational burden and ease of controller implementation because the intermediate controller variables can be constructed as an affine matrix-valued function in the polytopic coordinates of the scheduled parameter. In addition, an existing method for computing the intermediate controller variables, that is based on explicit controller formulas [43], is given in Algorithm C.0.2.

Define

$$P_1(\theta) = \begin{bmatrix} Y(\theta) & I_p \\ Y(\theta) & 0_{p \times p} \end{bmatrix} \quad (6.34)$$

By premultiplying the first row and postmultiplying the first column of (6.19) by

$P_1^T(\theta)$ and $P_1(\theta)$ respectively and substituting (6.17)–(6.24) in (6.19), we get [8]

$$\begin{pmatrix} \dot{X}(\theta) + \left(X(\theta)A(\theta) + \hat{B}_k(\theta)C_2 + (\star) \right) & \star \\ \hat{A}_k^T(\theta) + A(\theta) + B_2D_k(\theta)C_2 & -\dot{Y}(\theta) + \left(A(\theta)Y(\theta) + B_2\hat{C}_k(\theta) + (\star) \right) \\ B_1^T(\theta)X(\theta) + D_{21}^T\hat{B}_k^T(\theta) & B_1^T(\theta) + D_{21}^TD_k^T(\theta)B_2^T \\ C_1(\theta) + D_{12}D_k(\theta)C_2 & C_1(\theta)Y(\theta) + D_{12}\hat{C}_k(\theta) \\ \star & \star \\ \star & \star \\ -\gamma I & \star \\ D_{11}(\theta) + D_{12}D_k(\theta)D_{21} & -\gamma I \end{pmatrix} < 0 \quad (6.35)$$

where the notation \star represents a symmetric matrix block. Substituting (3.61) and (6.26)–(6.33) in (6.35), we have

$$\begin{aligned} & \sum_{i=1}^r \sum_{k=1}^r \alpha_i^2 \beta_k \begin{pmatrix} \tilde{X}_k + \left(\hat{X}_i \hat{A}_i + \tilde{B}_{k_i} C_2 + (\star) \right) & \star \\ \tilde{A}_{k_i}^T + \hat{A}_i + B_2 \tilde{D}_{k_i} C_2 & -\tilde{Y}_k + \left(\hat{A}_i \hat{Y}_i + B_2 \tilde{C}_{k_i} + (\star) \right) \\ \hat{B}_{1_i}^T \hat{X}_i + D_{21}^T \tilde{B}_{k_i}^T & \hat{B}_{1_i}^T + D_{21}^T \tilde{D}_{k_i}^T B_2^T \\ \hat{C}_{1_i} + D_{12} \tilde{D}_{k_i} C_2 & \hat{C}_{1_i} \hat{Y}_i + D_{12} \tilde{C}_{k_i} \\ \star & \star \\ \star & \star \\ -\gamma I & \star \\ \hat{D}_{11_i} + D_{12} \tilde{D}_{k_i} D_{21} & -\gamma I \end{pmatrix} \\ & + 2 \sum_{i=1}^{r-1} \sum_{j=i+1}^r \sum_{k=1}^r \alpha_i \alpha_j \beta_k \begin{pmatrix} \tilde{X}_k + \frac{1}{2} \left(\hat{X}_j \hat{A}_i + \tilde{B}_{k_j} C_2 + \hat{X}_i \hat{A}_j + \tilde{B}_{k_i} C_2 + (\star) \right) & \star \\ \frac{1}{2} \left(\tilde{A}_{k_j}^T + \hat{A}_i + B_2 \tilde{D}_{k_j} C_2 + \tilde{A}_{k_i}^T + \hat{A}_j + B_2 \tilde{D}_{k_i} C_2 \right) & -\tilde{Y}_k + \frac{1}{2} \left(\hat{A}_i \hat{Y}_j + B_2 \tilde{C}_{k_j} + \hat{A}_j \hat{Y}_i + B_2 \tilde{C}_{k_i} + (\star) \right) \\ \frac{1}{2} \left(\hat{B}_{1_i}^T \hat{X}_j + D_{21}^T \tilde{B}_{k_j}^T + \hat{B}_{1_j}^T \hat{X}_i + D_{21}^T \tilde{B}_{k_i}^T \right) & \frac{1}{2} \left(\hat{B}_{1_i}^T + D_{21}^T \tilde{D}_{k_j}^T B_2^T + \hat{B}_{1_j}^T + D_{21}^T \tilde{D}_{k_i}^T B_2^T \right) \\ \frac{1}{2} \left(\hat{C}_{1_i} + D_{12} \tilde{D}_{k_j} C_2 + \hat{C}_{1_j} + D_{12} \tilde{D}_{k_i} C_2 \right) & \frac{1}{2} \left(\hat{C}_{1_i} \hat{Y}_j + D_{12} \tilde{C}_{k_j} + \hat{C}_{1_j} \hat{Y}_i + D_{12} \tilde{C}_{k_i} \right) \\ \star & \star \\ -\tilde{Y}_k + \frac{1}{2} \left(\hat{A}_i \hat{Y}_j + B_2 \tilde{C}_{k_j} + \hat{A}_j \hat{Y}_i + B_2 \tilde{C}_{k_i} + (\star) \right) & \star \\ \frac{1}{2} \left(\hat{B}_{1_i}^T + D_{21}^T \tilde{D}_{k_j}^T B_2^T + \hat{B}_{1_j}^T + D_{21}^T \tilde{D}_{k_i}^T B_2^T \right) & \star \\ \frac{1}{2} \left(\hat{C}_{1_i} \hat{Y}_j + D_{12} \tilde{C}_{k_j} + \hat{C}_{1_j} \hat{Y}_i + D_{12} \tilde{C}_{k_i} \right) & \star \\ \star & \star \\ \star & \star \\ -\gamma I & \star \\ \frac{1}{2} \left(\hat{D}_{11_i} + D_{12} \tilde{D}_{k_j} D_{21} + \hat{D}_{11_j} + D_{12} \tilde{D}_{k_i} D_{21} \right) & -\gamma I \end{pmatrix} < 0 \quad (6.36) \end{aligned}$$

Inequality (6.36) can be also rewritten as

$$\sum_{i=1}^r \sum_{k=1}^r \alpha_i^2 \beta_k \left(\Psi_{cl_{ii}} + \mathcal{Q}^T \hat{K}_i^T \mathcal{P} + \mathcal{P}^T \hat{K}_i \mathcal{Q} \right) + 2 \sum_{i=1}^{r-1} \sum_{j=i+1}^r \sum_{k=1}^r \alpha_i \alpha_j \beta_k \left(\frac{1}{2} \left(\Psi_{cl_{ij}} \right. \right.$$

$$+ \mathcal{Q}^T \hat{K}_i^T \mathcal{P} + \mathcal{P}^T \hat{K}_i \mathcal{Q} + \Psi_{cl_{\clubsuit\spadesuit}} + \mathcal{Q}^T \hat{K}_j^T \mathcal{P} + \mathcal{P}^T \hat{K}_j \mathcal{Q} \Big) < 0 \quad (6.37)$$

where

$$\Psi_{cl_{\clubsuit\spadesuit}} = \begin{pmatrix} \tilde{X}_k + \hat{X}_{\clubsuit} \hat{A}_{\spadesuit} + (\star) & \star & \star & \star \\ \hat{A}_{\clubsuit} & -\tilde{Y}_k + \hat{A}_{\clubsuit} \hat{Y}_{\spadesuit} + (\star) & \star & \star \\ \hat{B}_{1\clubsuit}^T \hat{X}_{\spadesuit} & \hat{B}_{1\clubsuit}^T & -\gamma I & \star \\ \hat{C}_{1\clubsuit} & \hat{C}_{1\clubsuit} \hat{Y}_{\spadesuit} & \hat{D}_{11\clubsuit} & -\gamma I \end{pmatrix} \quad (6.38)$$

$$\mathcal{Q} = [\mathcal{C}, \quad \mathcal{D}_{21}, \quad 0_{(p+q_2) \times q_1}], \quad \tilde{\mathcal{B}} = \begin{bmatrix} I_p & 0 \\ 0 & B_2 \end{bmatrix} \quad (6.39)$$

$$\mathcal{P} = [\tilde{\mathcal{B}}^T, \quad 0_{(p+m_2) \times m_1}, \quad \mathcal{D}_{12}^T], \quad \hat{K}_i = \begin{pmatrix} \tilde{A}_{k_i} & \tilde{B}_{k_i} \\ \tilde{C}_{k_i} & \tilde{D}_{k_i} \end{pmatrix} \quad (6.40)$$

with the subscript $\clubsuit\spadesuit$ denote $(ii, ij, \text{ or } ji)$. By Lemma 3.3.1 and knowing the matrix vertices (\hat{X}_i, \hat{Y}_i) , $i = 1, 2, \dots, r$, the system matrix vertices \hat{K}_i can be determined from (6.37), that is an LMI in \hat{K}_i , at all vertices for which $(\hat{K}_1, \hat{K}_2, \dots, \hat{K}_r)$ have to satisfy all of $r^2(r+1)/2$ LMIs. Furthermore, knowing $\hat{A}_{k_i}, \dots, \hat{D}_{k_i}$, the controller system matrices $A_k(\theta, \dot{\theta}), \dots, D_k(\theta)$ can be computed on-line in real-time using (6.20)–(6.22) with instantaneous measurement values of θ and $\dot{\theta}$.

However, usually, the parameter derivatives either are not available or are difficult to estimate during system operation [8]. To avoid using the measured value of $\dot{\theta}$, we can constrain either $X(\theta)$ or $Y(\theta)$ to depend affinely on θ . This yields $\dot{X}(\theta)Y(\theta) + \dot{N}(\theta)M^T(\theta) = -(X(\theta)\dot{Y}(\theta) + N(\theta)\dot{r}^T(\theta)) = 0$ [8], hence, (6.20) becomes

$$A_k(\theta) = N^{-1}(\theta) \left(\hat{A}_k(\theta) - X(\theta)(A(\theta) - B_2 D_k(\theta) C_2) Y(\theta) - \hat{B}_k(\theta) C_2 Y(\theta) - X(\theta) B_2 \hat{C}_k(\theta) \right) M^{-T}(\theta) \quad (6.41)$$

By Lemma 2.2.14, the LMIs of (6.37) are solvable for \hat{K}_i if and only if there exist a pair of positive definite symmetric matrices $(X(\theta), Y(\theta))$ that satisfy the following LMIs:

$$\begin{aligned} & \sum_{i=1}^r \sum_{k=1}^r \alpha_i^2 \beta_k \left(\begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \hat{A}_i^T \hat{X}_i + \hat{X}_i \hat{A}_i + \tilde{X}_k & \hat{X}_i \hat{B}_{1i} & \hat{C}_{1i}^T \\ \hat{B}_{1i}^T \hat{X}_i & -\gamma I & \hat{D}_{11i}^T \\ \hat{C}_{1i} & \hat{D}_{11i} & -\gamma I \end{pmatrix} \begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix} \right) \\ & + 2 \sum_{i=1}^{r-1} \sum_{j=i+1}^r \sum_{k=1}^r \alpha_i \alpha_j \beta_k \left(\begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \frac{1}{2} (\hat{A}_i^T \hat{X}_j + \hat{A}_j^T \hat{X}_i + (\star)) + \tilde{X}_k & \frac{1}{2} (\hat{B}_{1i}^T \hat{X}_j + \hat{B}_{1j}^T \hat{X}_i) & \frac{1}{2} (\hat{C}_{1i} + \hat{C}_{1j}) \\ \frac{1}{2} (\hat{D}_{11i} + \hat{D}_{11j}) & -\gamma I & \star \\ \star & \star & -\gamma I \end{pmatrix} \begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix} \right) < 0 \quad (6.42) \\ & \sum_{i=1}^r \sum_{k=1}^r \alpha_i^2 \beta_k \left(\begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \hat{A}_i \hat{Y}_i + \hat{Y}_i \hat{A}_i^T - \tilde{Y}_k & \hat{Y}_i \hat{C}_{1i}^T & \hat{B}_{1i} \\ \hat{C}_{1i} \hat{Y}_i & -\gamma I & \hat{D}_{11i} \\ \hat{B}_{1i}^T & \hat{D}_{11i}^T & -\gamma I \end{pmatrix} \begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix} \right) \end{aligned}$$

$$\begin{aligned}
& + 2 \sum_{i=1}^{r-1} \sum_{j=i+1}^r \sum_{k=1}^r \alpha_i \alpha_j \beta_k \left(\begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \frac{1}{2} (\hat{A}_i \hat{Y}_j + \hat{A}_j \hat{Y}_i + (\star)) - \tilde{Y}_k \\ \frac{1}{2} (\hat{C}_{1_i} \hat{Y}_j + \hat{C}_{1_j} \hat{Y}_i) \\ \frac{1}{2} (\hat{B}_{1_i}^T + \hat{B}_{1_j}^T) \end{pmatrix} \right. \\
& \quad \left. \begin{pmatrix} \star & \star \\ -\gamma I & \star \\ \frac{1}{2} (\hat{D}_{11_i}^T + \hat{D}_{11_j}^T) & -\gamma I \end{pmatrix} \begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix} \right) < 0 \tag{6.43}
\end{aligned}$$

$$\sum_{i=1}^r \alpha_i \begin{pmatrix} \hat{X}_i & I \\ I & \hat{Y}_i \end{pmatrix} > 0 \tag{6.44}$$

where N_X and N_Y denote bases of the null spaces of $[C_2, D_{21}]$ and $[B_2^T, D_{12}^T]$, respectively. Note that, (6.44) ensures $X(\theta), Y(\theta) > 0$ and $X(\theta) - Y(\theta)^{-1} \geq 0$. By Lemma 3.3.1, (6.42)–(6.44) need only be evaluated at all vertices. Note that, (6.42)–(6.44) will become (3.77)–(3.79) when both X and Y are constant. Moreover, the quadratic H_∞ performance γ is determined from both $(X(\theta), Y)$ and $(X, Y(\theta))$ cases and the case that gives lowest γ is selected.

Theorem 6.2.1. *There exists an LPV controller $K(\theta)$ guaranteeing the closed-loop system, (3.59) and (6.14), quadratic H_∞ performance γ along all possible parameter trajectories, $\forall(\theta, \dot{\theta}) \in \Theta \times \Phi$, if and only if the following LMI conditions hold for some positive definite symmetric matrices $(X(\theta), Y(\theta))$, which further satisfy $\text{Rank}(X(\theta) - Y^{-1}(\theta)) \leq p$:*

$$\begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \hat{A}_i^T \hat{X}_i + \hat{X}_i \hat{A}_i + \tilde{X}_k & \hat{X}_i \hat{B}_{1_i} & \hat{C}_{1_i}^T \\ \hat{B}_{1_i}^T \hat{X}_i & -\gamma I & \hat{D}_{11_i}^T \\ \hat{C}_{1_i} & \hat{D}_{11_i} & -\gamma I \end{pmatrix} \begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix} < 0 \tag{6.45}$$

$$\begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \hat{A}_i \hat{Y}_i + \hat{Y}_i \hat{A}_i^T - \tilde{Y}_k & \hat{Y}_i \hat{C}_{1_i}^T & \hat{B}_{1_i} \\ \hat{C}_{1_i} \hat{Y}_i & -\gamma I & \hat{D}_{11_i} \\ \hat{B}_{1_i}^T & \hat{D}_{11_i}^T & -\gamma I \end{pmatrix} \begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix} < 0 \tag{6.46}$$

$$\begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \hat{A}_i^T \hat{X}_j + \hat{X}_j \hat{A}_i + \hat{A}_j^T \hat{X}_i + \hat{X}_i \hat{A}_j + 2\tilde{X}_k & \hat{B}_{1_i}^T \hat{X}_j + \hat{B}_{1_j}^T \hat{X}_i \\ \hat{C}_{1_i} + \hat{C}_{1_j} & \hat{X}_i \hat{B}_{1_j} + \hat{X}_j \hat{B}_{1_i} & \hat{C}_{1_i}^T + \hat{C}_{1_j}^T \\ -2\gamma I & \hat{D}_{11_i}^T + \hat{D}_{11_j}^T & -2\gamma I \\ \hat{D}_{11_i} + \hat{D}_{11_j} & -2\gamma I \end{pmatrix} \begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix} < 0 \tag{6.47}$$

$$\begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \hat{A}_i \hat{Y}_j + \hat{Y}_j \hat{A}_i^T + \hat{A}_j \hat{Y}_i + \hat{Y}_i \hat{A}_j^T - 2\tilde{Y}_k & \hat{C}_{1_i} \hat{Y}_j + \hat{C}_{1_j} \hat{Y}_i \\ \hat{B}_{1_i}^T + \hat{B}_{1_j}^T & \hat{Y}_i \hat{C}_{1_j}^T + \hat{Y}_j \hat{C}_{1_i}^T & \hat{B}_{1_i} + \hat{B}_{1_j} \\ -2\gamma I & \hat{D}_{11_i} + \hat{D}_{11_j} & -2\gamma I \\ \hat{D}_{11_i}^T + \hat{D}_{11_j}^T & -2\gamma I \end{pmatrix} \begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix} < 0 \tag{6.48}$$

$$\begin{pmatrix} X_i & I \\ I & Y_i \end{pmatrix} > 0 \quad (6.49)$$

for $i, k = 1, \dots, r$ and $1 \leq i < j \leq r$

Note that, Theorem 6.2.1 provides a new approach and an alternative to the multi-convexity approach [11] that is given in theorem C.0.1.

6.3 Numerical Example

We demonstrate the effectiveness of the approach through a simple numerical example taken from [61], described in Chapter 4, and where the Jacobian-based LPV model is given as (4.5). An LPV controller is synthesized with the criterion $\|[W_1 S, W_2 K S]^T\| < 1$. The performance weighting, W_1 , and robustness weighting, W_2 , taken from [61] are given as (4.7). The results and numerical features of the LPV synthesis technique for the case where the pair $(X(\theta), Y(\theta))$ are affine is presented in Table 6.1. It shows that, for this example, the number of LMIs and decision variables and the computational time are reduced. Furthermore, the achieved performance γ level is less conservative when using Theorem 6.2.1 compared with the multi-convexity technique [11]. The LMIs are solved using the MATLAB Robust Control Toolbox function [12], `mincx`, on a desktop PC (Intel Core(TM)2 CPU 2.13 GHz with 2 GB of RAM).

6.4 Longitudinal Affine LPV Model

In this thesis, a minimum least-squares method [58] is used to convert a nonlinearly parameter-dependent LPV model, shown in (5.13), into an affine LPV model. The method minimizes the sum of squared differences between a nonlinearly parameter-dependent LPV model and an affine LPV model. For example, consider $X_u(u, h)$ shown in (5.14), define $\tilde{X}_u(u, h)$ to be depended affinely on u and h as

$$\tilde{X}_u(u, h) = X_{u_0} + uX_{u_u} + hX_{u_h} \quad (6.50)$$

Then, the least-squares problem for $X_u(u, h)$ is to determine the best $[X_{u_0} \ X_{u_u} \ X_{u_h}]^T$ that minimizes the sum of squared differences between $X_u(u, h)$ and $\tilde{X}_u(u, h)$. Hence, the least-squares problem for $X_u(u, h)$ can be formulated as

$$Z = X\theta + v \quad (6.51)$$

where Z is an $N \times 1$ vector of values computed from (5.14), θ is a 3×1 vector of unknown parameters, X is an $N \times 3$ matrix of known data vectors or regressors, and v is an $N \times 1$ vector of equation errors as shown below

Table 6.1: Numerical comparisons of LPV synthesis techniques; an $(X(\theta), Y(\theta))$ case

Condition $(\theta, \dot{\theta}) \in [0, 10] \times [-5, 5]$	# of LMI	# of decision variables	CPU time (s)	performance γ
Multi-convexity [†]	16	46	0.4735	0.1083
Theorem 6.2.1	14	41	0.1969	0.0936
SQLF*	5	21	0.0269	0.1212
Condition $(\theta, \dot{\theta}) \in [0, 10^4] \times [-5, 5]$	# of LMI	# of decision variables	CPU time (s)	performance γ
Multi-convexity [†]	16	46	0.9212	4.8488
Theorem 6.2.1	14	41	0.2795	2.4576
SQLF*	5	21	0.0329	42.3240
Condition $(\theta, \dot{\theta}) \in [0, 10] \times [-10^6, 10^6]$	# of LMI	# of decision variables	CPU time (s)	performance γ
Multi-convexity [†]	16	46	0.6204	0.1333
Theorem 6.2.1	14	41	0.1985	0.1211
SQLF*	5	21	0.0267	0.1212

[†] [11, Theorem 5.3]

* SQLF, [10, Theorem 5.2]

$$\begin{aligned}
Z &= [X_u(u_1, h_1) \quad X_u(u_1, h_2) \quad \cdots \quad X_u(u_i, h_{j-1}) \quad X_u(u_i, h_j)]^T \\
X &= \begin{bmatrix} 1 & u_1 & h_1 \\ 1 & u_1 & h_2 \\ \vdots & & \\ 1 & u_i & h_{j-1} \\ 1 & u_i & h_j \end{bmatrix} \\
\theta &= [X_{u_0} \quad X_{u_u} \quad X_{u_h}]^T, \quad v = [v_1 \quad v_2 \quad \cdots \quad v_{N-1} \quad v_N]^T
\end{aligned}$$

where $N = i \times j$. The best estimator of θ minimizes the sum of squared differences; the cost function, J , is given by

$$J(\theta) = \frac{1}{2}(Z - X\theta)^T(Z - X\theta) \quad (6.52)$$

Differentiating (6.52) with respect to θ gives [58],

$$\frac{\partial J(\theta)}{\partial \theta} = -Z^T X + \theta^T X^T X \quad (6.53)$$

The necessary condition for minimizing the cost is given by $\partial J(\theta)/\partial \theta = 0$ giving the least-squares solution for the unknown parameter vector θ as

$$\hat{\theta} = (X^T X)^{-1} X^T Z \quad (6.54)$$

Having converted all of the stability and control derivatives, shown in (5.14)–(5.31), to be depended affinely on u and h as in a similar manner to (6.50), a nonlinearly parameter-dependent LPV model, shown in (5.13), can be converted into an affine LPV model as

$$\dot{x} = (A_0 + uA_u + hA_h)x + (B_0 + uB_u + hB_h)u \quad (6.55)$$

where $x = [u \quad w \quad q \quad \theta \quad h]^T$, $u = [\delta_e \quad \delta_{rpm}]^T$, $(u, h) \in [337.6, 759.5] \times [1000, 18000]$, and

$$A_0 = \begin{pmatrix} -0.019657 & 0.0077042 & -37.858 & -32.174 & 0 \\ -0.14414 & -0.48037 & -0.42767 & -2.4081 & 0 \\ 0.0023323 & -0.0098674 & -0.24883 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0.074845 & -0.99998 & 0 & -0.002407 & 0 \end{pmatrix} \quad (6.56)$$

$$A_u = \begin{pmatrix} -1.6403 \times 10^{-5} & 5.7706 \times 10^{-5} & 0.089211 & -3.4314 \times 10^{-7} & 0 \\ -1.1703 \times 10^{-5} & -0.0029387 & 0.99731 & 0.005674 & 0 \\ -5.5391 \times 10^{-6} & -6.2069 \times 10^{-5} & -0.0015653 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -0.00017635 & -1.0665 \times 10^{-8} & 0 & 1 & 0 \end{pmatrix} \quad (6.57)$$

$$A_h = \begin{pmatrix} 9.7563 \times 10^{-7} & -8.9741 \times 10^{-7} & -0.00070117 & 1.0864 \times 10^{-8} & 0 \\ 2.1189 \times 10^{-6} & 4.6897 \times 10^{-5} & 4.2769 \times 10^{-5} & -4.5033 \times 10^{-5} & 0 \\ 4.4412 \times 10^{-8} & 9.8674 \times 10^{-7} & 2.4885 \times 10^{-5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1.3997 \times 10^{-6} & 3.3767 \times 10^{-10} & 0 & -5.0696 \times 10^{-8} & 0 \end{pmatrix} \quad (6.58)$$

$$B_0 = \begin{pmatrix} 2.1684 & 0.001532 \\ 32.892 & 0 \\ 19.21 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \quad B_u = \begin{pmatrix} -0.012568 & 2.465 \times 10^{-6} \\ -0.19064 & 0 \\ -0.11134 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \quad (6.59)$$

$$B_h = \begin{pmatrix} 0.00010025 & -6.9612 \times 10^{-8} \\ 0.0015206 & 0 \\ 0.00088809 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \quad (6.60)$$

6.5 Gain-Scheduled H_∞ Autopilot Design

In this approach, the mixed-sensitivity criterion (5.42) is also employed in a similar manner to section 5.3 where the performance weighting, W_1 , and robustness weighting, W_2 , are

$$\begin{aligned} W_1(s) &= \begin{pmatrix} \frac{0.5s+1.333}{s+0.001333} & 0 \\ 0 & \frac{0.5s+0.4443}{s+4.443 \times 10^{-4}} \end{pmatrix} \\ W_2(s) &= \begin{pmatrix} \frac{100s+3.32}{0.001s+0.0664} & 0 \\ 0 & \frac{0.01s+6.663 \times 10^{-3}}{0.001s+1.333} \end{pmatrix} \\ W_{pre-filter}(s) &= \begin{pmatrix} \frac{500}{s+500} & 0 \\ 0 & \frac{1000}{s+1000} \end{pmatrix} \end{aligned} \quad (6.61)$$

Note that the values of weighting functions W_1 and W_2 are hand-tuned until the desired objectives of performance and robustness of the closed-loop system are achieved. After the longitudinal affine LPV model, shown in (6.55), is augmented with the weighting functions, shown in (6.61), a pair of positive definite symmetric matrices $(X(\theta), Y(\theta))$ can be determined in four cases, i.e. (X, Y) , $(X(\theta), Y)$, $(X, Y(\theta))$, and $(X(\theta), Y(\theta))$, using parameter-dependent Lyapunov function, Theorem 6.2.1, for which the performance measure (γ), shown in Table 6.2, can be compared. The LMIs are solved using the MATLAB Robust Control Toolbox function [12], `mincx`.

6.6 Nonlinear Simulation Results

For an entire flight envelope of the vehicle, the mismatch uncertainties between the affine LPV model, shown in (6.55), and the nonlinearly parameter-dependent LPV model, shown in (5.13), become more significant and degrade the transient performance. Leith and Leithead [61] and Chumalee and Whidborne [30] have shown that an affine LPV model can not always accurately represent the original nonlinear plant model. In severe cases, the mismatch uncertainties can cause closed-loop instability

Table 6.2: Performance γ comparison for different cases of $(X(u, h), Y(u, h))$

Flight condition $(u, h) \in [464.1, 548.5] \times [7500, 12500]$ $(\dot{u}, \dot{h}) \in [-1.26, 1.26] \times [-20, 50]$	$(X(u, h), Y(u, h))$	$(X(u, h), Y)$	$(X, Y(u, h))$	$(X, Y)^*$
Performance γ	2.8785	2.9079	2.8537	2.8830
Flight condition $(u, h) \in [337.6, 759.5] \times [1000, 18000]$ $(\dot{u}, \dot{h}) \in [-1.26, 1.26] \times [-20, 50]$	$(X(u, h), Y(u, h))$	$(X(u, h), Y)$	$(X, Y(u, h))$	$(X, Y)^*$
Performance γ	4.2583	9.1201	4.1252	994.9747
Flight condition $(u, h) \in [464.1, 548.5] \times [7500, 12500]$ $(\dot{u}, \dot{h}) \in [-10^6, 10^6] \times [-10^6, 10^6]$	$(X(u, h), Y(u, h))$	$(X(u, h), Y)$	$(X, Y(u, h))$	$(X, Y)^*$
Performance γ	2.9183	2.8979	2.8890	2.8830

* SQLF, [10, Theorem 5.2]

for the designed controller with the original nonlinear plant model. However, Lim and How [64] have shown that the mismatch uncertainties can be reduced until they are less significant when the scheduled parameters have a small variation. Extending the proposed method to be applicable to a general class of LPV system (whose system matrices can be nonlinearly dependent on the scheduled parameters θ) with a fuller range of operating condition requires further work which is discussed in the Conclusion.

Hence, the designed H_∞ gain-scheduling autopilot is validated for a bounded flight envelope, i.e. $(u, h) \in [464.1, 548.5] \times [7500, 12500]$, $(\dot{u}, \dot{h}) \in [-1.26, 1.26] \times [-20, 50]$, with the Jindivik nonlinear model [41] in a MATLAB Simulink simulation. In Figure 6.1, the transient responses of the simulated vehicle for small demanded changes in speed and altitude are shown for one particular point in the flight envelope. In addition, it also shows transient response of two LPV controllers, i.e. the first LPV controller (explicit formulas) where the intermediate controller variables ($\hat{A}_k(\theta)$, $\hat{B}_k(\theta)$, $\hat{C}_k(\theta)$ and $D_k(\theta)$) are computed using Algorithm 3.1 [43] and the second LPV controller (proposed method) where the matrix vertices of intermediate controller variables are determined from (6.37).

The first LPV controller's response is non-smooth because D_{21} matrix is not full-row rank, hence, cause a D_{21} singular problem. According to [43, Algorithm 3.1], the intermediate controller variables will be varied smoothly following the parameter θ , if the D_{12} and D_{21} matrices are full-column and full-row rank [8] respectively, shown in Algorithm C.0.2. In addition, Table 6.3 shows computational time of the intermediate controller variables using Algorithm 3.1 [43] and the proposed method using (6.26)–(6.29). One can see from Table 6.3 that our proposed method has a lower computational time. Finally, Figure 6.2 shows a simulated flight that cover a wide range of the flight envelope. It demonstrates that performance robustness was achieved over the defined flight envelope.

Table 6.3: Computational time of $\hat{A}_k(\theta)$, $\hat{B}_k(\theta)$, $\hat{C}_k(\theta)$ and $D_k(\theta)$

Method	CPU time (us)
Explicit formulas [†]	20830
Proposed method [*]	42

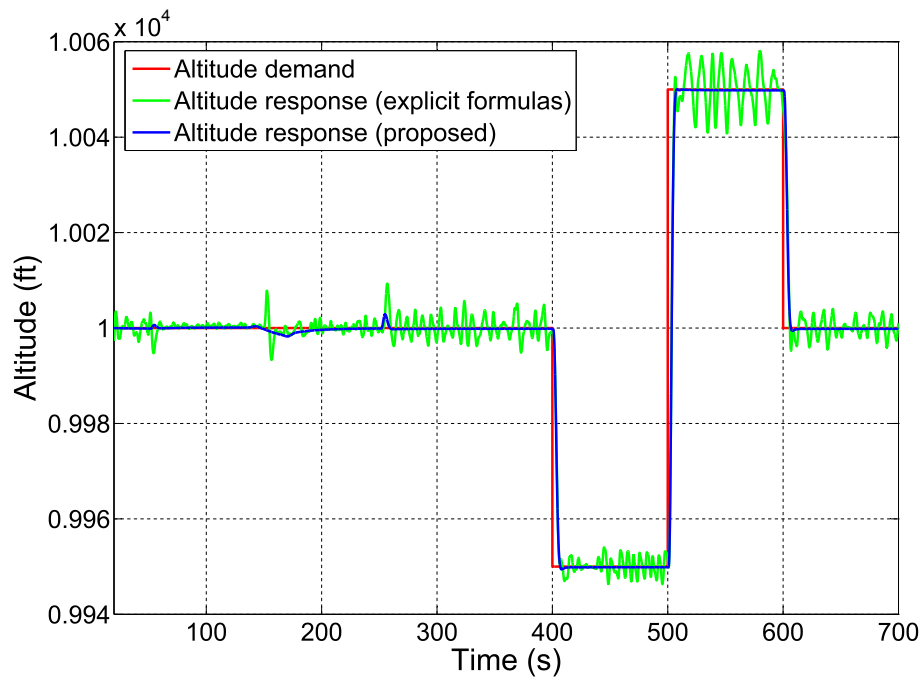
[†] [43, Algorithm 3.1]

^{*} (6.26)–(6.29)

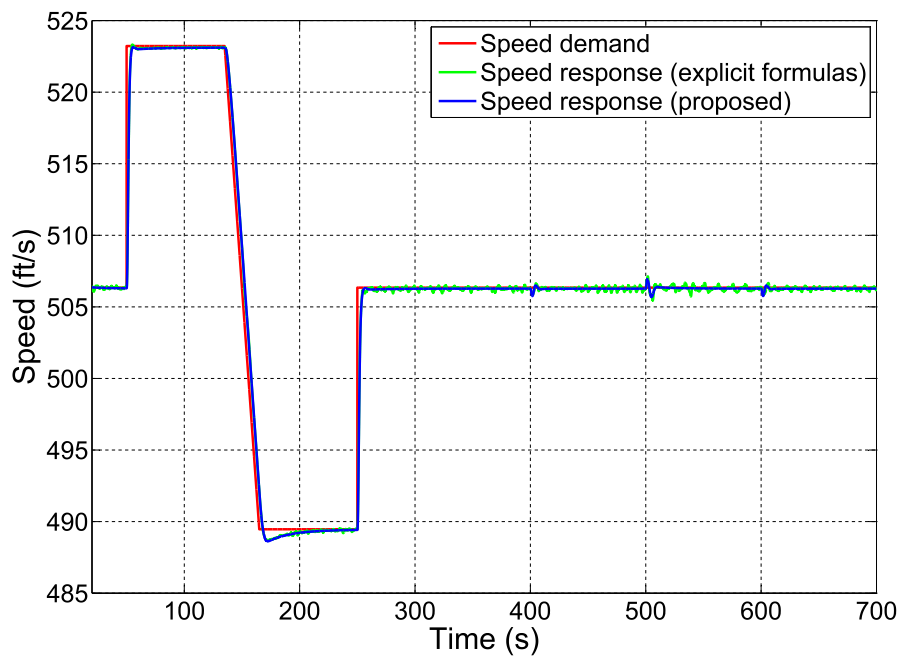
6.7 Conclusion

In this chapter, new sufficient conditions for gain-scheduled H_∞ performance analysis and synthesis for a class of affine LPV systems using parameter-dependent Lyapunov function are proposed, in Theorem 6.2.1. Compared with the multi-convexity technique [11] (Theorem C.0.1) fewer LMIs and decision variables are required and the computational time is lower while the achieved performance level is improved. The analysis and synthesis conditions are represented in the form of a finite number of LMIs. A numerical example is compared with the multi-convexity technique [11] results. In addition, the intermediate controller variables, i.e. $\hat{A}_k(\theta)$, $\hat{B}_k(\theta)$, $\hat{C}_k(\theta)$ and $D_k(\theta)$, can be constructed as an affine matrix-valued function in the polytopic coordinates of the scheduled parameter without the need for constraints on the D_{12} and D_{21} matrices. The proposed method was applied to synthesize a longitudinal LPV autopilot of the Jindivik UAV. The designed controller was tested with a full 6-DOF simulation of the vehicle and nonlinear simulation results show the effectiveness of the proposed method.

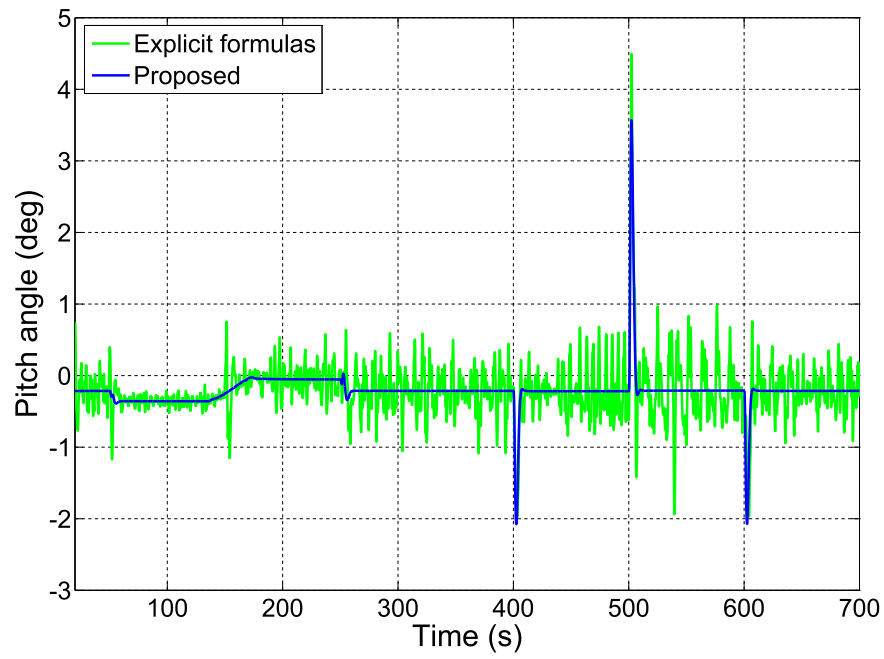
The main limitation is that the proposed method is not applicable to a general class of LPV systems which can be nonlinearly dependent on the time-varying parameters θ . However, the mismatch uncertainty between affine LPV model and nonlinearly parameter-dependent LPV model can be modelled as time-varying real parametric uncertainties and can be also included in an affine LPV model using a linear fractional transformation (LFT) for gain-scheduled control synthesis and analysis purpose. This approach will be presented in the next chapter.



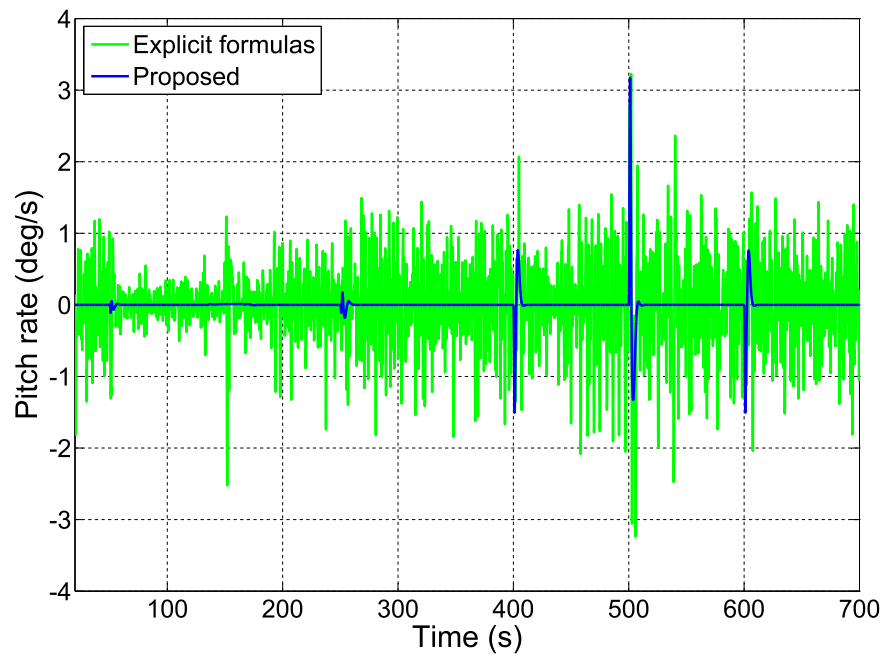
(a) Altitude (ft)



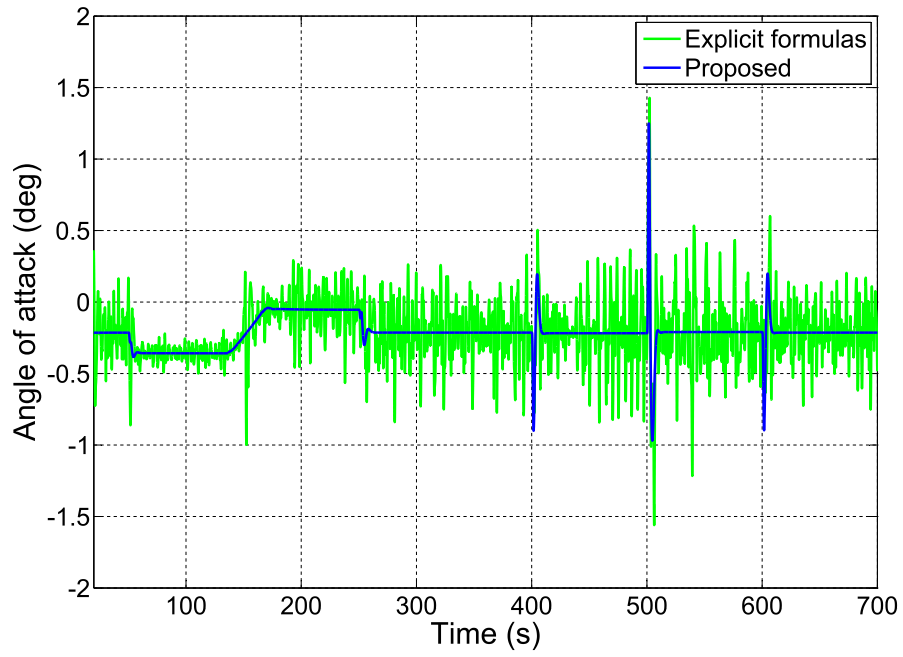
(b) Speed (ft/s)



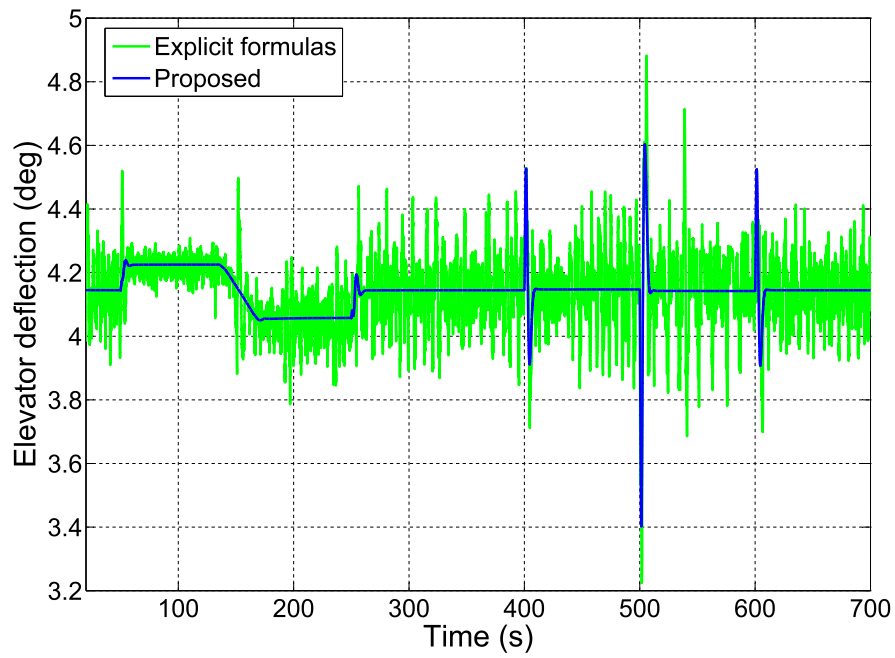
(c) Pitch angle (deg)



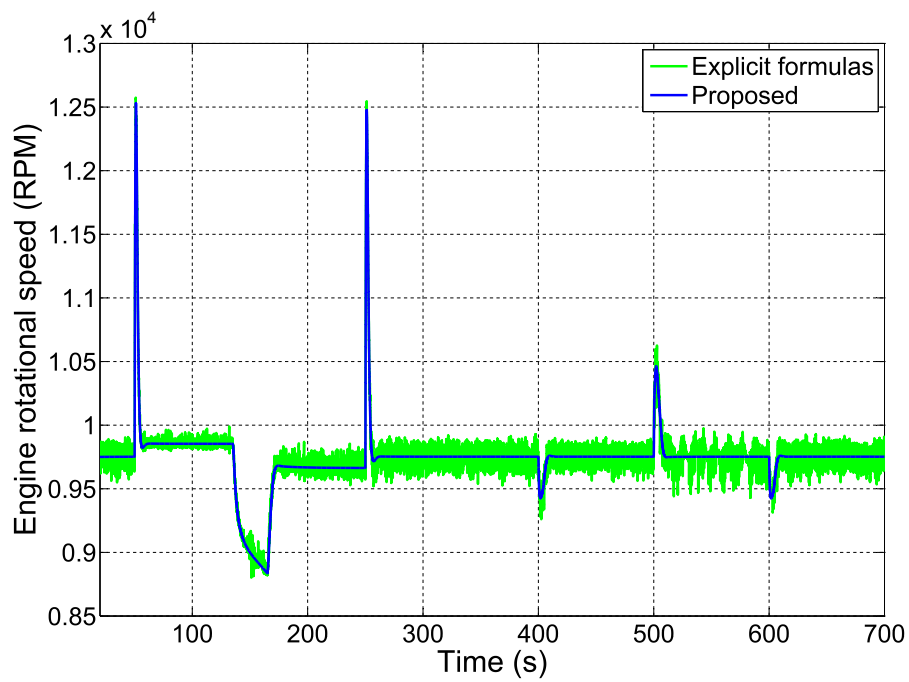
(d) Pitch rate (deg/s)



(e) Angle of Attack (deg)

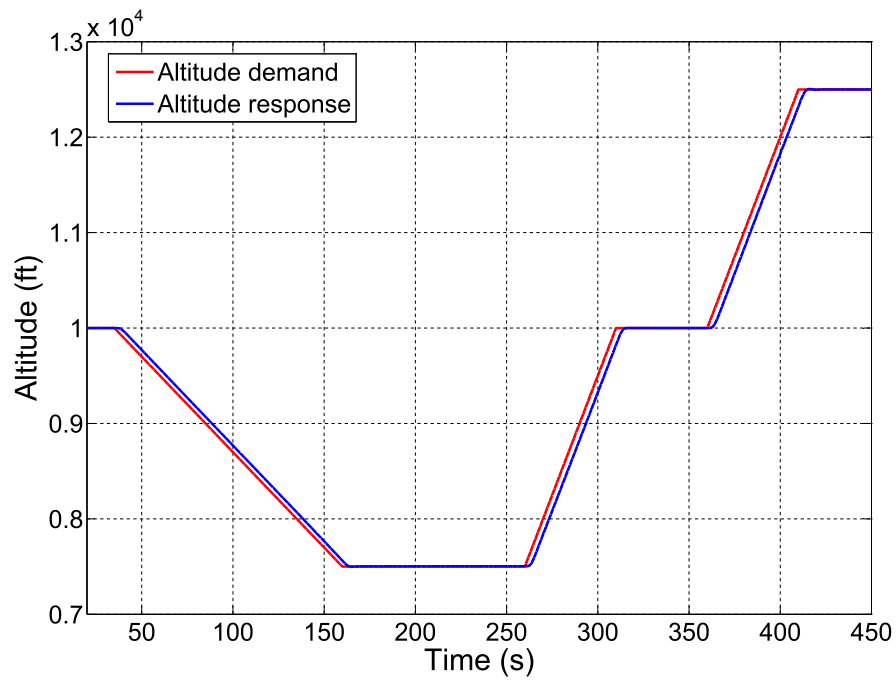


(f) Elevator deflection (deg)

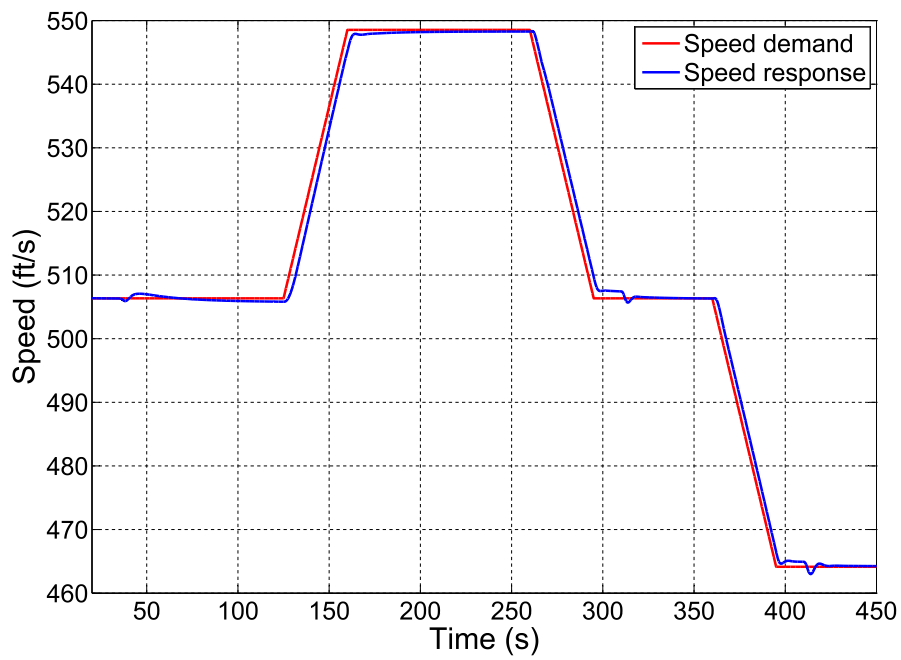


(g) Engine speed (RPM)

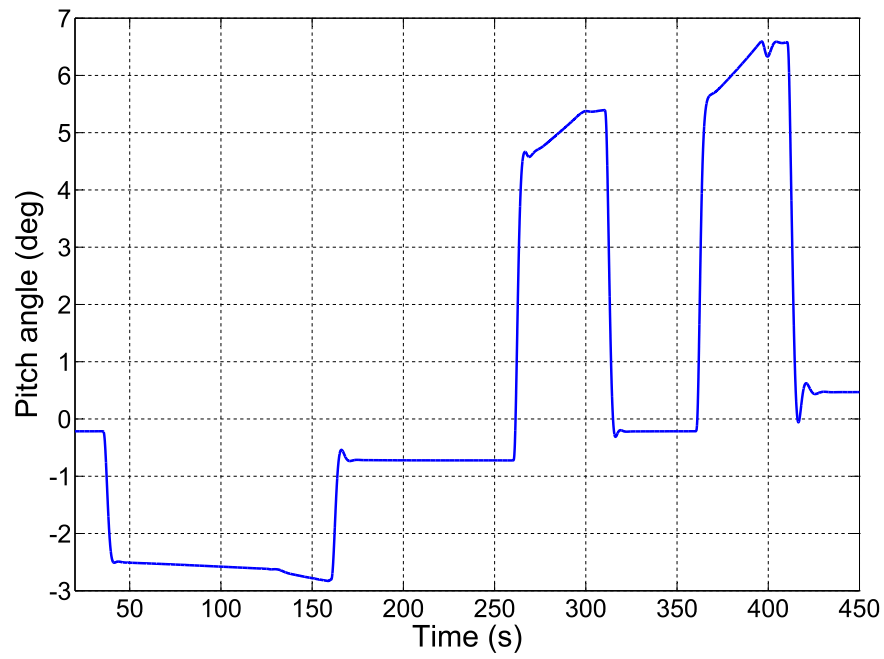
Figure 6.1: The transient performance of H_∞ gain-scheduling autopilot is validated with the Jindivik nonlinear dynamic model about one condition inside the flight envelope, i.e. speed = 506 ft/s and altitude = 10,000 ft



