

University of Strathclyde
Department of Mechanical and Aerospace Engineering

Nonlinear Feedback Controller Design Methods for Actuator and
Sensor Performance Limited Systems

Joseph Brindley B.Eng.

A thesis presented in fulfillment of the requirements
for the degree of Doctor of Philosophy

February 2014

Acknowledgments

I would firstly like to express my gratitude to the Scottish Funding Council and to the Building Research Establishment for providing financial support during the period of research.

I am extremely grateful to my supervisor Dr John Counsell, whose continual support, guidance and vision have made conducting this research a pleasurable and enriching experience. Without his influence I would not be set on the path I am today.

Many thanks to Dr Malcolm Macdonald for providing invaluable assistance and support whenever it was needed.

I would like to thank Dr John Holden of the Building Research Establishment for providing his expert advice and guidance throughout my research.

Thanks to John Pearce, who was always willing to share his invaluable knowledge on modelling and simulation with me.

Thanks to my colleagues in the BRE centre and throughout the department of Mechanical and Aerospace Engineering. Your support, advice and friendship made completing this work all the more enjoyable.

Finally, I would like to thank my family. Without their support I would not have been able to complete this thesis.

Abstract

This work describes the development of high performance, nonlinear controller design methodologies which aim to achieve high performance control with systems that have limited or low performance actuators and sensors.

The first class of actuator constraints are actuators or sensors with slow dynamics with respect to the desired controller response. In order to achieve high performance control with these systems a nonlinear control algorithm was to be designed that provided more robust performance without compromising the controller's speed of response.

The second class of problem involves actuators or sensors that are constrained by absolute limits. These are prevalent in almost all real systems and can be categorised as rate and amplitude limits. A control algorithm was to be designed that is able to operate for prolonged periods on either rate or amplitude limits without performance degradation. This was achieved by using a dynamic, controller output limiting design which aims to prevent the controller output from exceeding specified rate or amplitude limits.

These controller designs were applied to control problems in aerospace and energy systems. Specifically, automatic flight control and automatic internal climate control. Both case studies involve control systems that are highly constrained by their actuation systems. In the case of internal climate control the actuation systems have very large inertias and there can be significant sensor delays. For flight control, the power of the actuator's driving control surfaces is heavily constrained as well as the physical deflection limits of the surfaces. By employing these control algorithms, the control performance and robustness of these systems can be significantly improved. This was demonstrated by simulations of heating and ventilation control of a modern office building and a missile flight control system.

The contribution to knowledge, detailed in this thesis, is the development of nonlinear controller design methodologies which provide targeted solutions to some of the most widespread control problems encountered across a wide range

of applications; namely, the problem of achieving high performance control with low performance, or limited, actuation or sensor systems.

Table of Contents

Acknowledgments	ii
Abstract	ii
Table of Contents	v
List of Figures	viii
List of Tables	xiv
Nomenclature	xvi
1. Introduction	1
1.1 Motivation	1
1.2 Defining High Performance Control	3
1.3 The Challenge of Achieving High Performance Control	4
1.3.1 Accessing the Maximum Available Energy	5
1.3.2 Using Energy Efficiently	7
1.4 Thesis Aims	8
1.5 Thesis Methods and Structure	9
1.5.1 Outline of thesis chapters.....	10
Chapter 1 – Introduction	10
Chapter 2 – High Performance Control Methods	10
Chapter 3 – Variable Transient Response.....	11
Chapter 4 – Rate Actuated Inverse Dynamics	11
Chapter 5 – Case Study 1: Heating and Ventilation System	12
Chapter 6 – Case Study 2: Missile Flight Control	12
Chapter 7 – Conclusions and Further Work	13
2. An Overview of High Performance Control Methods	13
2.1 A High Performance Controller Framework	13
2.1.1 Accessing maximum available energy	14
2.1.2 Energy Efficiency.....	14
2.2 An Overview of High Performance Controller Design Methods	15
2.2.1 Relay and High Gain Control.....	15
2.2.2 Variable Structure Control and Sliding Mode Control	16

2.2.3 Optimal Control Methods.....	17
2.2.4 Inverse Dynamics.....	18
2.2.5 Controller Framework Selection.....	20
2.3 Robust Inverse Dynamics Estimation (RIDE) Design Method	21
3. Variable Transient Response.....	25
3.1 Introduction.....	25
3.1.1 Controller response and phase margins.....	25
3.2 Transient Response Shaping	27
3.2.1 Target Dynamics.....	30
3.3 Quasi-linear Equivalents	33
3.3.1 Exponential Input Describing Function (EIDF).....	34
3.3.2 Average of the Nonlinearity over the Error Space	36
3.4 Comparison of Approximate Linear Equivalents to Real Nonlinear Gain	37
3.4.1 Exponential Input, Variation of X	37
3.4.2 Exponential Input, Variation of Y	38
3.4.3 Exponential Input, Variation of E	40
3.4.4 Closed Loop Response, Variation of X	41
3.4.5 Closed Loop Response, Variation of Y	43
3.4.6 Closed Loop Response, Variation of E	44
3.5 VTR Controller Synthesis.....	45
3.5.1 Setpoint Dependence and Selection of e_{max}	48
3.5.2 Setpoint normalization	51
3.6 Design Process – Illustrative Example	52
3.7 Conclusions	57
4. Rate Actuated Inverse Dynamics	61
4.1 Introduction.....	61
4.2 State-Space Transform	64
4.3 Linear Stability Analysis	67
4.4 Sliding Mode Control	72
4.4.1 Definition of the Switching Surfaces	72
4.4.2 Existence of the Sliding Mode	73
4.4.3 Breaking the Sliding Mode.....	77
4.5 Variable Structure Controller Design for Rate and Amplitude Limits.....	79
4.5.1 Rate Limits	79
4.5.2 Amplitude Limits.....	80
4.5.3 Influence of controller gains.....	81
4.5.4 Implementation of Switching Logic.....	82
4.6 Robustness of the VSC Design	84
4.7 Pole Placement	85
4.8 Conclusions	87
5. Heating and Ventilation Control Case Study.....	90
5.1 Introduction.....	90
5.2 Nonlinear Building Energy System Mathematical Model.....	92

5.2.1 Thermodynamics	92
5.2.2 Actuation Systems	95
5.2.3 State-Space Representation	96
5.3 Linearised Building Energy System Mathematical Model	100
5.4 Modern Office Case Study.....	101
5.4.1 Simulated zone properties	102
5.5 Controller Design	103
5.5.1 Benchmark – Proportional Integral Control	103
5.5.2 RIDE Controller Design	104
5.6 Simulation.....	108
5.6.1 Simulation Parameters	109
5.6.2 Simulation Results – PI and RIDE	109
5.6.3 VTR Controller Design.....	117
5.6.4 Simulation Results - VTR.....	119
5.6.5 Results Discussion.....	124
5.7 Conclusions	132
6. Flight Control Case Study.....	134
6.1 Introduction	134
6.2 Missile Mathematical Model.....	136
6.2.1 Aerodynamics	136
6.2.2 Control Surface Actuation	139
6.3 Missile Flight Control Case Study	140
6.3.1 System Context.....	140
6.4 Controller Design	142
6.4.1 RIDE Controller	142
6.4.2 RAID Controller.....	144
6.4.3 Simulation Parameters and Integration Algorithm	146
6.5 Simulation Results	148
6.5.1 Large amplitude <i>Pitch Rate</i> responses, Large actuators	148
6.5.2 Large amplitude <i>Yaw Rate</i> responses – Large actuator	153
6.5.3 Large amplitude <i>Roll Rate</i> responses – Large Actuators	156
6.5.4 Large amplitude <i>Yaw Rate</i> responses – Small actuators.....	158
6.5.5 Large amplitude <i>Pitch Rate</i> responses – Small actuators.....	161
6.5.6 Large amplitude <i>Roll Rate</i> response – Small actuators	164
6.6 Results Summary	166
6.7 Conclusions	167
7. Conclusions and Further Work.....	169
7.1 Thesis summary – aims and achievements.....	169
7.2 Summary of Outcomes– Variable Transient Response	171
7.3 Summary of Outcomes– Rate Actuated Inverse Dynamics	173
7.4 Further Work	174
7.5 Overall Conclusions	176
Appendix – Simulation and Modelling.....	186

List of Figures

Figure 1.1 – BAE Harrier during vertical takeoff @ http://www.richard-seaman.com/	2
Figure 1.2 – Block diagram of a feedback control system.	5
Figure 2.1 – RIDE controller block diagram.....	24
Figure 3.1 – Phase margins with increasing values of integral gain.	29
Figure 3.2 – Transient change in damping ratio with error.	33
Figure 3.3 – Tansient change in natural frequency with error.	33
Figure 3.4 – RIDE control structure with integration of VTR gain.	34
Figure 3.5 – Exponential Input: $X=1, Y=1, E=1$	40
Figure 3.6 – Exponential Input: $X=2, Y=1, E=1$	40
Figure 3.7 – Exponential Input: $X=4, Y=1, E=1$	41
Figure 3.8 – Exponential Input: $X=1, Y=2, E=1$	41
Figure 3.9 – Exponential Input: $X=1, Y=4, E=1$	41
Figure 3.10 – Exponential Input: $X=1, Y=1, E=2.5$	42