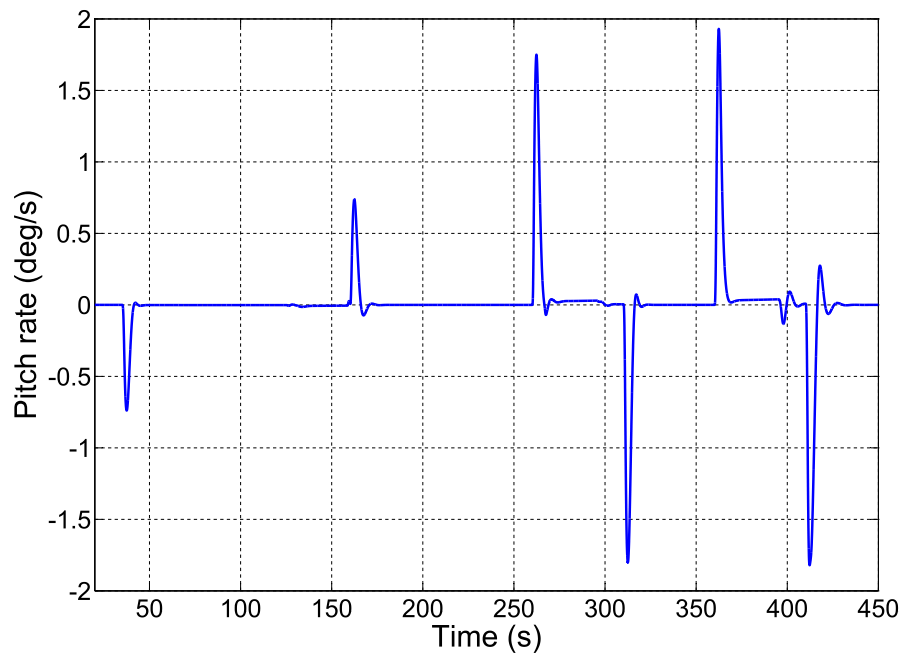
(a) Altitude (ft)



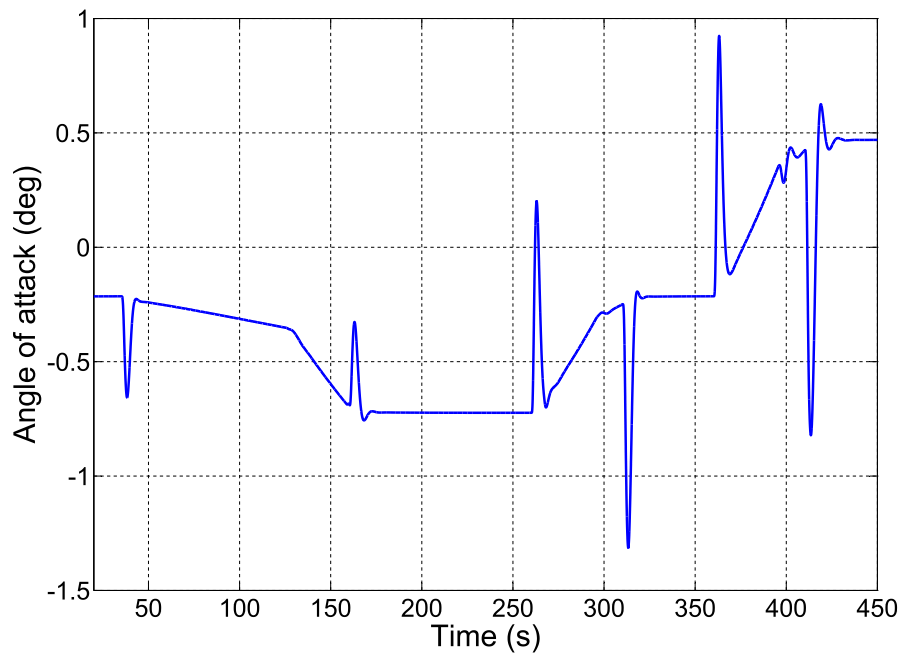
(b) Speed (ft/s)



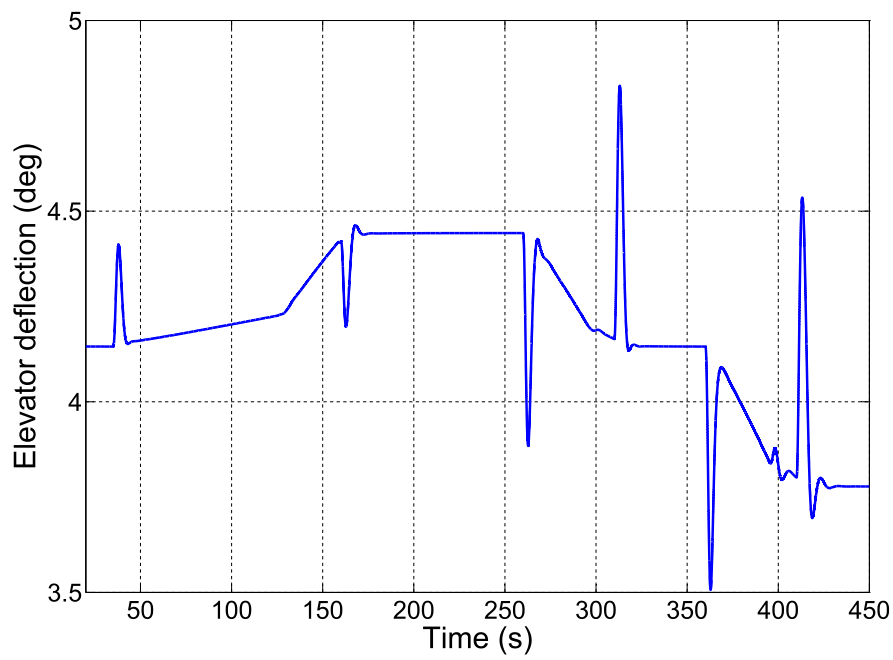
(c) Pitch angle (deg)



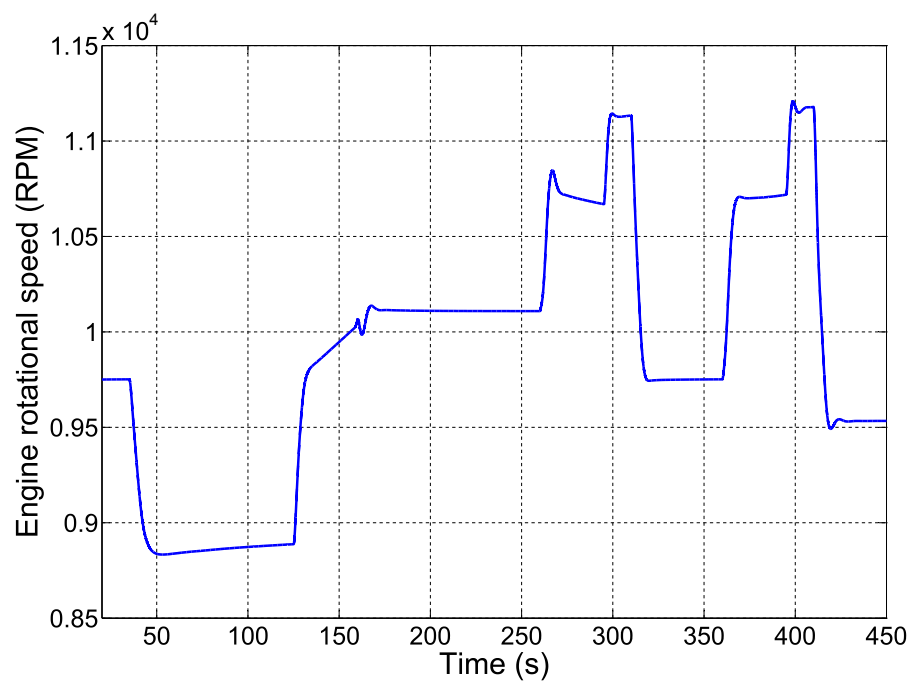
(d) Pitch rate (deg/s)



(e) Angle of Attack (deg)



(f) Elevator deflection (deg)



(g) Engine speed (RPM)

Figure 6.2: The desired performance and robustness objectives are achieved across the defined flight envelope

This page intentionally contains only this sentence.

Chapter 7

Robust Lateral LPV Autopilot Design

A major advantage of a class of affine LPV models is that sufficient conditions for gain-scheduled output feedback H_∞ control synthesis problem can be obtained in the form of a finite number of LMIs for both the single quadratic and parameter-dependent Lyapunov function cases. However, an affine LPV model can rarely accurately represent the original nonlinear model. There are always mismatch uncertainties between these two models. In this chapter, time-varying real parametric uncertainties are included in the system state-space model matrices as a linear fractional transformation (LFT) form in order to guarantee closed-loop stability and improve transient performance in presence of these mismatch uncertainties. Hence, a new class of LPV models is obtained called an uncertain affine LPV model which is less conservative than the existing parameter-dependent linear fractional transformation model (LPV/LFT).

New algorithms of robust stability analysis and gain-scheduled controller synthesis for this uncertain affine LPV model using single quadratic and parameter-dependent Lyapunov functions are proposed. The analysis and synthesis conditions are represented in the form of a finite number of LMIs. Moreover, a technique to construct the intermediate controller variables as an affine matrix-valued function in the polytopic coordinates of the scheduled parameter is also proposed. To demonstrate the impacts of the proposed scheme in robustness improvement of uncertain affine LPV systems, we compare our approach with the existing uncertain LPV/LFT approach on a numerical example. Furthermore, the proposed method is applied to synthesize a lateral autopilot, i.e. heading-hold, for a bounded flight envelope of the Jindivik UAV. The simulation results on a six degree-of-freedom Jindivik nonlinear model are presented to show the effectiveness of the approach.

7.1 Jacobian-Based Lateral LPV Model

Suppose an aircraft is assumed to be about a wings level and constant altitude and airspeed flight condition. In addition, assume it can manoeuvre only in the North-East plane, hence, all of the longitudinal states in (A.10) and (A.11) are frozen and equal to the trim values, i.e. $u = u_{\text{trim}}$, $w = w_{\text{trim}}$, $q = q_{\text{trim}} = 0$, $\alpha = \alpha_{\text{trim}} = \theta = \theta_{\text{trim}}$, and $h = h_{\text{trim}}$. Moreover, after substituting $\beta = \sin^{-1}(v/V)$, $V = \sqrt{u_{\text{trim}}^2 + v^2 + w_{\text{trim}}^2}$, and $\bar{q} = \frac{1}{2}\rho V^2$ into (A.10) and (A.11), we get equations of lateral motion as

$$\begin{aligned} \dot{v} = & \frac{\rho S}{2m} \left[C_{Y_0} + C_{Y_\beta} \sin^{-1} \left(\frac{v}{\sqrt{u_{\text{trim}}^2 + v^2 + w_{\text{trim}}^2}} \right) \right] (u_{\text{trim}}^2 + v^2 + w_{\text{trim}}^2) \\ & + \left[\left(\frac{\rho S b}{4m} C_{Y_p} \sqrt{u_{\text{trim}}^2 + v^2 + w_{\text{trim}}^2} \right) + w_{\text{trim}} \right] p \\ & + \left[\left(\frac{\rho S b}{4m} C_{Y_r} \sqrt{u_{\text{trim}}^2 + v^2 + w_{\text{trim}}^2} \right) - u_{\text{trim}} \right] r \\ & + (g \cos \theta_{\text{trim}}) \sin \phi + \frac{\rho S}{2m} (u_{\text{trim}}^2 + v^2 + w_{\text{trim}}^2) C_{Y_{\delta_a}} \delta_a \\ & + \frac{\rho S}{2m} (u_{\text{trim}}^2 + v^2 + w_{\text{trim}}^2) C_{Y_{\delta_r}} \delta_r \end{aligned} \quad (7.1)$$

$$\begin{aligned} \dot{p} = & \frac{\rho S b}{2I_x} \left[C_{l_0} + C_{l_\beta} \sin^{-1} \left(\frac{v}{\sqrt{u_{\text{trim}}^2 + v^2 + w_{\text{trim}}^2}} \right) \right] (u_{\text{trim}}^2 + v^2 + w_{\text{trim}}^2) \\ & + \frac{\rho S b^2}{4I_x} \left(\sqrt{u_{\text{trim}}^2 + v^2 + w_{\text{trim}}^2} \right) C_{l_p} p + \frac{\rho S b^2}{4I_x} \left(\sqrt{u_{\text{trim}}^2 + v^2 + w_{\text{trim}}^2} \right) C_{l_r} r \\ & + \frac{\rho S b}{2I_x} (u_{\text{trim}}^2 + v^2 + w_{\text{trim}}^2) C_{l_{\delta_a}} \delta_a + \frac{\rho S b}{2I_x} (u_{\text{trim}}^2 + v^2 + w_{\text{trim}}^2) C_{l_{\delta_r}} \delta_r \end{aligned} \quad (7.2)$$

$$\begin{aligned} \dot{r} = & \frac{\rho S b}{2I_z} \left[C_{n_0} + C_{n_\beta} \sin^{-1} \left(\frac{v}{\sqrt{u_{\text{trim}}^2 + v^2 + w_{\text{trim}}^2}} \right) \right] (u_{\text{trim}}^2 + v^2 + w_{\text{trim}}^2) \\ & + \frac{\rho S b^2}{4I_z} \left(\sqrt{u_{\text{trim}}^2 + v^2 + w_{\text{trim}}^2} \right) C_{n_p} p + \frac{\rho S b^2}{4I_z} \left(\sqrt{u_{\text{trim}}^2 + v^2 + w_{\text{trim}}^2} \right) C_{n_r} r \\ & + \frac{\rho S b}{2I_z} (u_{\text{trim}}^2 + v^2 + w_{\text{trim}}^2) C_{n_{\delta_a}} \delta_a + \frac{\rho S b}{2I_z} (u_{\text{trim}}^2 + v^2 + w_{\text{trim}}^2) C_{n_{\delta_r}} \delta_r \end{aligned} \quad (7.3)$$

$$\dot{\phi} = p + \left[(\tan \theta_{\text{trim}}) \cos \phi \right] r \quad (7.4)$$

$$\dot{\psi} = \left[\frac{\cos \phi}{\cos \theta_{\text{trim}}} \right] r \quad (7.5)$$

After (7.1)–(7.5) are linearized using the Jacobian linearization method about a wings level and constant altitude and airspeed flight condition, we get a lateral LTI

model as a state-space system of the form,

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} Y_v & Y_p & Y_r & Y_\phi & 0 \\ L_v & L_p & L_r & 0 & 0 \\ N_v & N_p & N_r & 0 & 0 \\ 0 & 1 & \phi_r & 0 & 0 \\ 0 & 0 & \psi_r & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \\ \psi \end{bmatrix} + \begin{bmatrix} Y_{\delta_a} & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (7.6)$$

Note that, $v_{\text{trim}} = p_{\text{trim}} = r_{\text{trim}} = \beta_{\text{trim}} = \phi_{\text{trim}} = \psi_{\text{trim}} = \delta_{a_{\text{trim}}} = \delta_{r_{\text{trim}}} = 0$, where,

$$Y_v = \frac{\rho S}{2m} \left(\sqrt{u_{\text{trim}}^2 + w_{\text{trim}}^2} \right) C_{Y_\beta} \quad (7.7)$$

$$Y_p = \left[\frac{\rho S b}{4m} \left(\sqrt{u_{\text{trim}}^2 + w_{\text{trim}}^2} \right) C_{Y_p} \right] + w_{\text{trim}} \quad (7.8)$$

$$Y_r = \left[\frac{\rho S b}{4m} \left(\sqrt{u_{\text{trim}}^2 + w_{\text{trim}}^2} \right) C_{Y_r} \right] - u_{\text{trim}} \quad (7.9)$$

$$Y_\phi = g \cos \theta_{\text{trim}} \quad (7.10)$$

$$Y_{\delta_a} = \frac{\rho S}{2m} \left(u_{\text{trim}}^2 + w_{\text{trim}}^2 \right) C_{Y_{\delta_a}} \quad (7.11)$$

$$Y_{\delta_r} = \frac{\rho S}{2m} \left(u_{\text{trim}}^2 + w_{\text{trim}}^2 \right) C_{Y_{\delta_r}} \quad (7.12)$$

$$L_v = \frac{\rho S b}{2I_x} \left(\sqrt{u_{\text{trim}}^2 + w_{\text{trim}}^2} \right) C_{l_\beta} \quad (7.13)$$

$$L_p = \frac{\rho S b^2}{4I_x} \left(\sqrt{u_{\text{trim}}^2 + w_{\text{trim}}^2} \right) C_{l_p} \quad (7.14)$$

$$L_r = \frac{\rho S b^2}{4I_x} \left(\sqrt{u_{\text{trim}}^2 + w_{\text{trim}}^2} \right) C_{l_r} \quad (7.15)$$

$$L_{\delta_a} = \frac{\rho S b}{2I_x} \left(u_{\text{trim}}^2 + w_{\text{trim}}^2 \right) C_{l_{\delta_a}} \quad (7.16)$$

$$L_{\delta_r} = \frac{\rho S b}{2I_x} \left(u_{\text{trim}}^2 + w_{\text{trim}}^2 \right) C_{l_{\delta_r}} \quad (7.17)$$

$$N_v = \frac{\rho S b}{2I_z} \left(\sqrt{u_{\text{trim}}^2 + w_{\text{trim}}^2} \right) C_{n_\beta} \quad (7.18)$$

$$N_p = \frac{\rho S b^2}{4I_z} \left(\sqrt{u_{\text{trim}}^2 + w_{\text{trim}}^2} \right) C_{n_p} \quad (7.19)$$

$$N_r = \frac{\rho S b^2}{4I_z} \left(\sqrt{u_{\text{trim}}^2 + w_{\text{trim}}^2} \right) C_{n_r} \quad (7.20)$$

$$N_{\delta_a} = \frac{\rho S b}{2I_z} \left(u_{\text{trim}}^2 + w_{\text{trim}}^2 \right) C_{n_{\delta_a}} \quad (7.21)$$

$$N_{\delta_r} = \frac{\rho S b}{2I_z} \left(u_{\text{trim}}^2 + w_{\text{trim}}^2 \right) C_{n_{\delta_r}} \quad (7.22)$$

$$\phi_r = \tan \theta_{\text{trim}} \quad (7.23)$$

$$\psi_r = \sec \theta_{\text{trim}} \quad (7.24)$$

Table 7.1: Stability and control derivative data (lateral mode) of Jindivik nonlinear model*

u (ft/s) h (ft)	464 9,900	464 10,500	464 11,100	497 9,900	497 10,500	497 11,100	531 9,900	531 10,500	531 11,100
w_{trim}^\dagger	3.0392	3.5079	3.9905	0.1690	0.6163	1.0735	-2.5063	-2.0815	-1.6408
θ_{trim}^\ddagger	0.00655	0.00756	0.00860	0.00034	0.00124	0.00216	-0.00472	-0.00392	-0.00309
Y_v	-0.1087	-0.1066	-0.1046	-0.1166	-0.1144	-0.1122	-0.1245	-0.1221	-0.1199
Y_p	2.9247	3.3996	3.8851	0.0455	0.4965	0.9576	-2.6444	-2.2138	-1.7736
Y_r	-464.1378	-464.1345	-464.1305	-497.9039	-497.9036	-497.9028	-531.6542	-531.6561	-531.6576
Y_ϕ	32.1735	32.1732	32.1730	32.1742	32.1741	32.1741	32.1738	32.1739	32.1740
Y_{δ_a}	0	0	0	0	0	0	0	0	0
Y_{δ_r}	15.6427	15.3502	15.0620	18.0007	17.6642	17.3325	20.5242	20.1405	19.7623
L_v	-0.0163	-0.0161	-0.0160	-0.0163	-0.0162	-0.0160	-0.0165	-0.0163	-0.0161
L_p	-1.4050	-1.3787	-1.3529	-1.5072	-1.4790	-1.4512	-1.6094	-1.5793	-1.5496
L_r	1.0956	1.0744	1.0536	1.1803	1.1576	1.1352	1.2647	1.2404	1.2166
L_{δ_a}	-23.4904	-23.0512	-22.6183	-27.0314	-26.5260	-26.0279	-30.8210	-30.2447	-29.6768
L_{δ_r}	-7.5644	-7.4230	-7.2836	-8.7047	-8.5419	-8.3815	-9.9250	-9.7394	-9.5565
N_v	0.0119	0.0116	0.0114	0.0129	0.0127	0.0124	0.0139	0.0137	0.0134
N_p	-0.0131	-0.0136	-0.0142	-0.0081	-0.0087	-0.0093	-0.0035	-0.0041	-0.0048
N_r	-0.2156	-0.2116	-0.2076	-0.2313	-0.2270	-0.2227	-0.2470	-0.2424	-0.2378
N_{δ_a}	-0.1990	-0.1798	-0.1608	-0.3544	-0.3322	-0.3104	-0.5206	-0.4954	-0.4705
N_{δ_r}	0.8753	0.8589	0.8428	1.0072	0.9884	0.9698	1.1484	1.1270	1.1058
ϕ_r	0.0066	0.0076	0.0086	0.0003	0.0012	0.0022	-0.0047	-0.0039	-0.0031
ψ_r	1	1	1	1	1	1	1	1	1

* about wings level and constant altitude and airspeed flight condition

 † ft/s ‡ rad

Table 7.2: Jindivik's lateral aerodynamic coefficients

Coefficient	C_{Y_0}	C_{Y_β}	C_{Y_p}	C_{Y_r}	$C_{Y_{\delta_a}}$	$C_{Y_{\delta_r}}$
Value	0	-0.3804	-0.0428	0.0022	0	0.1180
Coefficient	C_{l_0}	C_{l_β}	C_{l_p}	C_{l_r}	$C_{l_{\delta_a}}$	$C_{l_{\delta_r}}$
Value	0	-0.0589	-0.5704	0.4463	-0.1931	-0.0622
Coefficient	C_{n_0}	C_{n_β}	C_{n_p}	C_{n_r}	$C_{n_{\delta_a}}$	$C_{n_{\delta_r}}$
Value	0	0.0810	-0.0062	-0.1546	-0.0042	0.0127

Equations (7.7)–(7.24) are the stability and control derivatives (lateral mode), where values of w_{trim} and θ_{trim} can be calculated using (5.33) and (5.34) respectively. Furthermore, the stability and control derivatives in the above equations are nonlinearly dependent on only speed and altitude. With the speed and altitude frozen, (7.6) is a lateral LTI model. However, as the speed and altitude vary over the entire flight envelope, (7.6) becomes a lateral nonlinearly parameter-dependent LPV model.

Moreover, these equations show that the accuracy of this LPV model depends on the accuracy of the information that provides the aerodynamic coefficients. Using functions `trim` and `linmod`, the stability and control derivative values, shown in Table 7.1, are obtained about one flight condition (speed = 497 ft/s and altitude = 10,500 ft). After substituting the data from Table 7.1 into (7.7)–(7.24), we can determine approximate aerodynamic coefficients of the UAV as shown in Table 7.2.

Knowing the aerodynamic coefficients, the system matrices of the lateral nonlinearly parameter-dependent LPV model (7.6) at all points over the entire parameter spaces can be determined using (7.7)–(7.24). In addition, Figure 7.1 shows the variation of open-loop characteristic of the lateral nonlinearly parameter-dependent LPV model, i.e. $v(s)/\delta_r(s)$, over an entire flight envelope. According to Figure 7.1, one pole of spiral mode is open-loop unstable with constant damping ratio (ξ) of -1 and variation of the natural frequency (ω_n) from 0.0222 rad/s to 0.0535 rad/s, one pole of roll mode is open-loop stable with constant ξ of 1 and variation of ω_n from 0.831 rad/s to 3.21 rad/s, and the other two poles of Dutch roll mode are open-loop stable with variation of ξ and ω_n from 0.0449 to 0.0705 and 1.6 rad/s to 4.41 rad/s respectively. Moreover, the system of (7.6) also has non-minimum phase zeros as shown in figure 7.1.

7.2 Stability Analysis of Uncertain Affine LPV Systems

An uncertain affine LPV system that is extended from an affine LPV system [45] is given by

$$\dot{x} = \mathcal{A}(\theta, \delta)x, \quad x(0) = x_0 \quad (7.25)$$

where $\delta = [\delta_1, \dots, \delta_m]^T \in R^m$ is a vector of time-varying real parametric uncertainty which cannot be measured. The plant state matrix $\mathcal{A}(\theta, \delta)$ is assumed to depend affinely on both the scheduled parameters θ and parametric uncertainties δ . That is

$$\mathcal{A}(\theta, \delta) = \mathcal{A}(\theta) + \delta_1 \mathcal{A}_{\delta_1}(\theta) + \delta_2 \mathcal{A}_{\delta_2}(\theta) + \dots + \delta_m \mathcal{A}_{\delta_m}(\theta) \quad (7.26)$$

where

$$\begin{aligned} \mathcal{A}(\theta) &= \mathcal{A}_0 + \theta_1 \mathcal{A}_1 + \dots + \theta_n \mathcal{A}_n \\ \mathcal{A}_{\delta_i}(\theta) &= \mathcal{A}_{\delta_{i0}} + \theta_1 \mathcal{A}_{\delta_{i1}} + \dots + \theta_n \mathcal{A}_{\delta_{in}}, \quad i = 1, \dots, m \end{aligned} \quad (7.27)$$

with $\mathcal{A}_0, \mathcal{A}_1, \dots, \mathcal{A}_n$ are known fixed matrices. $\mathcal{A}_{\delta_{i0}}, \mathcal{A}_{\delta_{i1}}, \dots, \mathcal{A}_{\delta_{in}}, i = 1, \dots, m$, are also known fixed matrices. We assume that each uncertainty δ_i is assumed to lie between known bound values $\underline{\delta}_i$ and $\bar{\delta}_i$, $\delta_i \in [\underline{\delta}_i, \bar{\delta}_i]$, and δ lies in a polytope Δ , $\delta \in \Delta$.

Define the normalized parametric uncertainties $\tilde{\delta}_i \in [-1, 1]$, $i = 1, \dots, m$ as [83]

$$\tilde{\delta}_i = \frac{\delta_i - T_i}{S_i}, \quad T_i = \frac{\bar{\delta}_i + \underline{\delta}_i}{2}, \quad S_i = \frac{\bar{\delta}_i - \underline{\delta}_i}{2} \quad (7.28)$$

Substitute (7.28) into (7.26), we get

$$\mathcal{A}(\theta, \delta) = \mathcal{A}(\theta, \tilde{\delta}) = \mathcal{A}(\theta) + \tilde{\delta}_1 \mathcal{A}_{\delta_1}(\theta) + \tilde{\delta}_2 \mathcal{A}_{\delta_2}(\theta) + \dots + \tilde{\delta}_m \mathcal{A}_{\delta_m}(\theta) \quad (7.29)$$



Figure 7.1: The open-loop characteristic of transfer function $\frac{v(s)}{\delta_r(s)}$ of the Jindivik lateral LPV model

where

$$\begin{aligned}
 A(\theta) &= \left(A_0 + \sum_{i=1}^m T_i \mathcal{A}_{\delta_{i0}} \right) + \theta_1 \left(A_1 + \sum_{i=1}^m T_i \mathcal{A}_{\delta_{i1}} \right) + \cdots + \theta_n \left(A_n + \sum_{i=1}^m T_i \mathcal{A}_{\delta_{in}} \right) \\
 &= \tilde{A}_0 + \theta_1 \tilde{A}_1 + \cdots + \theta_n \tilde{A}_n \\
 A_{\delta_i}(\theta) &= S_i \mathcal{A}_{\delta_{i0}} + \theta_1 S_i \mathcal{A}_{\delta_{i1}} + \cdots + \theta_n S_i \mathcal{A}_{\delta_{in}}, \quad i = 1, \dots, m
 \end{aligned} \tag{7.30}$$

and

$$\tilde{A}_j = A_i + \sum_{i=1}^m T_i A_{\delta_{ij}}, \quad j = 0, 1, \dots, n$$

Based on the LFT technique [104], the parametric uncertainties $\tilde{\delta}$ in (7.29) (which are unknown but bounded) can be separated from the system state-space model matrices (which are known) as

$$\begin{aligned}
 \begin{bmatrix} \dot{x} \\ z_{\delta_1} \\ z_{\delta_2} \\ \vdots \\ z_{\delta_m} \end{bmatrix} &= \begin{bmatrix} A(\theta) & B_{\delta_1}(\theta) & B_{\delta_2}(\theta) & \cdots & B_{\delta_m}(\theta) \\ C_{\delta_1}(\theta) & D_{\delta_{11}}(\theta) & D_{\delta_{12}}(\theta) & \cdots & D_{\delta_{1m}}(\theta) \\ C_{\delta_2}(\theta) & D_{\delta_{21}}(\theta) & D_{\delta_{22}}(\theta) & \cdots & D_{\delta_{2m}}(\theta) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{\delta_m}(\theta) & D_{\delta_{m1}}(\theta) & D_{\delta_{m2}}(\theta) & \cdots & D_{\delta_{mm}}(\theta) \end{bmatrix} \begin{bmatrix} x \\ w_{\delta_1} \\ w_{\delta_2} \\ \vdots \\ w_{\delta_m} \end{bmatrix} \\
 \begin{bmatrix} w_{\delta_1} \\ w_{\delta_2} \\ \vdots \\ w_{\delta_m} \end{bmatrix} &= \begin{bmatrix} \tilde{\delta}_1 I_{s_1} & 0 & \cdots & 0 \\ 0 & \tilde{\delta}_2 I_{s_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \tilde{\delta}_m I_{s_m} \end{bmatrix} \begin{bmatrix} z_{\delta_1} \\ z_{\delta_2} \\ \vdots \\ z_{\delta_m} \end{bmatrix}
 \end{aligned} \tag{7.31}$$

where $w_{\delta_i}, z_{\delta_i} \in R^{s_i}$ and

$$\begin{aligned}
 B_{\delta_i}(\theta) &= B_{\delta_{i0}} + \theta_1 B_{\delta_{i1}} + \theta_2 B_{\delta_{i2}} + \cdots + \theta_n B_{\delta_{in}} \\
 C_{\delta_i}(\theta) &= C_{\delta_{i0}} + \theta_1 C_{\delta_{i1}} + \theta_2 C_{\delta_{i2}} + \cdots + \theta_n C_{\delta_{in}} \\
 D_{\delta_{ii}}(\theta) &= D_{\delta_{ii0}} + \theta_1 D_{\delta_{ii1}} + \theta_2 D_{\delta_{ii2}} + \cdots + \theta_n D_{\delta_{iin}}
 \end{aligned} \tag{7.32}$$

Note that $B_{\delta_i}(\theta), C_{\delta_i}(\theta), i = 1, \dots, m$, in (7.31) are the factors of $A_{\delta_i}(\theta)$ in (7.29) whereby either $B_{\delta_i}(\theta)$ or $C_{\delta_i}(\theta)$ depends affinely on the scheduled parameters θ . Moreover, $D_{\delta_{ii}}(\theta)$ is introduced in order that (7.31) is in a general state-space equation form. With notation

$$\begin{aligned}
 w_{\delta} &= [w_{\delta_1}^T \quad w_{\delta_2}^T \quad \cdots \quad w_{\delta_m}^T]^T \\
 z_{\delta} &= [z_{\delta_1}^T \quad z_{\delta_2}^T \quad \cdots \quad z_{\delta_m}^T]^T \\
 B_{\delta}(\theta) &= [B_{\delta_1}(\theta) \quad B_{\delta_2}(\theta) \quad \cdots \quad B_{\delta_m}(\theta)] \\
 C_{\delta}(\theta) &= [C_{\delta_1}^T(\theta) \quad C_{\delta_2}^T(\theta) \quad \cdots \quad C_{\delta_m}^T(\theta)]^T \\
 D_{\delta}(\theta) &= \begin{bmatrix} D_{\delta_{11}}(\theta) & D_{\delta_{12}}(\theta) & \cdots & D_{\delta_{1m}}(\theta) \\ D_{\delta_{21}}(\theta) & D_{\delta_{22}}(\theta) & \cdots & D_{\delta_{2m}}(\theta) \\ \vdots & \vdots & \ddots & \vdots \\ D_{\delta_{m1}}(\theta) & D_{\delta_{m2}}(\theta) & \cdots & D_{\delta_{mm}}(\theta) \end{bmatrix}
 \end{aligned} \tag{7.33}$$

Equation (7.31) can be rewritten as

$$\begin{aligned} \dot{x} &= A(\theta)x + B_\delta(\theta)w_\delta \\ z_\delta &= C_\delta(\theta)x + D_\delta(\theta)w_\delta \\ w_\delta &= \hat{\Delta}z_\delta \end{aligned} \quad (7.34)$$

where $w_\delta, z_\delta \in \mathbb{R}^s$, $s = s_1 + s_2 + \dots + s_m$, $\hat{\Delta} = \text{diag}(\tilde{\delta}_1 I_{s_1}, \tilde{\delta}_2 I_{s_2}, \dots, \tilde{\delta}_m I_{s_m})$, and $\|\hat{\Delta}\| \leq 1$. Figure 7.2 shows the structures of both uncertain affine LPV and uncertain LPV/LFT [9] models where $\tilde{\theta}_i \in [-1, 1]$, $i = 1, \dots, n$, is the normalized time-varying parameters and can be computed using (3.36). In general, the uncertain LPV/LFT [9] model is more conservative than the proposed uncertain affine LPV model since the scheduled parameters θ of an LPV/LFT model are in the uncertainty block in which the variations of parameters are allowed to be complex, thus it introduces conservatism when the scheduled parameters are real [10, 60, 83].

The plant state matrix $A(\theta)$ in (7.34) can also be written as a convex combination of the matrix vertices as

$$A(\theta) = \text{Co} \left\{ \hat{A}_1, \hat{A}_2, \dots, \hat{A}_r \right\} = \alpha_1 \hat{A}_1 + \alpha_2 \hat{A}_2 + \dots + \alpha_r \hat{A}_r \quad (7.35)$$

where $r = 2^n$, α_i is determined using (3.24) and (3.25) and

$$\begin{bmatrix} \hat{A}_1 \\ \hat{A}_2 \\ \hat{A}_3 \\ \vdots \\ \hat{A}_r \end{bmatrix} = \begin{bmatrix} 1 & \underline{\theta}_1 & \underline{\theta}_2 & \dots & \underline{\theta}_{n-1} & \underline{\theta}_n \\ 1 & \underline{\theta}_1 & \underline{\theta}_2 & \dots & \underline{\theta}_{n-1} & \underline{\theta}_n \\ 1 & \underline{\theta}_1 & \underline{\theta}_2 & \dots & \underline{\theta}_{n-1} & \underline{\theta}_n \\ \vdots & & & & & \\ 1 & \bar{\theta}_1 & \bar{\theta}_2 & \dots & \bar{\theta}_{n-1} & \bar{\theta}_n \end{bmatrix} \begin{bmatrix} \tilde{A}_0 \\ \tilde{A}_1 \\ \vdots \\ \tilde{A}_n \end{bmatrix} \quad (7.36)$$

Note that $B_\delta(\theta)$, $C_\delta(\theta)$ and $D_\delta(\theta)$ in (7.34) can also be written as a convex combination of the matrix vertices in a similar manner to (7.35).

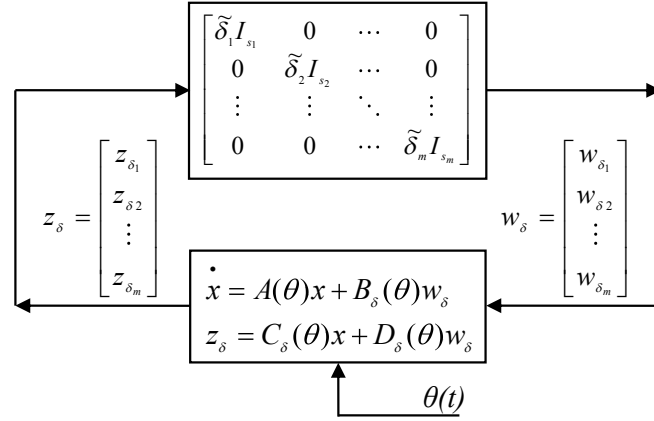
7.2.1 Robustness Analysis using SQLF

The system of (7.34) is said to be quadratically stable if there exists a quadratic Lyapunov function $V(x) = x^T P x$ whose derivative is negative, $d/dt(V(x)) < 0$, along all possible parameter trajectories, $\forall \theta \in \Theta$, for all $\hat{\Delta}$ with $\|\hat{\Delta}\| \leq 1 < 1/\gamma$. This is equivalent to the existence of a $P = P^T$ such that

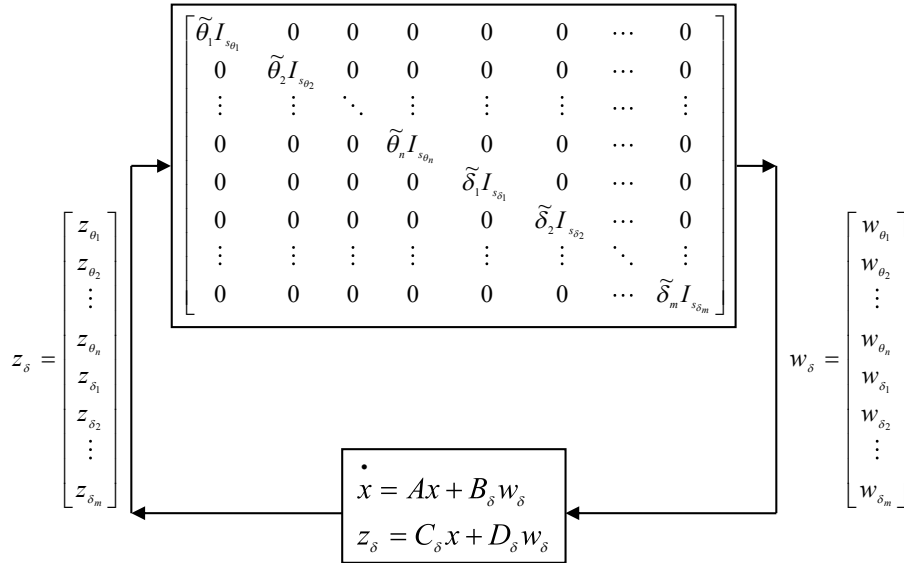
$$P > 0, \begin{bmatrix} x \\ w_\delta \end{bmatrix}^T \begin{bmatrix} A^T(\theta)P + PA(\theta) & PB_\delta(\theta) \\ B_\delta^T(\theta)P & 0 \end{bmatrix} \begin{bmatrix} x \\ w_\delta \end{bmatrix} < 0 \quad (7.37)$$

for all nonzero x satisfying [25]

$$\begin{aligned} w_\delta^T w_\delta &= (C_\delta(\theta)x + D_\delta(\theta)w_\delta)^T \hat{\Delta}^T \hat{\Delta} (C_\delta(\theta)x + D_\delta(\theta)w_\delta) \\ &< \frac{1}{\gamma^2} (C_\delta(\theta)x + D_\delta(\theta)w_\delta)^T (C_\delta(\theta)x + D_\delta(\theta)w_\delta) \end{aligned} \quad (7.38)$$



(a) Uncertain affine LPV structure



(b) Uncertain LPV/LFT structure

Figure 7.2: Structure comparisons of uncertain affine LPV and uncertain LPV/LFT [9] models.

where (7.38) can be written further as

$$\begin{bmatrix} x \\ w_\delta \end{bmatrix}^T \begin{bmatrix} -C_\delta(\theta)^T C_\delta(\theta) & -C_\delta(\theta)^T D_\delta(\theta) \\ -D_\delta^T(\theta) C_\delta(\theta) & \gamma^2 I - D_\delta^T(\theta) D_\delta(\theta) \end{bmatrix} \begin{bmatrix} x \\ w_\delta \end{bmatrix} < 0 \quad (7.39)$$

Following [25, pp.62-63], by applying the S -procedure, the quadratic stability of (7.34) is equivalent to the existence of P and λ satisfying

$$P > 0, \lambda \geq 0, \begin{bmatrix} A^T(\theta)P + PA(\theta) + \lambda C_\delta(\theta)^T C_\delta(\theta) & PB_\delta(\theta) + \lambda C_\delta(\theta)^T D_\delta(\theta) \\ B_\delta^T(\theta)P + \lambda D_\delta^T(\theta) C_\delta(\theta) & -\lambda \gamma^2 I + \lambda D_\delta^T(\theta) D_\delta(\theta) \end{bmatrix} < 0 \quad (7.40)$$

Having rearranged (7.40) by using the Schur complement [44, Lemma 3.2], (7.40) becomes the well-known bounded real lemma [10] inequality

$$\begin{pmatrix} A^T(\theta)P + PA(\theta) & PB_\delta(\theta) & C_\delta^T(\theta) \\ B_\delta^T(\theta)P & -\gamma I & D_\delta^T(\theta) \\ C_\delta(\theta) & D_\delta(\theta) & -\gamma I \end{pmatrix} < 0 \quad (7.41)$$

Note that P in (7.41) has been scaled by $1/\lambda$. The robust stability requirement is that $\gamma < 1$. However there generally exist an infinite number of the factor matrices pairs $(B_\delta(\theta), C_\delta(\theta))$ in which only some factor matrices pair give $\gamma < 1$. Instead of searching for such factor matrices pair manually, by introducing a scaling matrix $L^{1/2}$, we can select any factor matrices pair for which γ will always be less than unity if the system (7.34) is quadratically stable and the factor matrices pair can be determined using a singular value decomposition where $L^{1/2}$ denotes the unique positive definite square root of $L \in L_\Delta$. The set L_Δ is defined as

$$L_\Delta = \left\{ L > 0 : L\hat{\Delta} = \hat{\Delta}L, \forall \delta \in \Delta \right\} \subset \mathbb{R}^{s \times s} \quad (7.42)$$

Therefore, (7.34) can be modified further as

$$\begin{aligned} \dot{x} &= A(\theta)x + B_\delta(\theta)L^{-\frac{1}{2}}\dot{w}_\delta \\ \dot{z}_\delta &= L^{\frac{1}{2}}C_\delta(\theta)x + L^{\frac{1}{2}}D_\delta(\theta)L^{-\frac{1}{2}}\dot{w}_\delta \\ \dot{w}_\delta &= \hat{\Delta}\dot{z}_\delta \end{aligned} \quad (7.43)$$

where $\dot{z}_\delta = L^{\frac{1}{2}}\dot{z}_\delta$ and $w_\delta = L^{-\frac{1}{2}}\dot{w}_\delta$. With parameters from (7.43), (7.41) becomes a scaled bounded real lemma [8, 9] inequality

$$\begin{pmatrix} A^T(\theta)P + PA(\theta) & PB_\delta(\theta) & C_\delta^T(\theta) \\ B_\delta^T(\theta)P & -\gamma L & D_\delta^T(\theta) \\ C_\delta(\theta) & D_\delta(\theta) & -\gamma L^{-1} \end{pmatrix} < 0 \quad (7.44)$$

Substituting (7.35) into (7.44) and rearranging using the Schur complement [44, Lemma 3.2], we get

$$\sum_{i=1}^r \alpha_i \begin{pmatrix} \hat{A}_i^T P + P\hat{A}_i + \hat{C}_{\delta_i}^T L \hat{C}_{\delta_i} & P\hat{B}_{\delta_i} + \hat{C}_{\delta_i}^T L \hat{D}_{\delta_i} \\ \hat{B}_{\delta_i}^T P + \hat{D}_{\delta_i}^T L \hat{C}_{\delta_i} & -\gamma^2 L + \hat{D}_{\delta_i}^T L \hat{D}_{\delta_i} \end{pmatrix} < 0 \quad (7.45)$$

By Lemma 3.3.1, solving the above inequality for some positive definite symmetric Lyapunov and scaling matrices (P, L) need only be done at all vertices. Hence we get the following proposition.

Proposition 7.2.1. *The system of (7.34) is quadratically stable along all possible parameter trajectories, $\forall \theta \in \Theta$, for all $\hat{\Delta}$ with $\|\hat{\Delta}\| < 1/\gamma$ if and only if the following LMI conditions hold for some positive definite symmetric Lyapunov and scaling matrices (P, L)*

$$\begin{pmatrix} \hat{A}_i^T P + P \hat{A}_i + \hat{C}_{\delta_i}^T L \hat{C}_{\delta_i} & P \hat{B}_{\delta_i} + \hat{C}_{\delta_i}^T L \hat{D}_{\delta_i} \\ \hat{B}_{\delta_i}^T P + \hat{D}_{\delta_i}^T L \hat{C}_{\delta_i} & -\gamma^2 L + \hat{D}_{\delta_i}^T L \hat{D}_{\delta_i} \end{pmatrix} < 0 \quad (7.46)$$

where $i = 1, \dots, r$

Note that, the minimization of γ is achieved heuristically or by a simple grid search.

7.2.2 Robustness Analysis using PDLF

With parameter-dependent Lyapunov functions, $V(x, \theta) = x^T P(\theta)x$, (7.44) becomes

$$\begin{pmatrix} A^T(\theta)P(\theta) + P(\theta)A(\theta) + \dot{P}(\theta) & P(\theta)B_\delta(\theta) & C_\delta^T(\theta) \\ B_\delta^T(\theta)P(\theta) & -\gamma L & D_\delta^T(\theta) \\ C_\delta(\theta) & D_\delta(\theta) & -\gamma L^{-1} \end{pmatrix} < 0 \quad (7.47)$$

Substituting (7.35), (6.2) and (6.4) into (7.47), rearranging using the Schur complement [44, Lemma 3.2], and recalling that $\sum_{i=1}^r \alpha_i = 1$ and $\sum_{i=1}^r \beta_i = 1$, we get

$$\begin{aligned} & \sum_{i=1}^r \sum_{k=1}^r \alpha_i^2 \beta_k \begin{pmatrix} \hat{A}_i^T \hat{P}_i + \hat{P}_i \hat{A}_i + \tilde{P}_k + \hat{C}_{\delta_i}^T L \hat{C}_{\delta_i} & \hat{P}_i \hat{r}_{\delta_i} + \hat{C}_{\delta_i}^T L \hat{D}_{\delta_i} \\ \hat{r}_{\delta_i}^T \hat{P}_i + \hat{D}_{\delta_i}^T L \hat{C}_{\delta_i} & -\gamma^2 L + \hat{D}_{\delta_i}^T L \hat{D}_{\delta_i} \end{pmatrix} \\ & + \sum_{i=1}^{r-1} \sum_{j=i+1}^r \sum_{k=1}^r \alpha_i \alpha_j \beta_k \begin{pmatrix} \hat{A}_i^T \hat{P}_j + \hat{P}_j \hat{A}_i + \hat{A}_j^T \hat{P}_i + \hat{P}_i \hat{A}_j + 2\tilde{P}_k + \hat{C}_{\delta_i}^T L \hat{C}_{\delta_i} + \hat{C}_{\delta_j}^T L \hat{C}_{\delta_j} \\ \hat{r}_{\delta_i}^T \hat{P}_j + \hat{r}_{\delta_j}^T \hat{P}_i + \hat{D}_{\delta_i}^T L \hat{C}_{\delta_i} + \hat{D}_{\delta_j}^T L \hat{C}_{\delta_j} \\ \hat{P}_i \hat{r}_{\delta_j} + \hat{P}_j \hat{r}_{\delta_i} + \hat{C}_{\delta_i}^T L \hat{D}_{\delta_i} + \hat{C}_{\delta_j}^T L \hat{D}_{\delta_j} \\ -2\gamma^2 L + \hat{D}_{\delta_i}^T L \hat{D}_{\delta_i} + \hat{D}_{\delta_j}^T L \hat{D}_{\delta_j} \end{pmatrix} < 0 \end{aligned} \quad (7.48)$$

By Lemma 3.3.1, the above inequalities are sufficiently evaluated at all vertices for which the total number of LMIs to be solved is $r^2(r+3)/2$. Hence we get the following proposition.

Proposition 7.2.2. *The system of (7.34) is parameter-dependent stable along all possible parameter trajectories, $\forall (\theta, \hat{\theta}) \in \Theta \times \Phi$, for all $\hat{\Delta}$ with $\|\hat{\Delta}\| < 1/\gamma$ if and only if the following LMI conditions hold for some positive definite symmetric Lyapunov and scaling matrices (P, L)*

$$\begin{pmatrix} \hat{A}_i^T \hat{P}_i + \hat{P}_i \hat{A}_i + \tilde{P}_k + \hat{C}_{\delta_i}^T L \hat{C}_{\delta_i} & \hat{P}_i \hat{r}_{\delta_i} + \hat{C}_{\delta_i}^T L \hat{D}_{\delta_i} \\ \hat{r}_{\delta_i}^T \hat{P}_i + \hat{D}_{\delta_i}^T L \hat{C}_{\delta_i} & -\gamma^2 L + \hat{D}_{\delta_i}^T L \hat{D}_{\delta_i} \end{pmatrix} < 0 \quad (7.49)$$

$$\begin{pmatrix} \hat{A}_i^T \hat{P}_j + \hat{P}_j \hat{A}_i + \hat{A}_j^T \hat{P}_i + \hat{P}_i \hat{A}_j + 2\tilde{P}_k + \hat{C}_{\delta_i}^T L \hat{C}_{\delta_i} + \hat{C}_{\delta_j}^T L \hat{C}_{\delta_j} \\ \hat{r}_{\delta_i}^T \hat{P}_j + \hat{r}_{\delta_j}^T \hat{P}_i + \hat{D}_{\delta_i}^T L \hat{C}_{\delta_i} + \hat{D}_{\delta_j}^T L \hat{C}_{\delta_j} \\ \hat{P}_i \hat{r}_{\delta_j} + \hat{P}_j \hat{r}_{\delta_i} + \hat{C}_{\delta_i}^T L \hat{D}_{\delta_i} + \hat{C}_{\delta_j}^T L \hat{D}_{\delta_j} \\ -2\gamma^2 L + \hat{D}_{\delta_i}^T L \hat{D}_{\delta_i} + \hat{D}_{\delta_j}^T L \hat{D}_{\delta_j} \end{pmatrix} < 0 \quad (7.50)$$

where $i, k = 1, \dots, r$ and $1 \leq i < j \leq r$

Note that, the minimization of γ is achieved heuristically or by a simple grid search.

7.3 Controller Synthesis for Uncertain Affine LPV Systems

In the previous section, a sufficient condition for robust stability that can guarantee the closed-loop system property has been presented. This is especially relevant for robustness analysis using parameter-dependent Lyapunov functions, where a finite number of LMIs need only be evaluated at all vertices. By including time-varying real parametric uncertainties δ in an affine LPV model as shown in (7.34), our stability analysis can guarantee a larger stability margin over conventional stability analysis technique. Next, we consider the problem of designing a gain-scheduled output feedback H_∞ control with guaranteed L_2 -gain performance for uncertain affine LPV systems for which the proposed technique in the previous section can be directly extended to designing a gain-scheduled H_∞ controller. The material in this section is drawn from [8, 10, 43] and [44].