Figure 3.11 – Exponential Input: $X=1, Y=1, E=5$	42
Figure 3.12 – Closed Loop Response: $X=1, Y=1, E=1$	44
Figure 3.13 – Closed Loop Response: $X=2, Y=1, E=1$	44
Figure 3.14 – Closed Loop Response: $X=4, Y=1, E=1$	44
Figure 3.15 – Closed Loop Response: $X=1, Y=2, E=1$	45
Figure 3.16 – Closed Loop Response: $X=1, Y=4, E=1$	45
Figure 3.17 – Closed Loop Response: $X=1, Y=1, E=2.5$	46
Figure 3.18 – Closed Loop Response: $X=1, Y=1, E=4$	46
Figure 3.19 – Increase in overshoot as the setpoint is increased for fixed VTR parameters.	52
Figure 3.20 – Relationship between, X, Y and E for constant equivalent transient response.	53
Figure 3.21 – RIDE controller response.....	56

–	
–	
–	
–	
Figure 3.22	RIDE controller response with large actuator inertia. 57
Figure 3.23	Comparison of RIDE output with normal and reduced integral gain. 57
Figure 3.24	VTR controller response in comparison to RIDE. 59
Figure 3.25	Transient change in nonlinear function, N. 59
Figure 4.1	– Traditional anti-windup compensator with anti-windup gain K_{aw} 67
Figure 4.2	– Transformed RAID state space system. 70
Figure 4.3	– Graphic illustration of u_{eq} criteria for sliding mode. 81
Figure 4.4	– Illustration of the influence of controller gain on sliding mode behavior. 90
Figure 5.1	– Air temperature control, PI (no anti-windup), 2 min lag. 116
Figure 5.2	– Relative humidity control, PI (no anti-windup), 2 min lag. 116
Figure 5.3	– Air temperature control, PI, 2 min lag. 117
Figure 5.4	– Relative humidity control, PI, 2 min lag. 117
Figure 5.5	– Air temperature control, RIDE, 2 min lag. 118
Figure 5.6	– Relative humidity control, RIDE, 2 min lag. 118
Figure 5.7	– Air temperature control, PI (no anti-windup), 7 min lag. 119

–
–
–
–

Figure 5.8 – Relative humidity control, PI (no anti-windup), 7 min lag.	119
Figure 5.9 – Air temperature control, PI, 7 min lag.	120
Figure 5.10 – Relative humidity control, PI, 7 min lag.	120
Figure 5.11 – Air temperature control, RIDE, 7 min lag.	121
Figure 5.12 – Relative humidity control, RIDE, 7 min lag.	121
Figure 5.13 – Air temperature control, VTR ($K_I = K_{I\text{crit}}$), 7 min lag.	124
Figure 5.14 – Relative humidity control, VTR ($K_I = K_{I\text{crit}}$), 7 min lag.	125
Figure 5.15 – Air temperature control, VTR, 7 min lag.	125
Figure 5.16 – Relative humidity control, VTR, 7 min lag.	126
Figure 5.17 – Evolution of VTR gain N_{11} with temperature error.	126
Figure 5.18 – Evolution of VTR gain N_{11} with temperature error, detailed view.	127
Figure 5.19 Evolution of VTR gain N_{22} with humidity error.	127
Figure 5.20 Evolution of VTR gain N_{22} with humidity error, detailed view.	127
Figure 5.21 Air temperature control at temp. setpoint of 22.5 °C, RIDE, 7 min lag.	128
Figure 5.22 Relative humidity control at temp. setpoint of 22.5 °C, RIDE, 7 min lag.	128

–
–
–
–

Figure 5.23 – Summary of heating energy usage.....	138
Figure 5.24 – Summary of ventilation system usage.	138
Figure 6.1 – Missile body axis.	143
Figure 6.2 – Complete flight control system (modelled system in red).	147
Figure 6.3 – Expanded body-rate control system.	148
Figure 6.4 – RAID pole and zero locations.	153
Figure 6.5 – Body Rates / Large <i>Pitch Rate</i> Response / Large Actuator.	156
Figure 6.6 – Deflections / Large <i>Pitch Rate</i> Responses / Large Actuator.	157
Figure 6.7 – Rate of Deflections / Large <i>Pitch Rate</i> Responses / Large Actuator.	157
Figure 6.8 – u_{eq} sliding mode criteria / Large <i>Pitch Rate</i> Responses / Large Actuator.	158
Figure 6.9 – Body Rates / Large <i>Yaw Rate</i> Response / Large Actuator.	160
Figure 6.10 – Deflections / Large <i>Yaw Rate</i> Responses / Large Actuator.	161
Figure 6.11 – Rate of Deflections / Large <i>Yaw Rate</i> Responses / Large Actuator.	161
Figure 6.12 – u_{eq} sliding mode criteria / Large <i>Yaw Rate</i> Responses / Large Actuator.	162
Figure 6.13 – Body Rates / Large <i>Roll Rate</i> Response / Large Actuator.	163

–
–
–
–

Figure 6.14 – Deflections / Large <i>Roll Rate</i> Responses / Large Actuator.	164
Figure 6.15 – Rate of Deflections / Large <i>Roll Rate</i> Responses / Large Actuator.	164
Figure 6.16 – Body Rates / Large <i>Yaw Rate</i> Response / Small Actuator.	165
Figure 6.17 – Deflections / Large <i>Yaw Rate</i> Responses / Small Actuator.	166
Figure 6.18 – Rate of Deflections / Large <i>Yaw Rate</i> Responses / Small Actuator.	166
Figure 6.19 – u_{eq} sliding mode criteria / Large <i>Yaw Rate</i> Responses / Small Actuator.....	167
Figure 6.20 – Body Rates / Large <i>Pitch Rate</i> Response / Small Actuator.....	168
Figure 6.21 – Deflections / Large <i>Pitch Rate</i> Responses / Small Actuator.	169
Figure 6.22 – Rate of Deflections / Large <i>Pitch Rate</i> Responses / Small Actuator.	169
Figure 6.23 – u_{eq} sliding mode criteria / Large <i>Pitch Rate</i> Responses / Small Actuator.	170
Figure 6.24 – Body Rates / Large <i>Roll Rate</i> Response / Small Actuator.	171
Figure 6.25 – Deflections / Large <i>Roll Rate</i> Response / Small Actuator.	172
Figure 6.26 – Rates of deflection / Large <i>Roll Rate</i> Response / Small Actuator.	172
Figure A.1 – ESL modelling structure.	195
Figure A.2 – Flight control case study modelled system.....	195
Figure A.3 – HVAC case study modelled system.	195

–
–
–
–

List of Tables

Table 3.1 – Comparison of phase margin with integral gain.	29
Table 3.2 – Setpoint normalized values of X for values of Y.	54
Table 5.1 – Zone mass.	108
Table 5.2 – Zone thermal properties.	108
Table 5.3 – Zone area.	108
Table 5.4 – Occupant and environment properties.	108
Table 5.5 – Overall heat transfer coefficients.	108
Table 5.6 – Zone specific heat capacities.	108
Table 5.7 – Building heating and ventilation properties.	109
Table 5.8 – Closed loop tuning parameters for RIDE controller.	114

-
-
-
-

Table 5.9 – Total heating energy usage.
129

Table 5.10 – Total ventilation usage.
129

Nomenclature

Control system

- A - State matrix
- A* - Transformed RAID state matrix
- B - Input matrix
- B* - Transformed RAID input matrix
- B_{nl} - Nonlinear input matrix
- c - Partitioned slow states
- C - Output matrix
- C* - Transformed RAID output matrix
- CB_{trim} - Trimmed CB matrix
- d - Disturbance vector
- d* - Transformed RAID disturbance vector
- D - Disturbance matrix
- D* - Transformed RAID disturbance matrix
- e - Error vector
- f - Nonlinear state function
- E - Maximum error amplitude matrix
- e_{max} - Maximum error amplitude
- g - Feedback scalar gain
- K_d - Derivative gain matrix
- K_I - Integral gain matrix
- K_{I crit} - Critical integral gain
- K_P - Proportional gain matrix
- K_ω - VTR natural frequency coefficient
- K_v - VTR damping coefficient
- LL - Lower control signal limit
- M - Measurement matrix

M^*	-	Transformed RAID measurement matrix
N	-	Nonlinear VTR gain
N_{avg}	-	Quasi-linear VTR gain (average function method)
N_{DF}	-	Quasi-linear VTR gain (describing function method)
N_{QL}	-	Quasi-linear VTR gain
s	-	Laplace transform
sf	-	Safety factor
u	-	Plant input vector
u_c	-	Control signal vector
u_{DF}	-	Describing function input
u_{eq}	-	Equivalent Control
u_{eq}^*	-	RAID Equivalent Control
UL	-	Upper control signal limit
w	-	Feedback vector
x	-	State vector
x^*	-	Transformed RAID state vector
X	-	VTR parameter matrix X
X_{DF}	-	VTR parameter X designed using describing function method
X_{avg}	-	VTR parameter X designed using average function method
y	-	Output vector
Y	-	VTR parameter matrix Y
Y_{DF}	-	VTR parameter Y designed using describing function method
Y_{avg}	-	VTR parameter Y designed using average function method
z	-	Regulator vector
μ	-	Derivative scalar gain
ϵ_l	-	Lower switching surface
ϵ_u	-	Upper switching surface
ω	-	Natural frequency
ω_{VTR}	-	VTR natural frequency
ν	-	Damping ratio

$UVTR$ - VTR damping ratio

Energy system model

- \mathbf{A} - Linearised state matrix
- \mathbf{B} - Linearised input matrix
- A_f - Floor area (m^2)
- A_{tm} - Thermal mass area (m^2)
- A_r - Roof area (m^2)
- A_s - Structure area (m^2)
- A_w - Window area (m^2)
- C_a - Specific heat capacity of indoor air (J/kgK)
- C_{se} - Specific heat capacity of external structure (J/kgK)
- C_{si} - Specific heat capacity of internal structure (J/kgK)
- C_{tm} - Specific heat capacity of thermal mass (J/kgK)
- h_e - External heat transfer coefficient (W/m^2K)
- h_i - Internal heat transfer coefficient (W/m^2K)
- k - Wall thermal conductivity (W/mK)
- LL_{Qh} - Lower heating limit (W)
- UL_{Qh} - Upper heating limit (W)
- LL_{mv} - Lower ventilation limit (kg/s)
- UL_{mv} - Upper ventilation limit (kg/s)
- M_a - Mass of internal air (kg)
- M_{se} - Mass of external structure (kg)
- M_{si} - Mass of internal structure (kg)
- M_{tm} - Mass of thermal mass (kg)
- Q_f - Heat transfer through floor (J)
- Q_{dist} - Heat disturbances heat transfer (J)
- Q_{mv} - Mechanical ventilation heat transfer (J)
- Q_{nv} - Natural ventilation heat transfer (J)
- Q_{se} - Heat transfer through external structure (J)

Q_{si}	- Heat transfer through internal structure (J)
Q_{tm}	- Heat transfer through thermal mass (J)
Q_w	- Heat transfer through windows (J)
Q_{wall}	- Heat transfer through wall (J)
S_{occ}	- Occupant humidity gain (kg/kg)
T_a	- Internal air temperature (K)
T_g	- Ground temperature (K)
T_{se}	- External structure temperature (K)
T_{si}	- Internal structure temperature (K)
T_{tm}	- Thermal mass temperature (K)
U_f	- Floor heat transfer coefficient (W/m ² K)
U_r	- Roof heat transfer coefficient (W/m ² K)
U_w	- Window heat transfer coefficient (W/m ² K)
U_{tm}	- Thermal mass heat transfer coefficient (W/m ² K)
wt	- Structure wall thickness (m)
W_a	- Internal absolute humidity (kg/kg)
W_{ar}	- Internal relative humidity (%)
W_{ex}	- External absolute humidity (kg/kg)
τ_h	- Heating element time constant
τ_v	- Mechanical ventilation time constant

Aerodynamic model

F_y	- Horizontal aerodynamic force (N)
F_z	- Vertical aerodynamic force (N)
I_{xx}	- Moment of inertia about x axis (kg/m ²)
I_{yy}	- Moment of inertia about y axis (kg/m ²)
I_{zz}	- Moment of inertia about z axis (kg/m ²)
L	- Rolling aerodynamic moment (Nm)
N	- Yawing aerodynamic moment (Nm)
p	- Missile roll rate (rad/s)
P_d	- Dynamic pressure (Pa)

q	- Missile pitch rate (rad/s)
Q	- Pitching aerodynamic moment (Nm)
r	- Missile yaw rate (rad/s)
S_w	- Whetted surface area (m ²)
V_m	- Total missile forward velocity (m/s)
v_u	- Missile forward velocity (m/s)
v_v	- Missile sideslip velocity (m/s)
v_w	- Missile vertical velocity (m/s)
η	- Elevator deflection (rad)
ζ	- Rudder deflection (rad)
ξ	- Aileron deflection (rad)
λ	- Missile angle of incidence (rad)
ϕ	- Missile roll angle (rad)

Acronyms

HVAC - Heating Ventilation and Air Conditioning

MIMO - Multi Input Multi Output

NDI - Nonlinear Dynamic Inverse

PDF - Pseudo Derivative Feedback

PI - Proportional Integral

PID - Proportional Integral Derivative

RAID - Rate Actuated Inverse Dynamics

RAS - Rate and Amplitude Saturation

RIDE - Robust Inverse Dynamics Estimation

SISO - Single Input Single Output

SM - Sliding Mode

VSC - Variable Structure Control

VTR - Variable Transient Response

1. Introduction

1.1 Motivation

Control systems are a crucial component of almost all modern mechanical and electrical systems across a wide range of industries from automotive to aerospace, to energy systems and even domestic appliances. These controllers have been developed with the main purpose of enabling the performance of the systems they control to be improved. Performance can be defined as the ability of a system or machine to carry out its function. Through better control, the performance of a system or machine can be improved, and thus its functionality can be improved.

Primitive forms of control systems such as the centrifugal governors found on early steam engines enabled a huge increase in performance and usability, forming an essential part of the industrial revolution and creation of the modern world as we know it. The designs of control systems are constantly evolving in order to extract additional performance from systems. This can be encountered in everyday life as engine management systems are becoming ever more sophisticated in the quest to improve operating efficiency and to extract extra performance from the engine design. The leading edge of engineering technology also relies heavily on control systems to increase performance, such as in the Eurofighter Typhoon aircraft; deliberately designed to be aerodynamically unstable and so un-flyable without its control system. The field of aerospace engineering is always seeking to constantly extract more performance from the aircraft that are in service as this is a far more cost effective solution than to introduce new aircraft designs. Control systems play a key role in improving the performance and thus extending the operational life of aircraft. The BAE Harrier, seen in Figure 1.1, has seen a succession of control system improvements resulting in gradually less pilot

workload and increased overall performance. Enhancing system performance through better and more advanced control is especially important in applications such as missiles and projectiles where strict size and shape constraints and extremely high performance specifications mean that extracting the full capability from the existing system is extremely valuable. Across all applications and industries the benefits that the increased performance brings ranges from reduced running costs of automobile engines, more agile and easier to pilot aircraft, more efficient energy usage in industrial processes and cheaper energy generation.



Figure 1.1 – BAE Harrier during vertical takeoff ©<http://www.richard-seaman.com/>.

The cost of implementing new controller designs is relatively low, especially if a control system is already present; extracting more performance from the system

can be deceptively simple such as modifying a few lines of code programmed onto a micro-processor. Such small changes in controller design have the potential to yield huge benefits for the operation of the system. There is a constant drive to improve the design of the controller because it has the ability to unlock the maximum performance potential of the system it regulates.

1.2 Defining High Performance Control

A high performance controller can be defined as a control system that attempts to improve the performance of a machine or system beyond that which would result in it merely functioning. Rather, it attempts to enable the machine or system to function towards the limit of its capability.

In controller design there are many different indicators for controller performance such as speed of response, tracking, damping and disturbance rejection, amongst others. If the controller is viewed as a device that regulates the supply of energy to the plant then all these performance benchmarks can be encapsulated by assessing how that energy is used. Specifically, how much of the available energy is used and how efficiently it is used.

A system can be considered as having a certain amount of available energy which can be utilised to affect the dynamics of the plant. One characteristic of a high performance controller is that it has the ability to utilise the full amount of available energy if required. A lower performing controller cannot access the full amount of energy and so there is “un-tapped” potential dormant in the system.

It is not enough to just be able to use the maximum amount of energy available; for high performance control it is also important to use this energy in the most efficient manner possible. Whether the energy use is efficient, or not, is mainly

determined by the requested output of the system. For example, it may be possible for the controller to move the state of the plant from one point to another using the maximum amount of available energy; however, if the resultant state is not the commanded one, then the energy used has been wasted and so is, of course, not efficiently used. Generally, undesirable behaviour, such as overshoot or oscillations of requested system output, uses excess energy and so must be minimised if energy efficiency is to be maximised. Many systems require multiple variables to be controlled simultaneously and in these cases it is necessary to have some form of multivariable control to preserve energy efficiency. The objective of multivariable control is to remove any interactions between the controlled variables by compensating for any coupling in the controller design. This results in control signals becoming vectors and scalar controller gains becoming gain matrices.

Different systems will have different performance objectives and so will prioritise maximum energy usage over energy efficiency and vice versa. For instance, in a high performance combat aircraft or missile there is a certain amount of energy available to perform manoeuvres. So, in order to perform the most extreme manoeuvres, the maximum amount of available energy needs to be exploited. Conversely, in certain situations, such as building climate control or industrial process control, there is a need to minimise the total amount of energy used. In this case, efficient usage of the available energy is more important and will be a better definition of “high performance control”.

1.3 The Challenge of Achieving High Performance Control

The ultimate objective of a high performance controller is to enable the system it controls to reach its maximum performance potential. In order for the system to reach its full performance potential the controller must be designed so that the following objectives are met:

1. Be able to access the maximum amount of available energy, if required.
2. Use the available energy in the most efficient manner.

If these objectives are fully met then performance cannot be further improved and the system can be said to have reached its full potential.

The question is then what are the obstacles which must be overcome to achieve high performance control?

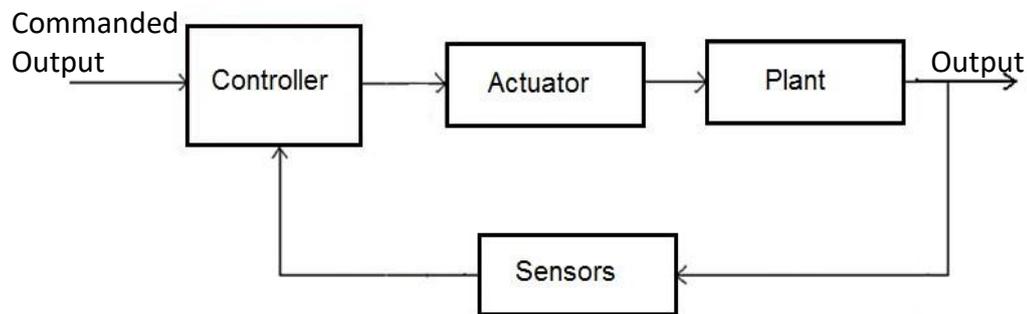


Figure 1.2 – Block diagram of a feedback control system.