Consider an uncertain affine LPV model that is extended from an affine LPV model (3.59) is given by

$$\begin{aligned} \dot{x} &= \mathcal{A}(\theta, \delta)x + \mathcal{B}_1(\theta, \delta)w + B_2u \\ z &= \mathcal{C}_1(\theta, \delta)x + \mathcal{D}_{11}(\theta, \delta)w + D_{12}u \\ y &= C_2x + D_{21}w \end{aligned} \quad (7.51)$$

where $\mathcal{B}_1(\theta, \delta)$, $\mathcal{C}_1(\theta, \delta)$, and $\mathcal{D}_{11}(\theta, \delta)$ can be written in a similar manner to (7.26). Based on the LFT technique [104], the uncertainty δ in (7.51) can be separated from the system matrices in a similar manner to (7.31) and (7.43), giving

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ z_{\delta_1} \\ \vdots \\ z_{\delta_m} \\ z \\ y \end{bmatrix} &= \begin{bmatrix} A(\theta) & B_{\delta_1}(\theta)L^{-\frac{1}{2}} & \dots & B_{\delta_m}(\theta)L^{-\frac{1}{2}} & B_1(\theta) & B_2 \\ L^{\frac{1}{2}}C_{\delta_1}(\theta) & L^{\frac{1}{2}}D_{\delta\delta_{11}}(\theta)L^{-\frac{1}{2}} & \dots & L^{\frac{1}{2}}D_{\delta\delta_{1m}}(\theta)L^{-\frac{1}{2}} & L^{\frac{1}{2}}D_{\delta 1_1}(\theta) & L^{\frac{1}{2}}D_{\delta 2} \\ \vdots & & & & & \\ L^{\frac{1}{2}}C_{\delta_m}(\theta) & L^{\frac{1}{2}}D_{\delta\delta_{1m}}(\theta)L^{-\frac{1}{2}} & \dots & L^{\frac{1}{2}}D_{\delta\delta_{mm}}(\theta)L^{-\frac{1}{2}} & L^{\frac{1}{2}}D_{\delta 1_m}(\theta) & L^{\frac{1}{2}}D_{\delta 2} \\ C_1(\theta) & D_{1\delta_1}(\theta)L^{-\frac{1}{2}} & \dots & D_{1\delta_m}(\theta)L^{-\frac{1}{2}} & D_{11}(\theta) & D_{12} \\ C_2 & D_{2\delta}L^{-\frac{1}{2}} & \dots & D_{2\delta}L^{-\frac{1}{2}} & D_{21} & 0 \end{bmatrix} \begin{bmatrix} x \\ w_{\delta_1} \\ \vdots \\ w_{\delta_m} \\ w \\ u \end{bmatrix} \\ \begin{bmatrix} w_{\delta_1} \\ w_{\delta_2} \\ \vdots \\ w_{\delta_m} \end{bmatrix} &= \begin{bmatrix} \tilde{\delta}_1 I_{s_1} & 0 & \dots & 0 \\ 0 & \tilde{\delta}_2 I_{s_2} & \dots & 0 \\ \vdots & & & \\ 0 & 0 & \dots & \tilde{\delta}_m I_{s_m} \end{bmatrix} \begin{bmatrix} z_{\delta_1} \\ z_{\delta_2} \\ \vdots \\ z_{\delta_m} \end{bmatrix} \end{aligned} \quad (7.52)$$

where $L \in L_\Delta$ is a scaling matrix that has to be determined, $w_{\delta_i}, z_{\delta_i} \in R^{s_i}$, $\tilde{\delta}_i \in [-1, 1]$, $i = 1, \dots, m$, the normalized parametric uncertainties which can be

determined using (7.28), and

$$\begin{bmatrix} A_{\delta_i}(\theta) & B_{1\delta_i}(\theta) \\ C_{1\delta_i}(\theta) & D_{11\delta_i}(\theta) \end{bmatrix} = \begin{bmatrix} B_{\delta_i}(\theta) \\ D_{1\delta_i}(\theta) \end{bmatrix} \begin{bmatrix} C_{\delta_i}(\theta) & D_{\delta 1_i}(\theta) \end{bmatrix} \quad (7.53)$$

Note that either $[B_{\delta_i}^T(\theta), D_{1\delta_i}^T(\theta)]^T$ or $[C_{\delta_i}(\theta), D_{\delta 1_i}(\theta)]$ depends affinely on θ . Moreover, $D_{\delta\delta_{ii}}(\theta)$, $D_{\delta 2}$, and $D_{2\delta}$ are introduced in order that (7.52) is in a general state-space equation form. With the notation of (7.33), we can rewrite (7.52) as

$$\begin{bmatrix} \dot{x} \\ z_\delta \\ z \\ y \end{bmatrix} = \begin{bmatrix} A(\theta) & B_\delta(\theta)L^{-\frac{1}{2}} & B_1(\theta) & B_2 \\ L^{\frac{1}{2}}C_\delta(\theta) & L^{\frac{1}{2}}D_{\delta\delta}(\theta)L^{-\frac{1}{2}} & L^{\frac{1}{2}}D_{\delta 1}(\theta) & L^{\frac{1}{2}}D_{\delta 2} \\ C_1(\theta) & D_{1\delta}(\theta)L^{-\frac{1}{2}} & D_{11}(\theta) & D_{12} \\ C_2 & D_{2\delta}L^{-\frac{1}{2}} & D_{21} & 0 \end{bmatrix} \begin{bmatrix} x \\ w_\delta \\ w \\ u \end{bmatrix} \quad (7.54)$$

$$w_\delta = \hat{\Delta}z_\delta$$

where $w_\delta, z_\delta \in \mathbb{R}^s$, $s = s_1 + s_2 + \dots + s_m$, $\hat{\Delta} = \text{diag}(\tilde{\delta}_1 I_{s_1}, \tilde{\delta}_2 I_{s_2}, \dots, \tilde{\delta}_m I_{s_m})$, and $\|\hat{\Delta}\| \leq 1$. Note that matrices $B_\delta(\theta)$, $B_1(\theta)$, $C_\delta(\theta)$, $C_1(\theta)$, $D_{\delta\delta}(\theta)$, $D_{\delta 1}(\theta)$, $D_{1\delta}(\theta)$, and $D_{11}(\theta)$ can also be written as a convex combination of the matrix vertices in a similar manner to (7.35) as

$$\begin{pmatrix} A(\theta) & B_\delta(\theta) & B_1(\theta) & B_2 \\ C_\delta(\theta) & D_{\delta\delta}(\theta) & D_{\delta 1}(\theta) & D_{\delta 2} \\ C_1(\theta) & D_{1\delta}(\theta) & D_{11}(\theta) & D_{12} \\ C_2 & D_{2\delta} & D_{21} & 0 \end{pmatrix} = \sum_{i=1}^r \alpha_i \begin{pmatrix} \hat{A}_i & \hat{B}_{\delta_i} & \hat{B}_{1_i} & B_2 \\ \hat{C}_{\delta_i} & \hat{D}_{\delta\delta_i} & \hat{D}_{\delta 1_i} & D_{\delta 2} \\ \hat{C}_{1_i} & \hat{D}_{1\delta_i} & \hat{D}_{11_i} & D_{12} \\ C_2 & D_{2\delta} & D_{21} & 0 \end{pmatrix} \quad (7.55)$$

7.3.1 Gain-Scheduled Controller Design using SQLF

The gain-scheduled output feedback H_∞ control problem using single quadratic Lyapunov functions is to compute a dynamic LPV controller, $K(\theta)$, with state-space equations

$$\begin{aligned} \dot{x}_k &= A_k(\theta)x_k + B_k(\theta)y \\ u &= C_k(\theta)x_k + D_k(\theta)y \end{aligned} \quad (7.56)$$

which stabilizes the closed-loop system, (7.54) and (7.56), and minimizes the closed-loop quadratic H_∞ performance, γ , so ensuring that the induced L_2 -norm of the operator mapping the disturbance signal into the controlled signal is bounded by γ

$$\int_0^{t_1} (z_\delta^T z_\delta + z^T z) dt \leq \gamma^2 \int_0^{t_1} (w_\delta^T w_\delta + w^T w) dt, \quad \forall t_1 \geq 0 \quad (7.57)$$

along all possible parameter trajectories, $\forall \theta \in \Theta$. Note that A and A_k have the same dimensions, since we restrict ourselves to the full-order case. With the notation

$$K(\theta) = \begin{pmatrix} A_k(\theta) & B_k(\theta) \\ C_k(\theta) & D_k(\theta) \end{pmatrix} = \sum_{i=1}^r \alpha_i K_i \quad (7.58)$$

$$K_i = \begin{pmatrix} A_{k_i} & B_{k_i} \\ C_{k_i} & D_{k_i} \end{pmatrix}, \quad i = 1, 2, \dots, r \quad (7.59)$$

where r is the total number of vertices and α_i is determined using (3.24) and (3.25). the closed-loop system, (7.54) and (7.56), is described by the state-space equations

$$\begin{bmatrix} \dot{x}_{cl} \\ z_\delta \\ z \end{bmatrix} = \begin{bmatrix} A_{cl}(\theta) & B_{\delta_{cl}}(\theta)L^{-\frac{1}{2}} & B_{1_{cl}}(\theta) \\ L^{\frac{1}{2}}C_{\delta_{cl}}(\theta) & L^{\frac{1}{2}}D_{\delta_{cl}}(\theta)L^{-\frac{1}{2}} & L^{\frac{1}{2}}D_{\delta_{1_{cl}}}(\theta) \\ C_{1_{cl}}(\theta) & D_{1\delta_{cl}}(\theta)L^{-\frac{1}{2}} & D_{11_{cl}}(\theta) \end{bmatrix} \begin{bmatrix} x_{cl} \\ w_\delta \\ w \end{bmatrix}$$

$$w_\delta = \hat{\Delta}z_\delta \quad (7.60)$$

where $x_{cl} = [x^T \ x_k^T]^T$ and

$$\begin{aligned} A_{cl}(\theta) &= \begin{bmatrix} A(\theta) & 0 \\ 0 & 0_{p \times p} \end{bmatrix} + \mathcal{B}K(\theta)\mathcal{C} = \sum_{i=1}^r \alpha_i \hat{A}_{cl_i}, \quad \hat{A}_{cl_i} = \begin{bmatrix} \hat{A}_i & 0 \\ 0 & 0_{p \times p} \end{bmatrix} + \mathcal{B}K_i\mathcal{C} \\ B_{\delta_{cl}}(\theta) &= \begin{bmatrix} B_\delta(\theta) \\ 0 \end{bmatrix} + \mathcal{B}K(\theta)\mathcal{D}_{\delta_{21}} = \sum_{i=1}^r \alpha_i \hat{B}_{\delta_{cl_i}}, \quad \hat{B}_{\delta_{cl_i}} = \begin{bmatrix} \hat{B}_{\delta_i} \\ 0 \end{bmatrix} + \mathcal{B}K_i\mathcal{D}_{\delta_{21}} \\ B_{1_{cl}}(\theta) &= \begin{bmatrix} B_1(\theta) \\ 0 \end{bmatrix} + \mathcal{B}K(\theta)\mathcal{D}_{1_{21}} = \sum_{i=1}^r \alpha_i \hat{B}_{1_{cl_i}}, \quad \hat{B}_{1_{cl_i}} = \begin{bmatrix} \hat{B}_{1_i} \\ 0 \end{bmatrix} + \mathcal{B}K_i\mathcal{D}_{1_{21}} \\ C_{\delta_{cl}}(\theta) &= [C_\delta(\theta) \ 0] + \mathcal{D}_{\delta_{12}}K(\theta)\mathcal{C} = \sum_{i=1}^r \alpha_i \hat{C}_{\delta_{cl_i}}, \quad \hat{C}_{\delta_{cl_i}} = [\hat{C}_{\delta_i} \ 0] + \mathcal{D}_{\delta_{12}}K_i\mathcal{C} \\ C_{1_{cl}}(\theta) &= [C_1(\theta) \ 0] + \mathcal{D}_{1_{12}}K(\theta)\mathcal{C} = \sum_{i=1}^r \alpha_i \hat{C}_{1_{cl_i}}, \quad \hat{C}_{1_{cl_i}} = [\hat{C}_{1_i} \ 0] + \mathcal{D}_{1_{12}}K_i\mathcal{C} \\ D_{\delta\delta_{cl}}(\theta) &= D_{\delta\delta}(\theta) + \mathcal{D}_{\delta_{12}}K(\theta)\mathcal{D}_{\delta_{21}} = \sum_{i=1}^r \alpha_i \hat{D}_{\delta\delta_{cl_i}}, \quad \hat{D}_{\delta\delta_{cl_i}} = \hat{D}_{\delta\delta_i} + \mathcal{D}_{\delta_{12}}K_i\mathcal{D}_{\delta_{21}} \\ D_{\delta 1_{cl}}(\theta) &= D_{\delta 1}(\theta) + \mathcal{D}_{\delta_{12}}K(\theta)\mathcal{D}_{1_{21}} = \sum_{i=1}^r \alpha_i \hat{D}_{\delta 1_{cl_i}}, \quad \hat{D}_{\delta 1_{cl_i}} = \hat{D}_{\delta 1_i} + \mathcal{D}_{\delta_{12}}K_i\mathcal{D}_{1_{21}} \\ D_{1\delta_{cl}}(\theta) &= D_{1\delta}(\theta) + \mathcal{D}_{1_{12}}K(\theta)\mathcal{D}_{\delta_{21}} = \sum_{i=1}^r \alpha_i \hat{D}_{1\delta_{cl_i}}, \quad \hat{D}_{1\delta_{cl_i}} = \hat{D}_{1\delta_i} + \mathcal{D}_{1_{12}}K_i\mathcal{D}_{\delta_{21}} \\ D_{11_{cl}}(\theta) &= D_{11}(\theta) + \mathcal{D}_{1_{12}}K(\theta)\mathcal{D}_{1_{21}} = \sum_{i=1}^r \alpha_i \hat{D}_{11_{cl_i}}, \quad \hat{D}_{11_{cl_i}} = \hat{D}_{11_i} + \mathcal{D}_{1_{12}}K_i\mathcal{D}_{1_{21}} \end{aligned} \quad (7.61)$$

with

$$\begin{aligned} \mathcal{B} &= \begin{bmatrix} 0 & B_2 \\ I_p & 0 \end{bmatrix}, \quad \mathcal{C} = \begin{bmatrix} 0 & I_p \\ C_2 & 0 \end{bmatrix}, \quad \mathcal{D}_{\delta_{21}} = \begin{bmatrix} 0 \\ D_{2\delta} \end{bmatrix}, \quad \mathcal{D}_{1_{21}} = \begin{bmatrix} 0 \\ D_{21} \end{bmatrix} \\ \mathcal{D}_{\delta_{12}} &= [0 \ D_{\delta 2}], \quad \mathcal{D}_{1_{12}} = [0 \ D_{12}], \end{aligned} \quad (7.62)$$

Based on the single quadratic Lyapunov functions $V(x) = x^T P x$, there is an LPV controller $K(\theta)$ that stabilizes the closed-loop system, (7.54) and (7.56), and ensures the L_2 -induced norm of the operator mapping the disturbance signal into the controlled signal is bounded by γ along all possible parameter trajectories if and only if there exists $P = P^T$ such that [45]

$$P > 0, \quad \frac{d}{dt}(x^T P x) + (z_\delta^T z_\delta + z^T z) - \gamma^2(w_\delta^T w_\delta + w^T w) < 0, \quad \forall \theta \in \Theta \quad (7.63)$$

Inequality (7.63) leads to the scaled bounded real lemma [8, 9] inequality

$$\begin{pmatrix} A_{cl}^T(\theta)P + PA_{cl}(\theta) & PB_{\delta_{cl}}(\theta) & PB_{1_{cl}}(\theta) & C_{\delta_{cl}}^T(\theta)L & C_{1_{cl}}^T(\theta) \\ B_{\delta_{cl}}^T(\theta)P & -\gamma L & 0 & D_{\delta_{cl}}^T(\theta)L & D_{1_{cl}}^T(\theta) \\ B_{1_{cl}}^T(\theta)P & 0 & -\gamma I & D_{\delta_{1_{cl}}}^T(\theta)L & D_{11_{cl}}^T(\theta) \\ LC_{\delta_{cl}}(\theta) & LD_{\delta_{cl}}(\theta) & LD_{\delta_{1_{cl}}}(\theta) & -\gamma L & 0 \\ C_{1_{cl}}(\theta) & D_{1_{cl}}(\theta) & D_{11_{cl}}(\theta) & 0 & -\gamma I \end{pmatrix} < 0 \quad (7.64)$$

Substituting (7.61) into (7.64), we get

$$\sum_{i=1}^r \alpha_i \begin{pmatrix} \hat{A}_{cl_i}^T P + P \hat{A}_{cl_i} & P \hat{B}_{\delta_{cl_i}} & P \hat{B}_{1_{cl_i}} & \hat{C}_{\delta_{cl_i}}^T L & \hat{C}_{1_{cl_i}}^T \\ \hat{B}_{\delta_{cl_i}}^T P & -\gamma L & 0 & \hat{D}_{\delta_{cl_i}}^T L & \hat{D}_{1_{cl_i}}^T \\ \hat{B}_{1_{cl_i}}^T P & 0 & -\gamma I & \hat{D}_{\delta_{1_{cl_i}}}^T L & \hat{D}_{11_{cl_i}}^T \\ L \hat{C}_{\delta_{cl_i}} & L \hat{D}_{\delta_{cl_i}} & L \hat{D}_{\delta_{1_{cl_i}}} & -\gamma L & 0 \\ \hat{C}_{1_{cl_i}} & \hat{D}_{1_{cl_i}} & \hat{D}_{11_{cl_i}} & 0 & -\gamma I \end{pmatrix} < 0 \quad (7.65)$$

Inequality (7.65) can be also rewritten as [44]

$$\sum_{i=1}^r \alpha_i \left(\Psi_{cl_i} + \mathcal{Q}^T K_i^T \mathcal{P}_{cl} + \mathcal{P}_{cl}^T K_i \mathcal{Q} \right) < 0 \quad (7.66)$$

where

$$\Psi_{cl_i} = \begin{pmatrix} \begin{bmatrix} \hat{A}_i & 0 \\ 0 & 0_{p \times p} \end{bmatrix}^T P + P \begin{bmatrix} \hat{A}_i & 0 \\ 0 & 0_{p \times p} \end{bmatrix} & P \begin{bmatrix} \hat{B}_{\delta_i} \\ 0 \end{bmatrix} & P \begin{bmatrix} \hat{B}_{1_i} \\ 0 \end{bmatrix} & [\hat{C}_{\delta_i} \ 0]^T L & [\hat{C}_{1_i} \ 0]^T \\ \begin{bmatrix} \hat{B}_{\delta_i} \\ 0 \end{bmatrix}^T P & -\gamma L & 0 & \hat{D}_{\delta_{\delta_i}}^T L & \hat{D}_{1_{\delta_i}}^T \\ \begin{bmatrix} \hat{B}_{1_i} \\ 0 \end{bmatrix}^T P & 0 & -\gamma I & \hat{D}_{\delta_{1_i}}^T L & \hat{D}_{11_i}^T \\ L [\hat{C}_{\delta_i} \ 0] & L \hat{D}_{\delta_{\delta_i}} & L \hat{D}_{\delta_{1_i}} & -\gamma L & 0 \\ [\hat{C}_{1_i} \ 0] & \hat{D}_{1_{\delta_i}} & \hat{D}_{11_i} & 0 & -\gamma I \end{pmatrix} \quad (7.67)$$

$$\mathcal{Q} = [\mathcal{C}, \ \mathcal{D}_{\delta_{21}}, \ \mathcal{D}_{1_{21}}, \ 0_{(p+q_2) \times (q_1+s)}] \quad (7.68)$$

$$\mathcal{P}_{cl} = [\mathcal{B}^T P, \ 0_{(p+m_2) \times (m_1+s)}, \ \mathcal{D}_{\delta_{12}}^T, \ \mathcal{D}_{112}^T] \quad (7.69)$$

Having determined the quadratic Lyapunov variable $P \in R^{2p \times 2p}$ and the scaling matrix $L \in L_{\Delta}$, the system matrix vertices K_i of the LPV controller $K(\theta)$ for each vertex Θ_i , $i = 1, \dots, r$, can be determined from (7.66) that is an LMI in K_i . By Lemma 3.3.1, the LMIs (7.66) are sufficiently evaluated at all vertices. Knowing K_i , the controller system matrices $A_k(\theta), \dots, D_k(\theta)$ can be computed on-line in real-time using (7.58) and an instantaneous measurement value of θ .

By Lemma 2.2.14 with a known scaling matrix L , LMIs (7.66) are solvable for K_i if and only if there exist a pair of positive definite symmetric matrices (X, Y) satisfying

the following LMIs:

$$\sum_{i=1}^r \alpha_i \left(\begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \hat{A}_i^T X + X \hat{A}_i & X \hat{B}_{\delta_i} & X \hat{B}_{1_i} & \hat{C}_{\delta_i}^T & \hat{C}_{1_i}^T \\ \hat{B}_{\delta_i}^T X & -\gamma L & 0 & \hat{D}_{\delta\delta_i}^T & \hat{D}_{1\delta_i}^T \\ \hat{B}_{1_i}^T X & 0 & -\gamma I & \hat{D}_{\delta 1_i}^T & \hat{D}_{11_i}^T \\ \hat{C}_{\delta_i} & \hat{D}_{\delta\delta_i} & \hat{D}_{\delta 1_i} & -\gamma L^{-1} & 0 \\ \hat{C}_{1_i} & \hat{D}_{1\delta_i} & \hat{D}_{11_i} & 0 & -\gamma I \end{pmatrix} \begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix} \right) < 0 \quad (7.70)$$

$$\sum_{i=1}^r \alpha_i \left(\begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \hat{A}_i Y + Y \hat{A}_i^T & Y \hat{C}_{\delta_i}^T & Y \hat{C}_{1_i}^T & \hat{B}_{\delta_i} & \hat{B}_{1_i} \\ \hat{C}_{\delta_i} Y & -\gamma L^{-1} & 0 & \hat{D}_{\delta\delta_i} & \hat{D}_{\delta 1_i} \\ \hat{C}_{1_i} Y & 0 & -\gamma I & \hat{D}_{1\delta_i} & \hat{D}_{11_i} \\ \hat{B}_{\delta_i}^T & \hat{D}_{\delta\delta_i}^T & \hat{D}_{\delta 1_i}^T & -\gamma L & 0 \\ \hat{B}_{1_i}^T & \hat{D}_{1\delta_i}^T & \hat{D}_{11_i}^T & 0 & -\gamma I \end{pmatrix} \begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix} \right) < 0 \quad (7.71)$$

$$\begin{pmatrix} X & I \\ I & Y \end{pmatrix} > 0 \quad (7.72)$$

where N_X and N_Y denote bases of the null spaces of $[C_2, D_{2\delta}, D_{21}]$ and $[B_2^T, D_{\delta 2}^T, D_{12}^T]$, respectively. Note that (7.72) ensures $X, Y > 0$ and $X - Y^{-1} \geq 0$. By Lemma 3.3.1, (7.70)–(7.72) are sufficiently evaluated at all vertices. Although (7.70)–(7.72) are not standard LMI problems, they can be solved by an iterative approach, referred to as D-K iteration [14]. Like the μ -synthesis algorithms, such a scheme is not guaranteed to converge to a global minimum [14], but may find a local minimum [9]. In spite of this drawback, the D-K iteration control design technique appears to work well on many engineering problems [14].

Theorem 7.3.1. *Given a scaling matrix L , there exists an LPV controller $K(\theta)$ that guarantees the closed-loop system, (7.54) and (7.56), quadratic H_∞ performance γ along all possible parameter trajectories, $\forall \theta \in \Theta$, if and only if the following LMI conditions hold for some positive definite symmetric matrices (X, Y) , which further satisfy $\text{rank}(X - Y^{-1}) \leq p$.*

$$\begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \hat{A}_i^T X + X \hat{A}_i & X \hat{B}_{\delta_i} & X \hat{B}_{1_i} & \hat{C}_{\delta_i}^T & \hat{C}_{1_i}^T \\ \hat{B}_{\delta_i}^T X & -\gamma L & 0 & \hat{D}_{\delta\delta_i}^T & \hat{D}_{1\delta_i}^T \\ \hat{B}_{1_i}^T X & 0 & -\gamma I & \hat{D}_{\delta 1_i}^T & \hat{D}_{11_i}^T \\ \hat{C}_{\delta_i} & \hat{D}_{\delta\delta_i} & \hat{D}_{\delta 1_i} & -\gamma L^{-1} & 0 \\ \hat{C}_{1_i} & \hat{D}_{1\delta_i} & \hat{D}_{11_i} & 0 & -\gamma I \end{pmatrix} \begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix} < 0 \quad (7.73)$$

$$\begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \hat{A}_i Y + Y \hat{A}_i^T & Y \hat{C}_{\delta_i}^T & Y \hat{C}_{1_i}^T & \hat{B}_{\delta_i} & \hat{B}_{1_i} \\ \hat{C}_{\delta_i} Y & -\gamma L^{-1} & 0 & \hat{D}_{\delta\delta_i} & \hat{D}_{\delta 1_i} \\ \hat{C}_{1_i} Y & 0 & -\gamma I & \hat{D}_{1\delta_i} & \hat{D}_{11_i} \\ \hat{B}_{\delta_i}^T & \hat{D}_{\delta\delta_i}^T & \hat{D}_{\delta 1_i}^T & -\gamma L & 0 \\ \hat{B}_{1_i}^T & \hat{D}_{1\delta_i}^T & \hat{D}_{11_i}^T & 0 & -\gamma I \end{pmatrix} \begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix} < 0 \quad (7.74)$$

$$\begin{pmatrix} X & I \\ I & Y \end{pmatrix} > 0 \quad (7.75)$$

for $i = 1, \dots, r$.

Algorithm 7.3.2. *Computation of X, Y and L*

Step 1: Setting $L = I$, solve (7.73)–(7.75) for X, Y by minimizing γ .

Step 2: Knowing X, Y , and L , solve (7.66) for K_i .

Step 3: With K_i fixed, solve (7.65) for L by minimizing γ (heuristic).

Step 4: With L fixed, solve (7.73)–(7.75) for new X, Y by minimizing γ .

Step 5: Iterate over Steps 2 to 4 until γ is converged to some minimum value.

Note that, by Lemma 3.3.1, (7.65) and (7.66) are sufficiently solved at all vertices.

7.3.2 Gain-Scheduled Controller Design using PDLF

With parameter-dependent Lyapunov functions, $V(x, \theta) = x^T P(\theta)x$, (7.63) becomes

$$P(\theta) > 0, \quad \frac{d}{dt}(x^T P(\theta)x) + (z_\delta^T z_\delta + z^T z) - \gamma^2(w_\delta^T w_\delta + w^T w) < 0, \quad \forall(\theta, \dot{\theta}) \in \Theta \times \Phi \quad (7.76)$$

A state-space equations of a dynamic LPV controller, $K(\theta)$ in (7.56), becomes

$$\begin{aligned} \dot{x}_k &= A_k(\theta, \dot{\theta})x_k + B_k(\theta)y \\ u &= C_k(\theta)x_k + D_k(\theta)y \end{aligned} \quad (7.77)$$

Note that, unlike the single quadratic Lyapunov functions case, $P(\theta)$, $A_k(\theta, \dot{\theta})$, $B_k(\theta)$, \dots , $D_k(\theta)$ and $A_{cl}(\theta, \dot{\theta})$, $B_{\delta_{cl}}(\theta)$, \dots , $D_{11_{cl}}$ do not depend affinely on the scheduled parameters θ . Inequality (7.76) becomes

$$\begin{pmatrix} A_{cl}^T(\theta, \dot{\theta})P(\theta) + P(\theta)A_{cl}(\theta, \dot{\theta}) + \dot{P}(\theta) & P(\theta)B_{\delta_{cl}}(\theta) & P(\theta)B_{1_{cl}}(\theta) & C_{\delta_{cl}}^T(\theta)L & C_{1_{cl}}^T(\theta) \\ B_{\delta_{cl}}^T(\theta)P(\theta) & -\gamma L & 0 & D_{\delta_{cl}}^T(\theta)L & D_{1_{cl}}^T(\theta) \\ B_{1_{cl}}^T(\theta)P(\theta) & 0 & -\gamma I & D_{\delta_{cl}}^T(\theta)L & D_{1_{cl}}^T(\theta) \\ LC_{\delta_{cl}}(\theta) & LD_{\delta_{cl}}(\theta) & LD_{1_{cl}}(\theta) & -\gamma L & 0 \\ C_{1_{cl}}(\theta) & D_{1_{cl}}(\theta) & D_{11_{cl}}(\theta) & 0 & -\gamma I \end{pmatrix} < 0 \quad (7.78)$$

where

$$\begin{aligned} A_{cl}(\theta, \dot{\theta}) &= \begin{bmatrix} A(\theta) + B_2 D_k(\theta) C_2 & B_2 C_k(\theta) \\ B_k(\theta) C_2 & A_k(\theta, \dot{\theta}) \end{bmatrix} \\ B_{\delta_{cl}}(\theta) &= \begin{bmatrix} B_\delta(\theta) + B_2 D_k(\theta) D_{2\delta} \\ B_k(\theta) D_{2\delta} \end{bmatrix}, \quad B_{1_{cl}}(\theta) = \begin{bmatrix} B_1(\theta) + B_2 D_k(\theta) D_{21} \\ B_k(\theta) D_{21} \end{bmatrix} \\ C_{\delta_{cl}}(\theta) &= [C_\delta(\theta) + D_{\delta 2} D_k(\theta) C_2 \quad D_{\delta 2} C_k(\theta)] \\ C_{1_{cl}}(\theta) &= [C_1(\theta) + D_{12} D_k(\theta) C_2 \quad D_{12} C_k(\theta)] \\ D_{\delta_{cl}}(\theta) &= D_{\delta \delta}(\theta) + D_{\delta 2} D_k(\theta) D_{2\delta}, \quad D_{\delta 1_{cl}}(\theta) = D_{\delta 1}(\theta) + D_{\delta 2} D_k(\theta) D_{21} \\ D_{1_{cl}}(\theta) &= D_{1\delta}(\theta) + D_{12} D_k(\theta) D_{2\delta}, \quad D_{11_{cl}}(\theta) = D_{11}(\theta) + D_{12} D_k(\theta) D_{21} \end{aligned} \quad (7.79)$$

Following section 6.2, inequality (7.78) leads to

$$\begin{pmatrix}
 \dot{X}(\theta) + X(\theta)A(\theta) + \hat{B}_k(\theta)C_2 + (\star) & \star & & & \\
 \hat{A}_k^T(\theta) + A(\theta) + B_2D_k(\theta)C_2 & -\dot{Y}(\theta) + A(\theta)Y(\theta) + B_2\hat{C}_k(\theta) + (\star) & & & \\
 B_\delta^T(\theta)X(\theta) + D_{2\delta}^T\hat{B}_k^T(\theta) & B_\delta^T(\theta) + D_{2\delta}^TD_k^T(\theta)B_2^T & & & \\
 B_{1\delta}^T(\theta)X(\theta) + D_{21}^T\hat{B}_k^T(\theta) & B_{1\delta}^T(\theta) + D_{21}^TD_k^T(\theta)B_2^T & & & \\
 LC_\delta(\theta) + LD_{\delta 2}D_k(\theta)C_2 & LC_\delta(\theta)Y(\theta) + LD_{\delta 2}\hat{C}_k(\theta) & & & \\
 C_1(\theta) + D_{12}D_k(\theta)C_2 & C_1(\theta)Y(\theta) + D_{12}\hat{C}_k(\theta) & & & \\
 \star & \star & \star & \star & \\
 \star & \star & \star & \star & \\
 -\gamma L & 0 & \star & \star & \\
 0 & -\gamma I & \star & \star & \\
 LD_{\delta\delta}(\theta) + LD_{\delta 2}D_k(\theta)D_{2\delta} & LD_{\delta 1}(\theta) + LD_{\delta 2}D_k(\theta)D_{21} & -\gamma L & \star & \\
 D_{1\delta}(\theta) + D_{12}D_k(\theta)D_{2\delta} & D_{11}(\theta) + D_{12}D_k(\theta)D_{21} & 0 & -\gamma I &
 \end{pmatrix} < 0 \quad (7.80)$$

where the notation \star represents a symmetric matrix block. Substituting (7.55) and (6.26)–(6.33) in (7.80), we have

$$\begin{aligned}
 & \sum_{i=1}^r \sum_{k=1}^r \alpha_i^2 \beta_k \begin{pmatrix}
 \tilde{X}_k + \hat{X}_i \hat{A}_i + \tilde{B}_{k_i} C_2 + (\star) & \star & & & \\
 \tilde{A}_{k_i}^T + \hat{A}_i + B_2 \tilde{D}_{k_i} C_2 & -\tilde{Y}_k + \hat{A}_i \hat{Y}_i + B_2 \tilde{C}_{k_i} + (\star) & & & \\
 \hat{B}_{\delta_i}^T \hat{X}_i + D_{2\delta}^T \tilde{B}_{k_i}^T & \hat{B}_{\delta_i}^T + D_{2\delta}^T \tilde{D}_{k_i}^T B_2^T & & & \\
 \hat{B}_{1_i}^T \hat{X}_i + D_{21}^T \tilde{B}_{k_i}^T & \hat{B}_{1_i}^T + D_{21}^T \tilde{D}_{k_i}^T B_2^T & & & \\
 L\hat{C}_{\delta_i} + LD_{\delta 2} \tilde{D}_{k_i} C_2 & L\hat{C}_{\delta_i} \hat{Y}_i + LD_{\delta 2} \tilde{C}_{k_i} & & & \\
 \hat{C}_{1_i} + D_{12} \tilde{D}_{k_i} C_2 & \hat{C}_{1_i} \hat{Y}_i + D_{12} \tilde{C}_{k_i} & & & \\
 \star & \star & \star & \star & \\
 \star & \star & \star & \star & \\
 -\gamma L & 0 & \star & \star & \\
 0 & -\gamma I & \star & \star & \\
 L\hat{D}_{\delta\delta_i} + LD_{\delta 2} \tilde{D}_{k_i} D_{2\delta} & L\hat{D}_{\delta 1_i} + LD_{\delta 2} \tilde{D}_{k_i} D_{21} & -\gamma L & \star & \\
 \hat{D}_{1\delta_i} + D_{12} \tilde{D}_{k_i} D_{2\delta} & \hat{D}_{11_i} + D_{12} \tilde{D}_{k_i} D_{21} & 0 & -\gamma I &
 \end{pmatrix} \\
 & + 2 \sum_{i=1}^{r-1} \sum_{j=i+1}^r \sum_{k=1}^r \alpha_i \alpha_j \beta_k \begin{pmatrix}
 \tilde{X}_k + \frac{1}{2} \left(\hat{X}_j \hat{A}_i + \tilde{B}_{k_j} C_2 + \hat{X}_i \hat{A}_j + \tilde{B}_{k_i} C_2 + (\star) \right) & & & & \\
 \frac{1}{2} \left(\tilde{A}_{k_j}^T + \hat{A}_i + B_2 \tilde{D}_{k_j} C_2 + \tilde{A}_{k_i}^T + \hat{A}_j + B_2 \tilde{D}_{k_i} C_2 \right) & & & & \\
 \frac{1}{2} \left(\hat{B}_{\delta_i}^T \hat{X}_j + D_{2\delta}^T \tilde{B}_{k_j}^T + \hat{B}_{\delta_j}^T \hat{X}_i + D_{2\delta}^T \tilde{B}_{k_i}^T \right) & & & & \\
 \frac{1}{2} \left(\hat{B}_{1_i}^T \hat{X}_j + D_{21}^T \tilde{B}_{k_j}^T + \hat{B}_{1_j}^T \hat{X}_i + D_{21}^T \tilde{B}_{k_i}^T \right) & & & & \\
 \frac{L}{2} \left(\hat{C}_{\delta_i} + D_{\delta 2} \tilde{D}_{k_j} C_2 + \hat{C}_{\delta_j} + D_{\delta 2} \tilde{D}_{k_i} C_2 \right) & & & & \\
 \frac{1}{2} \left(\hat{C}_{1_i} + D_{12} \tilde{D}_{k_j} C_2 + \hat{C}_{1_j} + D_{12} \tilde{D}_{k_i} C_2 \right) & & & &
 \end{pmatrix}
 \end{aligned}$$

$$\begin{array}{cc}
 \star & \star \\
 -\tilde{Y}_k + \frac{1}{2} \left(\hat{A}_i \hat{Y}_j + B_2 \tilde{C}_{k_j} + \hat{A}_j \hat{Y}_i + B_2 \tilde{C}_{k_i} + (\star) \right) & \star \\
 \frac{1}{2} \left(\hat{B}_{\delta_i}^T + D_{2\delta}^T \tilde{D}_{k_j}^T B_2^T + \hat{B}_{\delta_j}^T + D_{2\delta}^T \tilde{D}_{k_i}^T B_2^T \right) & -\gamma L \\
 \frac{1}{2} \left(\hat{B}_{1_i}^T + D_{21}^T \tilde{D}_{k_j}^T B_2^T + \hat{B}_{1_j}^T + D_{21}^T \tilde{D}_{k_i}^T B_2^T \right) & 0 \\
 \frac{L}{2} \left(\hat{C}_{\delta_i} \hat{Y}_j + D_{\delta 2} \tilde{C}_{k_j} + \hat{C}_{\delta_j} \hat{Y}_i + D_{\delta 2} \tilde{C}_{k_i} \right) & \frac{L}{2} \left(\hat{D}_{\delta \delta_i} + D_{\delta 2} \tilde{D}_{k_j} D_{2\delta} + \hat{D}_{\delta \delta_j} + D_{\delta 2} \tilde{D}_{k_i} D_{2\delta} \right) \\
 \frac{L}{2} \left(\hat{C}_{1_i} \hat{Y}_j + D_{12} \tilde{C}_{k_j} + \hat{C}_{1_j} \hat{Y}_i + D_{12} \tilde{C}_{k_i} \right) & \frac{L}{2} \left(\hat{D}_{1\delta_i} + D_{12} \tilde{D}_{k_j} D_{2\delta} + \hat{D}_{1\delta_j} + D_{12} \tilde{D}_{k_i} D_{2\delta} \right) \\
 \star & \star \quad \star \\
 \star & \star \quad \star \\
 0 & \star \quad \star \\
 -\gamma I & \star \quad \star \\
 \frac{L}{2} \left(\hat{D}_{\delta 1_i} + D_{\delta 2} \tilde{D}_{k_j} D_{21} + \hat{D}_{\delta 1_j} + D_{\delta 2} \tilde{D}_{k_i} D_{21} \right) & -\gamma L \quad \star \\
 \frac{L}{2} \left(\hat{D}_{11_i} + D_{12} \tilde{D}_{k_j} D_{21} + \hat{D}_{11_j} + D_{12} \tilde{D}_{k_i} D_{21} \right) & 0 \quad -\gamma I
 \end{array} < 0 \quad (7.81)$$

Inequality (7.81) can be also rewritten as

$$\begin{aligned}
 & \sum_{i=1}^r \sum_{k=1}^r \alpha_i^2 \beta_k \left(\Psi_{cl_{ii}} + \mathcal{Q}^T \hat{K}_i^T \mathcal{P} + \mathcal{P}^T \hat{K}_i \mathcal{Q} \right) + 2 \sum_{i=1}^{r-1} \sum_{j=i+1}^r \sum_{k=1}^r \alpha_i \alpha_j \beta_k \left(\frac{1}{2} \left(\Psi_{cl_{ij}} \right. \right. \\
 & \left. \left. + \mathcal{Q}^T \hat{K}_i^T \mathcal{P} + \mathcal{P}^T \hat{K}_i \mathcal{Q} + \Psi_{cl_{ji}} + \mathcal{Q}^T \hat{K}_j^T \mathcal{P} + \mathcal{P}^T \hat{K}_j \mathcal{Q} \right) \right) < 0 \quad (7.82)
 \end{aligned}$$

where

$$\Psi_{cl_{\clubsuit\spadesuit}} = \begin{pmatrix} \tilde{X}_k + \hat{X}_{\clubsuit} \hat{A}_{\spadesuit} + (\star) & \star & \star & \star & \star & \star \\ \hat{A}_{\clubsuit} & -\tilde{Y}_k + \hat{A}_{\clubsuit} \hat{Y}_{\spadesuit} + (\star) & \star & \star & \star & \star \\ \hat{B}_{\delta_{\clubsuit}}^T \hat{X}_{\spadesuit} & \hat{B}_{\delta_{\clubsuit}}^T & -\gamma L & 0 & \star & \star \\ \hat{B}_{1_{\clubsuit}}^T \hat{X}_{\spadesuit} & \hat{B}_{1_{\clubsuit}}^T & 0 & -\gamma I & \star & \star \\ L \hat{C}_{\delta_{\clubsuit}} & L \hat{C}_{\delta_{\clubsuit}} \hat{Y}_{\spadesuit} & L \hat{D}_{\delta \delta_{\clubsuit}} & L \hat{D}_{\delta 1_{\clubsuit}} & -\gamma L & \star \\ \hat{C}_{1_{\clubsuit}} & \hat{C}_{1_{\clubsuit}} \hat{Y}_{\spadesuit} & \hat{D}_{1\delta_{\clubsuit}} & \hat{D}_{11_{\clubsuit}} & 0 & -\gamma I \end{pmatrix} \quad (7.83)$$

$$\mathcal{Q} = [\mathcal{C}, \quad \mathcal{D}_{\delta 21}, \quad \mathcal{D}_{121}, \quad 0_{(p+q_2) \times (q_1+s)}], \quad \tilde{\mathcal{B}} = \begin{bmatrix} I_p & 0 \\ 0 & B_2 \end{bmatrix} \quad (7.84)$$

$$\mathcal{P} = [\tilde{\mathcal{B}}^T, \quad 0_{(p+m_2) \times (m_1+s)}, \quad \mathcal{D}_{\delta 12}^T, \quad \mathcal{D}_{112}^T], \quad \hat{K}_i = \begin{pmatrix} \tilde{A}_{k_i} & \tilde{B}_{k_i} \\ \tilde{C}_{k_i} & \tilde{D}_{k_i} \end{pmatrix} \quad (7.85)$$

with the subscript $\clubsuit\spadesuit$ denote $(ii, ij, \text{ or } ji)$. By Lemma 3.3.1 with a known scaling matrix L and matrix vertices (\hat{X}_i, \hat{Y}_i) , $i = 1, 2, \dots, r$, the system matrix vertices \hat{K}_i can be determined from (7.82), that is an LMI in \hat{K}_i , at all vertices for which $(\hat{K}_1, \hat{K}_2, \dots, \hat{K}_r)$ have to satisfy all of $r^2(r+1)/2$ LMIs. Furthermore, knowing $\tilde{A}_{k_i}, \dots, \tilde{D}_{k_i}$, the controller system matrices $A_k(\theta, \dot{\theta}), \dots, D_k(\theta)$ can be computed on-line in real-time using (6.20)–(6.22) with an instantaneous measurement value of θ and $\dot{\theta}$.

However, the parameter derivatives either are not available or are difficult to estimate during system operation [8]. We have (6.41) as a result to avoid using the measured value of $\dot{\theta}$. By Lemma 2.2.14 with a known scaling matrix L , the LMIs of (7.82) are solvable for \hat{K}_i if and only if there exist a pair of positive definite symmetric matrices $(X(\theta), Y(\theta))$ that satisfy the following LMIs:

$$\sum_{i=1}^r \sum_{k=1}^r \alpha_i^2 \beta_k \left(\begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \hat{A}_i^T X_i + X_i \hat{A}_i + \tilde{X}_k & X_i \hat{B}_{\delta_i} & X_i \hat{B}_{1_i} & \hat{C}_{\delta_i}^T \\ \hat{B}_{\delta_i}^T X_i & -\gamma L & 0 & \hat{D}_{\delta\delta_i}^T \\ \hat{B}_{1_i}^T X_i & 0 & -\gamma I & \hat{D}_{\delta 1_i}^T \\ \hat{C}_{\delta_i} & \hat{D}_{\delta\delta_i} & \hat{D}_{\delta 1_i} & -\gamma L^{-1} \\ \hat{C}_{1_i} & \hat{D}_{1\delta_i} & \hat{D}_{11_i} & 0 \end{pmatrix} \right. \\ \left. \begin{pmatrix} \hat{C}_{1_i}^T \\ \hat{D}_{1\delta_i}^T \\ \hat{D}_{11_i}^T \\ 0 \\ -\gamma I \end{pmatrix} \begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix} \right) + 2 \sum_{i=1}^{r-1} \sum_{j=i+1}^r \sum_{k=1}^r \alpha_i \alpha_j \beta_k \left(\begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \frac{1}{2} (\hat{A}_i^T \hat{X}_j + \hat{A}_j^T \hat{X}_i + (\star)) + \tilde{X}_k \\ \frac{1}{2} (\hat{B}_{\delta_i}^T \hat{X}_j + \hat{B}_{\delta_j}^T \hat{X}_i) \\ \frac{1}{2} (\hat{B}_{1_i}^T \hat{X}_j + \hat{B}_{1_j}^T \hat{X}_i) \\ \frac{1}{2} (\hat{C}_{\delta_i} + \hat{C}_{\delta_j}) \\ \frac{1}{2} (\hat{C}_{1_i} + \hat{C}_{1_j}) \end{pmatrix} \right. \\ \left. \begin{pmatrix} \star & \star & \star & \star \\ -\gamma L^{-1} & \star & \star & \star \\ 0 & -\gamma I & \star & \star \\ \frac{1}{2} (\hat{D}_{\delta\delta_i} + \hat{D}_{\delta\delta_j}) & \frac{1}{2} (\hat{D}_{\delta 1_i} + \hat{D}_{\delta 1_j}) & -\gamma L^{-1} & \star \\ \frac{1}{2} (\hat{D}_{1\delta_i} + \hat{D}_{1\delta_j}) & \frac{1}{2} (\hat{D}_{11_i} + \hat{D}_{11_j}) & 0 & -\gamma I \end{pmatrix} \begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix} \right) < 0 \quad (7.86)$$

$$\sum_{i=1}^r \sum_{k=1}^r \alpha_i^2 \beta_k \left(\begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \hat{A}_i Y_i + Y_i \hat{A}_i^T - \tilde{Y}_k & Y_i \hat{C}_{\delta_i}^T & Y_i \hat{C}_{1_i}^T & \hat{B}_{\delta_i} \\ \hat{C}_{\delta_i} Y_i & -\gamma L^{-1} & 0 & \hat{D}_{\delta\delta_i} \\ \hat{C}_{1_i} Y_i & 0 & -\gamma I & \hat{D}_{1\delta_i} \\ \hat{B}_{\delta_i}^T & \hat{D}_{\delta\delta_i}^T & \hat{D}_{1\delta_i}^T & -\gamma L \\ \hat{B}_{1_i}^T & \hat{D}_{\delta 1_i}^T & \hat{D}_{11_i}^T & 0 \end{pmatrix} \right. \\ \left. \begin{pmatrix} \hat{B}_{1_i} \\ \hat{D}_{\delta 1_i} \\ \hat{D}_{11_i} \\ 0 \\ -\gamma I \end{pmatrix} \begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix} \right) + 2 \sum_{i=1}^{r-1} \sum_{j=i+1}^r \sum_{k=1}^r \alpha_i \alpha_j \beta_k \left(\begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \frac{1}{2} (\hat{A}_i \hat{Y}_j + \hat{A}_j \hat{Y}_i + (\star)) - \tilde{Y}_k \\ \frac{1}{2} (\hat{C}_{\delta_i} \hat{Y}_j + \hat{C}_{\delta_j} \hat{Y}_i) \\ \frac{1}{2} (\hat{C}_{1_i} \hat{Y}_j + \hat{C}_{1_j} \hat{Y}_i) \\ \frac{1}{2} (\hat{B}_{\delta_i}^T + \hat{B}_{\delta_j}^T) \\ \frac{1}{2} (\hat{B}_{1_i}^T + \hat{B}_{1_j}^T) \end{pmatrix} \right. \\ \left. \begin{pmatrix} \star & \star & \star & \star \\ -\gamma L^{-1} & \star & \star & \star \\ 0 & -\gamma I & \star & \star \\ \frac{1}{2} (\hat{D}_{\delta\delta_i}^T + \hat{D}_{\delta\delta_j}^T) & \frac{1}{2} (\hat{D}_{1\delta_i}^T + \hat{D}_{1\delta_j}^T) & -\gamma L & \star \\ \frac{1}{2} (\hat{D}_{\delta 1_i}^T + \hat{D}_{\delta 1_j}^T) & \frac{1}{2} (\hat{D}_{11_i}^T + \hat{D}_{11_j}^T) & 0 & -\gamma I \end{pmatrix} \begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix} \right) < 0 \quad (7.87)$$

$$\sum_{i=1}^r \alpha_i \begin{pmatrix} \hat{X}_i & I \\ I & \hat{Y}_i \end{pmatrix} > 0 \quad (7.88)$$

where N_X and N_Y denote bases of the null spaces of $[C_2, D_{2\delta}, D_{21}]$ and $[B_2^T, D_{\delta 2}^T, D_{12}^T]$, respectively. Note that (7.88) ensures $X(\theta), Y(\theta) > 0$ and $X(\theta) - Y(\theta)^{-1} \geq 0$. By

Lemma 3.3.1, (7.86)–(7.88) are sufficiently evaluated at all vertices. Also note that (7.86)–(7.88) will become (7.70)–(7.72) when both X and Y are constant. Moreover, the quadratic H_∞ performance γ is determined for both the $(X(\theta), Y)$ and $(X, Y(\theta))$ cases and the case that gives lowest γ is selected.