1.3.1 Accessing the Maximum Available Energy

It is intuitive that the available energy of the system must have a limit. At any point in time the supply of energy to the plant is governed by the actuator. The actuator is any device or system that can be used to supply energy to the plant and thus alter its state. A block diagram schematic of the actuator and plant in relation to the complete control system is shown in Figure 1.2. All real actuators have a maximum and minimum limit to the amount of energy that can be supplied to the plant and hence it is this that limits the maximum amount of energy available to the control system. In terms of moving from one state to another as rapidly as

possible this is the ultimate barrier to further performance improvement. This limit also means that some states are not reachable as there is not enough energy available to the system. The limit on the maximum amount of energy that the actuator can supply is often described as an Amplitude Limit. Examples of actuator amplitude limits include the maximum and minimum deflections of aircraft control surfaces, maximum and minimum valve positions and maximum power output from heating elements.

The speed at which the system is able to transition from one state to another is heavily determined by the rate at which energy can be supplied to the plant. This is governed by the rate limit of the actuator. The actuator will spend the majority of its time operating between the amplitude limits, therefore, the rate limits will be highly significant because they will be dictating the maximum allowable energy consumption for the majority of the systems operation.

It is clear that if the maximum amount of available energy is to be accessed then the control system will be required to operate, without complications, when an actuator has reached its rate limit and then amplitude limit. This poses some problems for controller design as the controller's operation abruptly changes from continuous (in the non-limited region) to discontinuous when the limit is reached. Preventative measures have to be taken otherwise there is a risk that the actuator will either remain "stuck" on a limit for longer than is necessary or will cycle between limits, both of which can lead to excess energy use, or in the worst cases, catastrophic system failure. A well-known example of this problem is the JAS Gripen crash of 1993 (State and Iorga, 2012) and the Tornado pilot in the loop (PIO) incident (Fielding and Flux, 2003). In this case an oversight in the flight control system design caused the control surface actuators to cycle between limits causing an aircraft to stall and subsequently crash. This problem is compounded

when there are multiple actuators present in a system with multiple variables under control.

The severity of the consequences if the controller design does not adequately address the problem of amplitude limits has meant that a large amount of effort has been spent by both the research community and industry in addressing the issue. Therefore, whilst still addressed, it was not one of the main challenges investigated in this research. The issue of rate limits is still a major source of control failure, as seen by the Gripen crash, but solutions to this problem are not well represented in the literature. The problem of actuator rate limits in controller design was therefore one of the main topics for investigation in this thesis.

1.3.2 Using Energy Efficiently

The ability to access the maximum amount of available energy is of little value if it cannot be used efficiently. The most common causes of loss of energy efficiency, and thus performance degradation, are controller overshoot and oscillation. Both of these problems are often caused by inertia or time delays within the closed loop of the system, the most significant of which are present in the actuation and sensor systems.

In effect, this inertia also presents a limit to controller performance. As the responsiveness of the controller is made faster the significance of the actuator and sensor inertia is increased since the relative speed of the inertia to the controller is now slower. At a certain point the relative speed of the actuators and sensors will be so slow compared to the closed loop speed that oscillations and overshoot will be unavoidable. Minimisation of this effect will significantly improve the efficiency of the energy utilisation.

In summary, it can be reasoned that if high performance control is to be achieved then the main obstacles to overcome are the imperfections and limitations of actuation and sensor systems. Specifically, the amplitude and rate limits of the actuation system and the inertia present in the actuators and sensors.

1.4 Thesis Aims

The aim of this thesis is to describe a method of high performance controller design that will move closer to fulfilling the ultimate goal of high performance control: enabling the full potential of the system to be realised. This challenge will be met by overcoming the limitations of actuators and sensors inherent in all real systems and by satisfying the following requirements:

1. Being able to operate on multiple actuator amplitude and rate limits without detriment to controller operation. This will enable the maximum amount of available energy to be accessed when required.
2. Remaining on actuator amplitude and rate limits only for as long as is necessary. This will ensure that excessive energy is not used and energy efficiency is maximised.
3. Provide high performance control with minimum overshoot and oscillatory behaviour. This will bring the controller design closer to maximising energy efficiency.

4. To control multiple variables simultaneously with little or no interaction, thus utilising the available energy in a more efficient manner. This will require the controller design to be multi-input, multi-output in structure. All controlled outputs must be able to track a setpoint, such that the tracking error is zero in finite time.

5. The solutions should be applicable to many types of systems; both areas that traditionally use high performance control and also ones that do not - the areas that traditionally use lower performing controllers that stand to benefit most from improving controller performance. Therefore, the proposed controller designs should be limited in complexity wherever possible so that the controller design processes have applicability across a wide range of applications.

6. The solution should be robust to changes in operating conditions. Robustness is the invariance of the controller operation to assumptions made at the design stage. This is an important requirement, as details of the plant dynamics and operating conditions of real-life control systems are often uncertain at the design stage.

1.5 Thesis Methods and Structure

The high performance controller described in this thesis is structured in two layers. Firstly, a multivariable, nonlinear dynamic inverse (NDI) core. Secondly, modifications are built on top of the NDI core addressing the problems encountered with actuator and sensor limitations and inertia. The modifications

aim to preserve the positive attributes of the NDI controller (decoupled multivariable control, disturbance rejection and improved performance with nonlinear plant dynamics) whilst improving the performance when the actuator is severely limited in rate and amplitude and when the sensor and actuator have significant inertia. The two problems of actuator limitation and actuator and sensor inertia are addressed with separate modifications to the NDI core controller.

1.5.1 Outline of thesis chapters

Chapter 1 – Introduction

The importance and benefits of high performance control are presented along with its historical development. High performance control is defined in terms of energy utilisation and the problems associated with achieving high performance control are introduced. The aims of the thesis are defined and the controller design methods developed to achieve these aims are introduced. The structure of the thesis is described with a brief summary of each chapter.

Chapter 2 – High Performance Control Methods

An overview of high performance control methods is undertaken, including relay control, variable structure and sliding mode control, optimal control and Inverse Dynamics. Inverse Dynamics is selected as the structure on which the controller designs developed in this thesis are based. A robust and practical form of nonlinear Inverse Dynamics known as Robust Inverse Dynamics Estimation (RIDE) is described in detail. This chapter serves as an introduction to many of the control techniques encountered later in this thesis.

Chapter 3 – Variable Transient Response

Chapter 3 describes the controller design method of Variable Transient Response (VTR). VTR addresses the problem of achieving high performance control through efficient energy usage when there are significant inertias present in the actuator or sensor systems. The philosophy of VTR is described and its implementation within a multivariable nonlinear Inverse Dynamics framework is presented. The controller's performance and stability are analysed using classical and describing function methods. The analysis is used to develop a design methodology and systematic process for controller tuning.

Chapter 4 – Rate Actuated Inverse Dynamics

Chapter 4 describes the controller design method of Rate Actuated Inverse Dynamics (RAID). Using nonlinear Inverse Dynamics RAID addresses the problem of achieving high performance control when the system's actuators are severely limited in output rate and amplitude. Specifically, the RAID design aims to maintain control stability whilst fully saturating the system's actuators in both rate and amplitude. This is achieved by a Variable Structure Control (VSC) switching methodology and a transformation of a multivariable Inverse Dynamics control structure so that the input to the actuator is a rate of change. The closed loop pole and transmission zero locations are derived and the stability of the system is investigated. The conditions under which the controller can stability operate when the actuators are saturated in rate and amplitude are investigated, and corresponding criteria are created.

Chapter 5 – Case Study 1: Heating and Ventilation System

Chapter 5 presents a case study simulation of a high performance heating and ventilation system for a modern office space. The purpose of the case study is to assess the performance of the VTR controller design in tracking indoor air temperature and humidity with significant sensor lags present in the system. The performance of the VTR controller is compared to that of an “industry standard” single-input-single-output (SISO) Proportional Integral (PI) controller and a more advanced multivariable Inverse Dynamics based approach. The performance improvements gained by VTR over the latter approaches are investigated and the energy saved by using a VTR controller design is estimated.

Chapter 6 – Case Study 2: Missile Flight Control

Chapter 6 presents a simulated case study of a missile flight control system. A mathematical model of the missile aerodynamics, sensors and actuators is presented. The mathematical model of the missile is combined with a body-rate flight controller and simulations of the flight controller tracking demanded body rates are undertaken. The main purpose of the simulation is to assess the possible performance improvements that can be obtained with the RAID controller design when actuators are required to saturate in rate and amplitude. To better highlight any differences in performance two actuator types are simulated: (i) a larger high powered actuator with large rate of deflection limits and (ii) a smaller, lighter actuator with reduced rate of deflection limits. The performance of a RAID designed control system is compared with a benchmark RIDE controller to provide a comparison with previously published results.

Chapter 7 – Conclusions and Further Work

The final chapter summarises the work undertaken in this thesis. The achievements contained within each chapter are compared with the aims of the thesis to assess how they have been met. The outcomes of the research which resulted in the VTR and RAID controller design methods are discussed and potential further work and improvements are suggested. Finally the overall contribution of the work is summarised and the thesis is concluded.

2. An Overview of High Performance Control Methods

2.1 A High Performance Controller Framework

A controller framework must be selected in order to begin the process of creating a controller design method to fulfil the aims established in Chapter 1.

Controller design methods have been gradually evolving since the invention of feedback control and an evolution of design methods will accomplish the aims of this thesis. Therefore, a feedback controller design framework must be established, which will then form the foundations of the controller design methods presented in the subsequent chapters. To aid the selection process it is pertinent to specify some criteria that the controller design framework must possess in order for it to be considered. These criteria can fall under the two facets of high performance control defined in Chapter 1; namely, the ability to access the maximum available energy and the ability to use that energy as efficiently as possible.

2.1.1 Accessing maximum available energy

In order to access the maximum available amount of energy the actuators must be able to operate on, or close to, their amplitude and rate limits. It then follows that the control system must be able to operate without problems when this occurs. Problems with controller operation often occur when the control input exceeds the actuator's limit. To prevent this from happening some form of what is commonly called anti-windup is often implemented in the control system design.

The anti-windup design only becomes active when the control limits are reached and seeks to keep the control signal below the limit. Much research has been carried out on anti-windup designs for amplitude limits (Tabouriech and Turner, 2009), although there is comparatively little on rate and rate and amplitude input limitations (an in depth review of recent anti-windup designs is conducted in Chapter 4). Ideally, the controller framework should be suitable for a rate and amplitude limiting anti-windup design to be implemented.

2.1.2 Energy Efficiency

The ideal controller framework should provide the foundation for the most efficient use of the available energy, as stated in the thesis aims. The controller framework should be able to meet the stated energy efficiency aims as best as possible. In this way, there will be a strong base upon which the energy efficiency can be improved by the controller design methods developed in this thesis.

2.2 An Overview of High Performance Controller Design Methods

2.2.1 Relay and High Gain Control

One of the simplest methods of feedback control that could be classified as high performance is relay control, sometimes known as “Bang-Bang” control. In relay control, the control signal has two possible states: a high and low control action. If the error signal, e , is above a threshold then the state of the control signal switches to the upper value. If it is below the threshold then the control signal switches to the lower value.

This kind of control can be effective in certain circumstances (e.g. where a proportional actuation system is not available). It fulfils one of the criteria for high performance control in that, if the upper and lower control values are the upper and lower actuator amplitude limits, then it has the ability to access the maximum available energy. Furthermore, providing that the system is of minimum phase and the setpoint is reachable then a relay controller will be able to stabilise about the setpoint (Franklin et al., 2001). The main drawback of relay control is that its use of energy is extremely inefficient when attempting to track the setpoint. Close tracking of the setpoint is extremely difficult, especially if the upper and lower control values are large in magnitude. Despite these drawbacks relay control still be seen as a primitive form of high performance control and shares some aspects with variable structure control; i.e. high gain control and Inverse Dynamics which will be discussed later in this chapter.

A relay controller can be linearised about the point where it switches from one control value to the other. The results of the linearisation reveal that the relay

controller can be approximated by a linear controller with an infinite gain (Franklin et al., 2001). In effect a linear feedback controller with an infinite gain will be able to control any system with perfect disturbance rejection, setpoint tracking and an infinitely fast response time with no knowledge of the system dynamics required (Porter and Bradshaw, 1979), (Young et al., 1977) (this is providing that the system has stable transmission zeros). The question then has to be asked: why is controller design not as simple as using a feedback controller with an extremely high gain? The reality is that the physical limitations of the actuators and sensors in the system mean that this ideal is not achievable in practice. The higher order dynamics resulting from actuator and sensor inertia become excited with high gain and cause high frequency oscillations to occur, compromising system stability. By contrast, the response time of the system is limited by the physical power limitations of the actuators in the form of amplitude and rate limits. Thus, the goal of many high performance controller designs (since the conception of high gain feedback control) has been to achieve the ideal performance of the theoretical high gain controller, but without actually resorting to using high gain feedback.

2.2.2 Variable Structure Control and Sliding Mode Control

Variable Structure Control or VSC is a control method that shares some similarities with “Bang-Bang” or relay control. The distinguishing feature of VSC is that the controller can switch between two or more separate control laws depending on the state of the system; i.e., the structure of the controller is variable. Relay control is itself a type of VSC, but whereas relay control is limited to simply switching between two control values a VSC controller can switch between many different continuous control laws. In this manner VSC control can be both continuous and

discontinuous and consequently belongs to a group of controllers known as hybrid controllers.

The most common form of VSC is known as Sliding Mode (SM) control. Sliding mode control originated in Russia in the 1970s and was popularised in the following two decades by Vladimir Utkin in particular (Zinober, 1990). SM control relies on establishing what is known as a “sliding surface”, over which the controller switches between separate continuous control laws depending on the state of the system.

The main benefits of sliding mode control are (i) its excellent robustness with uncertain plant dynamics, (ii) relative simplicity and (iii) its robust performance with actuator amplitude limitations (Zinober, 1990). It has been extended for use with multivariable systems and remains an active area of research (Shtessel et al., 2002), (Utkin and Chang, 2002), (Yokoyama et al., 2010). As with relay control its main drawback is its potential inefficient use of energy; e.g. during sliding mode operation high frequency “chatter” can occur due to actuator and sensor inertia (Utkin, 2006).

Solutions to the problem of chatter usually focus on continuous control approximations of the controller during the sliding mode. The continuous average of the rapidly switching control during the sliding mode - known as the Equivalent Control or u_{eq} - is of particular interest in the design of Inverse Dynamics control systems and will be discussed further later in this chapter.

2.2.3 Optimal Control Methods

Optimal controller design methods originated in the 1980s and were popularised in the 1990s through formulating the control problem as a mathematical

optimisation problem (Lewis and Syrmos, 1995). This involves the formulation of a performance metric, which must then be minimised. Typically this is a weighted function of error and controller output. The controller is then said to be optimal if it is designed by minimising this function.

By optimising the controller across the whole frequency range and incorporating robustness metrics into the optimisation problem its sensitivity to disturbances and parameter variations can be minimised (Zhou et al., 1995). This results in the robustness of the controller being quantified in terms of bounds on the system uncertainties. One of the best known forms of optimal control is H^∞ and H^∞ loop shaping.

The main advantage of H^∞ based controller designs is its very efficient use of energy, which comes from its insensitivity to disturbances and coupling between variables. The most significant drawbacks are its lack of applicability when actuators are saturated and its mathematical complexity, which has limited its uptake in many industrial applications.

2.2.4 Inverse Dynamics

The intrinsic principal of control using Inverse Dynamics is simple in concept and yet a universal truth at the heart of all controller design. In control problems, a system (the plant) will accept some kind of input, the input will modify the state of the system and one or more outputs can be measured and their changes recorded. In essence, a control system is attempting to achieve the direct opposite of this process. A control system receives a requested plant output, processes this request and then outputs a plant input. Therefore, the control system is implicitly acting like an inverse of the plant. All control systems act in this manner, the differentiating characteristic of an Inverse Dynamics control system is that it

explicitly includes some form of a dynamic inverse of the plant in the controller design.

Inverse Dynamics controller design began in the robotics industry as an efficient method for computing required motor actions (Bayo et al., 1989). It was also popularised in high performance nonlinear flight control applications (Lane and Stengel, 1988) and remains a popular area of research in flight control (Menon et al., 2008), (Sieberling et al., 2010), (Shin et al., 2008), (Steer, 2001). In the most elementary form of Inverse Dynamics the inverse of the plant is used to cancel out the plant dynamics so that the control system designer is free to specify the controlled dynamics. This technique shares many similarities with the controller design method of feedback linearisation (Lee et al., 2009), (Pathak et al., 2005), (Zhengxian et al., 2007), (Lee et al, 2009). By performing this cancellation any couplings between controlled variables and disturbances are negated. If a nonlinear inverse of the plant is used, then the nonlinear dynamics of the plant are also negated. This is essentially a feed-forward design and so any variation of the real plant dynamics to the inverted set will cause performance degradation, hence much of the development of Inverse Dynamics control methods has focussed on achieving high performance with plant dynamics uncertainty. Proportional Integral (PI) feedback loops have been introduced which allow for the uncertainties in the model to be corrected as well as specifying desired closed loop dynamics (Qian and Stengel, 2005). Robust control methods such as H^∞ control have been used to design the gains in the feedback loops thus directly taking into account model uncertainty in the controller design, (Papageorgiu and Glover, 2005). Neural networks have also been utilised to improve the accuracy of the dynamic inverse (Plett, 2003). The amount of system knowledge required to perform a dynamic inverse was reduced by Robust Inverse Dynamic Estimation (RIDE) (Bradshaw and Counsell, 1992) where an estimate of the inverse was used which was then corrected by a Pseudo Derivative Feedback (PDF) feedback loop. The RIDE Inverse Dynamics method also

addressed the issue of actuator amplitude limits by incorporating a variable structure antiwindup design (Muir and Bradshaw, 1996). It has since been described in a discrete time form (Ding et al., 2006) and used as a benchmark for high performance flight control (Magni et al., 1997), (Fielding et al., 2002).

In summary, an Inverse Dynamics based controller design offers good energy efficiency through reduction of couplings, nonlinearities and disturbances and has the potential for effective treatment of actuator rate and amplitude limits.

Importantly, the controller design process is also relatively straightforward. (Magni et al. 1997).