Theorem 7.3.3. *Given a scaling matrix L , there exists an LPV controller $K(\theta)$ guaranteeing the closed-loop system, (7.54) and (7.56), quadratic H_∞ performance γ along all possible parameter trajectories, $\forall(\theta, \dot{\theta}) \in \Theta \times \Phi$, if and only if the following LMI conditions hold for some positive definite symmetric matrices $(X(\theta), Y(\theta))$, which further satisfy $\text{rank}(X(\theta) - Y^{-1}(\theta)) \leq p$:*

$$\begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \hat{A}_i^T X_i + X_i \hat{A}_i + \tilde{X}_k & X_i \hat{B}_{\delta_i} & X_i \hat{B}_{1_i} & \hat{C}_{\delta_i}^T & \hat{C}_{1_i}^T \\ \hat{B}_{\delta_i}^T X_i & -\gamma L & 0 & \hat{D}_{\delta\delta_i}^T & \hat{D}_{1\delta_i}^T \\ \hat{B}_{1_i}^T X_i & 0 & -\gamma I & \hat{D}_{\delta 1_i}^T & \hat{D}_{11_i}^T \\ \hat{C}_{\delta_i} & \hat{D}_{\delta\delta_i} & \hat{D}_{\delta 1_i} & -\gamma L^{-1} & 0 \\ \hat{C}_{1_i} & \hat{D}_{1\delta_i} & \hat{D}_{11_i} & 0 & -\gamma I \end{pmatrix} \begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix} < 0 \quad (7.89)$$

$$\begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \hat{A}_i Y_i + Y_i \hat{A}_i^T - \tilde{Y}_k & Y_i \hat{C}_{\delta_i}^T & Y_i \hat{C}_{1_i}^T & \hat{B}_{\delta_i} & \hat{B}_{1_i} \\ \hat{C}_{\delta_i} Y_i & -\gamma L^{-1} & 0 & \hat{D}_{\delta\delta_i} & \hat{D}_{\delta 1_i} \\ \hat{C}_{1_i} Y_i & 0 & -\gamma I & \hat{D}_{1\delta_i} & \hat{D}_{11_i} \\ \hat{B}_{\delta_i}^T & \hat{D}_{\delta\delta_i}^T & \hat{D}_{\delta 1_i}^T & -\gamma L & 0 \\ \hat{B}_{1_i}^T & \hat{D}_{1\delta_i}^T & \hat{D}_{11_i}^T & 0 & -\gamma I \end{pmatrix} \begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix} < 0 \quad (7.90)$$

$$\begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \left(\hat{A}_i^T \hat{X}_j + \hat{A}_j^T \hat{X}_i + (\star) \right) + 2\tilde{X}_k & \star \\ \hat{B}_{\delta_i}^T \hat{X}_j + \hat{B}_{\delta_j}^T \hat{X}_i & -2\gamma L \\ \hat{B}_{1_i}^T \hat{X}_j + \hat{B}_{1_j}^T \hat{X}_i & 0 \\ \hat{C}_{\delta_i} + \hat{C}_{\delta_j} & \hat{D}_{\delta\delta_i} + \hat{D}_{\delta\delta_j} \\ \hat{C}_{1_i} + \hat{C}_{1_j} & \hat{D}_{1\delta_i} + \hat{D}_{1\delta_j} \\ \star & \star & \star \\ \star & \star & \star \\ -2\gamma I & \star & \star \\ \hat{D}_{\delta 1_i} + \hat{D}_{\delta 1_j} & -2\gamma L^{-1} & \star \\ \hat{D}_{11_i} + \hat{D}_{11_j} & 0 & -2\gamma I \end{pmatrix} \begin{pmatrix} N_X & 0 \\ 0 & I \end{pmatrix} < 0 \quad (7.91)$$

$$\begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix}^T \begin{pmatrix} \left(\hat{A}_i \hat{Y}_j + \hat{A}_j \hat{Y}_i + (\star) \right) - 2\tilde{Y}_k & \star \\ \hat{C}_{\delta_i} \hat{Y}_j + \hat{C}_{\delta_j} \hat{Y}_i & -2\gamma L^{-1} \\ \hat{C}_{1_i} \hat{Y}_j + \hat{C}_{1_j} \hat{Y}_i & 0 \\ \hat{B}_{\delta_i}^T + \hat{B}_{\delta_j}^T & \hat{D}_{\delta\delta_i}^T + \hat{D}_{\delta\delta_j}^T \\ \hat{B}_{1_i}^T + \hat{B}_{1_j}^T & \hat{D}_{\delta 1_i}^T + \hat{D}_{\delta 1_j}^T \\ \star & \star & \star \\ \star & \star & \star \\ -2\gamma I & \star & \star \\ \hat{D}_{1\delta_i}^T + \hat{D}_{1\delta_j}^T & -2\gamma L & \star \\ \hat{D}_{11_i}^T + \hat{D}_{11_j}^T & 0 & -2\gamma I \end{pmatrix} \begin{pmatrix} N_Y & 0 \\ 0 & I \end{pmatrix} < 0 \quad (7.92)$$

$$\begin{pmatrix} \hat{X}_i & I \\ I & \hat{Y}_i \end{pmatrix} > 0 \quad (7.93)$$

for $i, k = 1, \dots, r$ and $1 \leq i < j \leq r$.

Algorithm 7.3.4. *Computation of $X_i, Y_i, i = 1, \dots, b$ and L*

Step 1: Setting $L = I$, solve (7.89)–(7.93) for X_i, Y_i by minimizing γ .

Step 2: Knowing X_i, Y_i , and L , solve (7.82) for \hat{K}_i .

Step 3: With \hat{K}_i fixed, solve (7.81) for L by minimizing γ (heuristic).

Step 4: With L fixed, solve (7.89)–(7.93) for new X_i, Y_i by minimizing γ .

Step 5: Iterate over Steps 2 to 4 until γ is converged to some minimum value.

Note that, by Lemma 3.3.1, (7.81) and (7.82) are sufficiently solved at all vertices.

7.4 Numerical Example

We demonstrate the robustness improvement through the numerical example of [61], described in Chapter 4. Consider the transfer function of the Jacobian-based LPV model taken from [30]

$$P(s, \theta) = \frac{1}{(s+1)(s+2\theta)}, \quad \theta \in [0, 10] \quad (7.94)$$

Having added a time-varying real parametric uncertainty δ , (7.94) becomes

$$P_\delta(s, \theta, \delta) = \frac{1}{(s+1)(s+2\theta+\delta)}, \quad (\theta, \delta) \in [0, 10] \times [-1, 1] \quad (7.95)$$

To demonstrate the impacts of the proposed scheme in robustness improvement of uncertain affine LPV systems, we compare our approach with the existing uncertain LPV/LFT approach [9, 24]. Figure 7.3 shows both uncertain affine LPV and LPV/LFT closed-loop systems.

First, consider uncertain LPV/LFT approach [9, 24], we have $\tilde{\theta} = (\theta-5)/5 \in [-1, 1]$, using (3.36), and $\tilde{\delta} = \delta \in [-1, 1]$. Having augmented $P(s)$ with two weighting function presented in [61], we get $P_a(s)$ with state-space realization

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} &= \begin{bmatrix} -1 & 0 & 0 & 0 \\ 1 & -10 & 0 & 0 \\ 0 & -0.5 & -0.002 & 0 \\ 0 & 0 & 0 & -1000 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} L^{-\frac{1}{2}} \begin{bmatrix} \tilde{w}_\theta \\ w_\theta \\ w_\delta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0.5 \\ 0 \end{bmatrix} w + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 4 & 0 \end{bmatrix} \begin{bmatrix} u \\ \tilde{u} \end{bmatrix} \\ \begin{bmatrix} \tilde{z}_\theta \\ z_\theta \\ z_\delta \end{bmatrix} &= L^{\frac{1}{2}} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -10 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + L^{\frac{1}{2}} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} L^{-\frac{1}{2}} \begin{bmatrix} \tilde{w}_\theta \\ w_\theta \\ w_\delta \end{bmatrix} + L^{\frac{1}{2}} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} w + L^{\frac{1}{2}} \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ \tilde{u} \end{bmatrix} \end{aligned}$$

$$\begin{aligned}
\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} L^{-\frac{1}{2}} \begin{bmatrix} \tilde{w}_\theta \\ w_\theta \\ w_\delta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} w + \begin{bmatrix} 0 & 0 \\ 0.02 & 0 \end{bmatrix} \begin{bmatrix} u \\ \tilde{u} \end{bmatrix} \\
\begin{bmatrix} y \\ \tilde{y} \end{bmatrix} &= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} L^{-\frac{1}{2}} \begin{bmatrix} \tilde{w}_\theta \\ w_\theta \\ w_\delta \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} w + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ \tilde{u} \end{bmatrix} \\
\begin{bmatrix} \tilde{w}_\theta \\ w_\theta \\ w_\delta \end{bmatrix} &= \begin{bmatrix} \tilde{\theta} & 0 & 0 \\ 0 & \tilde{\theta} & 0 \\ 0 & 0 & \tilde{\delta} \end{bmatrix} \begin{bmatrix} \tilde{z}_\theta \\ z_\theta \\ z_\delta \end{bmatrix}
\end{aligned} \tag{7.96}$$

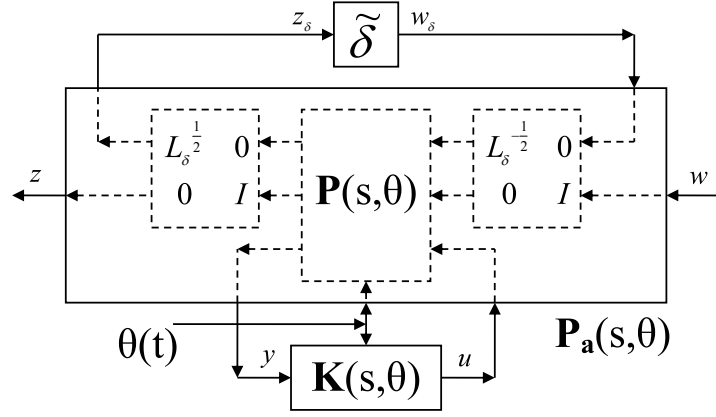
where

$$L = \begin{bmatrix} L_{1_\theta} & L_{2_\theta} & 0 \\ L_{2_\theta}^T & L_{3_\theta} & 0 \\ 0 & 0 & L_\delta \end{bmatrix}$$

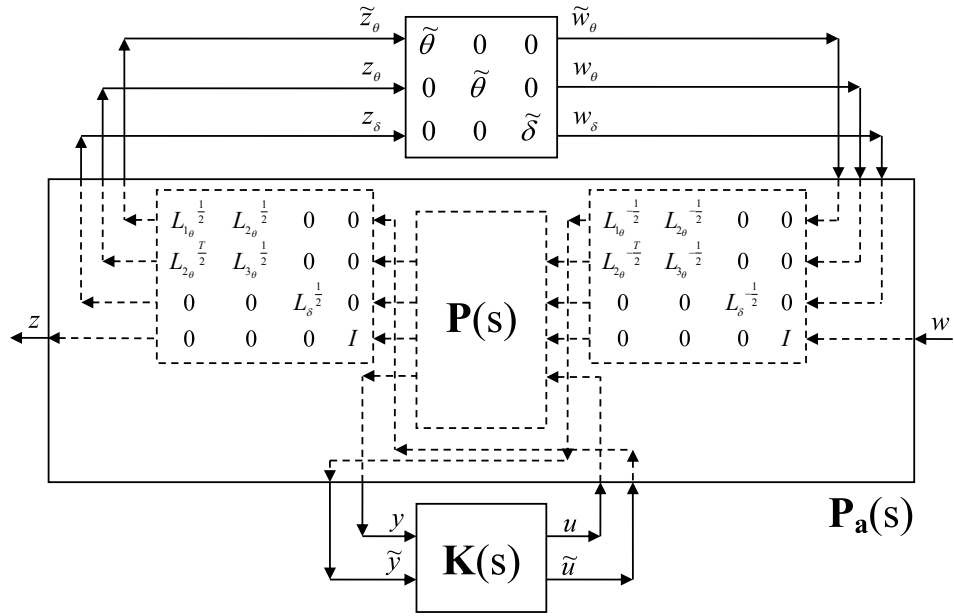
An LPV controller can be computed using Theorem 5.1 [9]. Unfortunately, the convexity of Theorem 5.1 is destroyed by the extra uncertainty block $\tilde{\delta}$, and D-K iterations are needed to compute adequate LPV controllers. Solving LMIs in Theorem 5.1 by using a MATLAB Robust Control Toolbox function [12], `mincx`, we got $\gamma = 0.5524$, a scaling matrix, and an LPV controller as shown below

$$\begin{aligned}
L &= \begin{bmatrix} 0.0007654 & -0.00099889 & 0 \\ -0.00099889 & 0.0028127 & 0 \\ 0 & 0 & 0.020756 \end{bmatrix} \\
\begin{bmatrix} \dot{x}_{k_1} \\ \dot{x}_{k_2} \\ \dot{x}_{k_3} \\ \dot{x}_{k_4} \end{bmatrix} &= \begin{bmatrix} -3.9848 & -63.001 & 307.21 & 228.16 \\ -1079.3 & -53336 & 1.8482 \times 10^5 & -5520.4 \\ 0.0062935 & 0.31046 & -1.0783 & 0.032158 \\ -11.939 & -252 & 1228.8 & -87.362 \end{bmatrix} \begin{bmatrix} x_{k_1} \\ x_{k_2} \\ x_{k_3} \\ x_{k_4} \end{bmatrix} + \begin{bmatrix} -15.149 \\ -29207 \\ 0.66995 \\ -60.592 \end{bmatrix} y \\
&\quad + \begin{bmatrix} 0.24715 \\ 467.41 \\ -0.0027188 \\ 0.98854 \end{bmatrix} \tilde{y} \\
u &= \begin{bmatrix} -2.9847 & -62.996 & 307.2 & 228.16 \end{bmatrix} \begin{bmatrix} x_{k_1} \\ x_{k_2} \\ x_{k_3} \\ x_{k_4} \end{bmatrix} - 15.146y + 0.24711\tilde{y} \\
\tilde{u} &= \begin{bmatrix} -0.21422 & -23.467 & 36.325 & -1.0938 \end{bmatrix} \begin{bmatrix} x_{k_1} \\ x_{k_2} \\ x_{k_3} \\ x_{k_4} \end{bmatrix} - 5.6313y + 0.093067\tilde{y} \\
\tilde{y} &= \tilde{\delta}\tilde{u}
\end{aligned} \tag{7.97}$$

Next, consider our approach, having augmented $P(s, \theta)$ with two weighting function



(a) Uncertain affine LPV closed-loop structure



(b) Uncertain LPV/LFT closed-loop structure

Figure 7.3: Structure comparisons of uncertain affine LPV and LPV/LFT [9] closed-loop systems.

presented in [61], we get $P_a(s, \theta)$ with state-space realization

$$\begin{aligned}
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} &= \begin{bmatrix} -1 & 0 & 0 & 0 \\ 1 & -2\theta & 0 & 0 \\ 0 & -0.5 & -0.002 & 0 \\ 0 & 0 & 0 & -1000 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} L_\delta^{-\frac{1}{2}} w_\delta + \begin{bmatrix} 0 \\ 0 \\ 0.5 \\ 0 \end{bmatrix} w + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 4 \end{bmatrix} u \\
z_\delta &= L_\delta^{\frac{1}{2}} \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + L_\delta^{\frac{1}{2}} \begin{bmatrix} 0 \end{bmatrix} L_\delta^{-\frac{1}{2}} w_\delta + L_\delta^{\frac{1}{2}} \begin{bmatrix} 0 \end{bmatrix} w + L_\delta^{\frac{1}{2}} \begin{bmatrix} 0 \end{bmatrix} u \\
\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} L_\delta^{-\frac{1}{2}} w_\delta + \begin{bmatrix} 0 \\ 0 \end{bmatrix} w + \begin{bmatrix} 0 \\ 0.02 \end{bmatrix} u \\
y &= \begin{bmatrix} 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} L_\delta^{-\frac{1}{2}} w_\delta + \begin{bmatrix} 1 \end{bmatrix} w + \begin{bmatrix} 0 \end{bmatrix} u \\
w_\delta &= \tilde{\delta} z_\delta
\end{aligned} \tag{7.98}$$

Solving Algorithm 7.3.2 with the same scaling matrix of δ , i.e. $L_\delta = 0.020756$, and the LMIs are solved using a MATLAB Robust Control Toolbox function [12], `mincx`, we got $\gamma = 0.2435$ and an LPV controller as shown below

$$\begin{aligned}
\dot{x}_k &= (\alpha_1 A_{k_1} + \alpha_2 A_{k_2}) x_k + (\alpha_1 B_{k_1} + \alpha_2 B_{k_2}) y \\
u &= (\alpha_1 C_{k_1} + \alpha_2 C_{k_2}) x_k + (\alpha_1 D_{k_1} + \alpha_2 D_{k_2}) y
\end{aligned} \tag{7.99}$$

where $\alpha_1 = (10 - \theta)/10$, $\alpha_2 = \theta/10$ and

$$\begin{aligned}
A_{k_1} &= \begin{pmatrix} -2.3776 & -11.695 & 201.38 & 235.44 \\ -821.01 & -16266 & 2.1605 \times 10^5 & -5851.3 \\ 0.018827 & 0.37233 & -4.9492 & 0.13401 \\ -5.5097 & -46.773 & 805.36 & -58.236 \end{pmatrix} \\
A_{k_2} &= \begin{pmatrix} -2.4698 & -12.45 & 249.52 & 234.91 \\ -1129.4 & -29334 & 2.9 \times 10^5 & -7638.8 \\ 0.02588 & 0.67111 & -6.6415 & 0.17489 \\ -5.8785 & -49.788 & 997.94 & -60.343 \end{pmatrix} \\
B_{k_1} &= \begin{pmatrix} 5.7865 \\ -5096.1 \\ 0.61651 \\ 23.146 \end{pmatrix}, \quad B_{k_2} = \begin{pmatrix} 0.87696 \\ -5852.3 \\ 0.634 \\ 3.5067 \end{pmatrix} \\
C_{k_1} &= [-1.3774 \quad -11.691 \quad 201.32 \quad 235.44] \\
C_{k_2} &= [-1.4695 \quad -12.442 \quad 249.44 \quad 234.92] \\
D_{k_1} &= 5.7879, \quad D_{k_2} = 0.87854
\end{aligned}$$

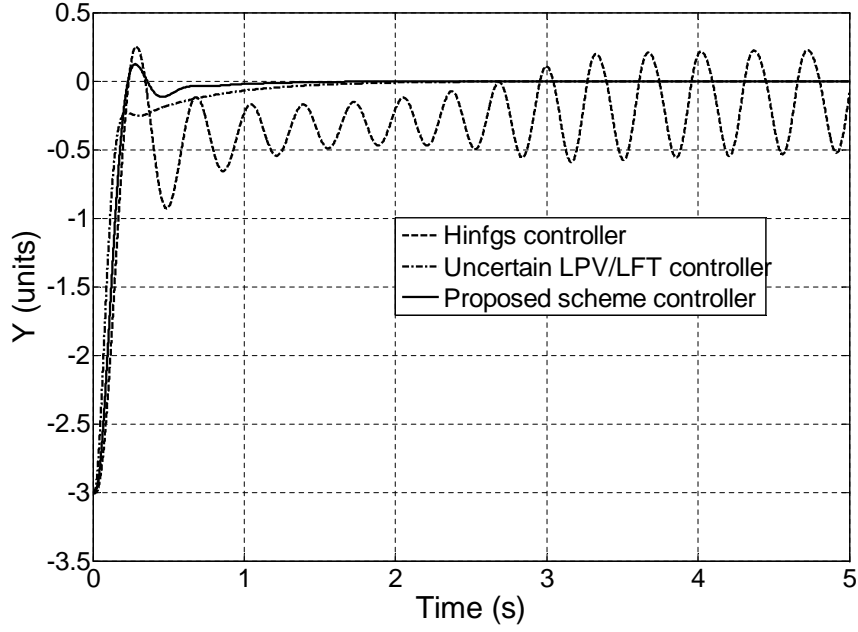


Figure 7.4: Nonlinear step response from -3 to 0 of the LPV controller with the original nonlinear plant.

Without including the uncertainty δ and using the MATLAB Robust Control Toolbox function [12], `hinfsgs`, an LPV controller was obtained with $\gamma = 0.1211$ [30]. However, the closed-loop system of the LPV controller with the original nonlinear model is unstable [30, 61].

Including the uncertainty δ , shown in (7.96) and (7.98), using Theorem 5.1 [9] and Theorem 7.3.1, two LPV controllers were obtained with $\gamma = 0.5524$ and 0.2435 , respectively. As an effect of including the uncertainty δ , we get more conservatism on γ but the stability property of the closed-loop system is guaranteed with a larger stability margin. Hence, the closed-loop instability for the controller with the nonlinear plant disappears without degrading the transient performance as shown in Figure 7.4. Moreover, this numerical example shows that our approach is less conservative than the uncertain LPV/LFT approach.

7.5 Lateral Uncertain Affine LPV Model

Following section 6.4, the lateral nonlinearly parameter-dependent LPV model, shown in (7.6), can be converted into an affine LPV model using a minimum least-squares method [58]. Moreover, the maximum and minimum of the least-squares conversion errors can be determined for all possible parameter trajectories, i.e. $\forall(u, h) \in [464.1, 548.5] \times [7500, 12500]$, by subtracting the converted (affine) stability and control derivatives from the stability and control derivatives in (7.7)–(7.24). For example, consider $Y_{\delta r}$ shown in (7.6), define $Y_{\delta r_{\text{Affine}}}$ to be depended

affinely on u and h , in a similar manner to (6.50), as

$$Y_{\delta r_{\text{Affine}}} = Y_{\delta r_0} + uY_{\delta r_u} + hY_{\delta r_h} \quad (7.100)$$

Then, $Y_{\delta r_0}$, $Y_{\delta r_u}$, and $Y_{\delta r_h}$ can be determined using a minimum least-squares method [58] as presented in section 6.4. Figure 7.5–a and–b shows a nonlinear $Y_{\delta r}$ and an affine $Y_{\delta r}$ that are determined using (7.12) and (7.100), respectively. The error (mismatch uncertainty) between a nonlinear $Y_{\delta r}$ and an affine $Y_{\delta r}$ can be computed as

$$\delta_{Y_{\delta r}} = Y_{\delta r} - Y_{\delta r_{\text{Affine}}} \quad (7.101)$$

The calculations of $\delta_{Y_{\delta r}}$ and $\tilde{\delta}_{Y_{\delta r}}$ (normalize), that is computed using (7.28), are shown in Figure 7.6–a and–b, respectively. Therefore, an uncertain affine $Y_{\delta r}$ can be written as

$$Y_{\delta r}(u, h, \tilde{\delta}) = (Y_{\delta r_0} + T_{Y_{\delta r}}) + uY_{\delta r_u} + hY_{\delta r_h} + S_{Y_{\delta r}}\tilde{\delta}_{Y_{\delta r}} \quad (7.102)$$

where

$$T_{Y_{\delta r}} = \frac{\bar{\delta}_{Y_{\delta r}} + \underline{\delta}_{Y_{\delta r}}}{2} \quad (7.103)$$

$$S_{Y_{\delta r}} = \frac{\bar{\delta}_{Y_{\delta r}} - \underline{\delta}_{Y_{\delta r}}}{2} \quad (7.104)$$

Figure 7.6–a also shows a maximum $\bar{\delta}_{Y_{\delta r}}$ and minimum $\underline{\delta}_{Y_{\delta r}}$ errors of $Y_{\delta r}$ that are equal to 0.3696 and -0.1323, respectively. Having normalized all of the mismatch uncertainties $\delta_{Y_v}, \delta_{Y_p}, \dots, \delta_{N_{\delta r}}$ using (7.28), and included the normalized mismatch uncertainties in the system state-space model matrices of the converted affine LPV model, we can compute the lateral uncertain affine LPV model, in a similar manner to (7.102), as

$$\dot{x} = A(u, h, \tilde{\delta})x + B(u, h, \tilde{\delta})u \quad (7.105)$$

where $x = [v \ p \ r \ \phi \ \psi]^T$, $u = [\delta_a \ \delta_r]^T$, $(u, h) \in [464.1, 548.5] \times [7500, 12500]$, i.e. bounded flight envelope, and

$$A(u, h, \tilde{\delta}) = \begin{pmatrix} -0.037327 & 37.833 & 0.0020293 & 32.174 & 0 \\ -0.0053012 & -0.48258 & 0.37758 & 0 & 0 \\ 0.0041288 & -0.0029707 & -0.074076 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} + u \begin{pmatrix} -0.00023363 & -0.089521 & -0.99999 & -1.4653 \times 10^{-6} & 0 \\ -3.3179 \times 10^{-5} & -0.0030204 & 0.0023632 & 0 & 0 \\ 2.5841 \times 10^{-5} & -1.8593 \times 10^{-5} & -0.00046363 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} + h \begin{pmatrix} 3.7146 \times 10^{-6} & 0.00070669 & -2.0194 \times 10^{-7} & 6.0454 \times 10^{-9} & 0 \\ 5.2754 \times 10^{-7} & 4.8023 \times 10^{-5} & -3.7575 \times 10^{-5} & 0 & 0 \\ -4.1087 \times 10^{-7} & 2.9563 \times 10^{-7} & 7.3716 \times 10^{-6} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\begin{aligned}
& + \begin{pmatrix} 0.00079214(\tilde{\delta}_{Y_v}) & 0.20339(\tilde{\delta}_{Y_p}) & 4.3064 \times 10^{-5}(\tilde{\delta}_{Y_r}) & 0.001143(\tilde{\delta}_{Y_\phi}) & 0 \\ 0.0001125(\tilde{\delta}_{L_v}) & 0.010241(\tilde{\delta}_{L_p}) & 0.0080129(\tilde{\delta}_{L_r}) & 0 & 0 \\ 8.7619 \times 10^{-5}(\tilde{\delta}_{N_v}) & 6.3043 \times 10^{-5}(\tilde{\delta}_{N_p}) & 0.001572(\tilde{\delta}_{N_r}) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\
B(u, h, \tilde{\delta}) = & \begin{pmatrix} 0 & -12.567 \\ 18.862 & 6.0757 \\ 0.23235 & -0.70258 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} + u \begin{pmatrix} 0 & 0.073385 \\ -0.11015 & -0.03548 \\ -0.0013568 & 0.0041028 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \\
& + h \begin{pmatrix} 0 & -0.00058483 \\ 0.00087781 & 0.00028275 \\ 1.0813 \times 10^{-5} & -3.2697 \times 10^{-5} \\ 0 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0.25095(\tilde{\delta}_{Y_{\delta_r}}) \\ 0.37667(\tilde{\delta}_{L_{\delta_a}}) & 0.12133(\tilde{\delta}_{L_{\delta_r}}) \\ 0.0046399(\tilde{\delta}_{N_{\delta_a}}) & 0.01403(\tilde{\delta}_{N_{\delta_r}}) \\ 0 & 0 \\ 0 & 0 \end{pmatrix}
\end{aligned}$$

Note that $\tilde{\delta}_{Y_v}, \tilde{\delta}_{Y_p}, \dots, \tilde{\delta}_{N_{\delta_r}} \in [-1, 1]$ and the uncertainty $\tilde{\delta}$ in (7.105) can be separated from the system matrices in a similar manner to (7.54).

7.6 Robust Gain-Scheduled H_∞ Autopilot Design

The mixed-sensitivity criterion (5.42) is also employed in a similar manner to section 5.3. Figure 7.7 shows the weighted open-loop interconnection for synthesis where

$$\begin{aligned}
z_\delta &= [z_{\delta_{Y_v}}, z_{\delta_{Y_p}}, \dots, z_{\delta_{N_{\delta_r}}}]^T, \quad z = [z_v, z_\psi, z_{\delta_a}, z_{\delta_r}]^T \\
w_\delta &= [w_{\delta_{Y_v}}, w_{\delta_{Y_p}}, \dots, w_{\delta_{N_{\delta_r}}}]^T, \quad w = [v_{\text{ref}}, \psi_{\text{ref}}]^T \\
y &= [v_{\text{ref}} - v, p, r, \phi, \psi_{\text{ref}} - \psi]^T, \quad u = [\delta_a, \delta_r]^T \\
\tilde{\delta} &= \text{diag}(\tilde{\delta}_{Y_v}, \tilde{\delta}_{Y_p}, \dots, \tilde{\delta}_{N_{\delta_r}}) \\
W_1(s) &= \begin{pmatrix} \frac{0.6667s+0.221}{s+0.00221} & 0 \\ 0 & \frac{s+1.105}{s+1.105 \times 10^{-3}} \end{pmatrix} \\
W_2(s) &= \begin{pmatrix} \frac{0.6s+0.442}{0.001s+1.105} & 0 \\ 0 & \frac{0.6s+0.442}{0.001s+1.105} \end{pmatrix} \\
W_{\text{pre-filter}}(s) &= \begin{pmatrix} \frac{500}{s+500} & 0 \\ 0 & \frac{500}{s+500} \end{pmatrix} \tag{7.106}
\end{aligned}$$

Note that the values of weighting functions W_1 and W_2 are hand-tuned until the desired objectives of performance and robustness of the closed-loop system are achieved. After the lateral uncertain affine LPV model, shown in (7.105), is augmented with the weighting functions, shown in (7.106), a pair of positive definite symmetric matrices $(X(\theta), Y(\theta))$ can be determined in four cases, i.e. (X, Y) , $(X(\theta), Y)$, $(X, Y(\theta))$, and $(X(\theta), Y(\theta))$, using Theorem 7.3.3 with the same scaling

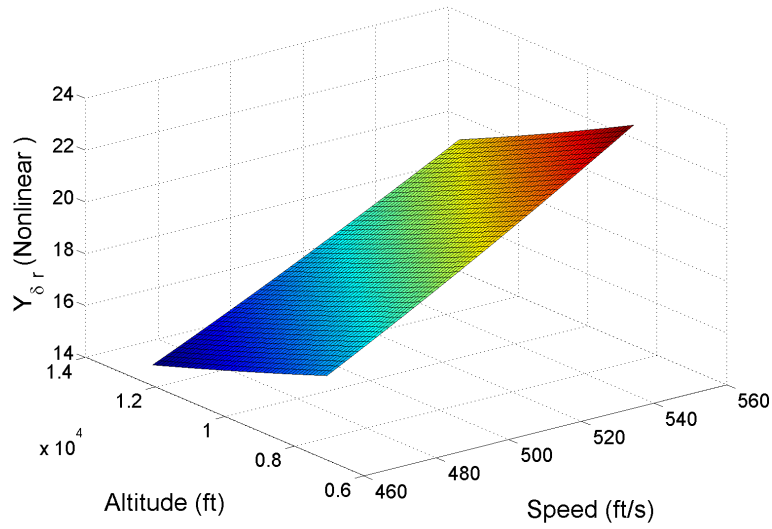
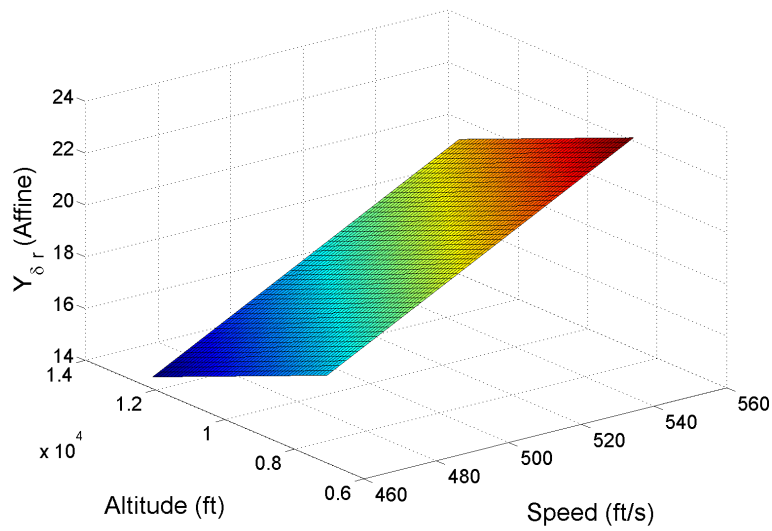
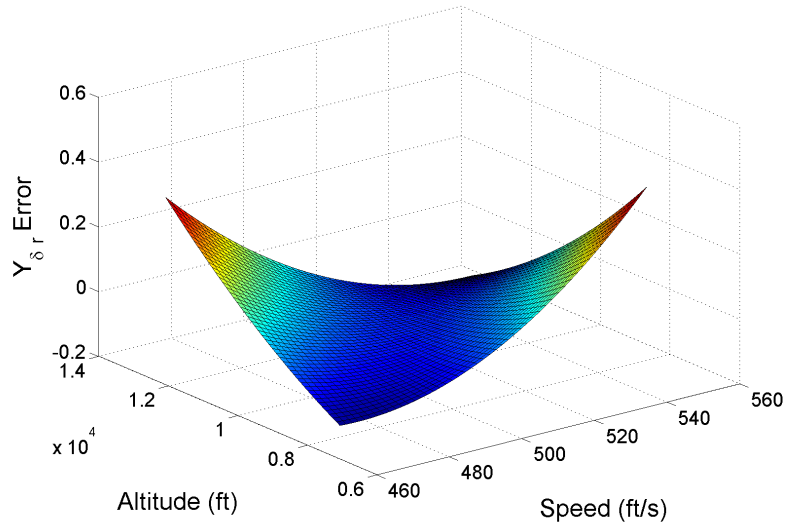
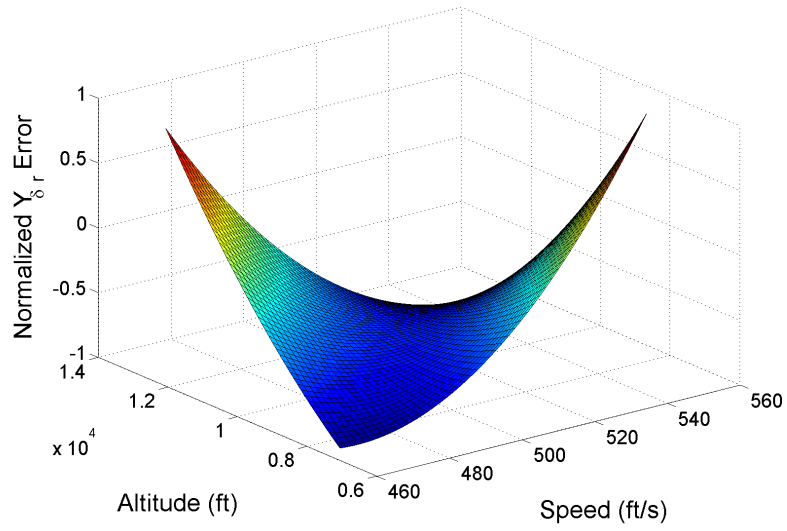

 (a) Nonlinear Y_{δ_r}

 (b) Affine Y_{δ_r}

 Figure 7.5: The variation of a nonlinear Y_{δ_r} and an affine Y_{δ_r} with speed and altitude



(a) $\delta_{Y_{\delta_r}}$



(b) $\tilde{\delta}_{Y_{\delta_r}}$ (Normalize)

Figure 7.6: The variation of $\delta_{Y_{\delta_r}}$ and $\tilde{\delta}_{Y_{\delta_r}}$ (normalize) with speed and altitude

Table 7.3: Performance γ comparison for different cases of $(X(u, h), Y(u, h))$

Flight condition $(u, h) \in [464.1, 548.5] \times [7500, 12500]$ $(\dot{u}, \dot{h}) \in [-1.26, 1.26] \times [-20, 50]$	$(X(u, h), Y(u, h))^{\dagger}$	$(X(u, h), Y)^{\dagger}$	$(X, Y(u, h))^{\dagger}$	$(X, Y)^{*}$
Performance γ	3.0399	3.8646	3.0706	4.0679

† Parameter-dependent Lyapunov functions, Theorem 7.3.3

$*$ Single quadratic Lyapunov function, Theorem 7.3.1

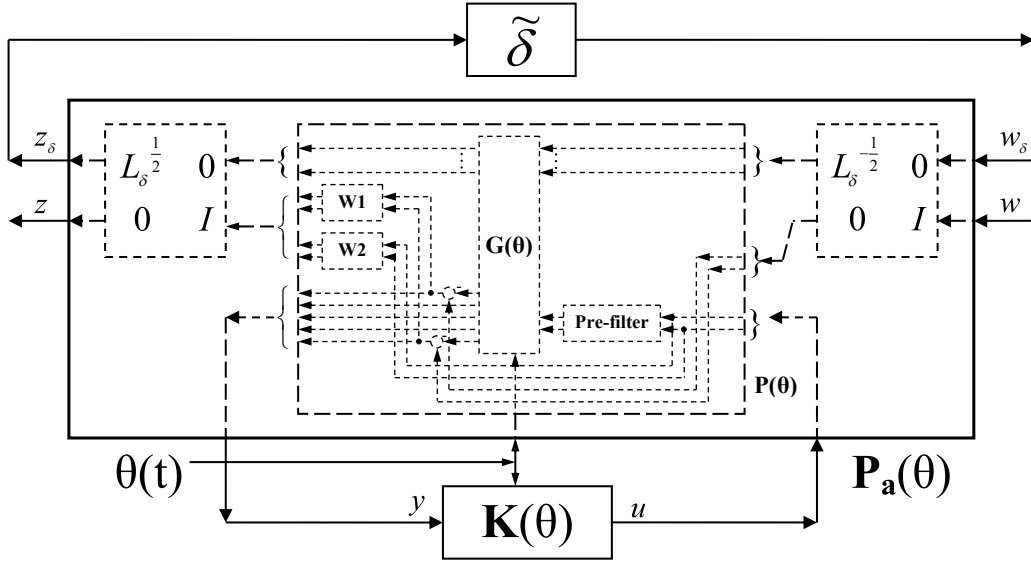


Figure 7.7: The weighted open-loop interconnection for the lateral uncertain affine LPV plant model.

matrix $L = \text{diag}(79.126, 0.43552, 91.049, 44.6, 70.304, 3.9041, 54.392, 1050, 1050.7, 1044.9, 3.5861, 1.1111, 3.0808, 1029.2, 816)$ for all cases, for which the performance measure (γ), shown in Table 7.3, can be compared. The LMIs are solved using the MATLAB Robust Control Toolbox function [12], `mincx`.

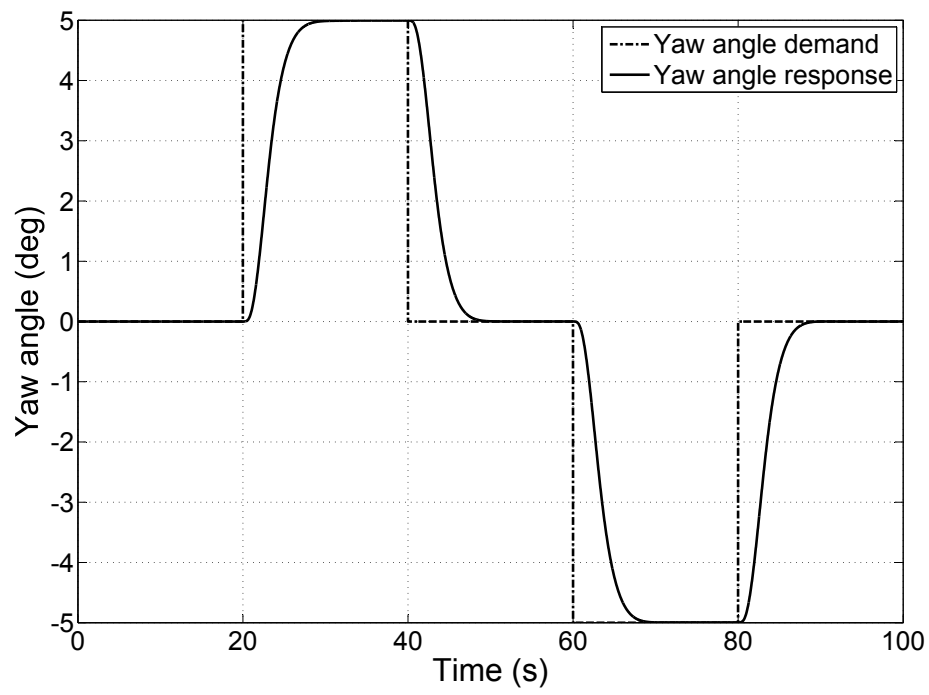
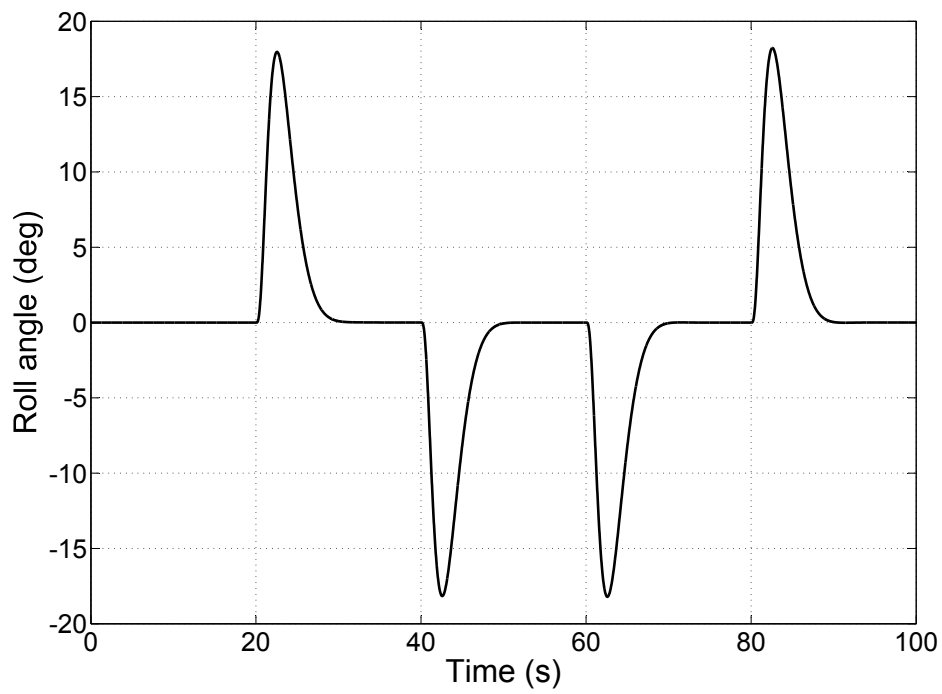
7.7 Nonlinear Simulation Results

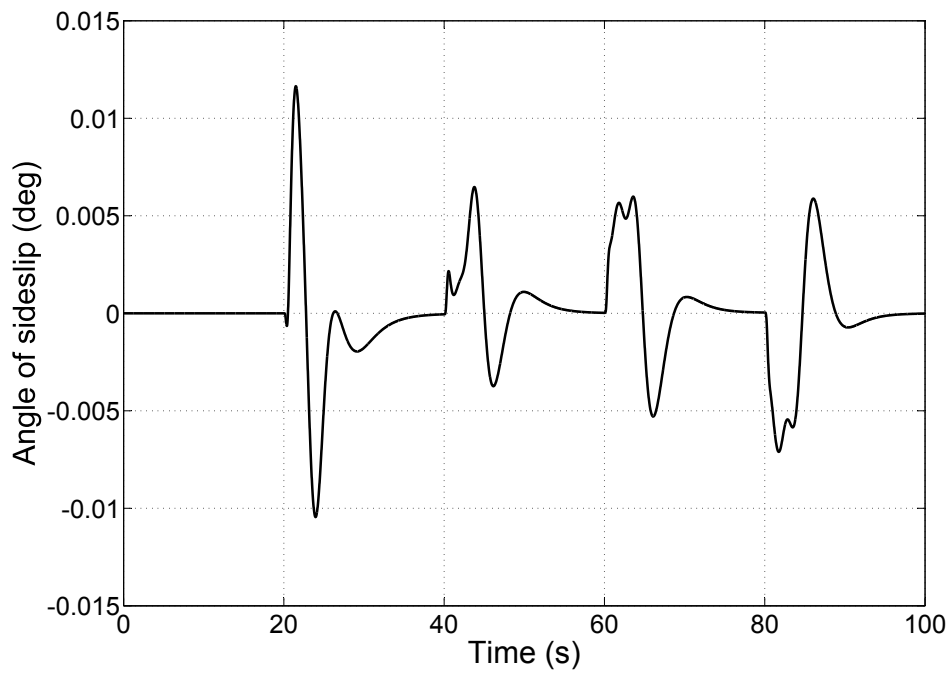
Since, for an LPV plant model with a large parameter variation region, it is often conservative to design a single LPV controller over the entire parameter space [64, 65, 66], the designed H_{∞} gain-scheduling autopilot is validated for a bounded flight envelope, i.e. $(u, h) \in [464.1, 548.5] \times [7500, 12500]$, $(\dot{u}, \dot{h}) \in [-1.26, 1.26] \times [-20, 50]$, with the Jindivik nonlinear model [41] in a MATLAB Simulink simulation. In Figure 7.8, the transient responses of the simulated vehicle for small demanded changes in yaw angle are shown for one particular point in the bounded flight envelope. Similar responses for other points in the flight envelope were obtained. Figure 7.9 shows

a rate one turn simulated flight, i.e. 3° per second turn, which completes a 360° turn in 2 minutes, for one particular point in the bounded flight envelope. Again, similar responses for other points in the flight envelope were obtained. These simulation results show that desired performance and robustness objectives are achieved over the defined flight envelope.

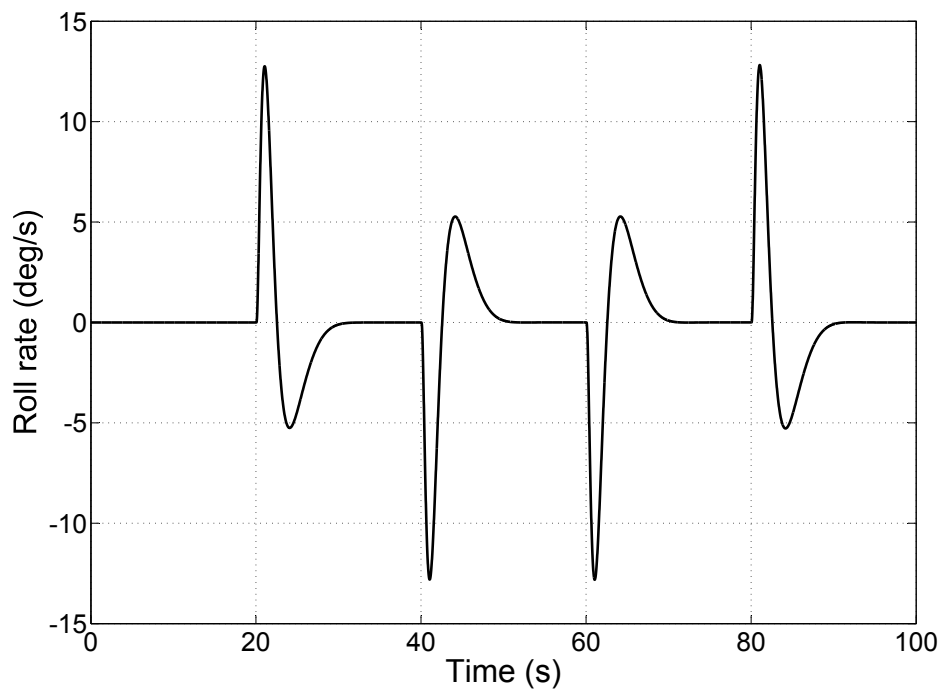
7.8 Conclusion

In this chapter, the mismatch uncertainties between a nonlinear model and an affine LPV model are handled by a new class of affine LPV systems which is called an uncertain affine LPV model. New sufficient conditions of gain-scheduled H_∞ performance analysis and synthesis, for this uncertain affine LPV model, using single quadratic or parameter dependent Lyapunov function are proposed for which the proposed scheme can guarantee robust stability and robust performance for all time-varying real parametric uncertainties which are $\|\tilde{\delta}\| < 1/\gamma$. The analysis and synthesis conditions are represented in the form of a finite number of LMIs. A numerical example was compared with uncertain LPV/LFT approach [9] results. In addition, the intermediate controller variables, i.e. $\hat{A}_k(\theta)$, $\hat{B}_k(\theta)$, $\hat{C}_k(\theta)$ and $D_k(\theta)$, can be constructed as an affine matrix-valued function in the polytopic coordinates of the scheduled parameter without the need for constraints on the D_{12} and D_{21} matrices. The proposed scheme was applied to synthesized a lateral LPV autopilot of the Jindivik UAV. The designed controller was tested with a full 6-DOF simulation of the vehicle and nonlinear simulation results show the effectiveness of the proposed method.