2.2.5 Controller Framework Selection

Whilst arguments could be made for many of the controller design methods to be used as the framework the design method of Inverse Dynamics was selected. A strong case could be made for using an optimal control framework because it is able to utilise energy very efficiently and has very good robustness properties. However, it was felt that the mathematical complexity would limit the general applicability of any developed controller designs. This is an important point as systems with severely limited and high inertial actuation and sensor systems often use relatively simple controller designs; consequently, if the developed controller designs are to be practical then they must reflect the reality that mathematically complex controller designs are not likely to be used in these situations. In summary, the Inverse Dynamics method was preferred as it offered the best balance between desired performance capability and minimal complexity.

2.3 Robust Inverse Dynamics Estimation (RIDE) Design Method

Having established that an Inverse Dynamics based control structure is to be used as the controller development framework, it is then necessary to decide on the form of the Inverse Dynamics.

As has previously been mentioned, the main drawback of early Inverse Dynamics based controller design methods was the reliance on an accurate mathematical model of the system to be controlled. As such its robustness was poor when faced with uncertain plant dynamics. One form of Inverse Dynamics that seeks to improve the robustness, and practicality, of an Inverse Dynamics controller design is Robust Inverse Dynamics Estimation or RIDE.

The RIDE controller design method addresses some of the problems encountered when designing controllers using a direct Inverse Dynamics method. Namely, RIDE (i) requires less knowledge of the system to invert the plant, (ii) there is improved robustness and (iii) the design procedure is simpler. This is achieved by combining a Pseudo Derivative Feedback (PDF) control structure (Phelan, 1977) with an inner loop which contains an estimate of the inverted dynamics of the plant. The block diagram of the RIDE controller structure is shown in Figure 2.1.

The following section serves as an introduction to the RIDE controller design process. The controller design methods presented in this thesis are built upon the foundation of the RIDE controller design method.

Modern controller design methods require that the plant or system to be controlled be represented in state space form. A nonlinear system can be represented in a generalised nonlinear state space form (Muir and Bradshaw, 1996) as shown in Equation 2.2.1.

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{B}_{nl}(\mathbf{x}(t))\mathbf{u}(t) + \mathbf{d}(t) \quad (2.2.1)$$

where $\mathbf{x}(t) \in R^n$, $\mathbf{u}(t) \in R^m$, $\mathbf{B}_{nl}(\mathbf{x}(t)) \in R^{n \times m}$ and $\mathbf{d}(t) \in R^n$, with linear output and feedback relationships where $\mathbf{y}(t) \in R^m$, $\mathbf{w}(t) \in R^m$, $\mathbf{C} \in R^{m \times n}$ and $\mathbf{M} \in R^{m \times n}$. f is a nonlinear function and the disturbance, \mathbf{d} , is assumed to be stochastic.

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \quad (2.2.2)$$

$$\mathbf{w}(t) = \mathbf{M}\mathbf{x}(t) \quad (2.2.3)$$

The nonlinear state equation (Equation 2.2.1) can be linearised about an operating point and represented in a general linear state-space form (Franklin et al., 2001), where $\mathbf{A} \in R^{n \times n}$ and $\mathbf{B} \in R^{n \times m}$. δ is the small perturbation operator.

$$\delta \dot{\mathbf{x}}(t) = \mathbf{A}\delta\mathbf{x}(t) + \mathbf{B}\delta\mathbf{u}(t) + \delta\mathbf{D}\mathbf{d}(t) \quad (2.2.4)$$

A dynamic inverse can be formed by setting $\dot{\mathbf{w}} = 0$

$$\mathbf{u}_{eq}(t) = -\mathbf{M}\mathbf{B}^{-1}(\mathbf{M}\mathbf{A}\mathbf{x}(t) - \mathbf{M}\mathbf{d}(t)) \quad (2.2.5)$$

This is known as the Equivalent Control, or \mathbf{u}_{eq} . The Equivalent Control is related to sliding mode control in that it is the continuous equivalent to the average of the control signal during the sliding mode (Zinober, 1990). The Equivalent Control only inverts the slow modes of the plant (Counsell, 1992) and so the fast modes must be considered separately.

As can be seen from Equation 2.2.5, the Equivalent Control requires full knowledge of the system dynamics in the form of the full state vector, the disturbances and the system matrices \mathbf{A} and \mathbf{B} . The measurement and knowledge of all these properties is often not achievable in many real systems. Whilst it may be possible to estimate some of these variables using Kalman filters (or similar) (Brown and Hwang, 1996), (Liu and Peng, 2002) it is still not a particularly

practical solution. A better solution is a simple rearranging of equations 2.2.5 and 2.2.1. Knowledge of the system can then be replaced with extra feedback loops, resulting in an estimate of the Equivalent Control, u_{eq} .

$$\mathbf{u}_{eq}(t) = -(\mathbf{MB})^{-1}\dot{\mathbf{w}}(t) + \mathbf{u}(t) \quad (2.2.6)$$

The resulting estimate of u_{eq} only requires knowledge of the input matrix B, the measurement matrix M, the feedback, w, and the actuator input, u. The effects of the inevitable modelling inaccuracies in M and B on the controller performance are compensated by the use of Pseudo Derivative Feedback (PDF) control loops. This in turn will correct any inaccuracies in the estimate of u_{eq} , hence improving the robustness of the controller design with uncertain plant parameters. The PDF control loops also attempt to decouple and invert the fast modes of the system, therefore, it is possible to completely decouple the closed loop system (Bradshaw and Counsell, 1992). The full control algorithm is described by the following equations:

$$\mathbf{u}_c(t) = \mathbf{z}(t) - \mathbf{K}_P\mathbf{w}(t) + \mathbf{u}_{eq}(t) \quad (2.2.7)$$

$$\dot{\mathbf{z}}(t) = \mathbf{K}_I\mathbf{e}(t) \quad (2.2.8)$$

$$\mathbf{e}(t) = \mathbf{y}_c(t) - \mathbf{w}(t) \quad (2.2.9)$$

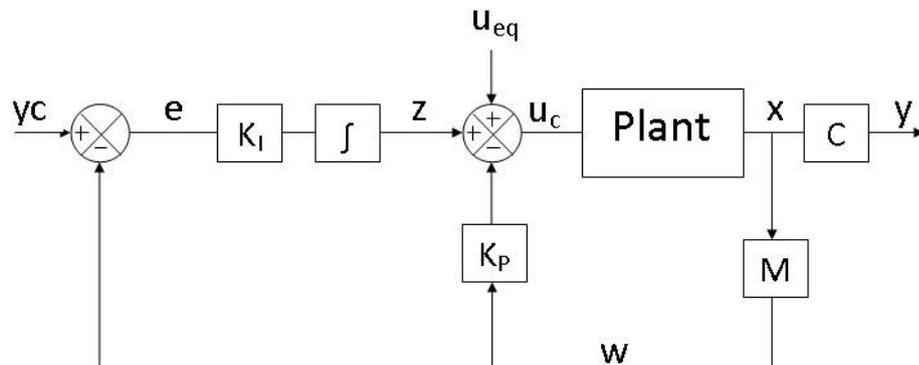


Figure 2.1 – RIDE controller block diagram.

If the gain matrices are set as shown in Equations 2.2.10 and 2.2.11 then the decoupled closed loop system can then be designed with a specified closed loop

natural frequency and damping ratio determined by the values of the diagonal gains σ and ρ .

$$\mathbf{K}_P = \rho[\mathbf{MB}]^{-1} \quad (2.2.10)$$

$$\mathbf{K}_I = \sigma[\mathbf{MB}]^{-1} \quad (2.2.11)$$

For the RIDE design to be successfully implemented there are two requirements which must be met: (i) the actuator and sensor dynamics must be considered fast with respect to the closed loop and the regulator transmission zeros with respect to the feedback vector must be stable (Counsell, 1992).

The feedback vector transmission zeros must be stable as the use of u_{eq} effectively places closed loop poles over the transmission zeros, thus cancelling the plant dynamics (Bradshaw and Counsell, 1992). In this respect it achieves the same outcome as using an infinitely large feedback gain, but without the impracticality of using an infinitely large gain.

The inclusion of the actuator and sensor dynamics directly in the design process would make achieving a dynamic inverse impossible as the relative degree would be greater than one. Therefore, they are excluded from the design process but must still be considered in specifying the closed loop bandwidth. There must be a large enough separation in bandwidth between the closed loop dynamics and the “un-modelled” actuator and sensor dynamics to avoid exciting the actuator and sensor dynamics. Exciting the higher order actuator and sensor dynamics will lead to high frequency oscillations and possible instability in the closed loop response. The greater the inertia of the actuators or sensors, the smaller the bandwidth and hence the smaller the useable closed loop bandwidth. Therefore, great care must be taken in choosing the scalar gains ρ and σ to ensure a robust final controller design.

3. Variable Transient Response

3.1 Introduction

The gain of a control system dictates how fast the controlled output will be able to reach its setpoint. A larger gain will result in a more responsive controller that is able to reach a setpoint faster. However, the gain cannot be increased without limit; if the gain is too large then undesirable overshoot of the setpoint or oscillatory behaviour can develop. The limit at which this occurs is greatly influenced by higher order (second order and above) dynamics e.g. the dynamics of the actuators and sensors or any lags and phase changing elements within the feedback system. Large inertias within these elements may mean that a high gain controller is not possible without exciting the system's higher order dynamics thereby incurring severe, high frequency oscillatory behaviour. Thus, a controller that is designed to be responsive may not be robust when implemented due to the presence of sensor lags and large actuator or sensor inertias.

In order to reduce the controller's sensitivity to higher order, high frequency dynamics, it is necessary to reduce the controller's gain. However, this will also reduce the controller's responsiveness. If the higher order un-modelled dynamics are significant enough then the gain may have to be set to a value which would result in an unacceptable speed of response. Therefore, reducing sensitivity to higher order dynamics comes at the expense of speed of response.

3.1.1 Controller response and phase margins

The Inverse Dynamics controller design process typically assumes that any actuators or sensors within the control loop operating at a sufficiently high

bandwidth compared to the closed loop so that they can be considered steady state. The validity of this assumption can vary depending on operating condition and can also vary during the lifetime of the controller (i.e. as systems degrade). A control system's sensitivity to elements which cause a phase lag can be defined by its phase margin (Franklin et al., 2001). The phase margin is a relative degree of stability; i.e. a large phase margin means that the performance of the controller is insensitive to phase changing elements such as the higher order un-modelled dynamics of sensors and actuators. Typically, a small phase margin will result in oscillatory controller behaviour if there are significant phase changing elements present. Thus, it can be inferred that a small phase margin equates to a controller that has reduced robustness properties.

CB

By examining the phase margin of a RIDE controller with a generic plant of $\frac{1}{s}$ at varying values of K_I (displayed in Figure 3.1 and Table 3.1) it is possible to gain a qualitative insight into the sensitivity to oscillations as the gain is changed.

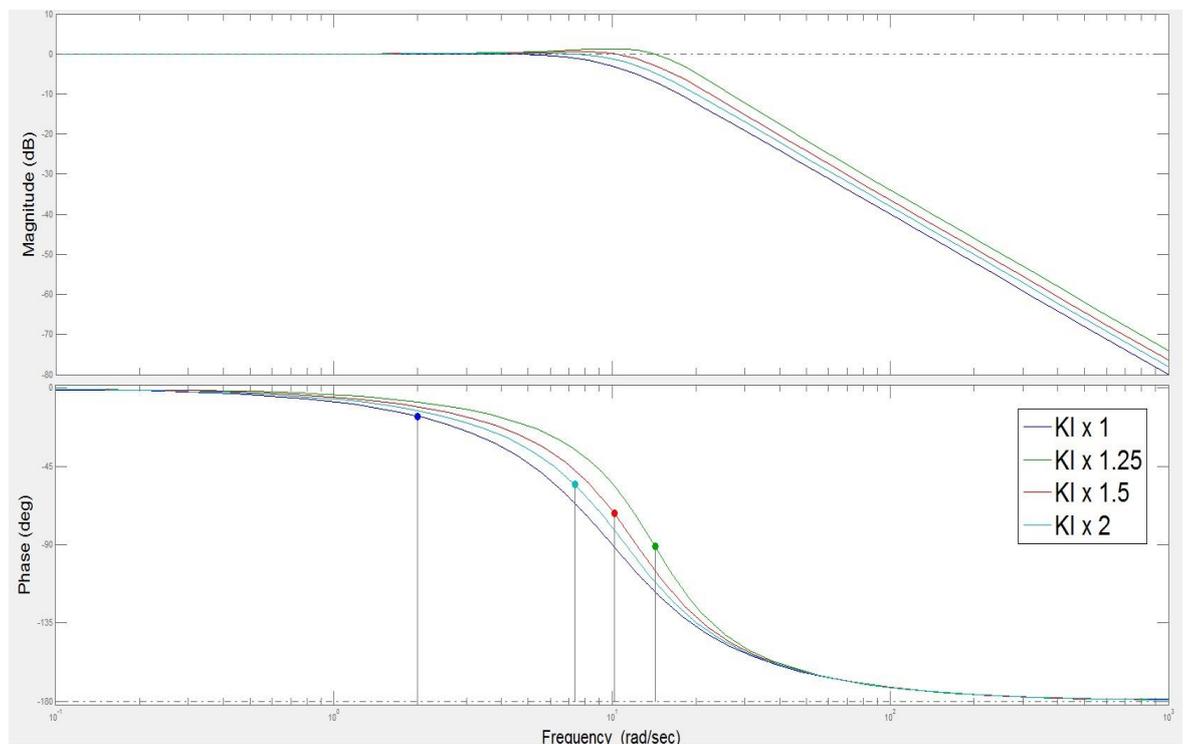


Figure 3.1 – Phase margins with increasing values of integral gain.

Table 3.1 – Comparison of phase margin with integral gain.

K _I multiplier	1	1.25	1.5	2
Phase Margin (deg)	164	125	108	89

The results demonstrate that by increasing the integral gain the phase margin is reduced but the controller bandwidth is increased. Conversely, reducing the integral gain reduces the controller bandwidth and increases the phase margin, thus, the controller's sensitivity to oscillations caused by fast un-modelled dynamics is reduced. Therefore, in general, a controller with a greater bandwidth is more responsive, but has increased sensitivity to high order dynamics.

3.2 Transient Response Shaping

In terms of transient response the bandwidth of a control system is equivalent to its natural frequency (Franklin et al., 2001). The previous sections in this chapter discussed the trade-off that exists in controller design between specifying an adequate natural frequency and insensitivity to actuator and sensor higher order dynamics. This trade-off can be circumvented by prioritising different transient response characteristics at different points during the controller's operation.

When the error is large, i.e. the output is far away from the setpoint, the priority is to reach the setpoint in an acceptable time. During this period responsiveness can be prioritised over robustness as this period should be brief and any oscillatory behaviour will be short-lived. Therefore a large natural frequency can be used and a reduced damping ratio is acceptable. When the output approaches the setpoint the reverse is true as the priority is robust and stable performance, therefore, in order to reduce sensitivity to higher frequency dynamics a reduced natural frequency must be specified. The above statements can be summarised by a transient change in system damping inversely proportional to the error ($y(t) - y_c$) and a directly proportional change in system natural frequency. In this case, a fast response time to the setpoint can be maintained whilst simultaneously having a lower natural frequency at the setpoint; consequently sensitivity to higher order un-modelled dynamics is reduced.

The desired change in transient response characteristics can be achieved by altering the controller gain. A time varying gain has often been used to change the transient characteristics of a control system during its operation. One of the most common forms of dynamic gain alteration is gain-scheduling, where the gain of a controller is varied as the system passes through operating points, with the aim of preserving linear operation in a nonlinear system. An overview of this technique is given by Rugh and Shamma (2000). This type of gain alteration is usually achieved through the use of interpolated look-up tables or a nonlinear function. Conventional gain scheduling has a much different time scale and objective to the transient response characteristics being proposed here, however, the use of a nonlinear function to change the controller gain is a potential method to achieve the desired variable transient response characteristics.

The relatively simple control method of bi-linear control (Goodhart et al., 1994), (where the controller gain has two values based on the system's state), attempts to achieve a similar transient response profile to that proposed by a VTR controller. The method is, however, quite unsophisticated. The abrupt change between the two gain values has the potential to cause undesirable controller behaviour which is difficult to predict and analyse. Consequently, it is difficult to design the desired transient response.

The controller design method of setpoint weighting attempts to improve the speed of response to large setpoints by varying the gain of the controller with the requested setpoint (Hang and Cao, 1996). This is more similar to gain scheduling in that a static gain is used during the controlled output's transition to the setpoint. A transient change in gain will be required to achieve the transient change in natural frequency required by a VTR controller.

Composite Nonlinear Feedback (CNF) (Lin et al., 1998), (He et al., 2005), (Peng et al., 2006), (Lan et al., 2010) is a very relevant controller design method that, by using a time varying nonlinear function, attempts to reduce settling time of a controlled response. The nonlinear function is dependent on the error such that when the error is large the nonlinear function is zero and has no influence. When the error is small the nonlinear function is large in magnitude, seeking to increase the damping of the controller and minimise overshoot. This is a fundamentally different philosophy than that proposed by VTR. Whereas VTR seeks to reduce the gain of controller close to the setpoint, CNF increases the controller action near the setpoint. This is analogous to increasing the proportional gain in the RIDE structure and thus increasing the damping.