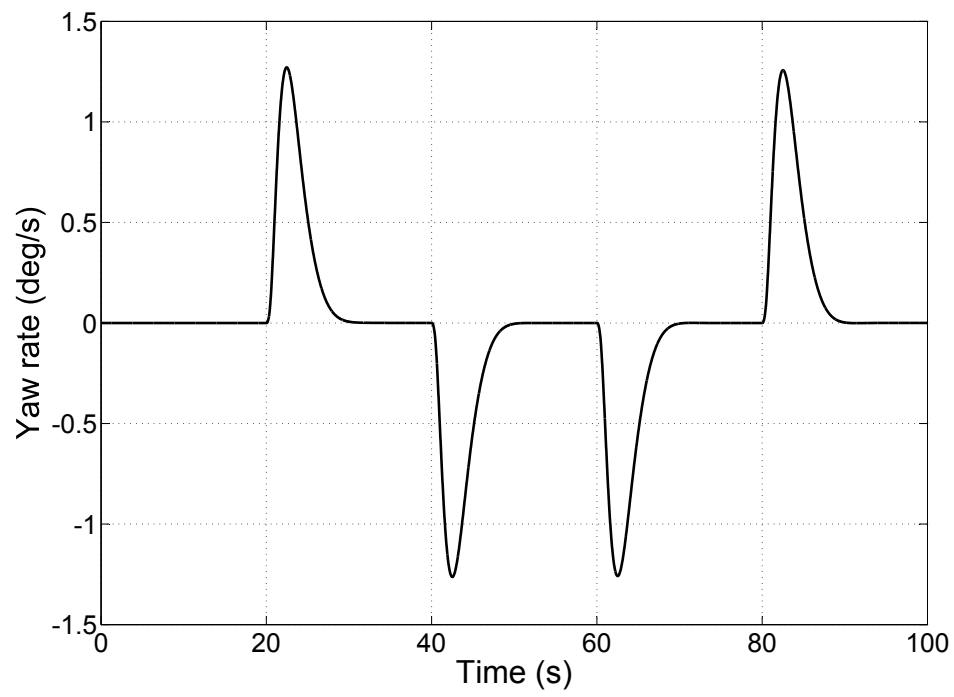
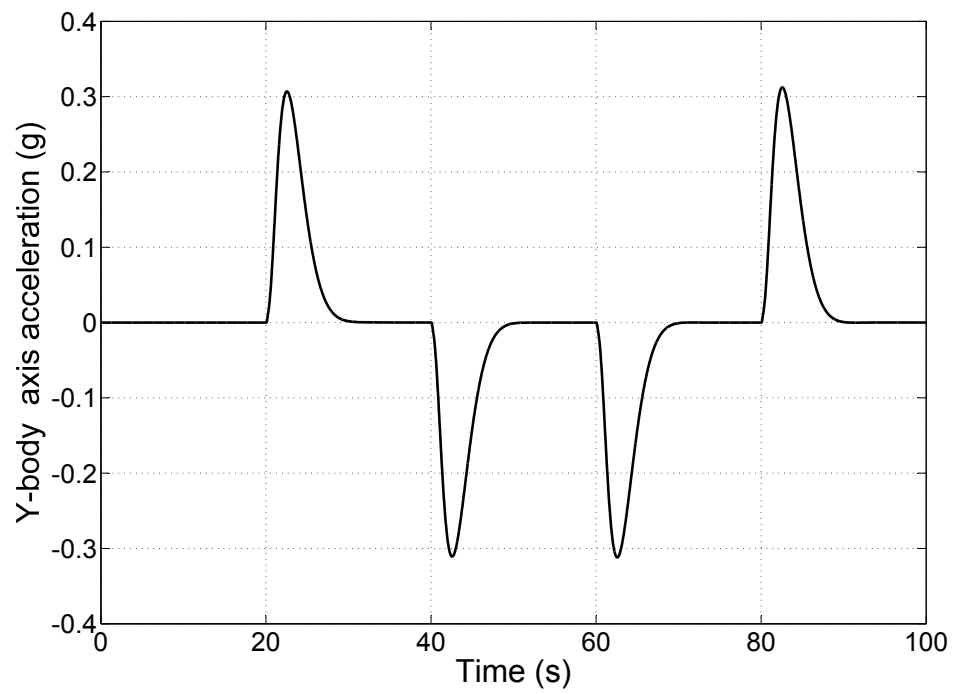
(a) ψ (b) ϕ



(c) β



(d) p

(e) r (f) a_y

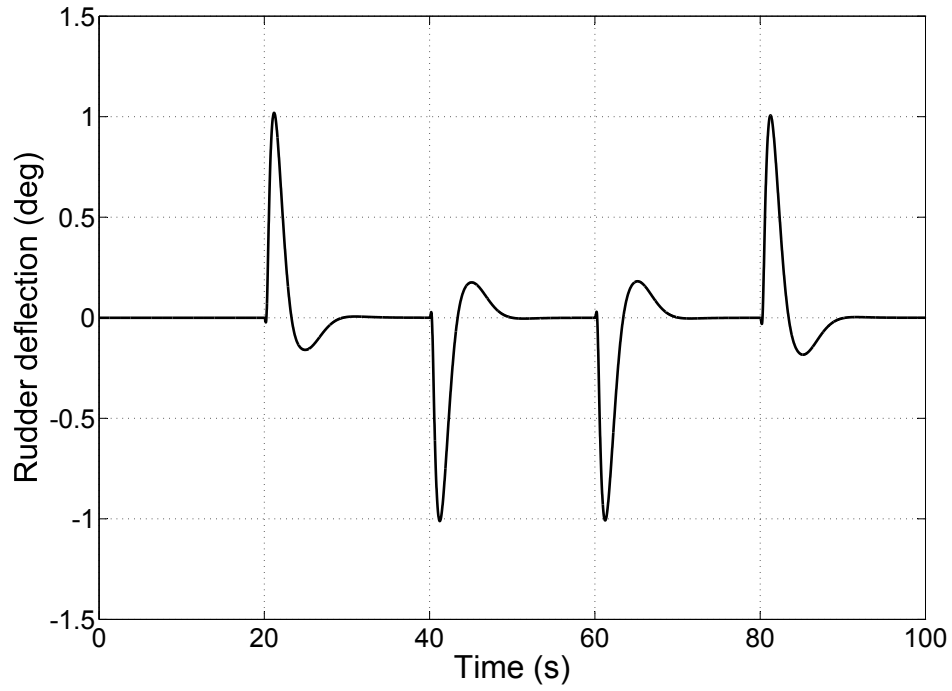
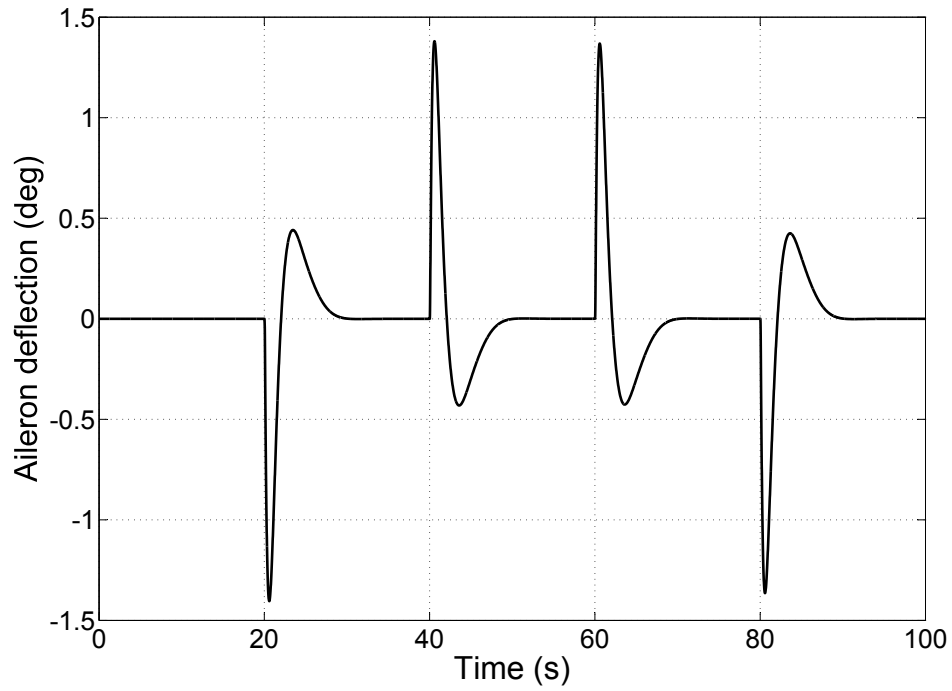
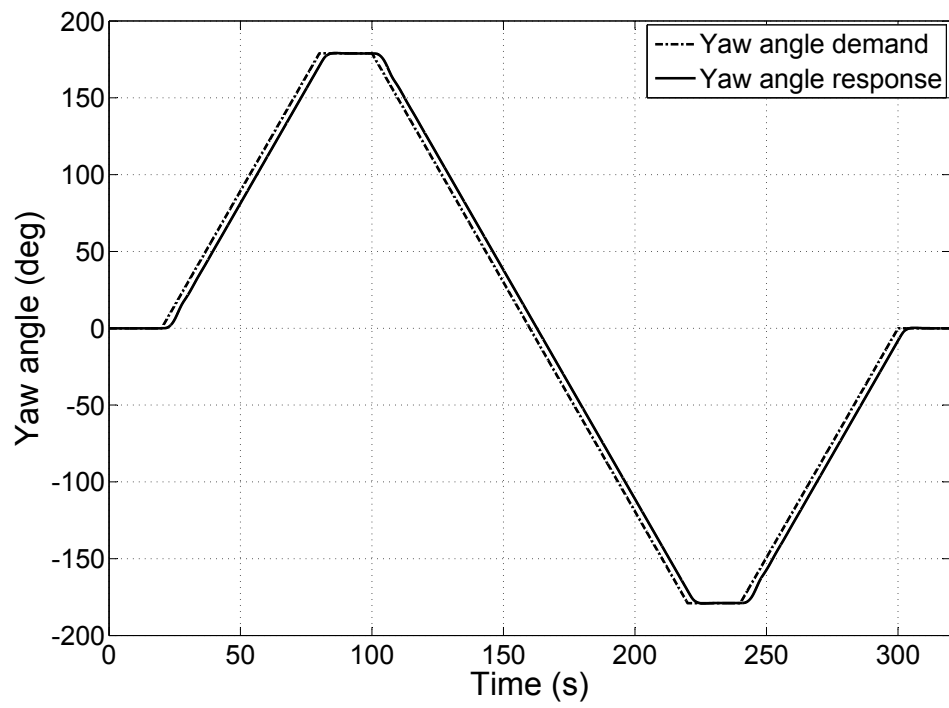
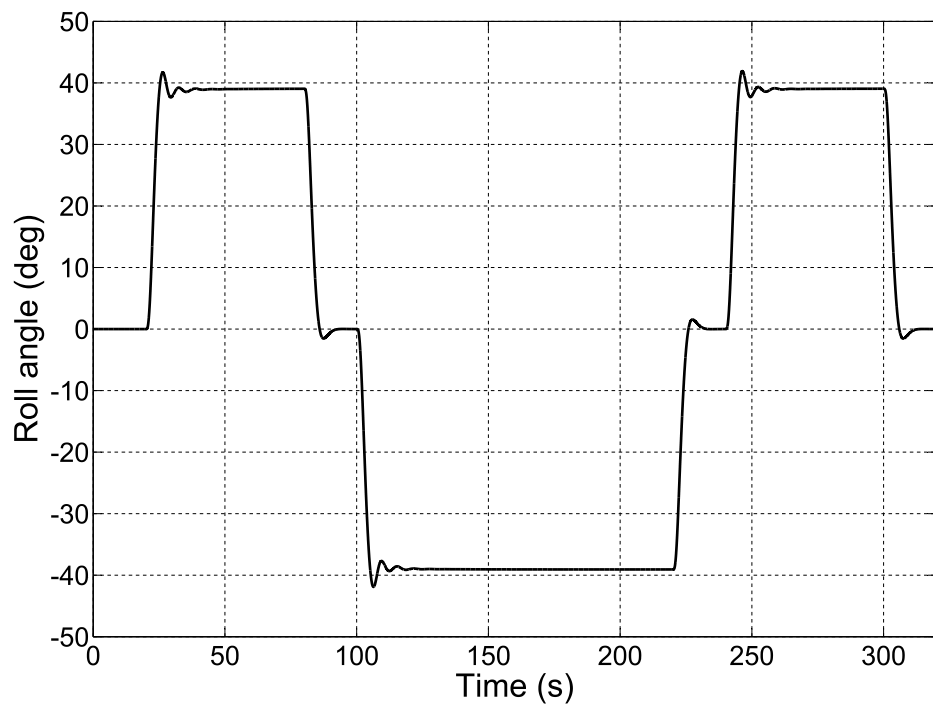
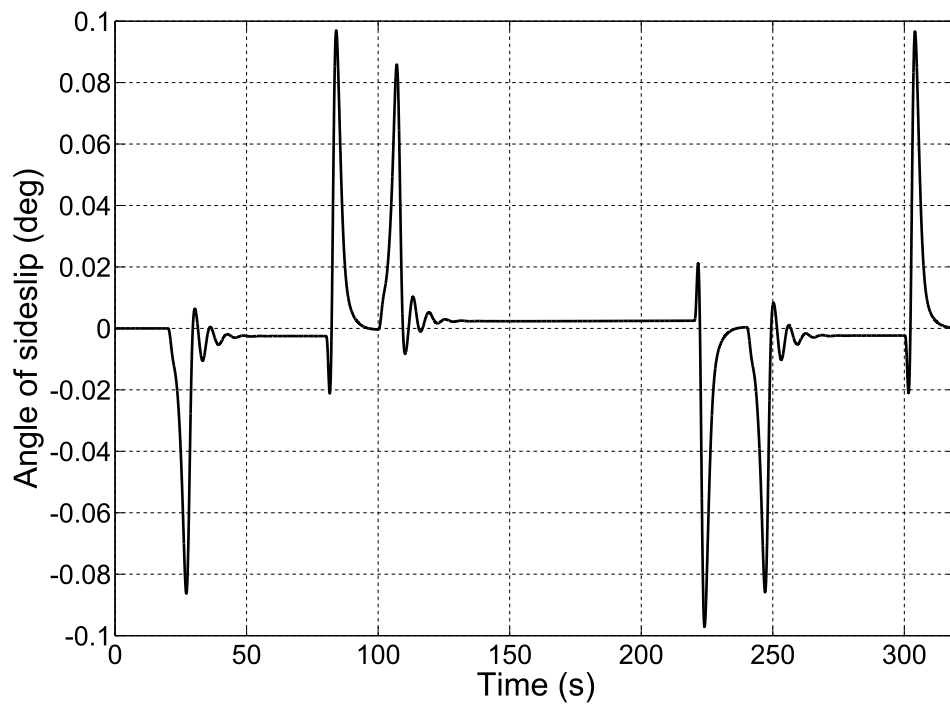
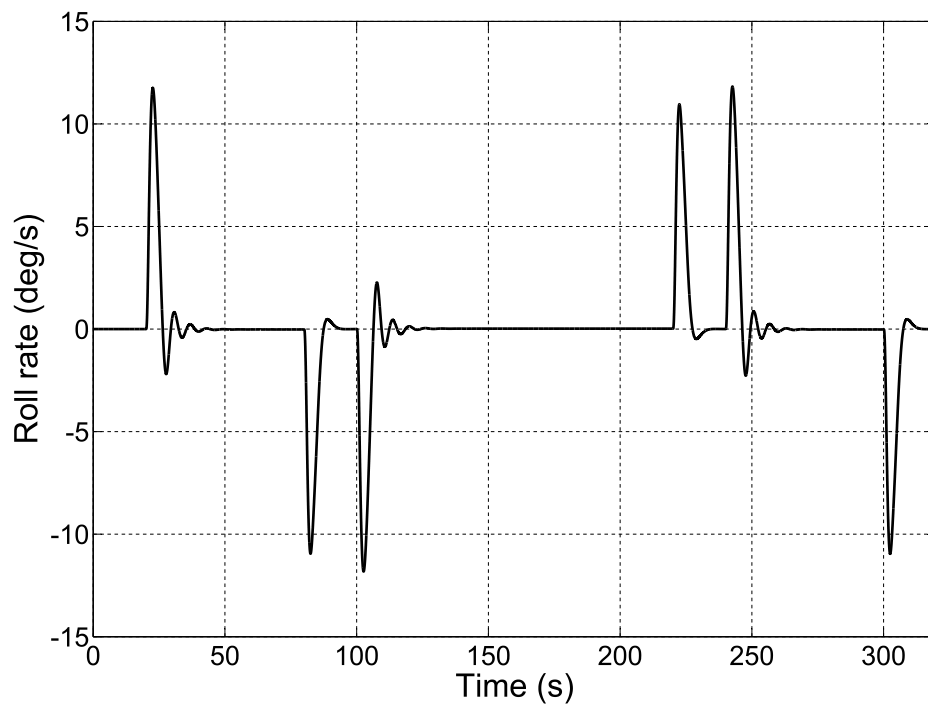
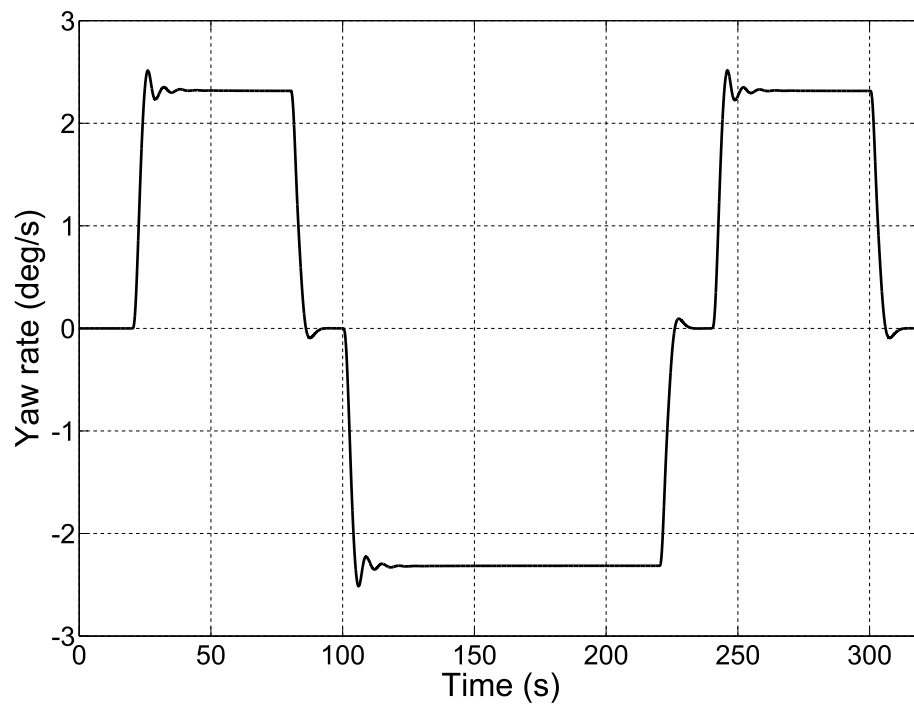
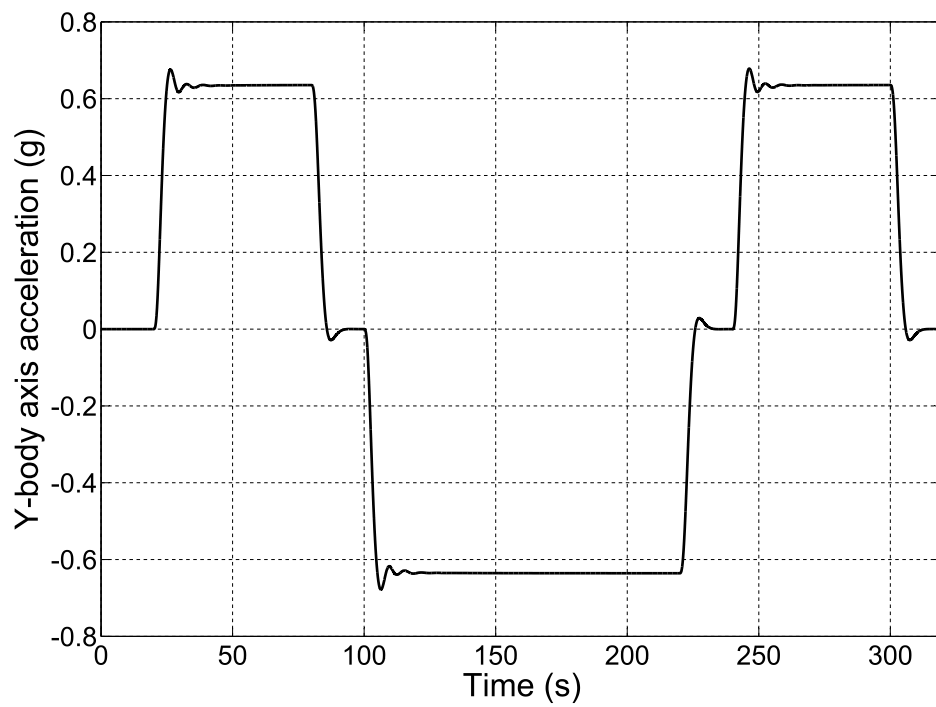
(g) δ_r (h) δ_a

Figure 7.8: The transient performance of H_∞ gain-scheduling autopilot is validated with the Jindivik nonlinear dynamic model about one condition inside the flight envelope, i.e. speed = 464 ft/s and altitude = 7,5000 ft.

(a) ψ (b) ϕ

(c) β (d) p

(e) r (f) a_y

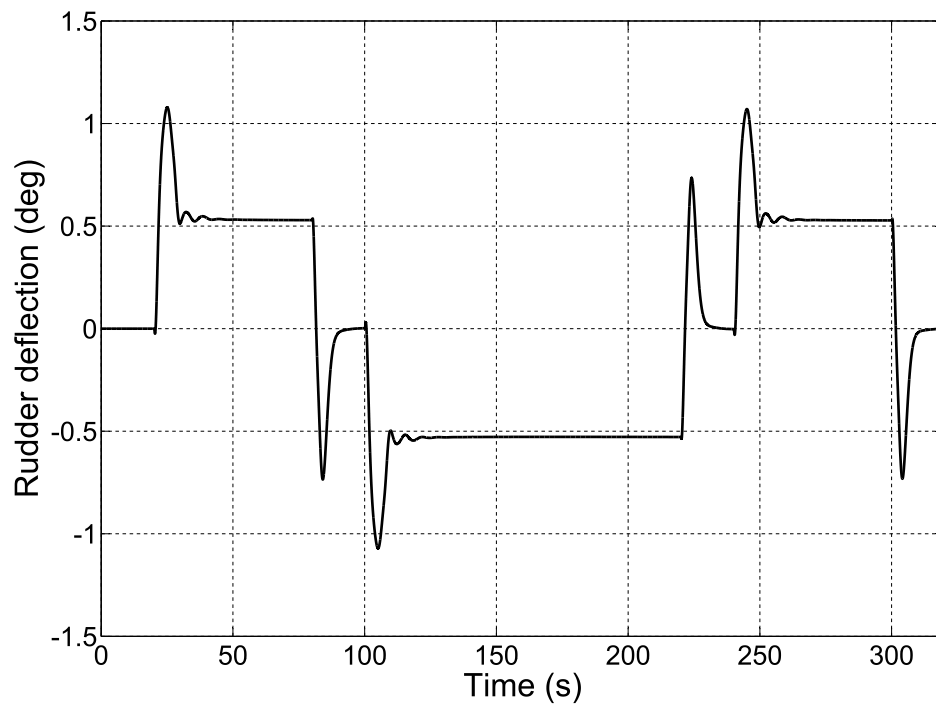
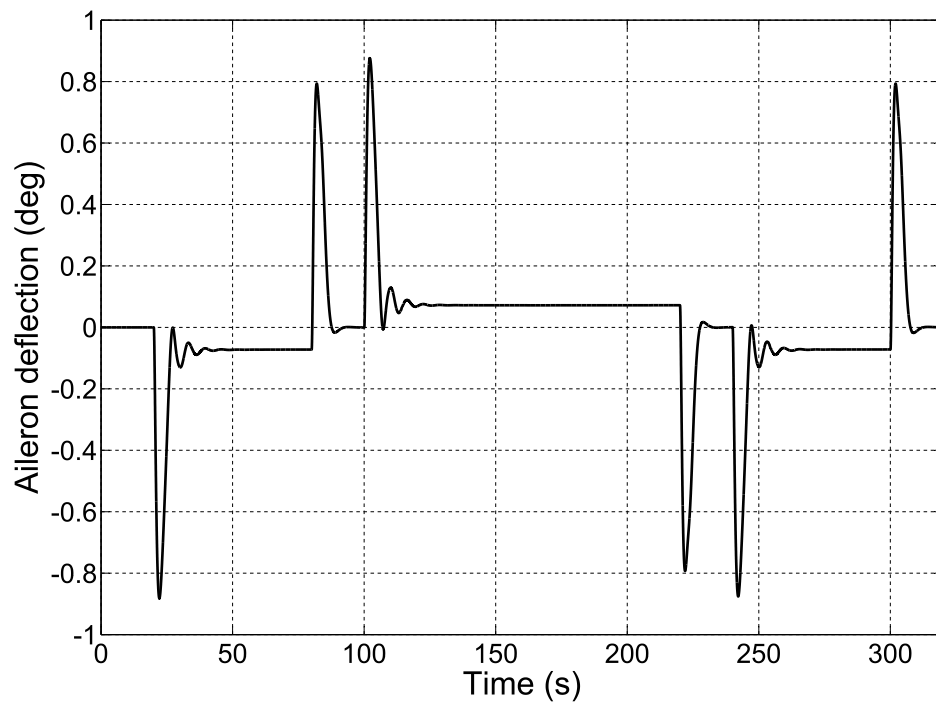
(g) δ_r (h) δ_a

Figure 7.9: The rate one turn of H_∞ gain-scheduling autopilot is validated with the Jindivik nonlinear dynamic model about one condition inside the flight envelope, i.e. speed = 506 ft/s and altitude = 10,000 ft.

Chapter 8

Conclusions

In this thesis, an LPV control approach has been employed to design a robust gain-scheduled flight controller for a conventional fixed-wing UAV. The work presented in this thesis was motivated by the shortcomings of the conventional gain-scheduling techniques that are both expensive and time-consuming for many UAV applications. The effectiveness of the proposed methods for designing flight controllers is verified and validated through the 6-DOF nonlinear model [41] in MATLAB Simulink environment of the Jindivik UAV.

However, the proposed methods are also applicable to a general class of conventional fixed-wing aircrafts. This chapter discusses some aspects of the work, lists the main contributions and provides suggestions for future work.

8.1 Conclusions & Discussions

- The dynamic characteristics of both lateral and longitudinal modes of an aircraft are represented by the stability and control derivatives, shown in (A.103) and (A.104), that vary following speed and altitude, shown in (5.14)–(5.36) and (7.7)–(7.24). With the speed and altitude fixed, both (A.103) and (A.104) are longitudinal and lateral LTI models, respectively. Suppose an aircraft is assumed about a wings level, constant altitude and airspeed flight condition, both nonlinearly parameter-dependent longitudinal and lateral LPV models, shown in (5.13) and (7.6) respectively, can be derived from a 6-DOF nonlinear model using Jacobian linearization.

Two interesting features of the derived LPV models are (i) they can accurately represent nonlinear dynamic characteristics of the 6-DOF nonlinear model better than the longitudinal and lateral LTI models, shown in (A.103)–(A.104) respectively, because they use the time-varying parameters θ (i.e. speed and altitude) to capture the nonlinear dynamic characteristics of the original nonlinear model and (ii) they are still a linear system where the system matrices are functions of speed and altitude. Note that, the speed and altitude are

also the state variables of the system. Comparison with the 6-DOF nonlinear models, the longitudinal and lateral LPV models are easier to prove stability using the single quadratic (Theorem 3.4.3) or parameter-dependent Lyapunov functions (Theorem 6.2.1).

- Equations (5.14)–(5.36) and (7.7)–(7.24) also show that the accuracy of both longitudinal and lateral LPV models, shown in (5.13) and (7.6) respectively, depend on the accuracy of the information that provides the aerodynamic and thrust coefficients. To estimate stability and control derivatives or aerodynamic coefficients of a conventional fixed-wing UAV, system identification techniques are preferred to the wind tunnel tests. However, a number of difficulties arise when system identification techniques are applied to certain UAVs, described in Appendix B. In addition, Appendix B shows that, when an UAV is flown as a racetrack manoeuvre pattern, those difficulties can be overcome.
- To synthesize an LPV controller based on Theorem 3.4.3 (SQLF) with a finite number of LMIs and avoiding the gridding parameter technique, the Tensor-Product (TP) model transformation can be applied to transform a nonlinearly parameter-dependent LPV model, shown in (5.13), into a TP-type convex polytopic model form, shown in (5.37). The TP-based LPV controller, shown in (5.46), can be constructed as a convex combination of the vertex coordinates of the scheduled parameter. Hence there is less computational on-line complexity at the gain-scheduling level than the grid-based LPV controller but its structure is still more complex than the affine-based LPV controller.
- An affine LPV model, shown in (6.55), is converted from a nonlinearly parameter-dependent LPV model, shown in (5.13), using the minimum least-squares method [58]. Based on an affine LPV model, an LPV controller can be synthesized with a finite number of LMIs using Theorem 3.4.3 (SQLF). However, the affine LPV model can rarely accurately represent the original nonlinear model and, in addition, the SQLF-based LPV controller is conservative when the parameters are time-invariant or slowly varying [45].
- The example from Leith and Leithead [61] is very interesting. The closed-loop instability of the LPV controller, shown in (4.11), with the original nonlinear model, shown in (4.1), occurs because the mismatch uncertainty between the Jacobian-based LPV model, shown in (4.5), and the original nonlinear model is in a region close to the right-half s -plane, described in Chapter 4. The mismatch uncertainties between a nonlinear model and an affine LPV model can be handled by a new class of affine LPV systems which is called an uncertain affine LPV model, described in Chapter 7. Synthesizing an LPV controller based on this uncertain affine LPV model, shown in (7.98), the stability property of the Leith and Leithead's example closed-loop system can be guaranteed with a larger stability margin. Hence, the closed-loop instability for the LPV controller, shown in (7.99), with the nonlinear plant disappears without degrading the transient performance as shown in Figure 7.4.
- Based on the lateral uncertain affine LPV model, shown in (7.105), an LPV

autopilot can be synthesized with a finite number of LMIs using Theorem 7.3.1 (SQLF) or 7.3.3 (PDLF) for which the resulting controller can guarantee the closed-loop system with a larger stability margin over conventional affine-based LPV controller. As shown by the simulation results of both longitudinal and lateral LPV autopilots in Chapters 5–7, the desired performance and robustness objectives are achieved across the defined flight envelope.

- Both aims and objectives of this thesis are achieved since the proposed schemes yield an LPV controller that can handle both uncertainties and nonlinearities of a 6-DOF nonlinear model with good command following, good disturbance attenuation, low sensitivity to measurement noise, reasonably small control efforts, and that is robustly stable to additive plant perturbations. We emphasize that the proposed schemes in this thesis can be a strong and very likely candidate for the next generation of flight control systems design for a conventional fixed-wing UAV.

8.2 Main Contributions

The main contributions of this thesis are the following:

- The determination of aircraft aerodynamic coefficient from wind-tunnel or computational fluid dynamics data can be an expensive and time-consuming procedure. A system identification techniques, i.e. equation-error method [58], is used to obtain this information from a racetrack manoeuvre with sufficient accuracy to design a satisfactory flight control system for an UAV [36].
- Knowing the aerodynamic coefficients, aircraft moments of inertias, and thrust coefficients, an LPV aircraft model can be derived from a 6-DOF nonlinear model using Jacobian linearization method. A tensor-product (TP) model transformation [15] is applied to transform a longitudinal nonlinearly parameter dependent LPV model to a TP-type convex polytopic model form. A gain-scheduled output feedback H_∞ controller [10] that is based on single quadratic Lyapunov functions is applied to the resulting TP convex polytopic model to yield a longitudinal LPV autopilot that guarantees the stability, robustness and performance properties of the closed-loop system [33].
- New sufficient conditions for gain-scheduled H_∞ performance analysis and synthesis for a class of affine LPV systems using parameter-dependent Lyapunov function are proposed, in Theorem 6.2.1. Compared with the multi-convexity technique [11], fewer linear matrix inequalities (LMIs) and decision variables are required and the computational time is lower while the achieved performance level is improved. The analysis and synthesis conditions are represented in the form of a finite number of LMIs. In addition, the intermediate controller variables, i.e. $\hat{A}_k(\theta)$, $\hat{B}_k(\theta)$, $\hat{C}_k(\theta)$ and $D_k(\theta)$, can be constructed as an affine matrix-valued function in the polytopic coordinates of the scheduled parameter

without the need for constraints on the D_{12} and D_{21} matrices. The proposed method is applied to synthesize a longitudinal LPV autopilot for a bounded flight envelope of a Jindivik UAV [34].

- The mismatch uncertainties between a nonlinear model and an affine LPV model are handled by a new class of affine LPV systems which is called an uncertain affine LPV model. New sufficient conditions for gain-scheduled H_∞ performance analysis and synthesis for this uncertain affine LPV model, using single quadratic or parameter-dependent Lyapunov functions are proposed. These are shown in Theorems 7.3.1 and 7.3.3 respectively, for which the proposed scheme can guarantee robust stability and robust performance for all possible time-varying real parametric uncertainties that are $\|\tilde{\delta}\| < 1/\gamma$. The analysis and synthesis conditions are represented in the form of a finite number of LMIs. A numerical example [61] is compared with uncertain LPV/LFT approach [9] results. The proposed scheme is applied to synthesize a lateral LPV autopilot for a bounded flight envelope of a Jindivik UAV [35].

8.3 Further Work

This thesis shows that an LPV controller can be synthesized as a single controller that will operate over the whole range of operating condition without having to create a scheduling scheme. Having assumed the complete measurement of the time-varying parameters θ is available for the controller to incorporate in the same LPV fashion as the plant model, the resulting LPV controller exploits all available information of θ to adjust its dynamic to the current plant dynamic on-line in real-time over the defined operating conditions. This provides smooth and automatic gain-scheduling with respect to θ .

However, there are still several directions that should be further researched and developed as outlined below.

- In this thesis, both longitudinal and lateral LPV models are derived from the 6-DOF nonlinear model of the Jindivik UAV using Jacobian linearization. However, the Jacobian-based LPV models only accurately represent the original nonlinear dynamics about the neighborhood of a set of equilibrium points and the time-varying parameters must vary slowly. Hence, the Jacobian linearization method is not suitable to derive an LPV aircraft model under the case of high angle of attack and extremely aggressive manoeuvring flight conditions. It is very interesting to apply the state transformation or function substitution methods to derive an LPV aircraft model under such highly nonlinear flight conditions. An initial research using the function substitution method to derive a quasi-LPV model of the F-16 aircraft that can cover the aircraft non-trim region has been proposed by Shin et al. [94].
- In this thesis, a small angle of attack flight condition is considered therefore there are only two scheduling parameters that are speed and altitude. However,

under the case of high angle of attack region, the angle of attack have a large variation hence it has to be included in the scheduling parameters where we believe that further improvements in the transient performance of both longitudinal and lateral LPV autopilots can be achieved if the autopilot is gain-scheduled with speed and altitude as well as the angle of attack. Then, it is also very interesting to design an LPV controller having speed, altitude and angle of attack as the scheduling parameters.

- Commonly, a feedforward controller is used to improve the transient performance of the closed-loop system. Hence, it is also very interesting to integrate the feedforward LPV controller into the gain-scheduled output feedback H_∞ control design framework where an initial research of this approach has been proposed by Prempain and Postlethwaite [85].
- Compared with an affine LPV model, a TP convex polytopic model more accurately represents the original nonlinear model. But, the TP polytopic model can not be written as an affine combination form. Therefore, to synthesize an LPV controller based on a TP polytopic model, a finite number of LMIs can be obtained only when using the single quadratic Lyapunov function. It is also very interesting (i) to extend the existing TP model transformation method [15] so that the resulting LPV model can be written in both convex and affine combination forms or (ii) to research a new convexifying techniques so that a finite number of LMIs can be obtained when synthesizing an LPV controller based on the existing TP convex polytopic model and using the parameter-dependent Lyapunov function.
- The TP-based LPV controller has a very high complexity. Hence controller complexity reduction is an important issue for the practical implementation of the method, and this aspect is also very interesting for further work.
- In this thesis, the effectiveness of the designed LPV autopilots, described in Chapters 5–7, is verified and validated only through the nonlinear simulation in MATLAB Simulink environment. It is also very interesting to test those designed LPV autopilots in the hardware-in-the-loop simulation.

This page intentionally contains only this sentence.

Appendix A

Aircraft Nonlinear Model

The full details of equations of motion for a conventional fixed-wing aircraft are well known and can be found in many textbooks, e.g. Cook [38], Klein and Morelli [58], Nelson [79], etc. The following sections give an overview of the equations of motion which is formulated as ordinary differential equations for the aircraft states with algebraic equations for the measured outputs. Before the equations of motion can be developed, it is necessary to define a suitable coordinate system and sign conventions (of the aircraft states, the control surfaces, and the measured outputs) for the formulation of the equations of motion.

A.1 Reference Frames & Sign Conventions

There are a variety of reference frames that would be used to describe aircraft movement and orientation for different purposes such as earth axes would be considered for a navigation control, and body axes would be considered for a stability control. The relevant frames are described in the following list [58]:

Earth axes: X_E, Y_E, Z_E . This reference frame is also called the topodetic axes. Its origin is at an arbitrary point on the earth surface, with positive X_E axis pointing toward north, positive Y_E axis pointing east, and positive Z_E axis pointing to the center of the earth as shown in Figure A.1. Earth axes are fixed with respect to the earth.

Vehicle-carried earth axes: X_V, Y_V, Z_V . Origin is at the centre of gravity of the aircraft (c.g.), orientation of the axes is parallel to earth axes (see Figure A.1). The centre of gravity is the point about which the aircraft would balance if suspended by a cable. This reference frame is used to show the rotational orientation of the aircraft relative to earth axes.

Body axes: X_B, Y_B, Z_B . Origin is at the aircraft c.g., with positive X_B axis pointing forward through the nose of the aircraft, positive Y_B axis out the right wing, and

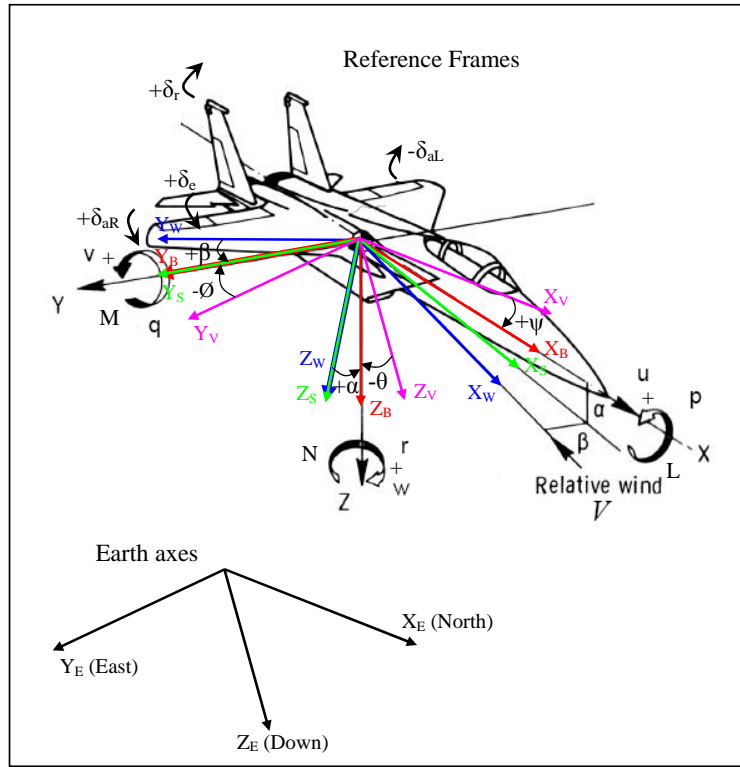


Figure A.1: Reference axes & sign conventions [58]

positive Z_B axis is directed through the underside. Body axes are fixed with respect to the aircraft body as shown in Figure A.1. The XZ plane is commonly a plane of symmetry for the aircraft.

Stability axes: X_S, Y_S, Z_S . This reference frame is a type of body axes, used in the study of small deviations from a nominal flight condition. The orientation of stability axes is related to a reference flight condition, usually defined at the start of a manoeuvre. Its origin is at the aircraft c.g., with positive X_S axis forward and aligned with the projection of the velocity vector of the aircraft c.g. through the air (also called the air-relative velocity) onto the XZ plane in body axes. The positive Y_S axis out the right wing, and positive Z_S axis is directed through the underside (see Figure A.1).

Wind axes: X_W, Y_W, Z_W . This reference frame, also called the flight-path axes, has its origin at the aircraft c.g., with positive X_W axis forward and aligned with the air-relative velocity vector, positive Y_W axis out the right wing, and positive Z_W axis through the underside in the XZ plane in body axes (see Figure A.1). The X_W axis is always tangent to the air-relative trajectory, hence wind axes are not fixed with respect to the aircraft body.

Sign conventions: For translational velocities at the aircraft c.g. along body axes, e.g. axial (u), lateral (v), and normal velocities (w) or applied forces at the aircraft c.g. along body axes, e.g. axial force (X), side force (Y), normal force (Z), the

positive sign convention follows the positive direction of a body axis (see Figure A.1). In addition, angular velocities about the aircraft c.g. along body axes, e.g. pitch rate (p), roll rate (q), and yaw rate (r) or applied moments about the aircraft c.g. along body axes, e.g. rolling moment (L), pitching moment (M), and yawing moment (N), the sign convention follows the right-hand rule. If the right-hand thumb is pointed in the positive direction of a body axis, the fingers curl in the direction of positive rotation (see Figure A.1).

Control surfaces are hinged surfaces that can be rotated about a hinge line to change the applied aerodynamic forces and moments on an aircraft. For control surfaces of a conventional aircraft, e.g. elevator δ_e , aileron δ_a , and rudder δ_r , the sign convention also follow the right-hand rule. For example, the positive sign of elevator control surfaces, If the right-hand thumb is pointed in the positive direction of a Y_B axis, the fingers curl in the direction of positive deflection (see Figure A.1). However, some control surfaces, i.e. the ailerons, are deflected simultaneously in an asymmetric manner, which means that the individual aileron control surfaces on each wing move in opposite directions. Following Klein and Morelli [58], a positive aileron deflection is defined as one-half the right aileron deflection minus the left aileron deflection,

$$\delta_a = \frac{1}{2} (\delta_{aR} - \delta_{aL}) \quad (\text{A.1})$$

In addition, the angle of attack (α) and sideslip angle (β) can be defined in terms of the velocity components of air-relative velocity (V) as shown in Figure A.1. The positive sign convention of α and β follow the positive sign of w and v respectively.

$$\begin{aligned} \alpha &= \arctan \frac{w}{u} \\ \beta &= \arcsin \frac{v}{V} \\ V &= \sqrt{u^2 + v^2 + w^2} \\ \begin{bmatrix} u \\ v \\ w \end{bmatrix} &= V \begin{bmatrix} \cos \alpha \cos \beta \\ \sin \beta \\ \sin \alpha \cos \beta \end{bmatrix} \end{aligned} \quad (\text{A.2})$$

Moreover, the relative orientation of the body axes to earth axes or to vehicle-carried earth axes are the same angles, known as Euler angles, e.g. pitch angle (θ), roll angle (ϕ), and yaw angle (ψ) as shown in Figure A.1. The positive sign convention of θ , ϕ , and ψ are defined by the rotation direction of the vehicle-carried earth axes to body axes, which follows the right-hand rule. If the right-hand thumb is pointed in the positive direction of a rotating axis, the fingers curl in the positive rotation direction about the rotating axis.

The sequence for rotating vehicle-carried earth axes into alignment with body axes starts with a yaw angle rotation ψ about the Z_V axis. In Figure A.1, it is a positive rotation direction according to right-hand rule stated above. This followed by a pitch angle rotation θ (a negative rotation direction) about the Y_V axis, completed by a roll angle rotation ϕ (also a negative rotation direction) about the X_V axis. Therefore, an arbitrary three dimensional vector at earth axes or vehicle-carried earth axes,

e.g. gravity force (F_G), can be transformed to body axes using a transformation matrix L_{BV} , known as Direction Cosine Matrix (DCM), as shown below,

$$L_{BV} = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix} \quad (\text{A.3})$$

$$F_G = \begin{bmatrix} mg_x \\ mg_y \\ mg_z \end{bmatrix}_B = L_{BV} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}_V = \begin{bmatrix} -mg \sin \theta \\ mg \sin \phi \cos \theta \\ mg \cos \phi \cos \theta \end{bmatrix} \quad (\text{A.4})$$

A.2 Aircraft Equation of Motion

The general motion of an aircraft, that is assumed a rigid body with fixed mass distribution and constant mass, can then be derived from Newton's second law of motion in translational and rotational forms. It is noted that all the given equations in this section are taken from Klein and Morelli [58],

$$\begin{aligned} F &= m\dot{V} + \omega \times mV \\ M &= I\dot{\omega} + \omega \times I\omega \\ I &= \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ I_{yx} & I_y & -I_{yz} \\ -I_{zx} & I_{zy} & I_z \end{bmatrix} \end{aligned} \quad (\text{A.5})$$

where F is the applied forces at the aircraft c.g. along body axes (i.e. X , Y , and Z), m is the mass of the aircraft, V is the translational velocities at the aircraft c.g. along body axes (i.e. u , v , and w), ω is the angular velocities about the aircraft c.g. along body axes (i.e. p , q , and r), M is the applied moments about the aircraft c.g. along body axes (i.e. L , M , and N), and I is the moment of inertia matrix of the aircraft. For a rigid body with symmetry relative to the XZ plane in body axes, the moment of inertia matrix I is symmetric and $I_{xy} = I_{yx} = I_{yz} = I_{zy} = 0$. The moment of inertia matrix I then reduces to,

$$I = \begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{xz} & 0 & I_z \end{bmatrix} \quad (\text{A.6})$$

where

$$\begin{aligned} I_x &= \int_{Volume} x^2 dm & I_y &= \int_{Volume} y^2 dm \\ I_z &= \int_{Volume} z^2 dm & I_{xz} &= \int_{Volume} xz dm \end{aligned}$$

Thus, writing (A.5) in terms of the variables defined in this section gives the force equations:

$$\begin{aligned} X &= m(\dot{u} + qw - rv) \\ Y &= m(\dot{v} + ru - pw) \\ Z &= m(\dot{w} + pv - qu) \end{aligned} \quad (\text{A.7})$$

and moment equations:

$$\begin{aligned} L &= \dot{p}I_x - \dot{r}I_{xz} + qr(I_z - I_y) - qpI_{xz} \\ M &= \dot{q}I_y + pr(I_x - I_z) + (p^2 - r^2)I_{xz} \\ N &= \dot{r}I_z - \dot{p}I_{xz} + pq(I_y - I_x) + qrI_{xz} \end{aligned} \quad (\text{A.8})$$

Reasonably, the forces acting on an aircraft in flight consist of aerodynamic (i.e. X_{aero} , Y_{aero} , and Z_{aero}), thrust (i.e. T_x , T_y , and T_z), and gravitational forces (i.e. mg_x , mg_y , and mg_z). Since gravity acts through the aircraft c.g., and the gravity field is assumed uniform, there is no gravity moment acting on the aircraft. Furthermore, to simplify the equation (A.7) and (A.8), the thrust from the propulsion system is assumed to act along the X_B body axis and through the c.g., thus give $T_y = T_z = 0$. The angular momentum due to the propulsion system is also neglected therefore there is only aerodynamic (i.e. L_{aero} , M_{aero} , and N_{aero}) moment acting on an aircraft. The aerodynamic forces and moments acting on the aircraft can be expressed in terms of nondimensional coefficients:

$$\begin{aligned} \begin{bmatrix} X_{\text{aero}} \\ Y_{\text{aero}} \\ Z_{\text{aero}} \end{bmatrix} &= \bar{q}S \begin{bmatrix} C_X \\ C_Y \\ C_Z \end{bmatrix} \\ \begin{bmatrix} L_{\text{aero}} \\ M_{\text{aero}} \\ N_{\text{aero}} \end{bmatrix} &= \bar{q}S \begin{bmatrix} bC_l \\ \bar{c}C_m \\ bC_n \end{bmatrix} \\ \bar{q} &= \frac{1}{2}\rho V^2 \end{aligned} \quad (\text{A.9})$$

where C_X , C_Y , C_Z , C_l , C_m , and C_n are the aerodynamic coefficients that primarily are a function of the Mach number, Reynolds number, angle of attack, and sideslip angle; They are secondary functions of the time rate of change of angle of attack and sideslip, and the angular velocity of the aircraft. \bar{q} is the dynamic pressure, V is the magnitude of the air-relative velocity (also called the airspeed), ρ is the air density, S is the wing reference area, b is the wing span, and \bar{c} is the mean aerodynamic chord of the wing.

Substituting the preceding expressions, including (A.4), into the dynamic equation (A.7) and (A.8) gives,

Force equations:

$$\begin{aligned}\dot{u} &= rv - qw + \frac{\bar{q}S}{m}C_X - g \sin \theta + \frac{T}{m} \\ \dot{v} &= pw - ru + \frac{\bar{q}S}{m}C_Y + g \cos \theta \sin \phi \\ \dot{w} &= qu - pv + \frac{\bar{q}S}{m}C_Z + g \cos \theta \cos \phi\end{aligned}\tag{A.10}$$

Moment equations:

$$\begin{aligned}\dot{p} - \frac{I_{xz}}{I_x}\dot{r} &= \frac{\bar{q}Sb}{I_x}C_l - \frac{(I_z - I_y)}{I_x}qr + \frac{I_{xz}}{I_x}qp \\ \dot{q} &= \frac{\bar{q}S\bar{c}}{I_y}C_m - \frac{(I_x - I_z)}{I_y}pr + \frac{I_{xz}}{I_y}(p^2 - r^2) \\ \dot{r} - \frac{I_{xz}}{I_z}\dot{p} &= \frac{\bar{q}Sb}{I_z}C_n - \frac{(I_y - I_x)}{I_z}pq - \frac{I_{xz}}{I_z}qr\end{aligned}\tag{A.11}$$

where $T = T_x$. However, there are a lot more variety of equations that are interesting and often used in flight simulation application such as by differentiating (A.2) with respect to time gives,

$$\begin{aligned}\dot{V} &= \frac{1}{V}(u\dot{u} + v\dot{v} + w\dot{w}) \\ \dot{\alpha} &= \left(\frac{u\dot{w} - w\dot{u}}{u^2 + w^2} \right) \\ \dot{\beta} &= \frac{(u^2 + w^2)\dot{v} - v(u\dot{u} + w\dot{w})}{V^2\sqrt{u^2 + w^2}}\end{aligned}\tag{A.12}$$

Substituting in (A.12) for \dot{u} , \dot{v} , and \dot{w} from (A.10), and for u , v , and w from (A.2), gives the wind axes force equations as,

$$\begin{aligned}\dot{V} &= -\frac{\bar{q}S}{m}C_{D_w} + \frac{T}{m}\cos \alpha \cos \beta + g(\cos \phi \cos \theta \sin \alpha \cos \beta) \\ &\quad + g(\sin \phi \cos \theta \sin \beta - \sin \theta \cos \alpha \cos \beta) \\ \dot{\alpha} &= -\frac{\bar{q}S}{mV \cos \beta}C_L + q - \tan \beta(p \cos \alpha + r \sin \alpha) - \frac{T \sin \alpha}{mV \cos \beta} \\ &\quad + \frac{g}{V \cos \beta}(\cos \phi \cos \theta \cos \alpha + \sin \theta \sin \alpha) \\ \dot{\beta} &= \frac{\bar{q}S}{mV}C_{Y_w} + p \sin \alpha - r \cos \alpha + \frac{g}{V} \cos \beta \sin \phi \cos \theta \\ &\quad + \frac{\sin \beta}{V} \left(g \cos \alpha \sin \theta - g \sin \alpha \cos \phi \cos \theta + \frac{T \cos \alpha}{m} \right)\end{aligned}\tag{A.13}$$

where

$$\begin{aligned}C_L &= -C_Z \cos \alpha + C_X \sin \alpha \\ C_D &= -C_X \cos \alpha - C_Z \sin \alpha \\ C_{D_w} &= C_D \cos \beta - C_Y \sin \beta \\ C_{Y_w} &= C_Y \cos \beta + C_D \sin \beta\end{aligned}\tag{A.14}$$

C_L and C_D are lift coefficient and drag coefficient in stability axes respectively, whereas they are obtained from body axes components by rotation through α and β . The positive sign convention of C_L and C_D are directed along the $-Z_S$ and $-X_S$ stability axes, respectively. Similarly, C_{D_W} and C_{Y_W} are lift coefficient and side force coefficient in wind axes respectively. The positive sign convention of C_{D_W} and C_{Y_W} are directed along the $-X_W$ and $+Y_W$ stability axes, respectively.

Furthermore, the *Rotational kinematic equations* relate the rate of change of the Euler angles to the body axis components of angular velocity.

$$\begin{aligned} p &= \dot{\phi} - \dot{\psi} \sin \theta \\ q &= \dot{\theta} \cos \phi + \dot{\psi} \sin \phi \cos \theta \\ r &= \dot{\psi} \cos \phi \cos \theta - \dot{\theta} \sin \phi \end{aligned} \quad (\text{A.15})$$

or

$$\begin{aligned} \dot{\phi} &= p + \tan \theta (q \sin \phi + r \cos \phi) \\ \dot{\theta} &= q \cos \phi - r \sin \phi \\ \dot{\psi} &= \frac{q \sin \phi + r \cos \phi}{\cos \theta} \end{aligned} \quad (\text{A.16})$$

Navigation equations are most often used to calculate the position of the aircraft. Its dynamics can be determined using a transformation matrix L_{VB} , that transforms an arbitrary three dimensional vector at body axes to earth axes or vehicle-carried earth axes:

$$\begin{aligned} L_{VB} &= L_{BV}^{-1} \\ &= \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \end{aligned} \quad (\text{A.17})$$

$$\begin{aligned} x_E &= u \cos \psi \cos \theta + v (\cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi) \\ &\quad + w (\cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi) \\ y_E &= u \sin \psi \cos \theta + v (\sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi) \\ &\quad + w (\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi) \\ \dot{h} &= u \sin \theta - v \cos \theta \sin \phi - w \cos \theta \cos \phi \end{aligned} \quad (\text{A.18})$$

where h is altitude (height above the ground), $= -Z_E$. Other commonly used variables, i.e. azimuth or heading angle (χ), and the flight path angle (γ), which can be computed directly from the aircraft's state. These are defined as,

$$\begin{aligned} \chi &= \beta + \psi \\ \gamma &= \arcsin \frac{\dot{h}}{V} = \theta - \alpha \end{aligned} \quad (\text{A.19})$$

A.3 Jindivik Nonlinear Mathematical Model

The Jindivik UAV, shown in Figure 1.1, is a remotely piloted fixed wing aircraft, which has been used as an aerial target drone in Australia and the UK for 50 years. Its specifications are given in Table A.1. Based on manufacturer's (Australian Government Aircraft Factory) wind tunnel data and subsequent flight trial validation, a 6-DOF nonlinear mathematical model of the UAV has been developed in the MATLAB Simulink environment by Fitzgerald [41], shown in Figure A.2. This section only gives a brief overview of the Jindivik nonlinear model; for further details refer to [41] and [42].

The aerodynamic model of the Jindivik nonlinear model is only valid within the following flight condition:

- Altitude between sea level and 20,000 ft.
- True airspeed between 180 knots and 530 knots.
- Bank angles up to 80°.
- Normal acceleration in the range -2g to +8g.
- Maximum Mach number 0.86.

A.3.1 Aerodynamic Force and Moment Models

The aerodynamic forces (i.e. X_{aero} , Y_{aero} , and Z_{aero}) and moments (i.e. L_{aero} , M_{aero} , and N_{aero}) acting on the vehicle are computed using (A.9). Following [42], (A.9) can be written further as

- Axial drag force due to aerodynamics, X_{aero} ,

$$X_{\text{aero}} = \frac{1}{2}\rho V^2 S [C_{L_{\text{wb}}} \sin(\alpha) - C_D \cos(\alpha)] + \frac{1}{2}\rho V^2 S_T [C_{L_T} \sin(\alpha_T - \alpha_R)] \quad (\text{A.20})$$

where C_D is the drag coefficient, C_{L_T} is the tailplane lift coefficient, $C_{L_{\text{wb}}}$ is the wing-body lift coefficient, $S = 76 \text{ ft}^2$ is the wing area, $S_T = 14 \text{ ft}^2$ is the tailplane area, $\alpha_R = -0.5 \text{ deg}$ is the tailplane rigging angle, and α_T is the tailplane angle of attack. α_T is defined as

$$\alpha_T = \alpha + \alpha_R - \varepsilon + q \frac{l_T}{V} + \text{lag}_T \quad (\text{A.21})$$

where $l_T = 9.49 \text{ ft}$ is the tail moment arm that is measured from wing quarter chord to tailplane quarter chord, ε is the downwash angle, and lag_T is the tailplane angle of attack lag angle due to downwash. ε is defined as

$$\varepsilon = \varepsilon_0 + \frac{\partial \varepsilon}{\partial \alpha} (\alpha - \alpha_0) \quad (\text{A.22})$$

Table A.1: The Jindivik UAV specifications

Description	Details
fuselage length	28 ft 8.75 in
wing span	18 ft 9.6 in
wing area	76 ft ²
wing chord	4 ft
maximum height	6 ft 9.85 in
weight	108.7 slug*
inertia I _x	2,228 slug-ft ² *
inertia I _y	1,789 slug-ft ² *
inertia I _z	3,934 slug-ft ² *
speed range	200 - 450 knot
service ceiling	18000 ft
range	820 miles
propulsion	Bristol Siddeley Viper Mk.201 turbojet; 1800 lb _f

* with fuel of 100 gallon

where α_0 is the zero lift angle of attack, ε_0 is the downwash angle at zero angle of attack, and $\partial\varepsilon/\partial\alpha$ is the rate of change of downwash angle with respect to angle of attack of the flexible aircraft. ε_0 is defined as

$$\varepsilon_0 = \varepsilon_{0\delta_f=1} + \left(\frac{\varepsilon_{0\delta_f=20} - \varepsilon_{0\delta_f=1}}{19} \right) (\delta_f - 1) \quad (\text{A.23})$$

where $\varepsilon_{0\delta_f=1}$ and $\varepsilon_{0\delta_f=20}$ are the downwash angles at zero angle of attack with flaps deflection $\delta_f = 1$ and $\delta_f = 20$ respectively from which they are defined as

$$\varepsilon_{0\delta_f=1} = 0.0323 - \frac{0.00611}{\beta_{pg}^2} + \frac{3.5 \times 10^{-5}}{\beta_{pg}^8} \text{ rad} \quad (\text{A.24})$$

$$\varepsilon_{0\delta_f=20} = -0.0909 \text{ rad} \quad (\text{A.25})$$

where the Prandtl-Glauert factor, β_{pg} , is defined as

$$\beta_{pg} = \sqrt{1 + M^2}, \quad M = \frac{V}{a} \quad (\text{A.26})$$

where M and a are the Mach number and the speed of sound, respectively. $\partial\varepsilon/\partial\alpha$ is defined as

$$\frac{\partial\varepsilon}{\partial\alpha} = K_{\#2} \frac{\partial\varepsilon}{\partial\alpha} \Big|_R \quad (\text{A.27})$$

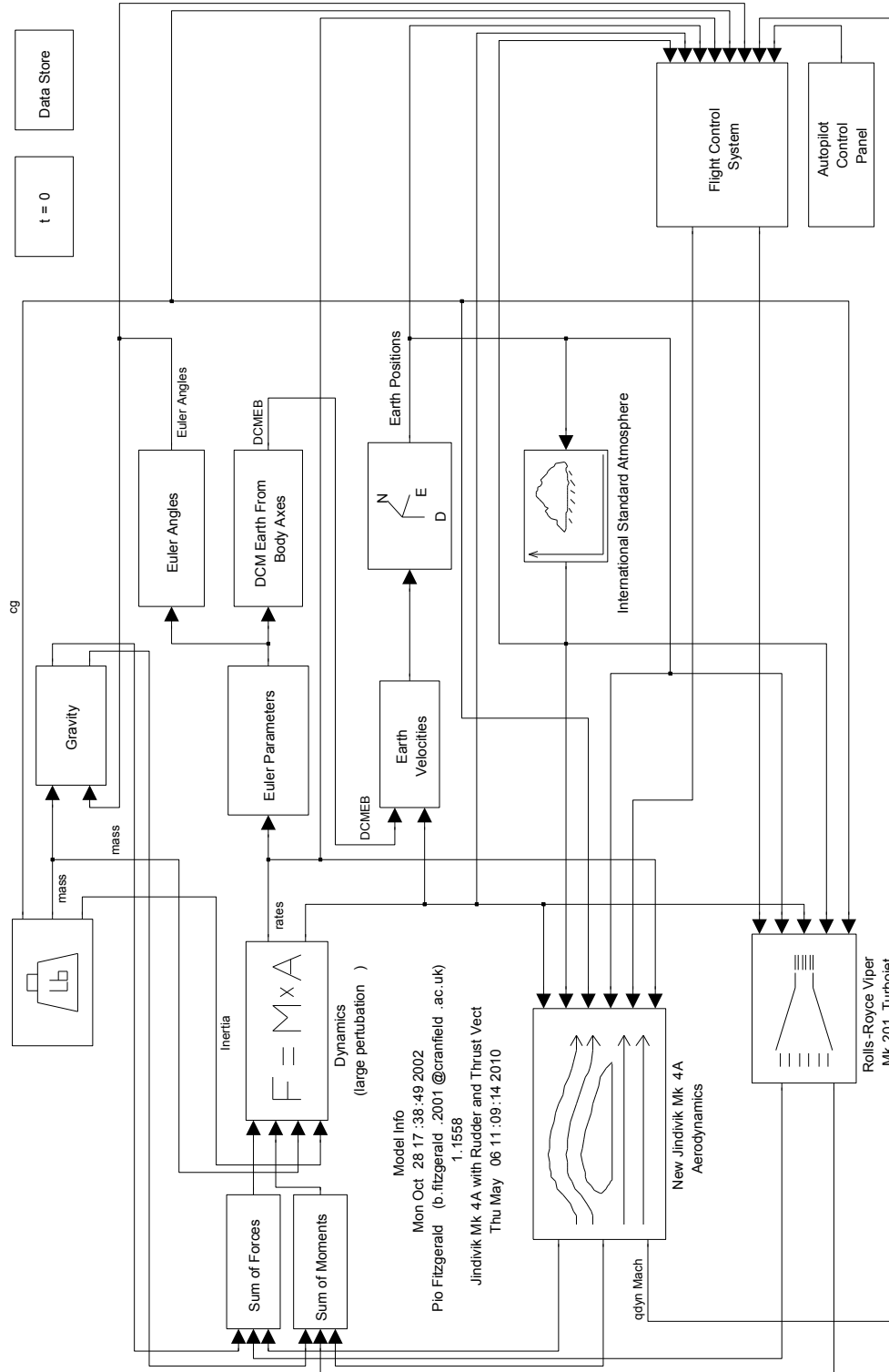


Figure A.2: The Jindivik nonlinear model with its autopilot in MATLAB Simulink environment [41]

where K_{ff2} is the flexible factor applied to the rigid body rate of change of downwash with respect to angle of attack and it is a function of Mach number and altitude as presented in Fitzgerald [42, Figure 3-6]. The rate of change of downwash angle with respect to angle of attack of the rigid aircraft, $\left. \frac{\partial \varepsilon}{\partial \alpha} \right|_R$, is defined as

$$\left. \frac{\partial \varepsilon}{\partial \alpha} \right|_R = \frac{\partial \varepsilon}{\partial \alpha_{\delta_f=1}} + \left(\frac{\frac{\partial \varepsilon}{\partial \alpha_{\delta_f=20}} - \frac{\partial \varepsilon}{\partial \alpha_{\delta_f=1}}}{19} \right) (\delta_f - 1) \quad (A.28)$$

where $\frac{\partial \varepsilon}{\partial \alpha_{\delta_f=1}}$ and $\frac{\partial \varepsilon}{\partial \alpha_{\delta_f=20}}$ are the rate of change of downwash angle with respect to angle of attack with flaps deflection $\delta_f = 1$ and $\delta_f = 20$ respectively in which they are defined as

$$\frac{\partial \varepsilon}{\partial \alpha_{\delta_f=1}} = a_{1_{wb}} \frac{\partial \varepsilon}{\partial C_L} \quad (A.29)$$

$$\frac{\partial \varepsilon}{\partial \alpha_{\delta_f=20}} = 0.412 \quad (A.30)$$

where $a_{1_{wb}}$ is the wing-body combination lift curve slope of the flexible aircraft and $\partial \varepsilon / \partial C_L$ is the rate of change of downwash angle with lift coefficient. $a_{1_{wb}}$ is defined as

$$a_{1_{wb}} = K_{ff1} a_{1_{wb_R}} \quad (A.31)$$

where K_{ff1} is the flexibility factor applied to the rigid wing-body combination lift curve slope and it is a function of Mach number and altitude as presented in Fitzgerald [42, Figure 3-5]. The wing-body combination lift curve slope of the rigid aircraft, $a_{1_{wb_R}}$, is defined as

$$a_{1_{wb_R}} = a_{1_{wb_R \delta_f=1}} + \left(\frac{a_{1_{wb_R \delta_f=20}} - a_{1_{wb_R \delta_f=1}}}{19} \right) (\delta_f - 1) \quad (A.32)$$

$$a_{1_{wb_R \delta_f=1}} = 5.05 \text{ rad}^{-1}, \quad a_{1_{wb_R \delta_f=20}} = 4.9 + 4.4(1 - \beta_{pg}) \text{ rad}^{-1} \quad (A.33)$$

$\partial \varepsilon / \partial C_L$ is defined as

$$\frac{\partial \varepsilon}{\partial C_L} = 0.0815 - \frac{4 \times 10^{-5}}{\beta_{pg}^8} \quad (A.34)$$

lag_T is defined as

$$\text{lag}_T = \dot{\alpha} \frac{\partial \varepsilon}{\partial \alpha} \frac{l_T}{V} \quad (A.35)$$

In addition, the lift and drag coefficients are defined as

$$C_L = C_{L_{wb}} + \frac{S_T}{S} C_{L_T} + C_{L_s} \quad (A.36)$$

$$C_D = C_{D_i} + C_{D_{Z_0}} + C_{D_{Re}} + C_{D_M} + C_{D_{C_L > C_{L_{crit}}}} + C_{D_s} + C_{D_{uc}} \quad (A.37)$$

where C_{L_s} is the lift coefficient increment due to the deflection of spoilers [42, pages 26-29], C_{D_s} is the drag coefficient increment due to symmetric spoiler

deflection [42, pages 30-31], $C_{D_{uc}}$ is the drag coefficient increment due to the extension of undercarriage [42, pages 31-32], C_{D_i} is the induced drag coefficient, $C_{D_{Z_0}}$ is the constant Reynolds number profile drag coefficient, $C_{D_{Re}}$ is the Reynolds number dependent profile drag coefficient, C_{D_M} is the drag coefficient due to Mach number, $C_{D_{C_L > C_{L_{crit}}}}$ is the increment of drag coefficient due to the lift coefficient, C_L , being greater than the critical lift coefficient, $C_{L_{crit}}$. $C_{L_{wb}}$ is defined as

$$C_{L_{wb}} = a_{1_{wb}}(\alpha_{wb} - \alpha_0) \quad (A.38)$$

$$\alpha_{wb} = \alpha + \alpha_w \quad (A.39)$$

where α_{wb} is the wing-body combination angle of attack and $\alpha_w = 1^\circ$ is the wing setting angle. C_{L_T} is defined as

$$C_{L_T} = a_{1_T}\alpha_T + a_{2_T}\delta_e \quad (A.40)$$

$$a_{1_T} = 2.4869 + 1.0314(1 - \beta_{pg}) \text{ rad}^{-1} \quad (A.41)$$

$$a_{2_T} = 1.6331 + 0.4298(1 - \beta_{pg}) \text{ rad}^{-1} \quad (A.42)$$

where a_{1_T} is the tailplane lift curve slope and a_{2_T} is the elevator effectiveness. $C_{D_i} + C_{D_{Z_0}}$ is defined as

$$C_{D_i} + C_{D_{Z_0}} = [C_{D_i} + C_{D_{Z_0}}]_{\delta f=1} + \left(\frac{[C_{D_i} + C_{D_{Z_0}}]_{\delta f=20} - [C_{D_i} + C_{D_{Z_0}}]_{\delta f=1}}{19} \right) (\delta f - 1) \quad (A.43)$$

where $[C_{D_i} + C_{D_{Z_0}}]_{\delta f=20}$ and $[C_{D_i} + C_{D_{Z_0}}]_{\delta f=1}$ are a function of C_L as presented in [42, Figure 3-7]. $C_{D_{Re}}$ is defined as

$$C_{D_{Re}} = 0.133 - 0.03134 \log(\text{Re}) + 0.001762 \log(\text{Re})^2 \quad (A.44)$$

where the Reynolds number, Re , is defined as

$$\text{Re} = \frac{\rho V c}{\eta} \quad (A.45)$$

where $c = 4$ ft is the wing chord and the dynamic viscosity, η , is defined as

$$\eta = 3.045 \times 10^{-8} \frac{T^{1.5}}{T + 110.4} \quad (A.46)$$

where T is the outside air temperature (at altitude) in kelvin. C_{D_M} is defined as

$$C_{D_M} = k_3 \left(\frac{1}{\beta_{pg} - 1} \right) + k_4 \left(\frac{1}{\beta_{pg}} - \frac{1}{\beta_{pg1}} \right) \quad (A.47)$$

where $k_3 = 0.004$, $k_4 = 0$ if $M < M_{crit}$ else $k_4 = 0.065$, M_{crit} is the aircraft Mach number in which the airflow first reaches the speed of sound, and the Prandtl-Glauert factor at the critical mach number, β_{pg1} , is defined as

$$\beta_{pg1} = \sqrt{1 - M_{crit}^2} \quad (A.48)$$

$$M_{crit} = 0.86 - 0.2C_L + 0.024C_L^2 \quad (A.49)$$

$C_{D_{C_L > C_{L_{\text{crit}}}}}$ is defined as

$$C_{D_{C_L > C_{L_{\text{crit}}}}} = k2 (C_L^2 - C_{L_{\text{crit}}}^2) \text{ if } C_L > C_{L_{\text{crit}}} \text{ else } C_{D_{C_L > C_{L_{\text{crit}}}}} = 0 \quad (\text{A.50})$$

where $k2 = 0.1$ and $C_{L_{\text{crit}}} = 0.6$

- Sideforce due to aerodynamics, Y_{aero} ,

$$Y_{\text{aero}} = \rho V S C_Y \quad (\text{A.51})$$

where the sideforce coefficient, C_Y , is defined as

$$C_Y = C_{Y_p} p \frac{b}{2} + C_{Y_v} v + C_{Y_{\delta_r}} \frac{1}{2} V \delta_r \quad (\text{A.52})$$

where $b = 18.8$ ft is the wingspan, $C_{Y_p} = -0.0215$ is the sideforce coefficient due to roll rate, $C_{Y_v} = -0.38$ is the sideforce coefficient due to lateral velocity, and $C_{Y_{\delta_r}} = 0.1176$ is the sideforce coefficient due to rudder deflection.

- Normal lift force due to aerodynamics, Z_{aero} ,

$$Z_{\text{aero}} = -\frac{1}{2} \rho V^2 S [C_{L_{wb}} \cos(\alpha) + C_D \cos(\alpha)] - \frac{1}{2} \rho V^2 S_T [C_{L_T} \sin(\alpha_T - \alpha_R)] \quad (\text{A.53})$$

- Rolling moment due to aerodynamics, L_{aero} ,

$$L_{\text{aero}} = \rho V S C_l \frac{b}{2} \quad (\text{A.54})$$

where the rolling moment coefficient, C_l , is defined as

$$C_l = (C_{l_r} r + C_{l_p} p) \frac{b}{2} + C_{l_{\delta_a}} V \delta_a + C_{l_v} V \beta + C_{l_{\delta_r}} V \delta_r \quad (\text{A.55})$$

where C_{l_r} , $C_{l_p} = -0.57$, $C_{l_{\delta_a}}$, C_{l_v} , and $C_{l_{\delta_r}} = -0.06215$ are the rolling moment coefficients due to yaw rate, roll rate, aileron deflection, lateral velocity, and rudder deflection respectively, and

$$C_{l_r} = 0.459 - 0.06 C_L \quad (\text{A.56})$$

$$C_{l_{\delta_a}} = -0.193 \left(1 - \frac{V^2}{V_{\delta_{aR}}^2} \right) \quad (\text{A.57})$$

$$C_{l_v} = -0.032 - 0.124 C_L \quad (\text{A.58})$$

where $V_{\delta_{aR}} = 630$ kts is the aileron reversal speed.

- Pitching moment due to aerodynamics, M_{aero} ,

$$M_{\text{aero}} = \frac{1}{2} \rho V^2 S C_m + \frac{1}{2} \rho V^2 S_T [-(l_T + (0.25 - h_0) \bar{c}) (C_{L_T} \cos(\alpha_T - \alpha_R)) - (T_{w1} - h_{w1}) C_{L_T} \sin(\alpha_T - \alpha_R)] \quad (\text{A.59})$$

where $\bar{c} = 4$ ft is the mean aerodynamic chord, $T_{wl} = 1.4362$ ft is the height of the tailplane from the waterline, $h_{wl} = 0$ is the height of the centre of gravity from the waterline, h_0 is the aerodynamic centre position as a percentage of the mean aerodynamic chord, and the pitching moment coefficient, C_m , is defined as

$$C_m = C_{m_{1/4}} + C_{Z_{wb}}(0.25 - h_{cg}) + C_{X_{wb}} \left(\frac{h_{wl} - c_{1/4_{wl}}}{\bar{c}} \right) \quad (A.60)$$

where $c_{1/4_{wl}} = 0$ is the height of the quarter chord from the waterline, h_{cg} is the centre of gravity position as a percentage of the mean aerodynamic chord [42, Figure 3-20], $C_{X_{wb}}$ is the wing-body combination coefficient of axial force, $C_{Z_{wb}}$ is the wing-body combination coefficient of normal force, and $C_{m_{1/4}}$ is the quarter chord pitching moment coefficient, and

$$C_{X_{wb}} = C_{L_{wb}} \sin \alpha - C_D \cos \alpha \quad (A.61)$$

$$C_{Z_{wb}} = -C_{L_{wb}} \cos \alpha - C_D \sin \alpha \quad (A.62)$$

$$C_{m_{1/4}} = C_{L_{wb}}(0.25 - h_0) + C_{m_0} \quad (A.63)$$

where C_{m_0} is the zero lift pitching moment. h_0 and C_{m_0} are defined as

$$h_0 = h_{0_{\delta_f=1}} + \left(\frac{h_{0_{\delta_f=20}} - h_{0_{\delta_f=1}}}{19} \right) (\delta_f - 1) \quad (A.64)$$

$$C_{m_0} = C_{m_{0_{\delta_f=1}}} + \left(\frac{C_{m_{0_{\delta_f=20}}} - C_{m_{0_{\delta_f=1}}}}{19} \right) (\delta_f - 1) \quad (A.65)$$

where $h_{0_{\delta_f=1}}$ and $h_{0_{\delta_f=20}} = 0.12$ are the aerodynamic centre position as a percentage of mean aerodynamic chord with flaps deflection $\delta_f = 1$ (as presented in [42, Figure 3-8]) and $\delta_f = 20$ respectively, and $C_{m_{0_{\delta_f=1}}}$ and $C_{m_{0_{\delta_f=20}}} = -0.174$ are the zero lift pitching moment with flaps deflection $\delta_f = 1$ and $\delta_f = 20$ respectively. $C_{m_{0_{\delta_f=1}}}$ is defined as

$$C_{m_{0_{\delta_f=1}}} = 0.009 - \frac{0.024}{\beta_{pg}} \quad (A.66)$$

- Yawing moment due to aerodynamics, N_{aero} ,

$$N_{aero} = \rho V S C_n \frac{b}{2} \quad (A.67)$$

where the yawing moment coefficient, C_n , is defined as

$$C_n = (C_{n_r} r + C_{n_p} p) \frac{b}{2} + C_{n_{\delta_a}} V \delta_a + C_{n_v} V \beta + C_{n_{\delta_r}} V \delta_r \quad (A.68)$$

where $C_{n_r} = -0.1545$, C_{n_p} , $C_{n_{\delta_a}}$, C_{n_v} , and $C_{n_{\delta_r}} = 0.0127$ are the yawing moment coefficients due to yaw rate, roll rate, aileron deflection, lateral velocity, and rudder deflection respectively, and

$$C_{n_p} = 0.0209 - 0.125 C_L \quad (A.69)$$

$$C_{n_{\delta_a}} = -0.015 + 0.05 C_L \quad (A.70)$$

$$C_{n_v} = C_{n_{v_{\delta_f=1}}} + \left(\frac{C_{n_{v_{\delta_f=20}}} - C_{n_{v_{\delta_f=1}}}}{19} \right) (\delta_f - 1) \quad (A.71)$$

where $C_{nv\delta_f=1}$ and $C_{nv\delta_f=20}$ are the coefficient of yawing moment due to lateral velocity with flaps deflection $\delta_f = 1$ and $\delta_f = 20$ respectively, and

$$C_{nv\delta_f=1} = 0.0855 - \frac{0.0725C_L^3}{|C_L|} \quad (\text{A.72})$$

$$C_{nv\delta_f=20} = 0.105 - \frac{0.032C_L^3}{|C_L|} \quad (\text{A.73})$$

A.3.2 Thrust Model

The Bristol Siddeley Viper Mk.201 turbojet engine has been used by the Jindivik Mk 4A UAV as a propulsion system. Fitzgerald [41, 42] presented a mathematical model of this turbojet engine as shown below; thrust, T , is defined as

$$T = T_G - D_{\text{int}} \quad (\text{A.74})$$

$$T_G = k_{\text{TG}} P \left(\frac{T_{G_{\text{nd}}} - P_{\text{intratio}}}{P_{\text{intratio}}} \right) \left(\frac{1}{144} \right) \quad (\text{A.75})$$

$$D_{\text{int}} = \frac{\dot{m}V}{g} \quad (\text{A.76})$$

where T_G is the gross thrust, $k_{\text{TG}} = 146.3$ is the gross thrust factor, P is the atmospheric pressure, $T_{G_{\text{nd}}}$ is the non dimensional gross thrust as a function of intake pressure ratio, P_{intratio} , and non dimensional engine speed, N_{nd} , as presented in [42, Figure 3-18], D_{int} is the engine intake drag, and \dot{m} is the engine mass flow. P_{intratio} is defined as

$$P_{\text{intratio}} = \frac{1}{P_{\text{rec}} P_{\text{ratio}}} \quad (\text{A.77})$$

$$P_{\text{ratio}} = (1 + 0.2M^2)^{3.5} \quad (\text{A.78})$$

$$\begin{aligned} \text{if } M > 0.5, \quad P_{\text{rec}} &= 0.9433 - 0.007(M - 0.5) \\ \text{else, } P_{\text{rec}} &= k_{P_{\text{rec}}}(0.825 + 0.00047V) \end{aligned} \quad (\text{A.79})$$

where P_{rec} is the engine pressure recovery, P_{ratio} is the pressure ratio, and $k_{P_{\text{rec}}} = 0.98$ is the intake pressure recovery factor. N_{nd} is defined as

$$N_{\text{nd}} = \frac{N1}{N1_{\text{max}}} \sqrt{\frac{288.2}{T_{\text{intake}}}} \quad (\text{A.80})$$

where $N1$ is the engine speed, $N1_{\text{max}} = 13,800$ is the engine maximum speed, and the engine intake temperature, T_{intake} , is defined as

$$T_{\text{intake}} = T(1 + 0.2M^2) \quad (\text{A.81})$$

\dot{m} is defined as

$$\dot{m} = k_{\dot{m}} k_h \frac{P_{\text{intake}}}{\sqrt{T_{\text{intake}}}} \quad (\text{A.82})$$

where $k_{\dot{m}}$ is the mass flow coefficient, k_h is the altitude correction factor as a function of engine intake pressure, P_{intake} , as presented in [42, Figure 3-17]. P_{intake} is defined as

$$P_{\text{intake}} = \frac{P}{144} (1 + 0.2M^2)^{3.5} \quad (\text{A.83})$$

$k_{\dot{m}}$ is defined as

$$\begin{aligned} &\text{if } \frac{N1}{14000} \sqrt{\frac{288.2}{T_{\text{intake}}}} > 1.03, \quad k_{\dot{m}} = 52 \\ &\text{else, } k_{\dot{m}} = 61.9 \left(\frac{N1}{14000} \sqrt{\frac{288.2}{T_{\text{intake}}}} \right) - 11.76 \end{aligned} \quad (\text{A.84})$$

In addition, the engine idle speed, $N1_{\text{idle}}$, is defined as

$$N1_{\text{idle}} = \left[0.418 + 0.34M + (0.178 - 0.15M) \frac{h}{20000} \right] N1_{\text{max}} \quad (\text{A.85})$$

A.3.3 Sensor Model

The sensors model implemented in the Jindivik Simulink model [41] includes angle of attack probe, sideslip vane, accelerometers, rate gyros, attitude gyros, static and dynamic pressure sensors, Mach meter, altimeter and velocity meters.

- An angle of attack probe is used to measure an angle of attack of the vehicle and is modelled as

$$\frac{\alpha_{\text{probe}}(s)}{\alpha_{\text{adj}}(s)} = \frac{1}{0.073s + 1} \quad (\text{A.86})$$

$$\alpha_{\text{adj}} = \alpha_{\text{cg}} - \frac{l_p q}{V} \quad (\text{A.87})$$

where α_{cg} is the angle of attack at the centre of gravity that is computed using (A.2), α_{adj} is the adjusted angle of attack due to the upwash induced by pitch rate, α_{probe} is the angle of attack output from the probe, and the distance from the centre of gravity to the probe, l_p , is defined as

$$l_p = 7.43 - h_{\text{cg}} c \quad (\text{A.88})$$

The probe output, α_{probe} , is limited in the range of $+57.3^\circ$ to -14.3° .

- An sideslip vane is used to measure a sideslip angle of the vehicle and is modelled as

$$\frac{\beta_{\text{vane}}(s)}{\beta_{\text{cg}}(s)} = \frac{1}{0.073s + 1} \quad (\text{A.89})$$

where β_{cg} is the angle of sideslip at the centre of gravity that is computed using (A.2). Note that, the factors such as offset from the centre of gravity and local aerodynamic influences are not modelled. The output of the sideslip vane is limited to $\pm 30^\circ$.

- Accelerometers are used to measure translational accelerations of the vehicle and are modelled as

$$\frac{a_{x_{acc}}(s)}{a_{x_{adj}}(s)} = \frac{a_{y_{acc}}(s)}{a_{y_{adj}}(s)} = \frac{a_{z_{acc}}(s)}{a_{z_{adj}}(s)} = \frac{34.557^2}{s^2 + 2(0.707)(34.557)s + 34.557^2} \quad (A.90)$$

$$\begin{bmatrix} a_{x_{adj}} \\ a_{y_{adj}} \\ a_{z_{adj}} \end{bmatrix} = \begin{bmatrix} a_{x_{cg}} \\ a_{y_{cg}} \\ a_{z_{cg}} \end{bmatrix} + \begin{bmatrix} -(q^2 + r^2) & (pq - \dot{r}) & (pr + \dot{q}) \\ (pq + \dot{r}) & -(p^2 + r^2) & (qr - \dot{p}) \\ (pr - \dot{q}) & (qr + \dot{p}) & -(p^2 + q^2) \end{bmatrix} \begin{bmatrix} x_{acc} \\ y_{acc} \\ z_{acc} \end{bmatrix} \quad (A.91)$$

$$\begin{bmatrix} a_{x_{cg}} \\ a_{y_{cg}} \\ a_{z_{cg}} \end{bmatrix} = \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \quad (A.92)$$

where $a_{x_{cg}}$, $a_{y_{cg}}$, and $a_{z_{cg}}$ are the X, Y, and Z-body axis translational accelerations at the centre of gravity respectively that are computed using (A.92). $a_{x_{adj}}$, $a_{y_{adj}}$, and $a_{z_{adj}}$ are the adjusted X, Y, and Z-body axis translational accelerations respectively due to the accelerometers are offset from the centre of gravity. $a_{x_{acc}}$, $a_{y_{acc}}$, and $a_{z_{acc}}$ are the X, Y, and Z-body axis translational accelerations output from the accelerometers respectively which $a_{x_{acc}}$ and $a_{y_{acc}}$ are limited to ± 4 g and $a_{z_{acc}}$ is limited to ± 10 g. x_{acc} , y_{acc} and z_{acc} are the X, Y, and Z-body axis distance from the centre of gravity to the accelerometers.

- Rate gyros are used to measure body axis roll, pitch, and yaw angular rates of the vehicle and are modelled as

$$\frac{p_{gyro}(s)}{p(s)} = \frac{90^2}{s^2 + 2(0.8)(90)s + 90^2} \quad (A.93)$$

$$\frac{q_{gyro}(s)}{q(s)} = \frac{r_{gyro}(s)}{r(s)} = \frac{200^2}{s^2 + 2(0.89)(200)s + 200^2} \quad (A.94)$$

where p , q , and r are the body axis roll, pitch, and yaw angular rates respectively that are computed using (A.11). p_{gyro} , q_{gyro} , and r_{gyro} are the body axis roll, pitch, and yaw angular rate outputs from the rate gyros which p_{gyro} is limited to $\pm 300^\circ$ while q_{gyro} and r_{gyro} are limited to $\pm 100^\circ$.

- Attitude Gyros are used to measure roll, pitch, and yaw angles of the vehicle and are modelled as

$$\frac{\phi_{gyro}(s)}{\phi(s)} = \frac{\theta_{gyro}(s)}{\theta(s)} = \frac{\psi_{gyro}(s)}{\psi(s)} = \frac{1}{0.025s + 1} \quad (A.95)$$

where ϕ , θ , and ψ are the roll, pitch, and yaw angles respectively that are computed using (A.16). ϕ_{gyro} , θ_{gyro} , and ψ_{gyro} are the roll, pitch, and yaw angle outputs from the attitude gyros which ϕ_{gyro} and ψ_{gyro} are limited to $\pm 180^\circ$ and θ_{gyro} is limited to $\pm 90^\circ$.

- Static and dynamic pressure sensors are used to measure static pressure of the atmosphere and dynamic pressures of the air flow respectively. The pressure sensors are modelled as

$$\frac{P_{press}(s)}{P(s)} = \frac{\bar{q}_{press}(s)}{\bar{q}(s)} = \frac{1}{0.025s + 1} \quad (A.96)$$

where P and \bar{q} are the static and dynamic pressures respectively that are computed following the International Standard Atmosphere [42, page 47]. P_{press} and \bar{q}_{press} are the static and dynamic pressure output from the static and dynamic pressure sensors respectively.

- Mach meter is used to measure Mach number of the air speed and is modelled as

$$\frac{M_{\text{meter}}(s)}{M(s)} = \frac{1}{0.025s + 1} \quad (\text{A.97})$$

where M is the Mach number which is computed following the International Standard Atmosphere [42, page 47]. M_{meter} is the Mach number output from the Mach meter.

- Altimeter is used to measure altitude of the vehicle and is modelled as

$$\frac{h_{\text{meter}}(s)}{h(s)} = \frac{1}{0.025s + 1} \quad (\text{A.98})$$

where h is the altitude which is computed using (A.18). h_{meter} is the altitude output from the altimeter.

- Velocity meter is used to measure true air speed of the vehicle and is modelled as

$$\frac{V_{\text{meter}}(s)}{V(s)} = \frac{1}{0.03s + 1} \quad (\text{A.99})$$

where V is the true air speed which is computed using (A.12). V_{meter} is the true air speed output from the velocity meter.

A.3.4 Actuator Model

Actuator models of elevator, aileron, rudder, trailing edge flap, thrust vectoring paddles, and undercarriage have been implemented in the Jindivik Simulink model [41]. However, in this thesis, the trailing edge flap, thrust vectoring paddles, and undercarriage are not used in which the flap angle is set to 0° and the undercarriage is in the retracted position.

- Elevator actuator dynamics are modelled as

$$\frac{\delta_{e_{\text{res}}}(s)}{\delta_{e_{\text{cmd}}}(s)} = \frac{30.74^2}{s^2 + 2(0.509)(30.74)s + 30.74^2} \quad (\text{A.100})$$

where $\delta_{e_{\text{res}}}$ and $\delta_{e_{\text{cmd}}}$ are the elevator angle command and response respectively. The rate limit is $\pm 40^\circ/\text{sec}$ and the position limit is 15° to -25° .

- Aileron actuator dynamics are modelled as

$$\frac{\delta_{a_{\text{res}}}(s)}{\delta_{a_{\text{cmd}}}(s)} = \frac{75^2}{s^2 + 2(0.59)(75)s + 75^2} \quad (\text{A.101})$$

where $\delta_{a_{\text{res}}}$ and $\delta_{a_{\text{cmd}}}$ are the aileron angle command and response respectively. The rate limit is $\pm 100^\circ/\text{sec}$ and the position limit is $\pm 40^\circ$.

- Rudder actuator dynamics are modelled as

$$\frac{\delta_{r_{\text{res}}}(s)}{\delta_{r_{\text{cmd}}}(s)} = \frac{72.1^2}{s^2 + 2(0.69)(72.1)s + 72.1^2} \quad (\text{A.102})$$

where $\delta_{r_{\text{res}}}$ and $\delta_{r_{\text{cmd}}}$ are the rudder angle command and response respectively. The rate limit is $\pm 82^\circ/\text{sec}$ and the position limit is $\pm 35^\circ$.

A.4 Open-Loop Characteristics of the Jindivik Nonlinear Model

To investigate open-loop dynamic characteristics of the Jindivik nonlinear model [41] about one particular flight condition, two major lateral and longitudinal modes of the vehicle can be determined using MATLAB functions, `trim` and `linmod`. In this section, we consider the following flight configuration: (i) fuel 100 gallons, (ii) flap angle 0° , (iii) the undercarriage in the retracted position, and (iv) wings level and constant altitude 10,000 ft and airspeed 506.3 ft/s straight flight condition. As a result of `trim` and `linmod`, the state-space system forms of these two modes are,

- (i) a longitudinal LTI model:

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0.0028 & 0.0138 & 1.8610 & -32.1739 \\ -0.1353 & -1.5112 & 504.9849 & 0.1204 \\ -0.0079 & -0.0307 & -0.7974 & 0 \\ 0 & 0 & 1.0000 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} -3.4470 & 0.00369 \\ -48.1562 & 0 \\ -28.1521 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_{\text{rpm}} \end{bmatrix} \quad (\text{A.103})$$

- (ii) a lateral LTI model:

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -0.1182 & -2.0208 & -506.3395 & 32.1739 \\ -0.0171 & -1.5279 & 1.1940 & 0 \\ 0.0132 & -0.0113 & -0.2345 & 0 \\ 0 & 1.0000 & -0.0037 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \end{bmatrix} + \begin{bmatrix} 0 & 18.5577 \\ -27.8679 & -8.9740 \\ -0.3003 & 1.0384 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (\text{A.104})$$

In addition, Table A.2 presents the mode characteristics of the determined longitudinal and lateral Dynamics. It can be seen that the Phugoid and spiral modes are an open-loop unstable. Moreover, Figures A.3-A.6 show open-loop dynamic responses of the vehicle due to the step inputs of elevator, engine speed, aileron, and rudder. Obviously, the determination of natural frequency ω_n of the short period mode from Figures A.3-b and A.3-c, that is 3.83 (rad/s), agrees quite well with the presented data in Table A.2. Figures A.5-a, A.6-c, and A.6-e also show that the open-loop transfer functions of $v(s)/\delta_a(s)$, $r(s)/\delta_r(s)$, and $\psi(s)/\delta_r(s)$ are non-minimum phase zero. Furthermore, the $\omega_n = 2.60$ (rad/s) of the Dutch roll mode that is determined from Figures A.5-a, A.6-a, and A.6-c, agrees quite well with the presented data in Table A.2.

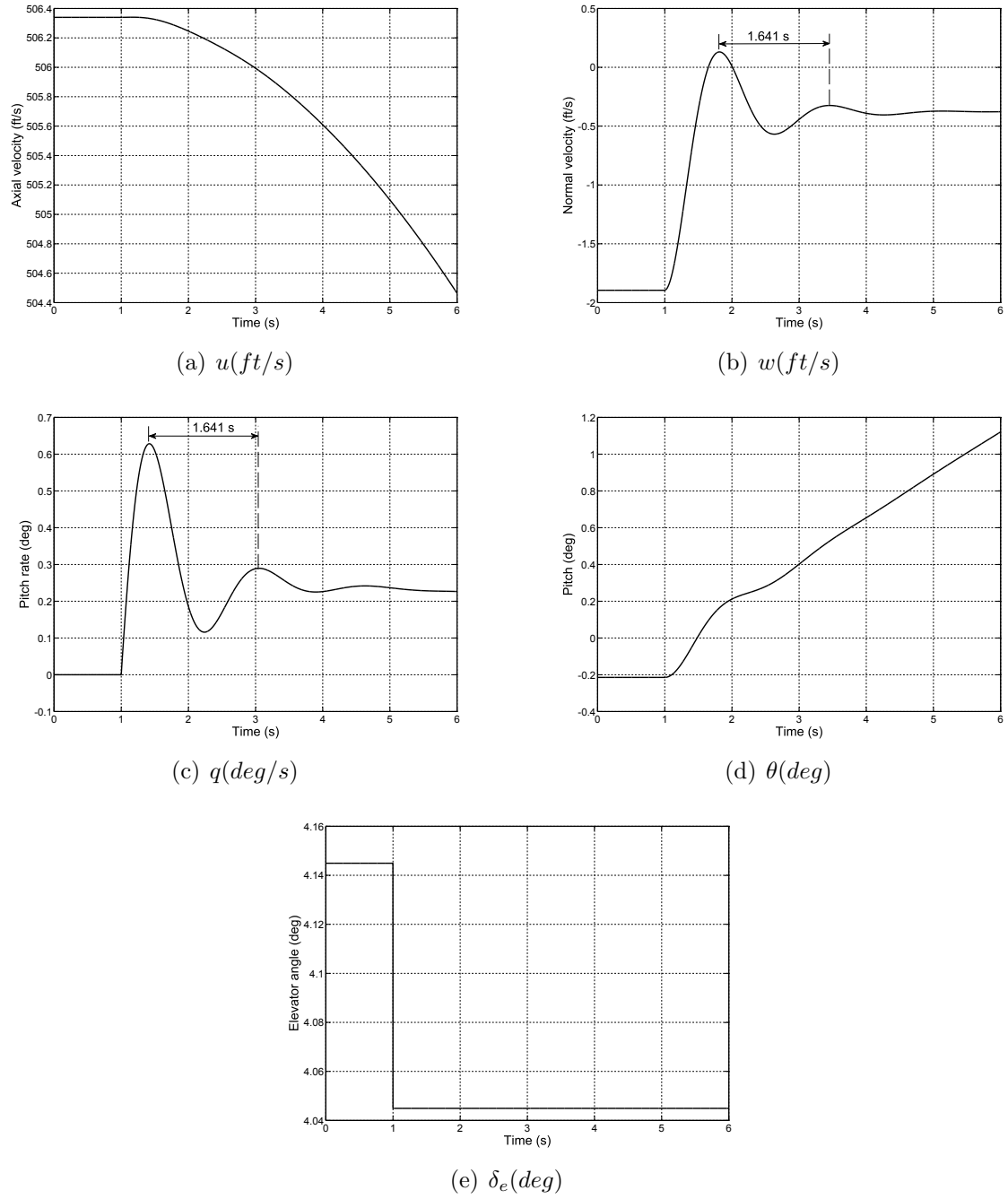


Figure A.3: Aircraft open-loop dynamic responses to 0.1 degree elevator step

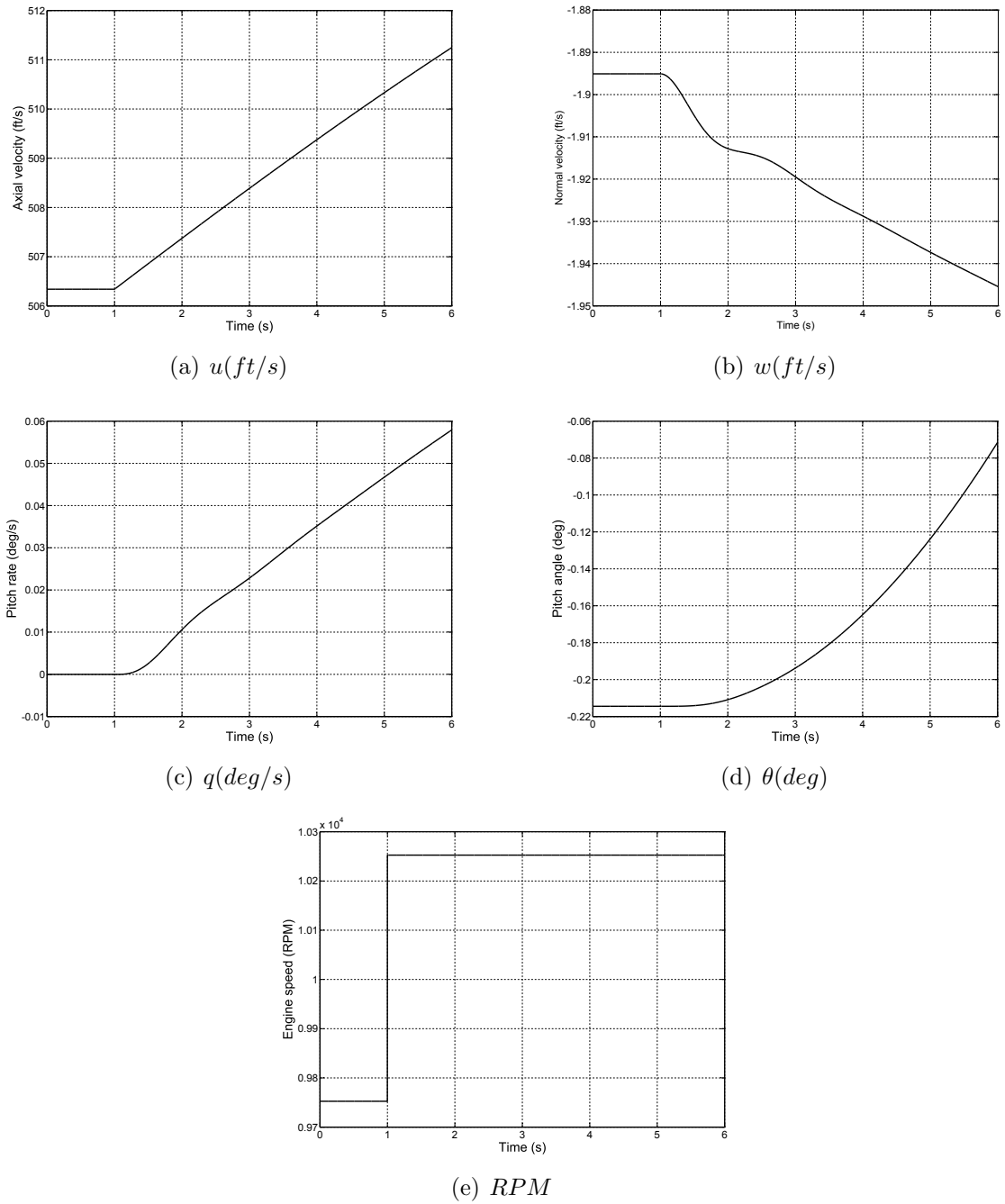


Figure A.4: Aircraft open-loop dynamic responses to 500 RPM engine speed step

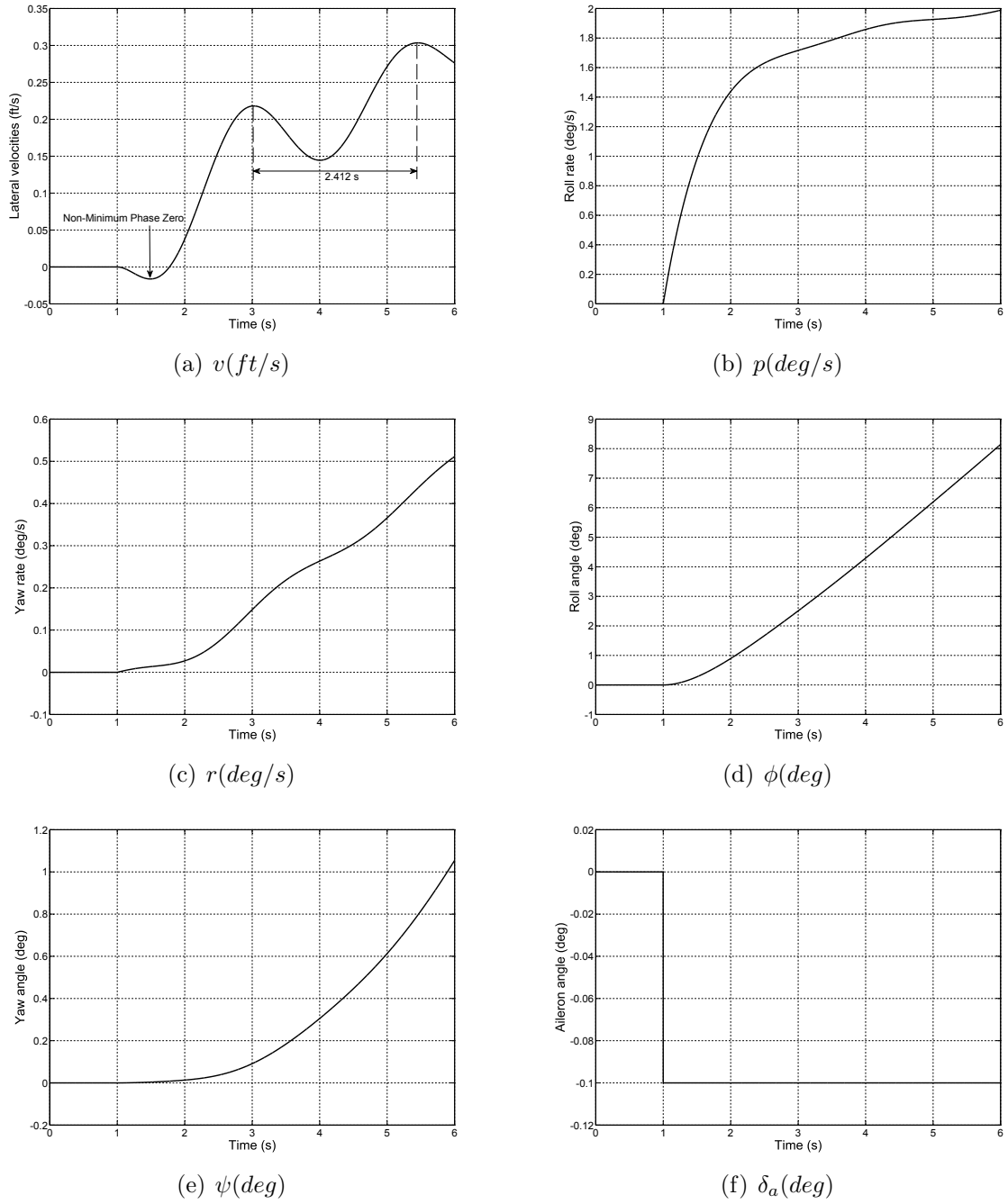


Figure A.5: Aircraft open-loop dynamic responses to 0.1 degree aileron step

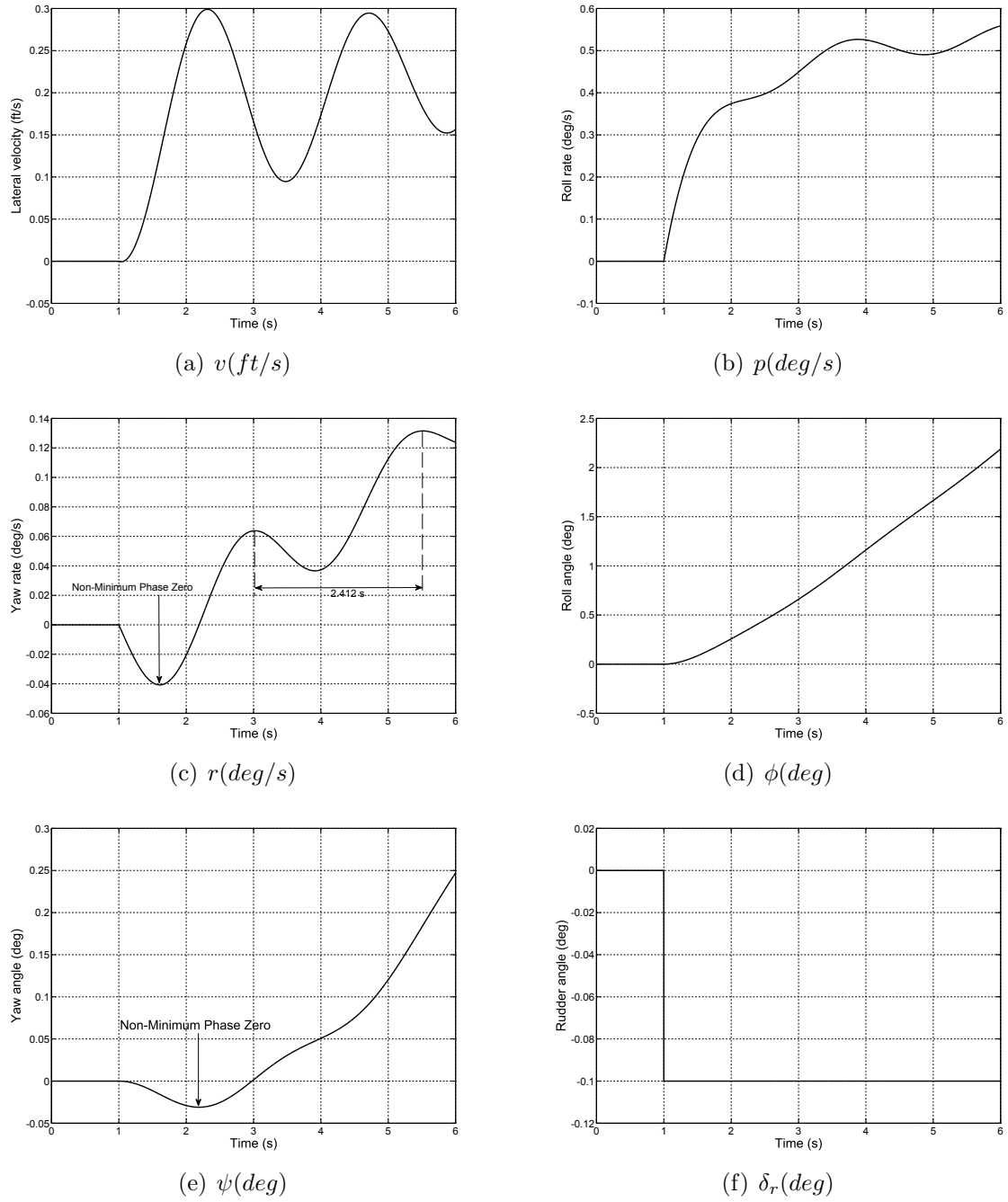


Figure A.6: Aircraft open-loop dynamic responses to 0.1 degree rudder step

Table A.2: Mode characteristics of the determined longitudinal and lateral dynamics*

Mode	Pole	Damping (ξ)	Frequency (rad/s)
Phugoid Mode	$-0.117, 0.129$	$1, -1$	$0.117, 0.129$
Short Period Mode	$-1.16 \pm 3.93i$	0.283	4.09
Spiral Mode	0.0336	-1	0.0336
Roll Mode	-1.63	1	1.63
Dutch Roll Mode	$-0.140 \pm 2.61i$	0.0534	2.62

* about wings level and constant altitude and airspeed flight condition

Appendix B

UAV aerodynamic model identification from a racetrack manoeuvre

The determination of aircraft stability and control derivatives from wind-tunnel and computational fluid dynamics data can be an expensive and time-consuming procedure. This appendix shows how linear system identification techniques can be used to obtain this information with sufficient accuracy to design a satisfactory flight control system for an aerial target of the Royal Thai Air Force. Just one flight was undertaken, using a racetrack manoeuvre, to provide the data for identification and validation. The system parameters were identified and a flight control system was designed. Hardware-in-the-loop simulation was used to perform initial tests on the controller and to test the control system hardware and software. A second flight was performed to test the resulting controller, and a satisfactory performance was obtained without the need to adjust the controller gains.

B.1 Introduction

Flight control systems are typically designed and validated using six-degree-of-freedom (6-DOF) dynamic models of the aircraft. Traditionally, the model parameters are determined using wind tunnel tests by measuring aerodynamic forces and moments introduced on an aircraft. Furthermore, aircraft moments of inertia are calculated and the aircraft engine model determined from experimental data. However these standard processes are both expensive and time-consuming, and may not be affordable or practicable for many UAV applications.

System identification techniques provide an alternative approach that can be used to estimate stability and control derivatives or aerodynamic coefficients of both manned and unmanned aircrafts from flight data [48, 53, 57, 78]. Typical tasks of comprising the system identification process are experiment design, data compatibility analysis,

model structure determination, parameter estimation, and model validation [58, 69, 78]. An important part of the experiment design is to select input waveforms that are used in the excitation of maneuver suitable for model structure determination and parameter estimation because the shape of an input signal has a major impact on the accuracy of estimated parameters from dynamic flight measurements [69, 74, 78].

A data compatibility analysis is usually applied to the measured aircraft responses in order to remove bias and scale factor errors and reconstruct the measured response data [28, 58, 78]. Having checked the data compatibility, the model structure determination [22, 76, 71] and parameter estimation [28, 53, 57, 72] can be executed in order to determine the most appropriate form of the equations to describe the measured responses and to estimate the numerical values of the coefficients appearing in the equations, respectively.

The model validation is the last step in the identification process. The identified model should have parameters that are physically reasonable values with acceptable accuracy and it should have good prediction capabilities. For these reasons, the predicted responses are usually compared with a separate set of the measured responses that is not used in the identification process. If the validation process is successful then the analysis is stopped. Otherwise, the identification process is repeated using a different equation structure or parameter estimation technique. In severe cases, the whole process may have to be performed again with a different shape of the input waveform [58, 69, 75, 78].

In practice, to obtain accurate and reliable results from system identification techniques, an aircraft is typically required to fly about a trim condition and to manoeuvre by deflecting each control surfaces deflections at a time with a suitable input waveform of appropriate amplitude [58] in order to excite certain dynamic modes. A steady wings-level flight condition is generally most suitable. The inputs can be optimal input waveforms [69, 70, 74], 3-2-1-1 input waveform [28, 73, 84], etc. Klein [58] suggests that manoeuvres about trim should not exceed angles of attack and sideslip of $\pm 5^\circ$, angular rates of $\pm 20^\circ/\text{s}$, and translational accelerations of $\pm 0.3\text{ g}$.

A number of difficulties arise when system identification techniques are applied to certain UAVs. Before the automatic control system is designed, the UAV must be flown as a remotely piloted vehicle. Without pilot-eye-view video images and telemetered data, these requirements can be difficult and time-consuming to perform by a remote ground pilot using only radio control; this is especially the case for fast, large and heavy UAVs, because the remote pilot does not receive the same motion cues that they would if onboard the aircraft. Furthermore extensive test ranges must be used which is very costly.

In this thesis, the aerodynamic and propulsion coefficients of a RTAF aerial target were identified from a racetrack manoeuvre. Thus the aircraft is always in visual contact with the remote pilot, and this reduces the range requirements. In addition, ground pilots are generally familiar with the racetrack pattern which is easy to perform. However, the aircraft is not always flying straight and level which is the normal condition for aircraft parameter identification methods. Moreover, due to the



Figure B.1: RTAF aerial target

schedule and limited budget of this work, only one flight test for identification was undertaken. The study shows that, for this vehicle, the necessary parameters can be identified with sufficient accuracy to design a satisfactory flight control system.

B.2 Flight System Configuration

The RTAF aerial target is a conventional fixed wing aircraft, shown in Figure B.1. It is powered by an AR731 rotary engine from UAV Engines Ltd. The elevator and aileron deflections and throttle setting are effected by Futaba S9206 servo motors. Note that the aerial target does not have a rudder. Its specifications are given in Table B.1.

B.2.1 Flight Control Computer

The flight computer board was developed especially for this work by the Science and Weapon System Development Center (SWSDC), RTAF, shown in Figure B.2. This board is an embedded flight computer which does not have any operating system. It is equipped with only three main chips: an Intel 80C196MH with a 16 MHz clock speed as a processor, a STMicroelectronics PSD4235G2 as a flash in-system programmable (ISP) peripheral, and a Texas Instruments TL16C554 as an asynchronous communication element (ACE) peripheral. The number of chips has been kept as low as possible so that the board is simple and reliable.

B.2.2 Avionic Instrumentation

High quality instrumentation was implemented for measuring the flight data. The sampling rate of measurement and the type of sensor output signal were the main criteria for selecting the vehicle instrumentations. As suggested by [58], the measurement sampling rate was selected as $25f_{\max}$ where f_{\max} represents the maximum



Figure B.2: SWSDC flight controller

Table B.1: RTAF aerial target specifications

Description	Details
Wing span (b)	3.21 m
Wing area (S)	1.57 m ²
Wing chord (\bar{c})	0.547 m
Stabiliser span	1.25 m
Fuselage length	2.67 m
Maximum height	0.82 m
Weight (m)	65 kg
Speed range	40 - 150 knot
Endurance	45 minute

frequency of the rigid body modes. In general, the frequencies of the rigid-body dynamic modes are below 2 Hz, which means that the sampling rate should be at least 50 Hz.

The chosen instrumentation packages are listed in Table B.6. Digital output devices were preferred because digital signals are less prone to electromagnetic interference or radio frequency interference (RFI) than the analogue. In addition, the anti-aliasing filters did not have to be designed and ADCs implemented. However the mini air-data boom provides an analogue output. A second-order low-pass Butterworth filter [56] was used to for its anti-aliasing filters. The break frequency was chosen as $5f_{\max}$ as suggested in [58].

B.2.3 Radio Telemetry

High quality radio frequency telemetries (RF Modem) were also required in this work. The RFM96W from Pacific Crest Corporation was used for the manual piloted command uplink and was configured as 412.025 MHz, 9,600 baud for RS-232 interface, and 9,600 baud for the link rate. The data packages of the piloted command were transmitted (uplink) every 80 ms. The RFM96WSS (frequency hopping

spread spectrum) from the same company was used for the flight data downlink and was configured as 902-928 MHz, 19,200 baud for RS-232 interface and 37.5 kbps raw data for the link rate. The flight data downlink packages were transmitted, displayed and recorded every 40 msec on the ground station computer, shown in Figure B.3 which also shows an overview of the flight system configuration of the RTAF aerial target.

B.2.4 Racetrack manoeuvre

The flight data (racetrack manoeuvre) that was used for the identification are shown in Figure B.4. Additional results can be seen in [31, 32]. Control surface deflection sensors were not used, however the control surface defections are generated by simulation from a linear model of the servo dynamics and the known pulse-width modulation (PWM) signals. A technique to identify a servo linear dynamic model is presented in [54]. The Futaba S9206 linear dynamic model was assumed as a first-order transfer function, $1/(\tau_a s + 1)$, where the time constant (actuator lag), τ_a , was 20 ms. The measured maximum and minimum deflections of the elevator, aileron, and throttle setting were equal to $\pm 32.4^\circ$, $\pm 19.96^\circ$ and 51° and 0° respectively. The maximum angle rate magnitude of the Futaba S9206 servo is $315^\circ/\text{s}$, so the maximum rate magnitudes of the elevator, aileron, and throttle control were determined as $126^\circ/\text{s}$, $78.75^\circ/\text{s}$, and $105^\circ/\text{s}$ respectively because the ratio of elevator, aileron, and throttle deflections to servo output deflections were 1:2.5, 1:4, and 1:3 respectively.

The altitude and velocity along the body x -axis are calculated as [1]

$$h = 145442.2 \left[1 - \left(\frac{SP}{29.92126} \right)^{0.190261} \right] \quad (\text{B.1})$$

$$u = \left[38.967854 \sqrt{T + 273.15} \right] \sqrt{5 \left(\frac{DP}{SP} + 1 \right)^{2/7} - 1} \quad (\text{B.2})$$

where h is the altitude in feet, SP is the static pressure in inches of mercury (inHg), u is the velocity along the x -body axis in knots, T is the outside air temperature (at altitude) in degrees Celsius, and DP is the dynamic pressure in inHg. To simplify (B.2), T is assumed to be constant and equal to a measurement temperature on the ground that was 27.6°C .

Although the static pressure was not calibrated to perform altitude error corrections, the calculated altitude using the static pressure data in (B.1) agreed quite well with the global positioning system (GPS) altitude as shown in Figure B.4. In addition, because of installation constraints, the mini air-data boom was installed on the wing tip of the vehicle as shown in Figure B.3. Due to upwash, such a location can affect the accuracy of the altitude and u -velocity measurements. It is a general issue for small UAVs that the air-data probes can not always be placed in aerodynamically optimal locations.

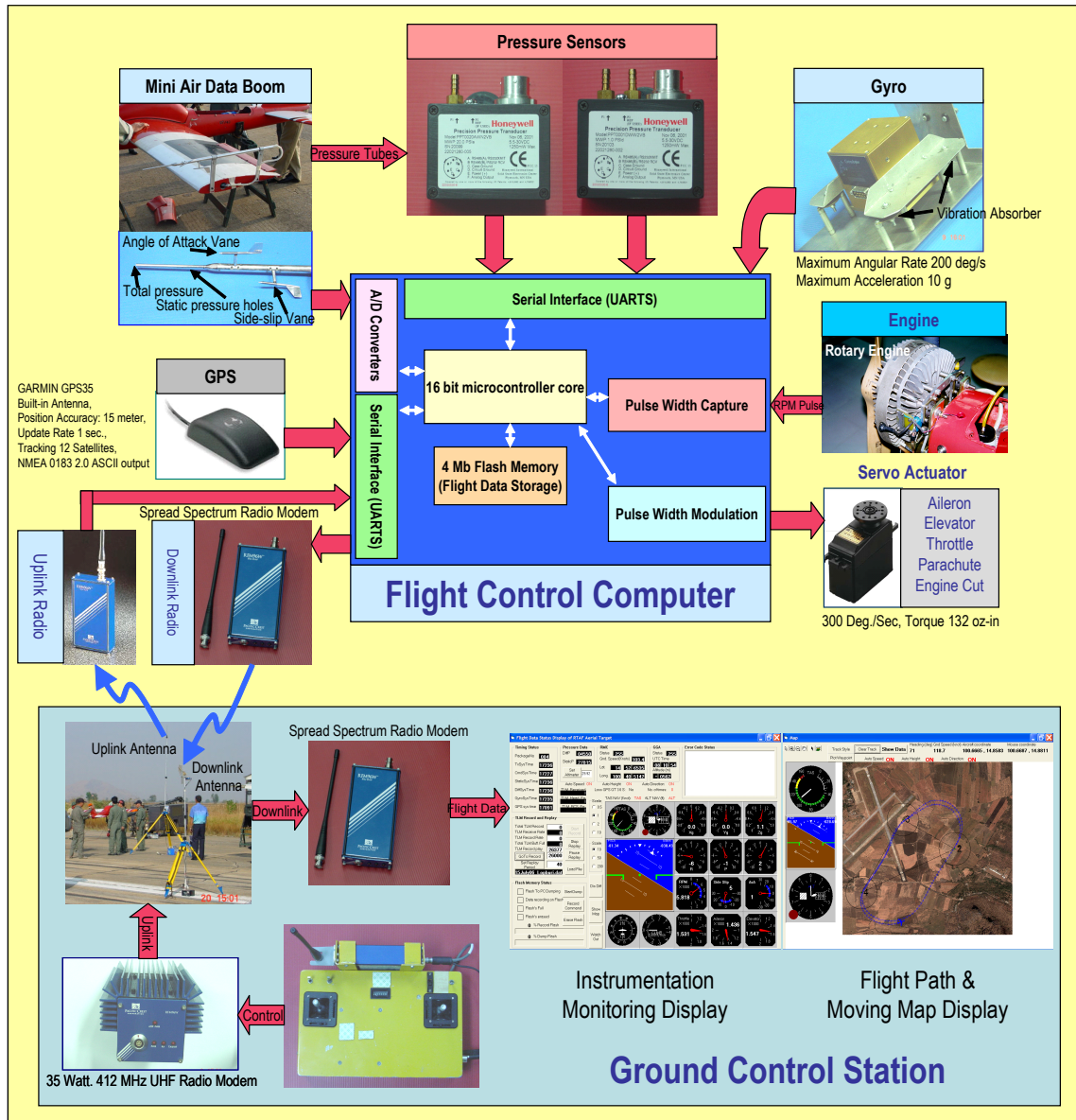
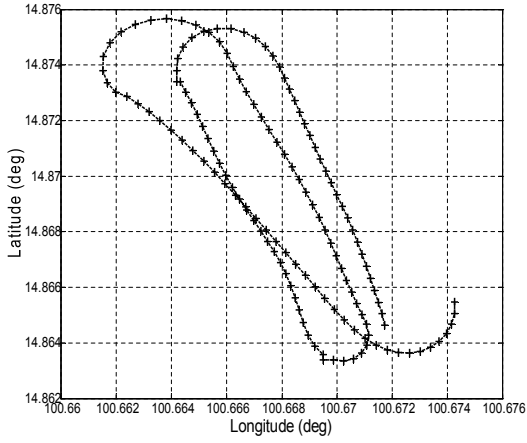


Figure B.3: Flight system configuration



(a) 2D flight path

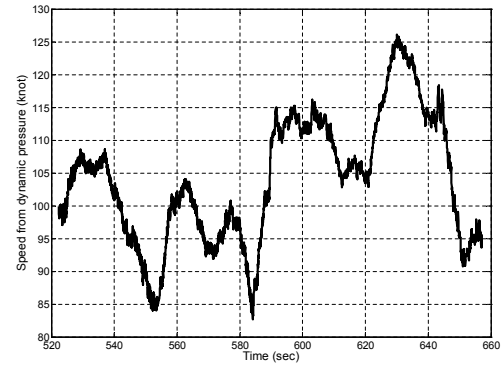
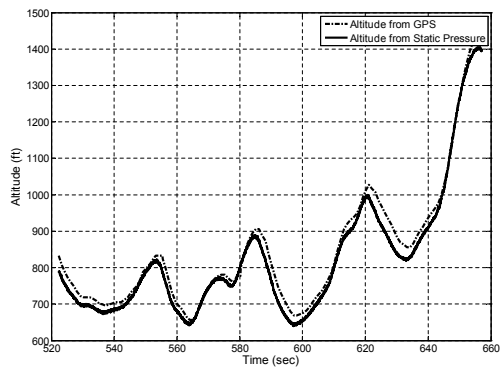
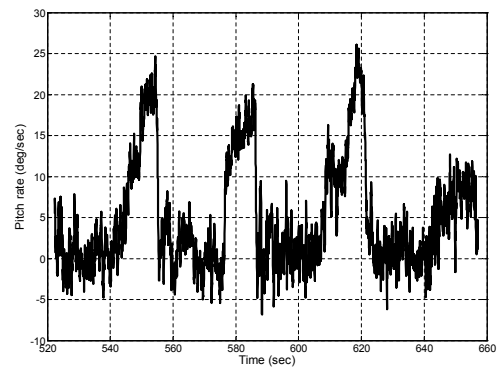
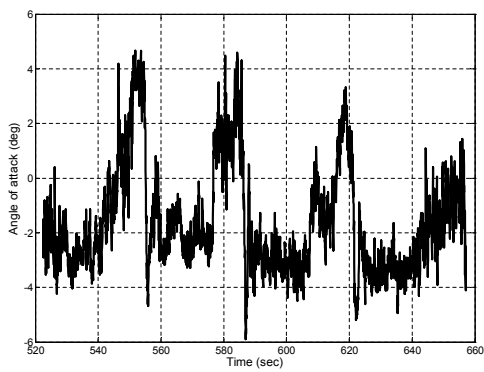
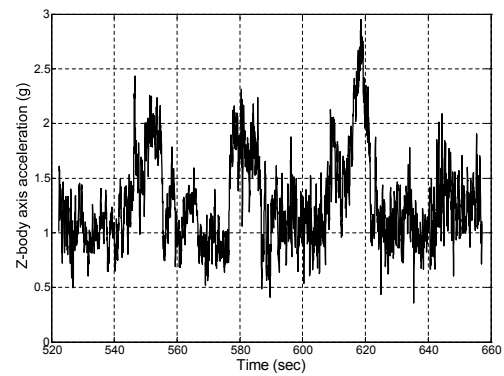
(b) u (c) h (d) q (e) α (f) a_z

Figure B.4: Flight data (racetrack pattern) for identification

B.3 Aircraft Parameter Estimation

B.3.1 Model Postulation

Based on *a priori* knowledge about the standard 6-DOF equations of motion, the aerodynamic force and moment coefficients can be calculated from flight data as shown below [77]

$$C_X = \frac{ma_x - T}{\bar{q}S} \quad (\text{B.3})$$

$$C_Y = \frac{ma_y}{\bar{q}S} \quad (\text{B.4})$$

$$C_Z = \frac{ma_z}{\bar{q}S} \quad (\text{B.5})$$

$$C_L = C_X \sin \alpha - C_Z \cos \alpha \quad (\text{B.6})$$

$$C_D = -C_X \cos \alpha - C_Z \sin \alpha \quad (\text{B.7})$$

$$C_l = \frac{I_x}{\bar{q}Sb} \left[\dot{p} - \frac{I_{xz}}{I_x} (pq + \dot{r}) + \frac{(I_z - I_y)}{I_x} qr \right] \quad (\text{B.8})$$

$$C_m = \frac{I_y}{\bar{q}S\bar{c}} \left[\dot{q} + \frac{(I_x - I_z)}{I_y} pr + \frac{I_{xz}}{I_y} (p^2 - r^2) \right] \quad (\text{B.9})$$

$$C_n = \frac{I_z}{\bar{q}Sb} \left[\dot{r} - \frac{I_{xz}}{I_z} (\dot{p} - qr) + \frac{(I_y - I_x)}{I_z} pq \right] \quad (\text{B.10})$$

Variables a_x , a_y , a_z , p , q , r , α , and \bar{q} are measured by the instrumentation shown in Table B.6. Parameters m , S , b , and \bar{c} are given in Table B.1.

The inertias I_x , I_y , and I_z are measured by a torsional pendulum experiment, i.e. the bifilar pendulum [103] and trifilar pendulum [6], using the relation

$$I = \frac{mgT^2R^2}{4\pi^2L} \quad (\text{B.11})$$

where I (kg-m²) is the measured moment of inertia, T (s) is the period of oscillation, R (m) is the distance between cables and the distance from a cable to the center of three cables in the case of bifilar and trifilar pendulums respectively, and L (m) is the cable length. In this work, the trifilar pendulum was used to measure the inertias of the aerial target and the measured values of I_x , I_y , and I_z , were equal to 16.79 kg-m², 39.44 kg-m², and 51.67 kg-m² respectively. I_{xz} was assumed zero.

The engine thrust, T (N), as a function of engine rotational speed was measured at steady engine rotational speed using a force gauge by letting the aerial target pull the force gauge only in the x -axis direction. The results of the measurement are shown in Figure B.5. A simple linear thrust model that fits the shown data was determined as

$$T = (0.0716 \times \text{RPM}) - 188.6 \quad (\text{B.12})$$

where RPM is the measurement of engine rotational speed in revolutions per minute and lies in the range 3500-7000.

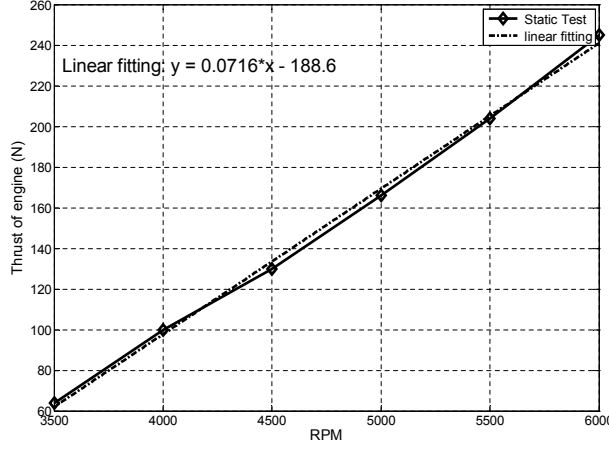


Figure B.5: Static thrust measurement result

B.3.2 Flight Data Post-Processing

All of the measured aircraft response data have to be synchronized. However, as shown in Table B.6, the sampling rates for each avionic instrumentation package are different. Hence linear interpolation was used to re-sample the data at the same frequency.

To remove the noise, e.g. process noise (atmospheric disturbance), engine vibration, and sensor noise, that lie outside the bandwidth of the data of interest, the received data were filtered using the standard MATLAB Filter Design Toolbox function [4], **butter** (i.e. Butterworth infinite impulse response digital filter), since Butterworth filters give a magnitude response that is maximally flat in the passband and is monotonic overall. In addition, they sacrifice rolloff steepness for monotonicity in the passband and stopband. In this work, the filters for V , u , ϕ , and θ were order 3 with a cutoff frequency of 2.35 Hz; the filters for a_x , a_y , a_z , p , q , r , α , and β were order 3 with a cutoff frequency of 6.35 Hz, and the filters for δ_a , δ_e , and δ_{th} were order 2 with a cutoff frequency of 4.75 Hz. A power spectral density of the filtered flight data shown in Figure B.6.

The angle of attack, sideslip angle, and translational accelerations were corrected to the center of gravity using [27, 58]

$$\alpha = \alpha_E + \frac{qx_\alpha}{V} - \frac{py_\alpha}{V} \quad (\text{B.13})$$

$$\beta = \beta_E + \frac{rx_\beta}{V} - \frac{pz_\beta}{V} \quad (\text{B.14})$$

$$g \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = g \begin{bmatrix} a_{x_E} \\ a_{y_E} \\ a_{z_E} \end{bmatrix} + \begin{bmatrix} (q^2 + r^2) & -(pq - \dot{r}) & -(pr + \dot{q}) \\ -(pq + \dot{r}) & (p^2 + r^2) & -(qr - \dot{p}) \\ -(pr - \dot{q}) & -(qr + \dot{p}) & (p^2 + q^2) \end{bmatrix} \begin{bmatrix} x_a \\ y_a \\ z_a \end{bmatrix} \quad (\text{B.15})$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} u_E \\ v_E \\ w_E \end{bmatrix} + \begin{bmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{bmatrix} \begin{bmatrix} x_{\text{air}} \\ y_{\text{air}} \\ z_{\text{air}} \end{bmatrix} \quad (\text{B.16})$$

where the subscript E denotes the measured value from the experiment. $[x_\alpha \ y_\alpha \ z_\alpha]^T$, $[x_\beta \ y_\beta \ z_\beta]^T$, $[x_a \ y_a \ z_a]^T$, and $[x_{\text{air}} \ y_{\text{air}} \ z_{\text{air}}]^T$ denote the position vectors of the angles of attack and sideslip, accelerometer, and air-data probes relative to the centre of gravity in the body axes respectively.

A data compatibility analysis should be applied to the measured aircraft responses in order to verify the data accuracy because the measured response data contains bias and scale factor errors. The purpose of data compatibility analysis is to remove the bias and scale factor errors and reconstruct the measured responses [58, 78]. The measurement equation model for aircraft sensors with typical instrumentation errors is defined as [28]

$$y_m = (1 + \lambda_y)y + b_y \quad (\text{B.17})$$

where y_m denotes the measurement of the true value of variable y , λ_y is the scale factor error, and b_y is the bias error. Following [58], the state-space form of the translational and rotational kinematics differential equations and the measured output equations used for data compatibility analysis is

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} 0 & r_E - b_r & -(q_E - b_q) & 0 \\ -(r_E - b_r) & 0 & p_E - b_p & 0 \\ q_E - b_q & -(p_E - b_p) & 0 & 0 \\ \sin \theta & -\cos \theta \sin \phi & -\cos \theta \cos \phi & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ h \end{bmatrix} + \begin{bmatrix} -g \sin \theta + g a_{x_E} - b_{a_x} \\ g \cos \theta \sin \phi + g a_{y_E} - b_{y_x} \\ g \cos \theta \cos \phi + g a_{z_E} - b_{z_x} \\ 0 \end{bmatrix} \quad (\text{B.18})$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan \theta \sin \phi & \tan \theta \cos \phi \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} p_E - b_p \\ q_E - b_q \\ r_E - b_r \end{bmatrix} \quad (\text{B.19})$$

$$V_E(i) = (1 + \lambda_V) \sqrt{u^2(i) + v^2(i) + w^2(i)} + b_V \quad (\text{B.20})$$

$$\beta_E(i) = (1 + \lambda_\beta) \tan^{-1} \left[\frac{v(i)}{u(i)} \right] + b_\beta \quad (\text{B.21})$$

$$\alpha_E(i) = (1 + \lambda_\alpha) \tan^{-1} \left[\frac{w(i)}{u(i)} \right] + b_\alpha \quad (\text{B.22})$$

$$\phi_E(i) = (1 + \lambda_\phi) \phi(i) + b_\phi \quad (\text{B.23})$$

$$\theta_E(i) = (1 + \lambda_\theta) \theta(i) + b_\theta \quad (\text{B.24})$$

$$\psi_E(i) = (1 + \lambda_\psi) \psi(i) + b_\psi \quad (\text{B.25})$$

$$h_E(i) = (1 + \lambda_h) h(i) + b_h \quad (\text{B.26})$$

where the subscript E again indicates measured values from the experiment, $\lambda_{(\cdot)}$ is the unknown scale factor error, and $b_{(\cdot)}$ is the unknown bias error. The constant unknown instrumentation error parameters in (B.18)–(B.26) were estimated using a function, `dcmp`, of the MATLAB software package SIDPAC, which is documented in and included with [58]. The function `dcmp` is an output-error parameter identification technique that is based on the principle of maximum likelihood method, for further details refer to [58]. The results are shown in Table B.2.

Klein [58] discusses the expected errors of a typical instrumentation package, and this was used to determine which bias and scale factor terms should be estimated. The

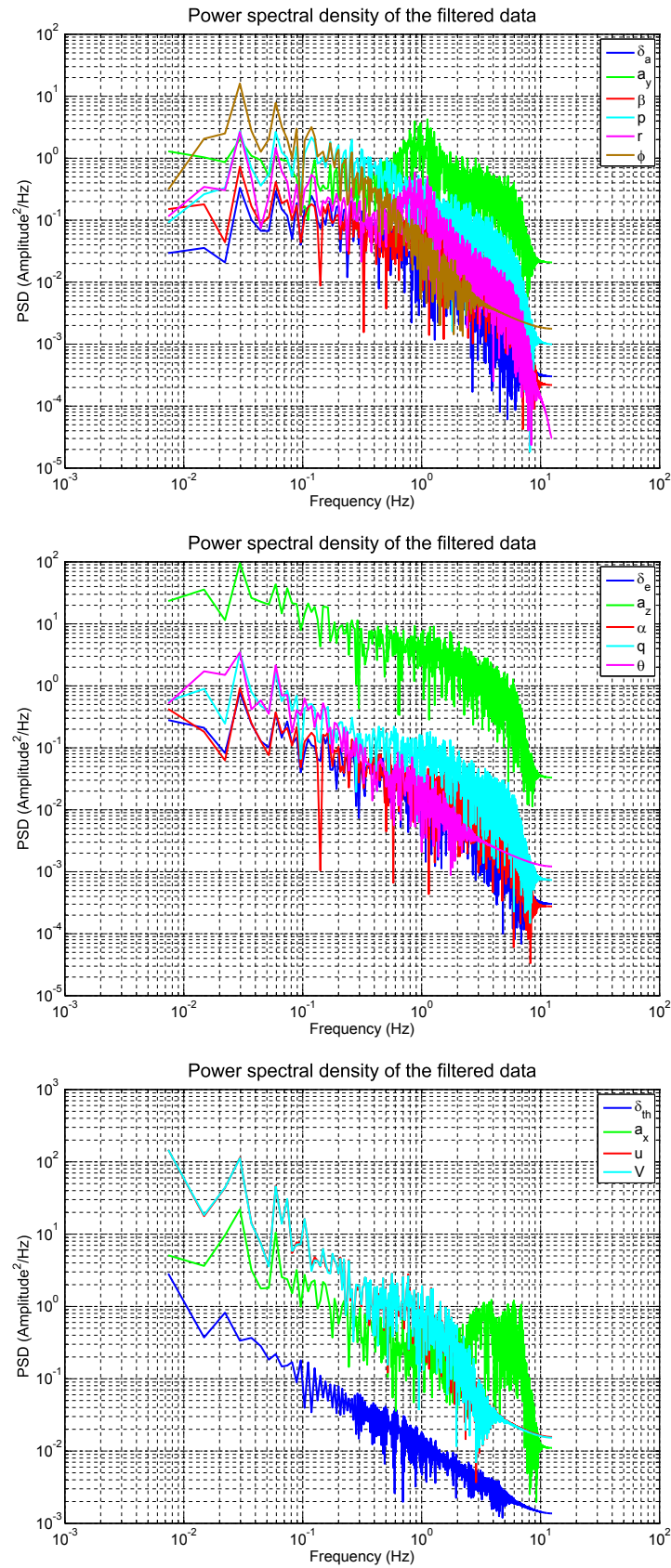


Figure B.6: Power spectral densities

Table B.2: Estimated instrumentation error parameters

b_{a_x} (m/s ²)	-0.8264 ± 0.0135
b_{a_y} (m/s ²)	-2.3694 ± 0.0129
b_{a_z} (m/s ²)	-0.5380 ± 0.0059
b_p (rad/s)	-0.0008 ± 0.0000
b_q (rad/s)	-0.0012 ± 0.0000
b_r (rad/s)	-0.0138 ± 0.0001
b_β (rad)	0.0193 ± 0.0034
b_α (rad)	0.0145 ± 0.0050
λ_ϕ	0.0071 ± 0.0003
λ_θ	0.0894 ± 0.0025
b_ϕ (rad)	-0.0309 ± 0.0003
b_θ (rad)	-0.0067 ± 0.0004

bias and scale factor of yaw angle were not estimated since there was no measurement data of the yaw angle. The selection of bias and scale factor terms in function `dcmp` were hand-tuned until the reasonable parameters, shown in Table B.2, were obtained with the best match between the measured and reconstructed responses, shown in Figure B.7. The match between the measured and reconstructed responses for V , α , and β , shown in Figure B.7–a, were not as good as the match between the measured and reconstructed responses for ϕ and θ , shown in Figure B.7–b. This is because the random measurement errors of the translational accelerations and angular rates were neglected in (B.18). Moreover, Figure B.4 also shows that all of the measured responses have a significant noise; this is especially the case for the translational accelerations. Although all of the measured responses were filtered, the process noise (i.e. atmospheric turbulence) typically resides in the same frequency band as the aircraft dynamics making it difficult to filter out post-flight without also filtering out the data of interest. Furthermore, the upwash also induced some errors on the u -velocity.

B.3.3 Equation-error Method

Although the exact forms of aerodynamic coefficients structures are not certain, typical linear model structures are suggested in [50, 58, 77]. The linear model structures that were used in this work, from [50], are

$$C_D = C_{D_0} + C_{D_\alpha} \alpha + C_{D_{\delta_e}} \delta_e \quad (\text{B.27})$$

$$C_Y = C_{Y_0} + C_{Y_\beta} \beta + C_{Y_p} \left(\frac{pb}{2V} \right) + C_{Y_r} \left(\frac{rb}{2V} \right) + C_{Y_{\delta_a}} \delta_a \quad (\text{B.28})$$

$$C_L = C_{L_0} + C_{L_\alpha} \alpha + C_{L_{\dot{\alpha}}} \left(\frac{\dot{\alpha} \bar{c}}{2V} \right) + C_{L_q} \left(\frac{q \bar{c}}{2V} \right) + C_{L_{\delta_e}} \delta_e \quad (\text{B.29})$$

$$C_l = C_{l_0} + C_{l_\beta} \beta + C_{l_p} \left(\frac{pb}{2V} \right) + C_{l_r} \left(\frac{rb}{2V} \right) + C_{l_{\delta_a}} \delta_a \quad (\text{B.30})$$

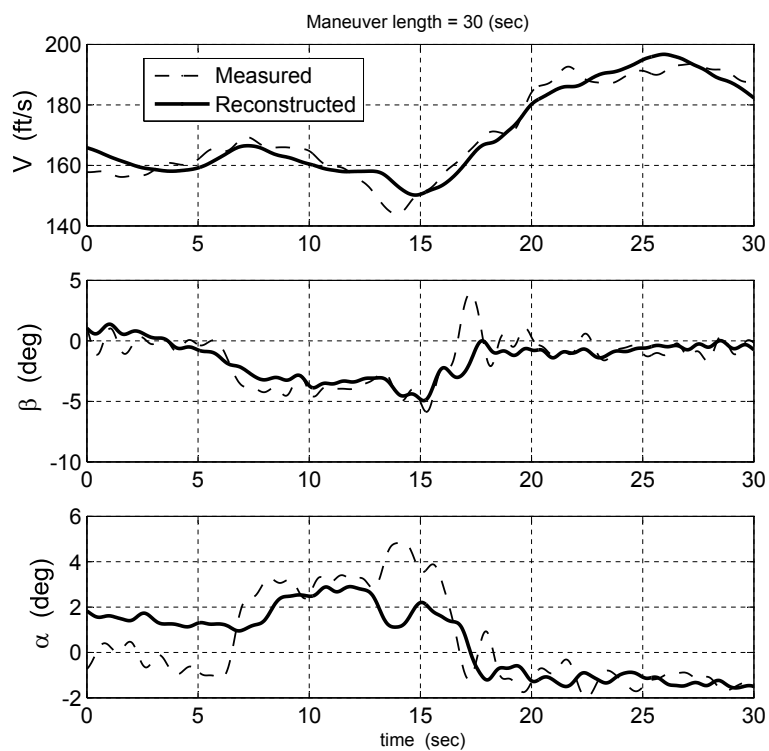
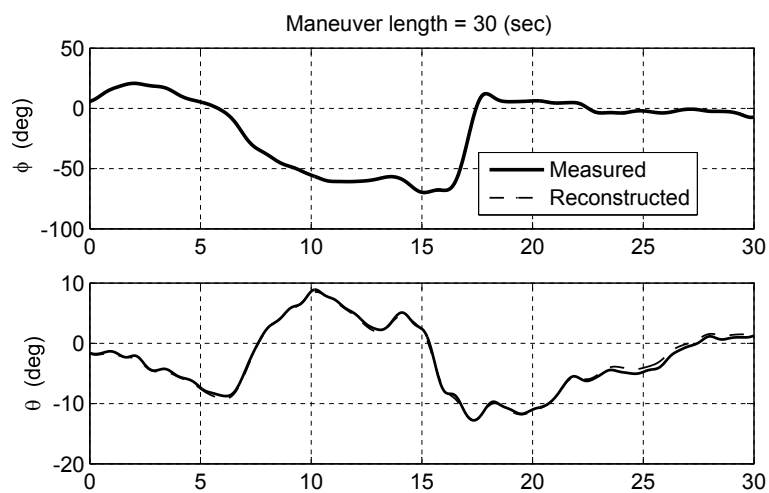
(a) V , β , and α (b) ϕ and θ

Figure B.7: Data compatibility analysis

$$C_m = C_{m_0} + C_{m_\alpha}\alpha + C_{m_{\dot{\alpha}}}\left(\frac{\dot{\alpha}\bar{c}}{2V}\right) + C_{m_q}\left(\frac{q\bar{c}}{2V}\right) + C_{m_{\delta_e}}\delta_e \quad (\text{B.31})$$

$$C_n = C_{n_0} + C_{n_\beta}\beta + C_{n_p}\left(\frac{pb}{2V}\right) + C_{n_r}\left(\frac{rb}{2V}\right) + C_{n_{\delta_a}}\delta_a \quad (\text{B.32})$$

$$RPM = C_0 + C_\alpha\alpha + C_VV + C_{\delta_{th}}\delta_{th} \quad (\text{B.33})$$

After substituting the aerodynamic coefficients that were calculated using (B.3)–(B.10) and the measured state and control variables in (B.27)–(B.33), we obtain a set of equations of the unknown aerodynamic parameters for which are solved using an equation-error method (least-squares method). For example, following [58], the least-squares problem for the lift coefficient C_L is formulated using the model structure in (B.29) as

$$\mathbf{Z} = \mathbf{X}\boldsymbol{\theta} + \mathbf{v} \quad (\text{B.34})$$

where \mathbf{Z} is an $N \times 1$ vector of values computed from (B.6), $\boldsymbol{\theta}$ is a 5×1 vector of unknown parameters, \mathbf{X} is an $N \times 5$ matrix of measurement data vectors or regressors, and \mathbf{v} is an $N \times 1$ vector of equation errors as shown below

$$\begin{aligned} \mathbf{Z} &= [C_L(1) \quad C_L(2) \quad \cdots \quad C_L(N)]^T \\ \boldsymbol{\theta} &= [C_{L_0} \quad C_{L_\alpha} \quad C_{L_{\dot{\alpha}}} \quad C_{L_q} \quad C_{L_{\delta_e}}]^T \\ \mathbf{X} &= \begin{bmatrix} 1 & \alpha(1) & \frac{\bar{c}\dot{\alpha}(1)}{2V} & \frac{\bar{c}q(1)}{2V} & \delta_e(1) \\ 1 & \alpha(2) & \frac{\bar{c}\dot{\alpha}(2)}{2V} & \frac{\bar{c}q(2)}{2V} & \delta_e(2) \\ \vdots & & & & \\ 1 & \alpha(N) & \frac{\bar{c}\dot{\alpha}(N)}{2V} & \frac{\bar{c}q(N)}{2V} & \delta_e(N) \end{bmatrix} \\ \mathbf{v} &= [v(1) \quad v(2) \quad \cdots \quad v(N)]^T \end{aligned}$$

The best estimator of $\boldsymbol{\theta}$ minimizes the sum of squared differences between the dependent variable measurements \mathbf{Z} and the model; the cost function, J , is given by

$$J(\boldsymbol{\theta}) = \frac{1}{2}(\mathbf{Z} - \mathbf{X}\boldsymbol{\theta})^T(\mathbf{Z} - \mathbf{X}\boldsymbol{\theta}) \quad (\text{B.35})$$

Differentiating (B.35) with respect to $\boldsymbol{\theta}$ gives [27],

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -\mathbf{Z}^T \mathbf{X} + \boldsymbol{\theta}^T \mathbf{X}^T \mathbf{X} \quad (\text{B.36})$$

The necessary condition for minimizing the cost is given by $\partial J(\boldsymbol{\theta})/\partial \boldsymbol{\theta} = 0$ giving the least-squares solution for the unknown parameter vector $\boldsymbol{\theta}$ as

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Z} \quad (\text{B.37})$$

The estimated parameter covariance matrix from [58] is

$$[\mathbf{C}_{ij}] = \hat{\sigma}^2 (\mathbf{X}^T \mathbf{X})^{-1}, \quad i, j = 1, 2, \dots, n_P \quad (\text{B.38})$$

where n_P is the dimension of the unknown parameter vector and

$$\hat{\sigma}^2 = \frac{(\mathbf{Z} - \mathbf{X}\hat{\boldsymbol{\theta}})^T(\mathbf{Z} - \mathbf{X}\hat{\boldsymbol{\theta}})}{N - n_P} \quad (\text{B.39})$$

Table B.3: Parameter correlation coefficient matrix (coefficients of drag, lift, and pitching moment)

	$C_{(D,L,m)_0}$	$C_{(D,L,m)_\alpha}$	$C_{(L,m)_\dot{\alpha}}$	$C_{(L,m)_q}$	$C_{(D,L,m)_{\delta_e}}$
$C_{(D,L,m)_0}$	1	0.3355	0.1262	-0.4696	-0.2884
$C_{(D,L,m)_\alpha}$	-	1	0.0614	-0.1809	0.4946
$C_{(L,m)_\dot{\alpha}}$	-	-	1	-0.2966	-0.1702
$C_{(L,m)_q}$	-	-	-	1	0.6297
$C_{(D,L,m)_{\delta_e}}$	-	-	-	-	1

The standard error of the estimated parameters is [77]

$$s(\hat{\theta}_j) = \sqrt{\mathbf{C}_{jj}}, \quad j = 1, 2, \dots, n_P \quad (\text{B.40})$$

The correlation coefficient between two estimated parameters is [27]

$$\rho_{ij} = \frac{\mathbf{C}_{ij}}{\sqrt{\mathbf{C}_{ii}\mathbf{C}_{jj}}} \quad (\text{B.41})$$

The correlation coefficient is a measure of the pair-wise correlation between the two parameters. A value of $\rho_{ij} \geq 0.9$ means that the two regressors, \mathbf{X}_i and \mathbf{X}_j , are linearly dependent and are in some way related to each other. In that case, some additional action must be taken [58].

B.3.4 Results

A single flight was undertaken lasting 1,038 sec. The flight data of the period from 522-657 sec was used for identification. Some of the measurements are shown in Figure B.4. The identification results are presented in Table B.7. Figure B.8 show how well of the match between the estimated and calculated aerodynamic coefficients using the flight data of period 560-620 sec. The correlation coefficients of these results are presented in Tables B.3 to B.5. The results were also validated by comparing the estimated aerodynamic coefficients with the values calculated by (B.3)–(B.10) using the flight data of period 860-870 sec. Some results are shown in Figure B.9. Additional results can be seen in [31, 32]. The offset in the RPM plot occurs because engine nonlinear effects are likely to be significant at low engine speeds (below 4,500 RPM), the engine speed for the identification period ranged between 5,200 RPM to 6,800 RPM, but was below 4,400 RPM for the validation period. Moreover, the regressor showing the correlation between V and δ_{th} , shown in Table B.5, shows these are linearly dependent. This could cause an inaccurate estimated parameter. Although we can not suppose the identified results are the exact aerodynamic coefficient parameters of the vehicle, these results should be reliable and accurate enough for control synthesis and analysis.

Table B.4: Parameter correlation coefficient matrix (coefficients of sideforce, rolling, and yawing moments)

	$C_{(Y,l,n)_0}$	$C_{(Y,l,n)_\beta}$	$C_{(Y,l,n)_p}$	$C_{(Y,l,n)_r}$	$C_{(Y,l,n)_{\delta_a}}$
$C_{(Y,l,n)_0}$	1	0.0872	0.3235	-0.0151	-0.3727
$C_{(Y,l,n)_\beta}$	-	1	-0.3362	-0.2669	0.0268
$C_{(Y,l,n)_p}$	-	-	1	0.2240	-0.8636
$C_{(Y,l,n)_r}$	-	-	-	1	-0.0492
$C_{(Y,l,n)_{\delta_a}}$	-	-	-	-	1

Table B.5: Parameter correlation coefficient matrix (engine speed coefficient)

	C_0	C_α	C_V	$C_{\delta_{th}}$
C_0	1	0.0035	0.0206	-0.2157
C_α	-	1	0.3702	0.2067
C_V	-	-	1	0.9272
$C_{\delta_{th}}$	-	-	-	1

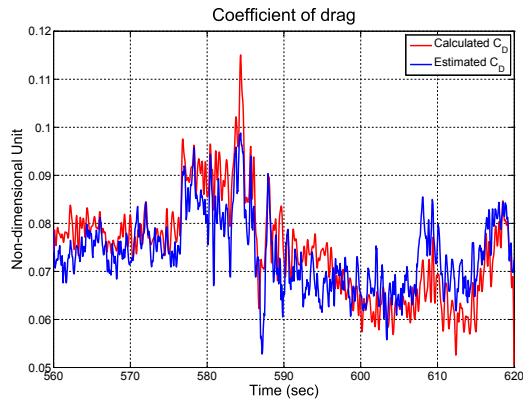
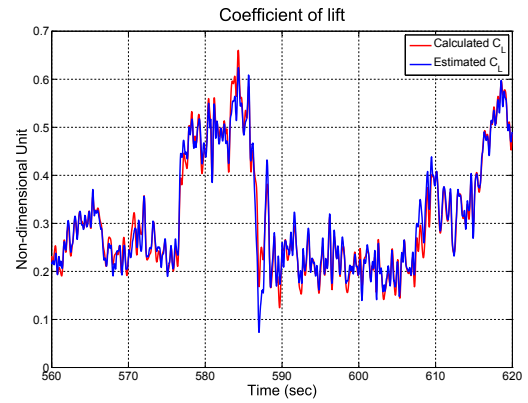
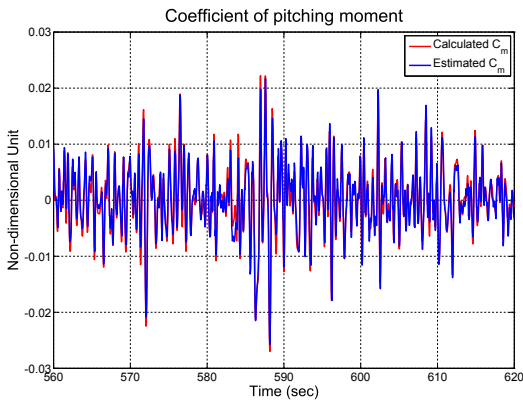
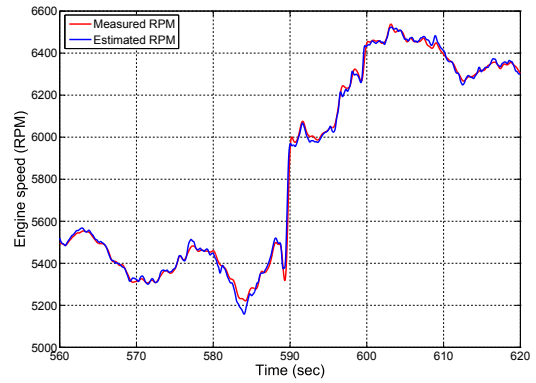
(a) C_D (b) C_L (c) C_m (d) RPM

Figure B.8: Estimation of aerodynamic coefficients

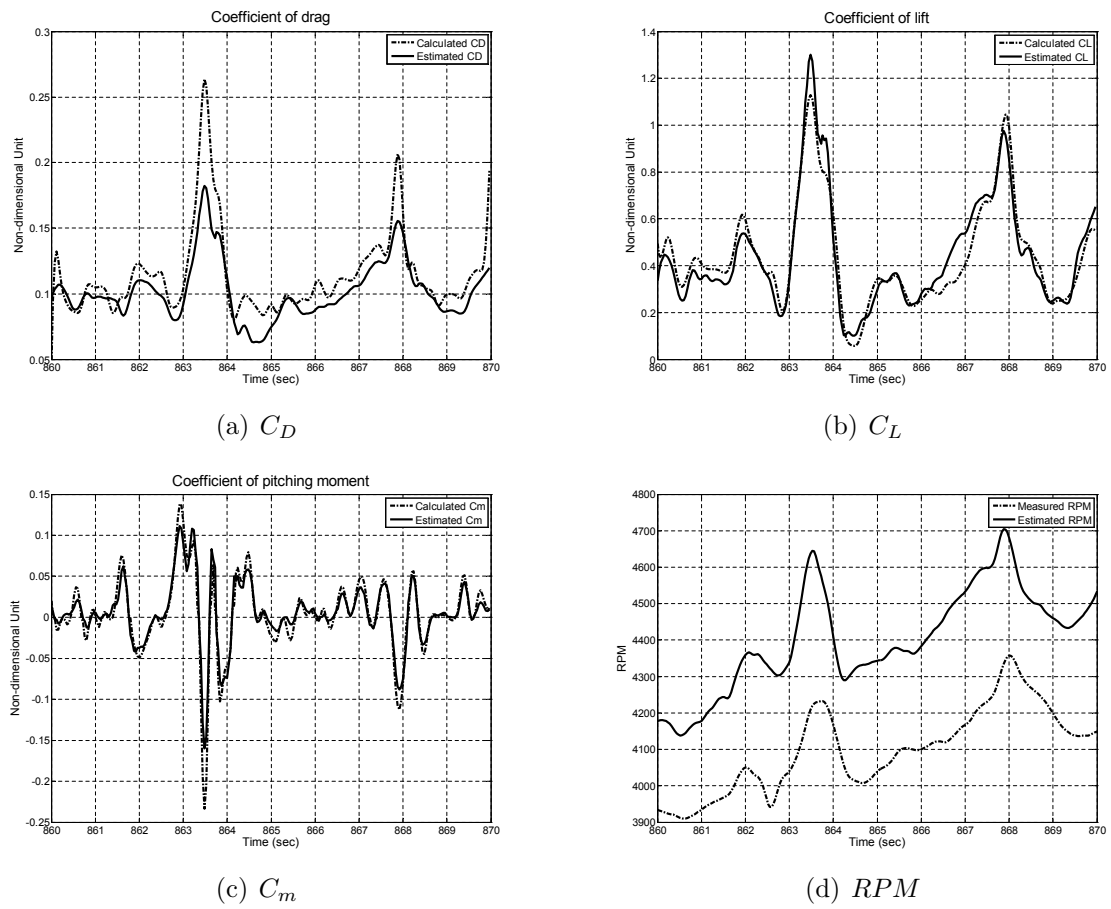


Figure B.9: Validation of aerodynamic coefficients

B.4 Design, HIL simulation and flight test

B.4.1 PID Autopilot Design

Three autopilot functions, altitude-hold, speed-hold, and GPS waypoint navigation, were designed using a proportional, integral and derivative (PID) control methodology drawn from [59]. The PID autopilot were implemented as a discrete-time controller with antiwindup. The details of its structure are shown in Figure B.10. All gain values in each loop were tuned manually, based on the identified 6-DOF nonlinear model, until the autopilot's functions performances are satisfied. The angle calculation box in Figure B.10–b calculates an angle v from a current position (Latitude, Longitude) of the vehicle to the command position (destination). The purpose of pitch and roll limiters in Figures B.10–a and B.10–b is to ensure that the aerial target will manoeuvre with small amplitude of controls, i.e. elevator, aileron, and throttle. Hence, its dynamic will not be faraway from trim condition (or small-disturbance condition).

B.4.2 Hardware-In-the-Loop (HIL) Simulation

In practice, before a real flight test, HIL simulation is often used to validate the reliability of the hardware and software of the flight control system as well as the effectiveness of the designed flight control law. The details of the HIL simulation are shown in Figure B.11. Based on the identified aerodynamic coefficients, the 6-DOF nonlinear model of the vehicle was written in MATLAB/Simulink Real-Time Workshop [2] environment with a 32-bit xPC Target system being an Intel Pentium III computer.

Two PCI-CTR05 counter/timer boards from Measurement Computing Corporation were installed inside the xPC target computer and used to capture the elevator, aileron, and throttle PWM signals that were generated from the flight control computer. Five RS-232 serial port signals were required (see Figure B.11), so a PCI-ESC-100 serial board from Quatech Inc. were also installed to provide these. For the 6-DOF nonlinear Simulink model, the xPC Target Toolbox blocks `PCI-CTR05 PWM` and `ESC-100 Quatech` [3] were used to program the PCI-CTR05 boards and PCI-ESC-100 board respectively. The 6-DOF nonlinear Simulink model was automatically compiled by VisualC and downloaded to the xPC target using the MATLAB function, `xpcexplr` [3].

The designed PID autopilot was programmed manually using the Phyton 80C196 C language development kit. The sensor interface modules of the flight computer board were programmed using assembly language. The Intel Hex-file format was the result of compiling both assembly and C program modules. This Intel Hex-file was burned to the flash in-system programmable peripheral (PSD4235G2) on the flight computer board.

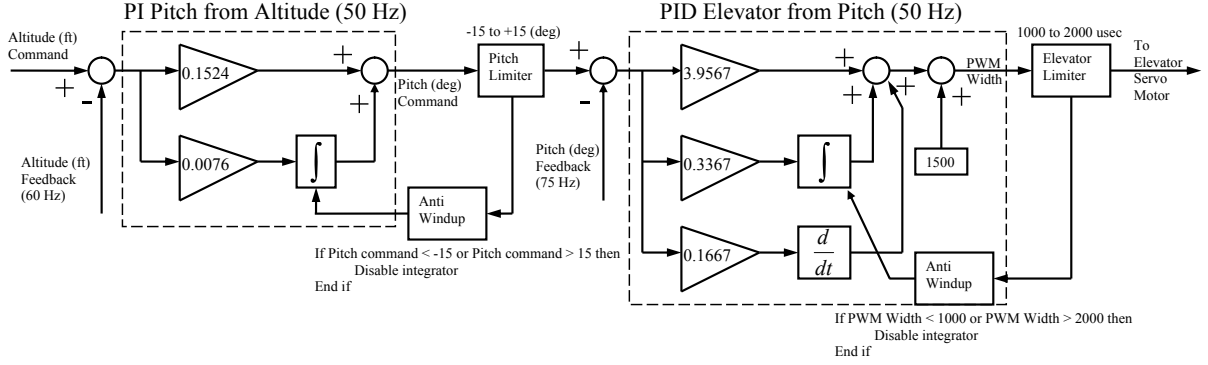
The ground station application programs, i.e. instrumentation monitoring and moving map displays, were programmed using Visual Basic 6.0. Furthermore, this HIL simulation actually can simulate flying in both manual and automatic modes. Mode selection is controlled by three switches, i.e. altitude-hold on-off, speed-hold on-off, and GPS waypoint navigation on-off, on the joystick control unit. Microsoft Flight Simulator was used for visualisation. An experienced ground pilot used the manual mode to simulate flying with the flight dynamic visualization display, shown in Figure B.11, in order to validate the identified 6-DOF nonlinear model.

B.4.3 Flight Test

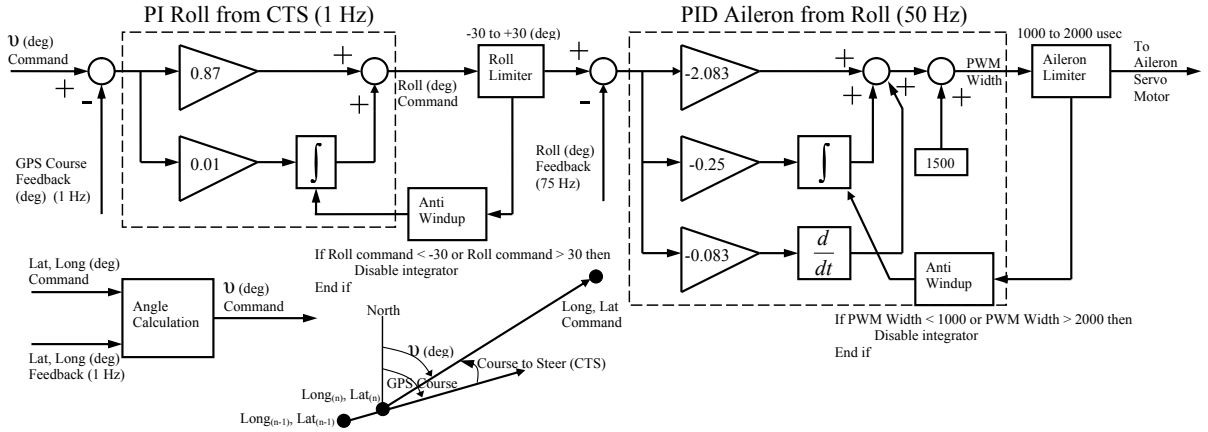
After the HIL simulation results were deemed successful and satisfactory, a real flight test was performed. The altitude command was set to 2,000 ft, the speed command to 105 knot, and three GPS waypoint navigation settings of (lat: $N14^{\circ}52.2240'$, long: $E100^{\circ}40.5360'$), (lat: $N14^{\circ}52.5120'$, long: $E100^{\circ}40.0080'$), and (lat: $N14^{\circ}51.7860'$, long: $E100^{\circ}39.5760'$). Some of the flight test data are presented in Figure B.12. It can be seen that the PID autopilot successfully performed its functions without any need to adjust the controller gains. Additional results can be seen in [31, 32].

B.5 Conclusions

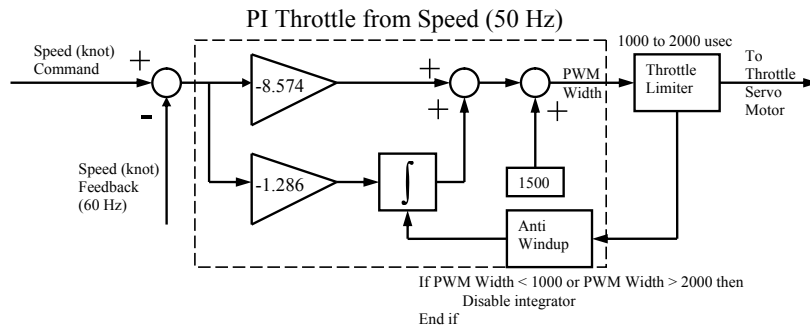
This appendix demonstrates how system identification techniques can be used for UAV control system design and development in a cost-effective manner. An ordinary piloted manoeuvre and off-trim condition flight data (racetrack manoeuvre) was studied and identified in order to estimate the aerodynamic coefficients of the RTAF aerial target. As shown by the flight test results of the PID autopilot, the identified 6-DOF non-linear model was sufficiently reliable and accurate for the design of a satisfactory control system. In this work, only two flight tests had to be undertaken. The first flight test was done to record flight data by controlling the aerial target manually for a flight duration of 12 min. The second flight test was done to validate the PID autopilot for a flight duration of 26 min. However, both flight tests were performed in one flight condition; it is envisaged that gain-scheduling will be required to cover a fuller range of flight conditions. An advanced robust gain-scheduling technique, namely linear parameter-varying control, should be employed where the details of LPV control approach are already presented in the main chapters of this thesis.



(a) Altitude-hold

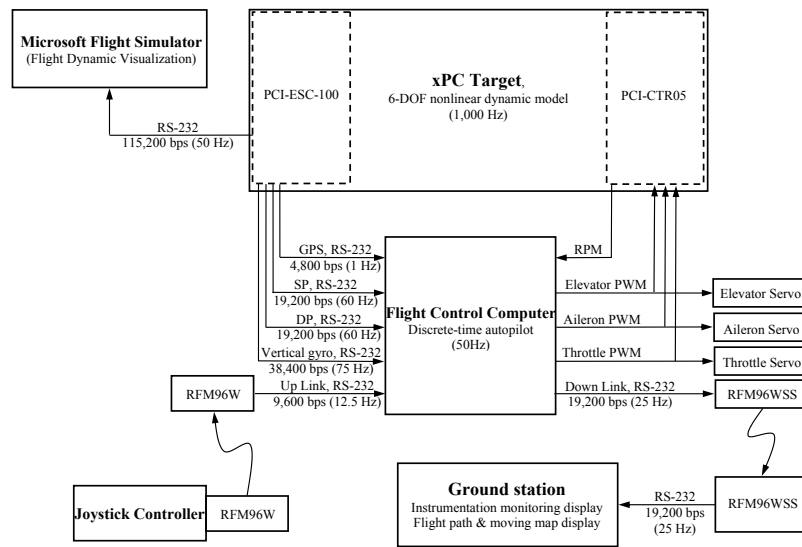


(b) GPS waypoint navigation

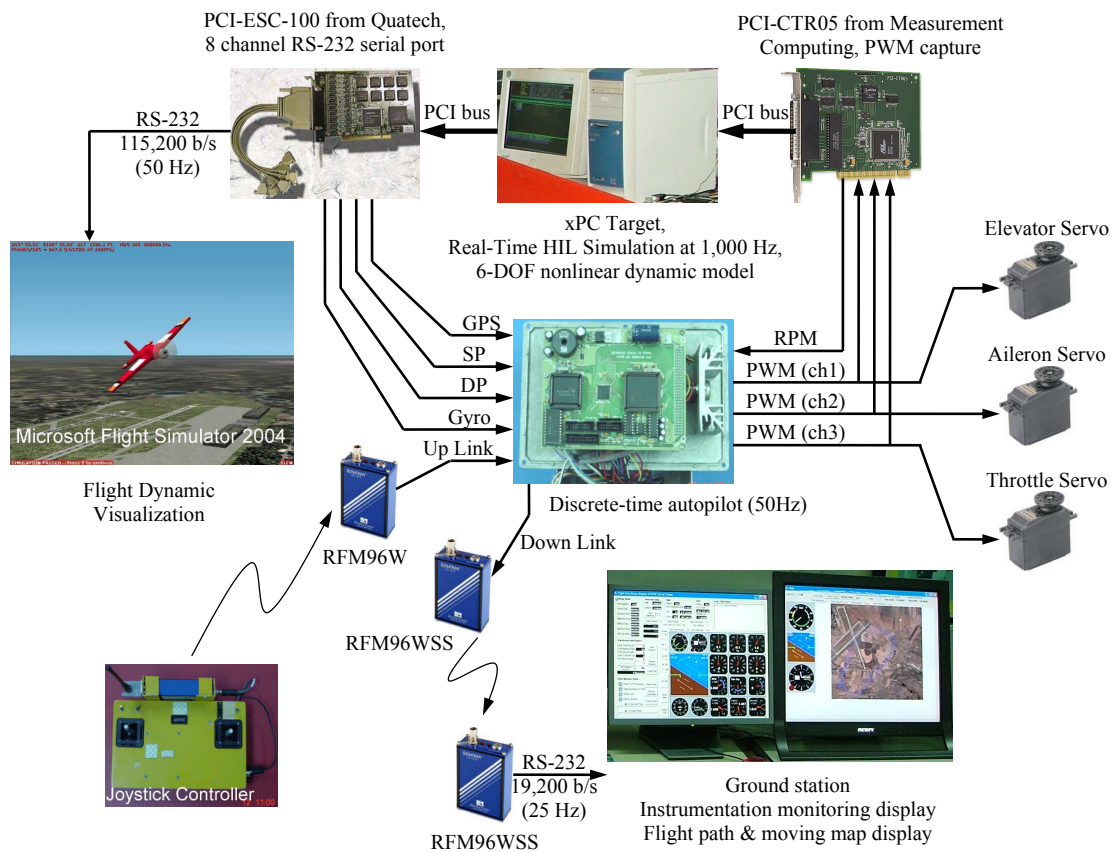


(c) Speed-hold

Figure B.10: Structure design of the autopilot

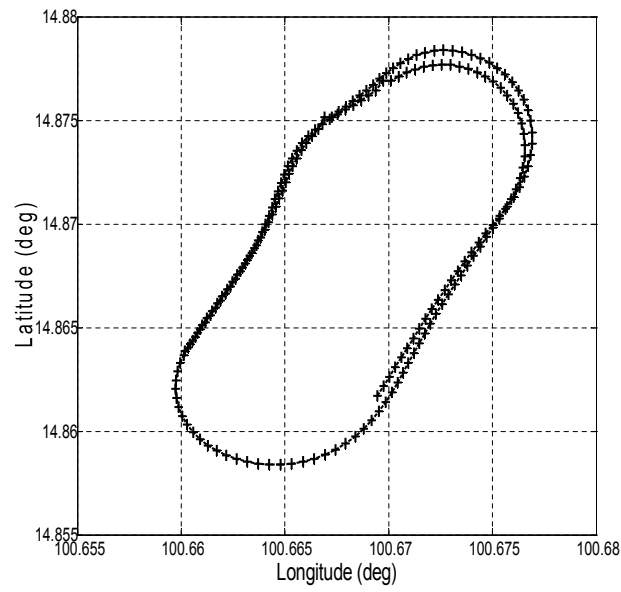


(a) Block diagram

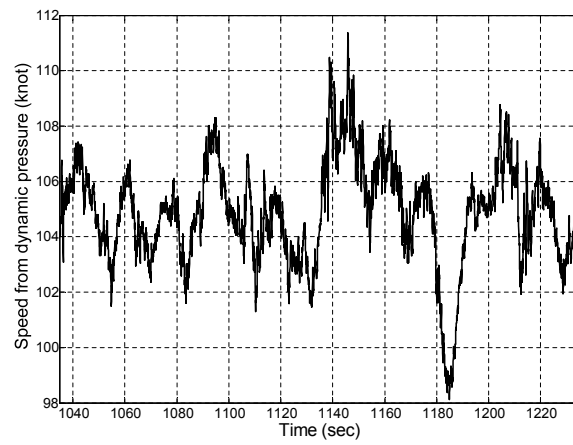


(b) Image diagram

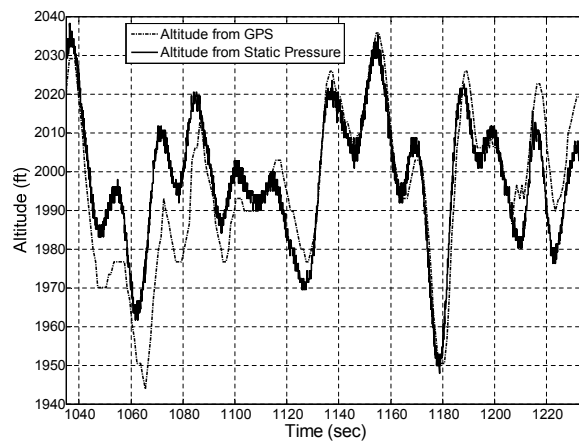
Figure B.11: Real-time hardware-in-the-loop simulation environment



(a) 2D flight path



(b) u



(c) h

Figure B.12: Validation of the PID autopilot through flight test

Table B.6: The Details of Vehicle Instrumentation

Quantity Measured Data	Sensor Model	Manufacturer	Type of Sensor Output Signal	Sampling Frequency of Measurement
$\theta, \phi, p, q, r, a_x, a_y, a_z$	VG400CA-200	Crossbow Technology, Inc.	Digital, RS-232, 38,400 bps	75 Hz
Static pressure	PPT0020AWN2VB	Honeywell International Inc.	Digital, RS-232, 19,200 bps	60 Hz
Dynamic pressure	PPT0001DWW2VB	Honeywell International Inc.	Digital, RS-232, 19,200 bps	60 Hz
α, β	100400 mini air data boom	SpaceAge Control, Inc.	Analogue, 0-5 Volt	50 Hz
Global positioning data	GPS35-LVS	Garmin Ltd.	Digital, RS-232, 4,800 bps	1 Hz
Engine speed	Ignition Module	on engine	Digital, Pulse with period varying	50 Hz

Table B.7: The Identification Results

	C_{D_0}	C_{D_α}	$C_{D_{\delta_e}}$	C_{Y_0}	C_{Y_β}	C_{Y_p}	C_{Y_r}	$C_{Y_{\delta_a}}$
value	0.076693	0.3579	-0.13474	-0.0025983	-0.064828	-0.099401	0.74232	0.084732
std. error	1.4825×10^{-7}	0.011832	0.017693	3.5998×10^{-8}	0.0033294	0.016875	0.025609	0.0042899
	C_{L_0}	C_{L_α}	$C_{L_{\delta_a}}$	C_{L_q}	$C_{L_{\delta_e}}$	C_{l_0}	C_{l_β}	C_{l_p}
value	0.26704	2.3446	-4.1801	13.631	-0.65913	-3.9032×10^{-4}	-0.0034639	-0.091252
std. error	4.1089×10^{-7}	0.032794	1.0845	2.1308	0.049038	5.5085×10^{-9}	5.0948×10^{-4}	0.0025822
	C_{l_r}	$C_{l_{\delta_a}}$	C_{m_0}	C_{m_α}	$C_{m_{\delta_e}}$	C_{m_q}	$C_{m_{\delta_e}}$	C_{n_0}
value	0.014748	0.024692	-7.2386×10^{-5}	-0.03385	-3.2026	-2.8481	-0.064212	-3.7794×10^{-5}
std. error	0.0039187	6.5645×10^{-4}	3.1908×10^{-8}	0.0025467	0.084219	0.16547	0.0038081	5.9323×10^{-9}
	C_{n_β}	C_{n_p}	C_{n_r}	$C_{n_{\delta_a}}$	C_0	C_α	C_V	$C_{\delta_{th}}$
value	0.010383	-0.01673	-0.099772	-4.7205×10^{-4}	790.59	335.53	41.426	4529.6
std. error	5.4868×10^{-4}	0.0027809	0.0042202	7.0696×10^{-4}	7.2182×10^{-4}	12.051	0.49215	50.667

This page intentionally contains only this sentence.

Appendix C

Explicit Controller Formulas for PDLF-based Gain-Scheduled H_∞ Synthesis

This appendix aims to provide a brief overview of a multi-convexity method [11] and an explicit controller formulas [43] that are used for synthesizing and constructing a PDLF-based LPV controller, respectively. The material in this appendix is taken from [8, 11] and [43].

Consider a given affine LPV plant model with state-space realization of the form (3.59). $A(\cdot)$, $B_1(\cdot)$, $C_1(\cdot)$ and $D_{11}(\cdot)$ matrices in (3.59) can be written as an affine and convex combinations in a similar manner to (3.20) and (3.22), respectively:

$$\begin{aligned} A(\theta) &= A_0 + \sum_{i=1}^n \theta_i A_i = \sum_{j=1}^r \alpha_j \hat{A}_j \\ B_1(\theta) &= B_{1_0} + \sum_{i=1}^n \theta_i B_{1_i} = \sum_{j=1}^r \alpha_j \hat{B}_{1_j} \\ C_1(\theta) &= C_{1_0} + \sum_{i=1}^n \theta_i C_{1_i} = \sum_{j=1}^r \alpha_j \hat{C}_{1_j} \\ D_{11}(\theta) &= D_{11_0} + \sum_{i=1}^n \theta_i D_{11_i} = \sum_{j=1}^r \alpha_j \hat{D}_{11_j} \end{aligned} \tag{C.1}$$

where n is the total number of the time-varying parameters, θ , $r = 2^n$ is the total number of vertices, and α_j is determined using (3.24) and (3.25).

The gain-scheduled output feedback H_∞ control problem using the parameter-dependent Lyapunov function is to compute a dynamic LPV controller, $K(\theta)$, with state-space equations of the form (6.14). $A_k(\theta, \theta)$, $B_k(\theta)$, and $C_k(\theta)$ in (6.14) are determined using (6.20)–(6.22) respectively. $X(\theta)$ and $Y(\theta)$ in (6.20)–(6.22) can also be written

as

$$\begin{aligned}
X(\theta) &= X_0 + \sum_{i=1}^n \theta_i X_i = \sum_{j=1}^r \alpha_j \hat{X}_j \\
Y(\theta) &= Y_0 + \sum_{i=1}^n \theta_i Y_i = \sum_{j=1}^r \alpha_j \hat{Y}_j \\
\dot{X}(\theta) &= \sum_{i=1}^n \dot{\theta}_i X_i = \sum_{k=1}^r \beta_k \tilde{X}_k \\
\dot{Y}(\theta) &= \sum_{i=1}^n \dot{\theta}_i Y_i = \sum_{k=1}^r \beta_k \tilde{Y}_k
\end{aligned} \tag{C.2}$$

where β_k can be determined in a similar manner to α_j using (3.24) and (3.25). Note that \tilde{X}_k and \tilde{Y}_k , $k = 1, \dots, r$, map to X_i and Y_i , $i = 1, \dots, n$, respectively in a similar manner to (6.5) and \hat{X}_j and \hat{Y}_j , $j = 1, \dots, r$, map to X_i and Y_i , $i = 0, \dots, n$, respectively in a similar manner to (6.3)

Moreover, $X(\theta)$, $Y(\theta)$, $N(\theta)$, $M(\theta)$, $\hat{A}_k(\theta)$, $\hat{B}_k(\theta)$, $\hat{C}_k(\theta)$, and $D_k(\theta)$ in (6.20)–(6.22) are determined as shown below:

Theorem C.0.1. (*Multi-convexity, [11, Theorem 5.3]*) *There exists an LPV controller (6.14) solution to the LPV control problem with guaranteed L_2 -gain performance with level γ along all possible parameter trajectories, $\forall(\theta, \dot{\theta}) \in \Theta \times \Phi$, whenever there exist symmetric matrices X_0, X_1, \dots, X_n and Y_0, Y_1, \dots, Y_n and scalars $\lambda_0, \lambda_1, \dots, \lambda_n, \mu_0, \mu_1, \dots, \mu_n$ and σ such that*

$$\begin{bmatrix} \tilde{X}_k + \hat{X}_j \hat{A}_j + \hat{A}_j^T \hat{X}_j & \hat{X}_j \hat{B}_{1j} & \hat{C}_{1j}^T \\ \hat{B}_{1j}^T \hat{X}_j & -\gamma I & \hat{D}_{11j}^T \\ \hat{C}_{1j} & \hat{D}_{11j} & -\gamma I \end{bmatrix} - \sigma \begin{bmatrix} C_2^T \\ D_{21}^T \\ 0 \end{bmatrix} \begin{bmatrix} C_2 & D_{21} & 0 \end{bmatrix} < - \left(\lambda_0 + \sum_{i=1}^n \theta_i^2 \lambda_i \right) I \tag{C.3}$$

$$\begin{bmatrix} -\tilde{Y}_k + \hat{Y}_j \hat{A}_j^T + \hat{A}_j \hat{Y}_j & \hat{Y}_j \hat{C}_{1j}^T & \hat{B}_{1j} \\ \hat{C}_{1j}^T \hat{Y}_j & -\gamma I & \hat{D}_{11j} \\ \hat{B}_{1j}^T & \hat{D}_{11j}^T & -\gamma I \end{bmatrix} - \sigma \begin{bmatrix} B_2 \\ D_{12} \\ 0 \end{bmatrix} \begin{bmatrix} B_2^T & D_{12}^T & 0 \end{bmatrix} < - \left(\mu_0 + \sum_{i=1}^n \theta_i^2 \mu_i \right) I \tag{C.4}$$

$$\begin{bmatrix} \hat{X}_j & I \\ I & \hat{Y}_j \end{bmatrix} > 0 \tag{C.5}$$

$$\begin{bmatrix} X_i A_i + A_i^T X_i & X_i B_{1i} \\ B_{1i}^T X_i & 0 \end{bmatrix} - \sigma \begin{bmatrix} C_2^T C_2 & C_2^T D_{21} \\ D_{21}^T C_2 & D_{21}^T D_{21} \end{bmatrix} \geq -\lambda_i I \tag{C.6}$$

$$\begin{bmatrix} Y_i A_i^T + A_i Y_i & Y_i C_{1i}^T \\ C_{1i} Y_i & 0 \end{bmatrix} - \sigma \begin{bmatrix} B_2 B_2^T & B_2 D_{12}^T \\ D_{12} B_2^T & D_{12} D_{12}^T \end{bmatrix} \geq -\mu_i I \tag{C.7}$$

$$\lambda_0 \geq 0, \quad \lambda_i \geq 0, \quad \mu_0 \geq 0, \quad \mu_i \geq 0 \tag{C.8}$$

for $j, k=1, 2, \dots, r$ and $i=1, 2, \dots, n$

Having determined $X(\theta)$ and $Y(\theta)$ using Theorem C.0.1, $N(\theta)$ and $M(\theta)$ can be determined from the factorization problem:

$$I - X(\theta)Y(\theta) = N(\theta)M^T(\theta) \quad (\text{C.9})$$

Algorithm C.0.2. (*Explicit Controller Formulas, [8, 43]*) Computation of $\hat{A}_k(\theta)$, $\hat{B}_k(\theta)$, $\hat{C}_k(\theta)$, and $D_k(\theta)$.

Step 1: Set $D_k(\theta) = (D_{12}^+ D_{12}) D_0(\theta) (D_{21} D_{21}^+)$, where $D_0(\theta)$ is any matrix such that $\sigma_{\max}(D_{11}(\theta) + D_{12} D_0(\theta) D_{21}) < \gamma$. This amounts to solving a Parrott problem.

Step 2: Compute the least-squares solutions of

$$\begin{pmatrix} 0 & [D_{21} & 0] \\ [D_{21}^T] & [-\gamma I & D_{cl}^T(\theta)] \\ 0 & [D_{cl}(\theta) & -\gamma I] \end{pmatrix} \begin{pmatrix} \Theta_{B_1}(\theta) \\ [*] \end{pmatrix} = - \begin{pmatrix} C_2 \\ B_1^T(\theta)X(\theta) \\ C_1(\theta) + D_{12}D_k(\theta)C_2 \end{pmatrix} \quad (\text{C.10})$$

$$\begin{pmatrix} 0 & [0 & D_{12}^T] \\ [0] & [-\gamma I & D_{cl}^T(\theta)] \\ [D_{12}] & [D_{cl}(\theta) & -\gamma I] \end{pmatrix} \begin{pmatrix} \Theta_{C_1}(\theta) \\ [*] \end{pmatrix} = - \begin{pmatrix} B_2^T \\ [(B_1(\theta) + B_2D_k(\theta)D_{21})^T] \\ C_1(\theta)Y(\theta) \end{pmatrix} \quad (\text{C.11})$$

where $D_{cl}(\theta) = D_{11}(\theta) + D_{12}D_k(\theta)D_{12}$ and $*$ denotes matrices without interest here.

Step 3: If $\pi_{21}C_2 = 0$, set $\Theta_{B_2}(\theta) = 0$. Otherwise, compute $\Theta_{B_2}(\theta)$ such that

$$\psi + C_2^T \pi_{21} \Theta_{B_2}(\theta) + \Theta_{B_2}^T(\theta) \pi_{21} C_2 < 0 \quad (\text{C.12})$$

where $\pi_{21} = I - D_{21}D_{21}^+$ and

$$\begin{aligned} \psi &= A^T(\theta)X(\theta) + X(\theta)A(\theta) + \dot{X}(\theta) + \Theta_{B_1}^T(\theta)C_2 + C_2^T\Theta_{B_1}(\theta) \\ &+ \begin{bmatrix} B_1^T(\theta)X(\theta) + D_{21}^T\Theta_{B_1}(\theta) \\ C_1(\theta) + D_{12}D_k(\theta)C_2 \end{bmatrix}^T \begin{bmatrix} \gamma I & -D_{cl}^T(\theta) \\ -D_{cl}(\theta) & \gamma I \end{bmatrix}^{-1} \begin{bmatrix} B_1^T(\theta)X(\theta) + D_{21}^T\Theta_{B_1}(\theta) \\ C_1(\theta) + D_{12}D_k(\theta)C_2 \end{bmatrix} \end{aligned} \quad (\text{C.13})$$

Similarly, set $\Theta_{C_2}(\theta) = 0$ if $\pi_{12}B_2^T = 0$. Otherwise, compute $\Theta_{C_2}(\theta)$ such that

$$\Pi + B_2\pi_{12}\Theta_{C_2}(\theta) + \Theta_{C_2}^T(\theta)\pi_{12}B_2^T < 0 \quad (\text{C.14})$$

where $\pi_{12} = I - D_{12}^+ D_{12}$ and

$$\begin{aligned} \Pi &= A(\theta)Y(\theta) + Y(\theta)A^T(\theta) - \dot{Y}(\theta) + B_2\Theta_{C_1}^T(\theta) + \Theta_{C_1}^T(\theta)B_2^T \\ &+ \begin{bmatrix} (B_1(\theta) + B_2D_k(\theta)D_{21})^T \\ C_1(\theta)Y(\theta) + D_{12}\Theta_{C_1}(\theta) \end{bmatrix}^T \begin{bmatrix} \gamma I & -D_{cl}^T(\theta) \\ -D_{cl}(\theta) & \gamma I \end{bmatrix}^{-1} \begin{bmatrix} (B_1(\theta) + B_2D_k(\theta)D_{21})^T \\ C_1(\theta)Y(\theta) + D_{12}\Theta_{C_1}(\theta) \end{bmatrix} \end{aligned} \quad (\text{C.15})$$

Step 4: Compute $\hat{A}_k(\theta)$, $\hat{B}_k(\theta)$, and $\hat{C}_k(\theta)$ as

$$\hat{C}_k(\theta) = \Theta_{C_1}(\theta) + \pi_{12}\Theta_{C_2}(\theta) \quad (\text{C.16})$$

$$\hat{B}_k(\theta) = (\Theta_{B_1}(\theta) + \pi_{21}\Theta_{B_2}(\theta))^T \quad (\text{C.17})$$

$$\begin{aligned} \hat{A}_k(\theta) &= -(A(\theta) + B_2D_k(\theta)C_2)^T + \left[(X(\theta)B_1(\theta) + \hat{B}_k(\theta)D_{21}) \quad (C_1(\theta) + D_{12}D_k(\theta)C_2)^T \right] \\ &\times \begin{bmatrix} -\gamma I & D_{cl}^T(\theta) \\ D_{cl}(\theta) & -\gamma I \end{bmatrix}^{-1} \begin{bmatrix} (B_1(\theta) + B_2D_k(\theta)D_{21})^T \\ C_1(\theta)Y(\theta) + D_{12}\hat{C}_k(\theta) \end{bmatrix} \end{aligned} \quad (\text{C.18})$$

This page intentionally contains only this sentence.

Bibliography

- [1] *U.S. Standard Atmosphere 1976*. NASA-TM-X-74335, 1976.
- [2] *Real-Time Workshop^R 7 User's Guide*. The Mathworks, Inc., Natick, MA, USA, 2008.
- [3] *xPC TargetTM 4 User's Guide*. The Mathworks, Inc., Natick, MA, USA, 2008.
- [4] *Filter Design ToolboxTM 4 User's Guide*. The Mathworks, Inc., Natick, MA, USA, 2009.
- [5] F. Amato. *Robust Control of Linear Systems Subject to Uncertain Time-Varying Parameters*. Springer-Verlag Berlin Heidelberg, 2006.
- [6] N. Anderson, B. Hagenauer, R. Erickson, and S. Bhandari. Flight-testing of a UAV aircraft for autonomous operation using piccolo II autopilot. In *AIAA paper 2008-6568*, 2008.
- [7] P. Apkarian. On the discretization of LMI-synthesized linear parameter-varying controllers. *Automatica*, 33(4):655–661, Apr 1997.
- [8] P. Apkarian and R. Adams. Advanced gain-scheduling techniques for uncertain systems. *IEEE Transactions on Control Systems Technology*, 6(1):21–32, Jan 1998.
- [9] P. Apkarian and P. Gahinet. A convex characterization of gain-scheduled H_∞ controllers. *IEEE Transactions on Automatic Control*, 40(5):853–864, May 1995.
- [10] P. Apkarian, P. Gahinet, and G. Becker. Self-scheduled H_∞ control of linear parameter-varying systems: A design example. *Automatica*, 31(9):1251–1261, 1995.
- [11] P. Apkarian and H. D. Tuan. Parameterized LMIs in control theory. *SIAM Journal on Control and Optimization*, 38(4):1241 – 1264, 2000.
- [12] G. Balas, R. Chiang, A. Packard, and M. Safonov. *Robust Control Toolbox 3: Users Guide*. The Mathworks, Inc., Natick, MA, USA, 2007.

- [13] G. J. Balas. Linear, parameter-varying control and its application to a turbofan engine. *International Journal of Robust and Nonlinear Control*, 12:763–796, 2002.
- [14] G. J. Balas, J. C. Doyle, K. Glover, A. Packard, and R. Smith. *μ -Analysis and Synthesis Toolbox 3: Users Guide*. The Mathworks, Inc., Natick, MA, USA, 2001.
- [15] P. Baranyi. TP model transformation as a way to LMI based controller design. *IEEE Transaction on Industrial Electronics*, 51(2):387–400, April 2004.
- [16] P. Baranyi. Tensor-product model-based control of two-dimensional aeroelastic system. *Journal of Guidance, Control, and Dynamics*, 29(2):391–400, May 2005.
- [17] P. Baranyi, Z. Petres, and S. Nagy. TPtool Tensor Product MATLAB toolbox. In [Online] Available: <http://tptool.sztaki.hu/>, 2009.
- [18] P. Baranyi, Z. Petres, P. Vrkonyi, P. Korondi, and Y. Yam. Determination of different polytopic models of the prototypical aeroelastic wing section by TP model transformation. *Journal of Advanced Computational Intelligence*, 10(4):486–493, 2006.
- [19] B. R. Barmish. Stabilization of uncertain systems via linear control. *IEEE Transactions on Automatic Control*, 28(8):848–850, 1983.
- [20] D. G. Bates. H_∞ control design. In *Mathematical Methods for Robust and Nonlinear Control*. Springer-Verlag Berlin Heidelberg, 2007.
- [21] D. G. Bates and I. Postlethwaite. *Robust Multivariable Control of Aerospace Systems*. Delft University Press, 2002.
- [22] V. Klein and J. G. Batterson and P. C. Murphy. Airplane model structure determination from flight data. *Journal of Aircraft*, 20(5):469–474, 1983.
- [23] G. Becker and A. Packard. Robust performance of linear parametrically varying systems using parametrically-dependent linear feedback. *Systems & Control Letters*, 23:205–215, 1994.
- [24] J.-M. Biannic, P. Apkarian, and W. L. Garrard. Parameter varying control of a high performance aircraft. *Journal of Guidance, Control and Dynamics*, 20(2):225–231, 1997.
- [25] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics, 1994.
- [26] S. Boyd and Q. Yang. Structured and simultaneous Lyapunov functions for system stability problems. *International Journal of Control*, 49:2215–2240, 1989.

-
- [27] S. D. Carnduff. *System Identification of Unmanned Aerial Vehicles*. PhD thesis, School of Engineering, Cranfield University, UK, August 2008.
 - [28] S. D. Carnduff, S. D. Erbsloeh, A. K. Cooke, and M. V. Cook. Characterizing stability and control of subscale aircraft from dynamic motion in a wind tunnel. *Journal of Aircraft*, 64(1):137–147, 2009.
 - [29] L. H. Carter and J. S. Shamma. Gain-scheduled bank-to-turn autopilot design using linear parameter varying transformations. *Journal of Guidance, Control, and Dynamics*, 19(5):1056–1063, 1996.
 - [30] S. Chumalee and J. F. Whidborne. Pole placement controller design for linear parameter varying plants. In *Proceedings of the UKACC International Conference on Control 2008*, Manchester, UK, September 2008.
 - [31] S. Chumalee and J. F. Whidborne. Experimental development of an UAV nonlinear dynamic model. In *Proceedings of the 24th Bristol International Unmanned Air Vehicle Systems (UAVS) Conference*, pages 21.1–21.12, Bristol, UK, March 2009.
 - [32] S. Chumalee and J. F. Whidborne. Identification and control of RTAF aerial target. In *Proceedings of the European Control Conference 2009*, Budapest, Hungary, August 2009.
 - [33] S. Chumalee and J. F. Whidborne. LPV autopilot design of a Jindivik UAV. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Chicago, Illinois, August 2009.
 - [34] S. Chumalee and J. F. Whidborne. Gain-scheduled H_∞ autopilot design via parameter-dependent Lyapunov functions. *Journal of Guidance, Control, and Dynamics*, 2010. (submitted).
 - [35] S. Chumalee and J. F. Whidborne. Robust flight control for uncertain affine linear parameter-varying models. *International Journal of Control*, 2010. (submitted).
 - [36] S. Chumalee and J. F. Whidborne. Unmanned aerial vehicle aerodynamic model identification from a racetrack manoeuvre. *J. Aerospace Engineering, Proc. IMechE Vol. 224 Part G*, pages 831–842, 2010.
 - [37] S. T. Clausen, P. Andersen, and J. Stoustrup. Robust control. Technical report, Department of Control Engineering, Institute of Electronic Systems Aalborg University, Denmark, 2001.
 - [38] M. V. Cook. *Flight dynamics principles*. Elsevier/Butterworth-Heinemann, 2007.
 - [39] M. Corno, S. M. Savaresi, and G. J. Balas. On linear-parameter-varying (LPV) slip-controller design for two-wheeled vehicles. *International Journal of Robust and Nonlinear Control*, pages 1–24, 2008.

- [40] E. Feson, P. Apkarian, and P. Gahinet. S -procedure for the analysis of control systems with parametric uncertainties via parameter-dependent Lyapunov functions. In *American Control Conference*, 1995.
- [41] P. Fitzgerald. *Flight Control System Design for Autonomous UAV Carrier Landing*. PhD thesis, School of Engineering, Cranfield University, UK, October 2004.
- [42] P. Fitzgerald. Unmanned combat air vehicle technology demonstrator six degree of freedom flight dynamics simulation model. Technical report, College of Aeronautics, Cranfield University, UK, April 2004. Report NFP 0401.
- [43] P. Gahinet. Explicit controller formulas for LMI-based H_∞ synthesis. *Automatica*, 32(7):1007–1014, 1996.
- [44] P. Gahinet and P. Apkarian. A linear matrix inequality approach to H_∞ control. *International Journal of Robust and Nonlinear Control*, 4:421–448, 1994.
- [45] P. Gahinet, P. Apkarian, and M. Chilali. Affine parameter-dependent Lyapunov functions and real parametric uncertainty. *IEEE Transactions on Automatic Control*, 41(3):436–442, Mar 1996.
- [46] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali. *LMI Control Toolbox: Users Guide, version 1*. The Mathworks, Inc., Natick, MA, USA, 1995.
- [47] D.-W. Gu, P. Hr. Petkov, and M. M. Konstantinov. *Robust control design with Matlab*. Springer-Verlag London Limited, 2005.
- [48] P. G. Hamel and R. V. Jategaonkar. Evolution of flight vehicle system identification. *Journal of Aircraft*, 33(1):9–28, 1996.
- [49] G. Herrmann, M. C. Turner, and I. Postlethwaite. Linear matrix inequalities in control. In *Mathematical Methods for Robust and Nonlinear Control*. Springer-Verlag Berlin Heidelberg, 2007.
- [50] J. C. Hoff. Initial evaluation of the modified step-wise regression procedure to estimate aircraft stability and control parameters from flight test data. Technical report, College of Aeronautics Reports, Cranfield University, UK, 1993.
- [51] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.
- [52] R. A. Hyde. *H_∞ Aerospace Control Design : A VSTOL Flight Application*. Springer-Verlag Berlin Heidelberg, 1995.
- [53] K. W. Iliff. Parameter estimation for flight vehicles. *Journal of Guidance Control and Dynamics*, 12(5):609–622, 1989.

-
- [54] J. S. Jang. *Nonlinear Control Using Discrete-Time Dynamic Inversion under Input Saturation: Theory and Experiment on The Stanford Dragonfly UAVs*. PhD thesis, Department of Aeronautics and Astronautics, Stanford University, USA, 2003.
 - [55] D. Jung and P. Tsiotras. Modeling and hardware-in-the-loop simulation for a small unmanned aerial vehicle. In *AIAA 2007-2768*, 2007.
 - [56] J. Karki. Active low-pass filter design. In *Texas Instruments Application Note SLOA049A*, October 2000.
 - [57] V. Klein and J. G. Batterson. Aerodynamic parameters estimated from flight and wind tunnel data. *Journal of Aircraft*, 23(4):306–312, 1986.
 - [58] V. Klein and E. A. Morelli. *Aircraft System Identification: Theory and Practice*. AIAA Education Series, AIAA, Reston, VA, USA, 2006.
 - [59] R. Lancaster. Formation flight autopilot design for the GAF Jindivik Mk 4A UAV. Master’s thesis, School of Engineering, Cranfield University, UK, September 2004.
 - [60] C. H. Lee and M. J. Chung. Gain-scheduled state feedback control design technique for flight vehicles. *IEEE Transactions on Aerospace and Electronic Systems*, 37(1):173–182, Jan 2001.
 - [61] D. J. Leith and W. E. Leithead. Counter-example to a common LPV gain-scheduling design approach. Technical report, Industrial Control Center, University of Strathclyde, UK, 1999.
 - [62] D. J. Leith and W. E. Leithead. Survey of gain-scheduling analysis and design. *International Journal of Control*, 73(11):1001–1025, 2000.
 - [63] S. Lim and J. P. How. Analysis of linear parameter-varying systems using a non-smooth dissipative systems framework. *International Journal of Robust and Nonlinear Control*, 12:1067–1092, October 2002.
 - [64] S. Lim and J. P. How. Modeling and H_∞ control for switched linear parameter-varying missile autopilot. *IEEE Transactions on Control Systems Technology*, 11(6):830–838, November 2003.
 - [65] B. Lu and F. Wu. Switching LPV control designs using multiple parameter-dependent Lyapunov functions. *Automatica*, 40:1973–1980, 2004.
 - [66] B. Lu, F. Wu, and S. W. Kim. Switching LPV control of an F-16 aircraft via controller state reset. *IEEE Transactions on Control Systems Technology*, 14(2):267–277, Mar 2006.
 - [67] J.-F. Magni, S. Bennani, and J. Terlouw. *Robust Flight Control: A Design Challenge*. Springer-Verlag, 1997.

- [68] A. Marcos and G. J. Balas. Development of linear-parameter-varying models for aircraft. *Journal of Guidance Control and Dynamics*, 27(2):218–228, 2004.
- [69] R. K. Mehra. Optimal input signals for parameter estimation in dynamic systems - survey and new results. *IEEE Transactions on Automatic Control*, 19(6):753–768, December 1974.
- [70] E. A. Morelli. *Practical Input Optimization for Aircraft Parameter Estimation Experiments*. PhD thesis, George Washington University, Joint Institute for the Advancement of Flight Systems, USA, 1990.
- [71] E. A. Morelli. Global nonlinear aerodynamic modeling using multivariate orthogonal functions. *Journal of Aircraft*, 32(2):270–77, 1995.
- [72] E. A. Morelli. Accuracy of aerodynamic model parameters estimated from flight test data. *Journal of Guidance Control and Dynamics*, 20(1):74–80, 1997.
- [73] E. A. Morelli. Advances in experiment design for high performance aircraft. In *RTO Meeting Proceedings on System Identification for Integrated Aircraft Development and Flight Testing*, pages 1–17. RTO-MP 11, Paper 8, 1998.
- [74] E. A. Morelli. Flight test of optimal inputs and comparison with conventional inputs. *Journal of Aircraft*, 36(2):389–397, 1999.
- [75] E. A. Morelli. Real-time parameter estimation in the frequency domain. *Journal of Guidance, Control, and Dynamics*, 23(5):812–818, 2000.
- [76] E. A. Morelli. Low-order equivalent system identification for the Tu-144LL supersonic transport aircraft. *Journal of Guidance Control and Dynamics*, 26(2):354–362, 2003.
- [77] E. A. Morelli. Practical aspects of the equation-error method for aircraft parameter estimation. In *AIAA Paper 2006-6144*, 2006.
- [78] E. A. Morelli and V. Klein. Application of system identification to aircraft at NASA Langley research center. *Journal of Aircraft*, 42(1):12–25, 2005.
- [79] R. C. Nelson. *Flight Stability and Automatic Control*. McGraw-Hill, 1998.
- [80] K. Z. Ostergaard, J. Stoustrup, and P. Brath. Linear parameter varying control of wind turbines covering both partial load and full load conditions. *International Journal of Robust and Nonlinear Control*, 19:92–16, 2009.
- [81] A. Packard. Gain scheduling via linear fractional transformations. *Systems & Control Letters*, 22:79–92, 1994.
- [82] A. Packard, K. Zhou, P. Pandey, and G. Becker. A collection of robust control problems leading to LMI's. In *Proceedings of the 30th Conference on Decision and Control*, pages 1245–1250, Brighton, England, December 1991.

-
- [83] P. C. Pellanda, P. Apkarian, and H. D. Tuan. Missile autopilot design via a multi-channel LFT/LPV control method. *International Journal of Robust and Nonlinear Control*, 12:1–20, 2002.
 - [84] E. Plaetschke and G. Schultz. Practical input signal design in parameter identification. In *AGARD-LS-104*, pages 1–19, 1979.
 - [85] E. Prempain and I. Postlethwaite. A feedforward control synthesis approach for LPV systems. In *Proceeding of the American Control Conference*, pages 3589–3594, Seattle, Washington, USA, June 11–13, 2008.
 - [86] E. Prempain and I. Postlethwaite. L_2 and H_2 performance analysis and gain-scheduling synthesis for parameter-dependent systems. *Automatica*, 44:2081–2089, 2008.
 - [87] W. J. Rugh and J. S. Shamma. Research on gain scheduling. *Automatica*, 36(10):1401–1425, 2000.
 - [88] C. Scherer. *The Riccati Inequality and State-Space H_∞ -Optimal Control*. PhD thesis, Universitat Wurzburg, Germany, 1990.
 - [89] C. Scherer and S. Weiland. Linear matrix inequalities in control. Technical report, Delft Center for Systems and Control, Delft University of Technology, The Netherlands, 2004.
 - [90] J. S. Shamma and M. Athans. Analysis of gain scheduled control for nonlinear plants. *IEEE Transactions on Automatic Control*, 35(8):898–907, 1990.
 - [91] J. S. Shamma and M. Athans. Guaranteed properties of gain scheduled control for linear parameter-varying plants. *Automatica*, 27:559–564, 1991.
 - [92] J. S. Shamma and J. Cloutier. Gain-scheduled missile autopilot design using linear parameter varying transformations. *Journal of Guidance, Control, and Dynamics*, 16(2):256–261, 1993.
 - [93] J.-Y. Shin. Analysis of linear parameter varying system models based on reachable sets. In *Proceeding of the American Control Conference*, pages 35–40, Anchorage, AK, USA, May 8–10, 2002.
 - [94] J.-Y. Shin, G. J. Balas, and M. A. Kaya. Blending methodology of linear parameter varying control synthesis of F-16 aircraft system. *Journal of Guidance, Control, and Dynamics*, 25(6):1040–1048, 2002.
 - [95] W. Tan, A. Packard, and G. Balas. Quasi-LPV modeling and LPV control of a generic missile. In *Proceedings of the American Control Conference*, pages 3692–3696, Chicago, IL, 2000.
 - [96] F. Wang and V. Balakrishnan. Improved stability analysis and gain-scheduled controller synthesis for parameter-dependent systems. *IEEE Transactions on Automatic Control*, 47(5):720–734, May 2002.

- [97] F. Wu. A generalized LPV system analysis and control synthesis framework. *International Journal of Control*, 74(7):745–759, May 2001.
- [98] F. Wu and K. Dong. Gain-scheduling control of LFT systems using parameter-dependent Lyapunov functions. *Automatica*, 42:39–50, 2006.
- [99] F. Wu, A. Packard, and G. Balas. LPV control design for pitch-axis missile autopilots. In *Proceedings of the 34th Conference on Decision and Control*, pages 188–193, New Orleans, LA, 1995.
- [100] F. Wu, A. Packard, and G. Balas. Systematic gain-scheduling control design: A missile autopilot example. *Asian Journal of Control*, 4(3):341–347, Sep 2002.
- [101] F. Wu, X. H. Yang, A. Packard, and G. Becker. Induced L_2 -norm control for LPV systems with bounded parameter variation rates. *International Journal of Robust and Nonlinear Control*, pages 983–998, 1996.
- [102] P. Yan and H. Ozbay. On switching H_∞ controllers for a class of linear parameter varying systems. *Systems & Control Letters*, 56:504–511, 2007.
- [103] S. Yoon, S.-K. Lee, J.-Y. Kong, and K.-W. Nam. Development of a HLA-based simulation game for RC airplanes and UAVs. In *AIAA paper 2004-4804*, 2004.
- [104] K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control*. Englewood Cliffs, NJ: Prentice-Hall, 1995.