3.2.1 Target Dynamics

Relationships between natural frequency and error, and damping and error can be proposed that meet the required transient response shapes, where X and Y are tuneable coefficient matrices and K is a function of the plant. The selection of the X and Y parameters will shape how the transient characteristics change with error, as is illustrated by Figure 3.2 and Figure 3.3. X will determine the magnitude of the relationships between natural frequency, damping ratio and error, whereas Y determines the minimum natural frequency and the maximum damping ratio. During the steady state (when the setpoint is being tracked) the error becomes zero and hence the dynamics are only affected by the value of Y . Lower values of Y will result in a lower steady state natural frequency and a high damping ratio.

$$\omega_{VTR} \text{ target } (\mathbf{e}(t)) = \mathbf{K}_\omega(\mathbf{X}\mathbf{e}(t)^2 + \mathbf{Y}) \quad (3.2.1)$$

$$\nu_{VTR} \text{ target } (\mathbf{e}(t)) = \mathbf{K}_\nu(\mathbf{X}\mathbf{e}(t)^2 + \mathbf{Y})^{-1} \quad (3.2.2)$$

$$\mathbf{X} = \begin{bmatrix} X_i & 0 & 0 & 0 \\ 0 & X_{i+1} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & X_n \end{bmatrix} \quad (3.2.3)$$

$$\mathbf{Y} = \begin{bmatrix} Y_i & 0 & 0 & 0 \\ 0 & Y_{i+1} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & Y_n \end{bmatrix} \quad (3.2.4)$$

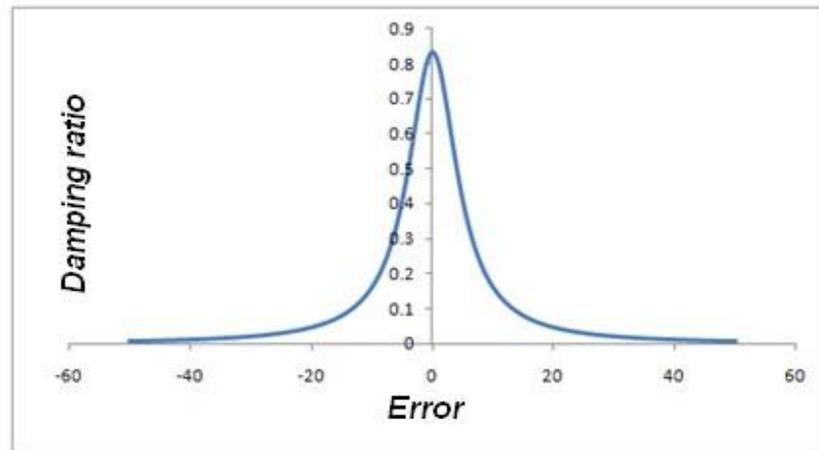


Figure 3.2 – Transient change in damping ratio with error.

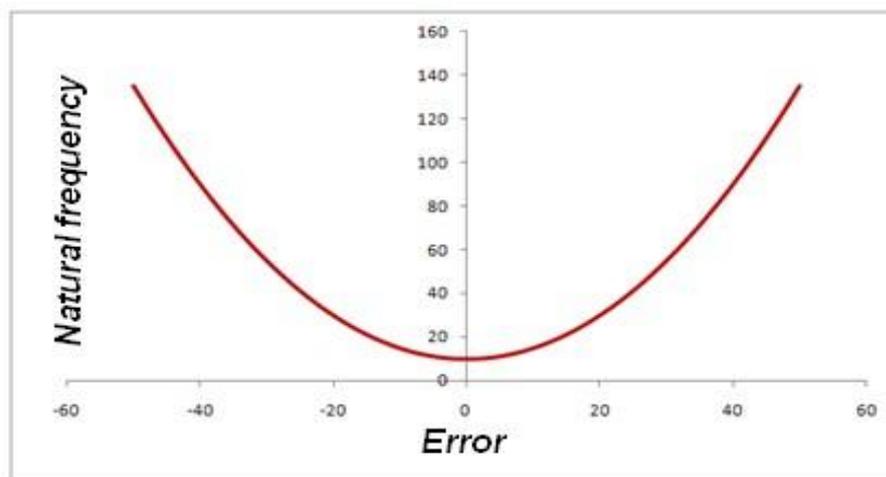


Figure 3.3 – Transient change in natural frequency with error.

Considering the RIDE control structure, this relationship can be achieved if a nonlinear gain matrix is placed before the integrator in the forward path of the outer loop, as shown in Figure 3.4. The regulator is now

$$\dot{z}(t) = N(e(t))K_I e(t) \quad (3.2.5)$$

and the total feed-forward gain is

$$K_{ff}(e(t)) = N(e(t))K_I \quad (3.2.6)$$

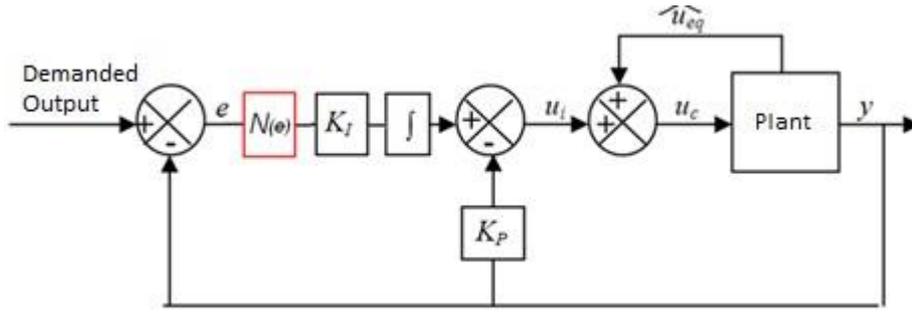


Figure 3.4 – RIDE control structure with integration of VTR gain.

With the nonlinear gain inserted into the outer loop of the RIDE control structure and assuming that $M = C$ the closed loop natural frequency and damping ratio are given by the following relations

$$\omega_{VTR}(e(t)) = \sqrt{CBK_I N(e(t))} \quad (3.2.7)$$

$$\nu_{VTR}(e(t)) = K_P CB \left(2 \sqrt{K_I CB N(e(t))} \right)^{-1} \quad (3.2.8)$$

Equations 3.2.7 and 3.2.8 demonstrate that the addition of a nonlinear, time dependent gain is able to make the transient closed loop characteristics of the system also time dependent. The selection of this time dependent gain must now be chosen so that the target dynamics expressed in Equations 3.2.1 and 3.2.2 are achieved.

If the nonlinear gain is chosen as in Equation 3.2.9, it can be substituted into Equations 3.2.7 and 3.2.8 and the resulting transient characteristics can be compared with those of the target characteristics.

$$N(e(t)) = (Xe(t)^2 + Y)^2 \quad (3.2.9)$$

$$\omega_{VTR}(e(t)) = \sqrt{K_I CB (Xe(t)^2 + Y)} \quad (3.2.10)$$

$$\mathbf{v}_{VTR}(\mathbf{e}(t)) = \mathbf{K}_P \mathbf{C} \mathbf{B} (2\sqrt{\mathbf{K}_I \mathbf{C} \mathbf{B}})^{-1} (\mathbf{X} \mathbf{e}(t)^2 + \mathbf{Y})^{-1} \quad (3.2.11)$$

Comparing Equations 3.2.10 and 3.2.11 with the target transient characteristics it can be seen that with this choice of \mathbf{N} the desired target dynamics are attainable.

The system constants \mathbf{K}_ω and \mathbf{K}_v are given by

$$\mathbf{K}_\omega = \sqrt{\mathbf{K}_I \mathbf{C} \mathbf{B}} \quad (3.2.12)$$

$$\mathbf{K}_v = \mathbf{K}_P \mathbf{C} \mathbf{B} (2\sqrt{\mathbf{K}_I \mathbf{C} \mathbf{B}})^{-1} \quad (3.2.13)$$

3.3 Quasi-linear Equivalents

Equations 3.2.10 and 3.2.11 describe the dynamic relationship between controller error and damping ratio and natural frequency. These relationships describe the transient response characteristics as if “frozen in time” for a given error magnitude. What is also required are relationships that describe the transient response of a VTR controller over a range of error, typically for the controlled output starting at an initial value and eventually reaching the setpoint. This can be described as the emergent transient behaviour of the controller. This overall emergent behaviour will be nonlinear in nature and so cannot be fully described by linear constructs such as damping ratio and natural frequency. These linear descriptions of the systems transient response are extremely useful for expressing the overall behaviour of the controller and are the most common way of describing the transient response. Therefore determining a linear approximation of the nonlinear VTR response is necessary for the controller design process as well as to provide theoretical insight into the controller’s behaviour. This is particularly important as tuning X and Y manually, without any theoretical insight into the overall shape of the resulting dynamics, may be extremely difficult.

Finding linear approximations of nonlinear systems has always been a major part of controller design and synthesis. Very few nonlinear analysis techniques exist and the ones that do, such as Lyapunov based methods (Freeman and Kokotovic, 1996), are often mathematically very complex and limited in their usability. The describing function method is a middle ground between linear and nonlinear analysis techniques as it allows nonlinear elements in control systems to be made quasi-linear and thus analysed using conventional methods.

3.3.1 Exponential Input Describing Function (EIDF)

Describing functions are most often employed in frequency domain analysis of control systems where the effects of a nonlinearity, such as a relay or a saturation element, need to be investigated and its effect on the frequency response of the controller determined. The describing function replaces the real nonlinear element in the system with a quasi linear approximation that can then be used in the analysis of the whole system or its effects studied in isolation. This technique can be used to replace the nonlinear gain element in the VTR design with a quasi-linear equivalent which will then allow for a linear approximation of the overall nonlinear dynamics to be used in the controller design and analysis. In the case of VTR we are most interested in the time domain response of the system and so a special type of describing function known as an Exponential Input Describing Function (EIDF) will be used (Gelb and Vander Velde, 1968), (Janabi and Gray, 1991). This describing function analysis is based on the supposition that as $t \rightarrow \infty$ $e \rightarrow 0$, which, providing that the controller is tracking, is a reasonable assumption. The method also assumes that this decay is exponential and that the input into the describing function block is the error signal.

Since the nonlinear gain N receives the error signal, the input, u_{DF} , to the describing function equivalent, N_{DF} , can be given by the following relation.

$$\mathbf{u}_{DF}(t) = \mathbf{E} \exp^{-t} \quad (3.3.1)$$

Where \mathbf{E} is a matrix giving the maximum amplitude of the error during the controlled response for each controller channel

$$\mathbf{E} = \begin{bmatrix} |\text{emax}_i| & 0 & 0 & 0 \\ 0 & |\text{emax}_{i+1}| & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & |\text{emax}_n| \end{bmatrix} \quad (3.3.2)$$

It can be assumed that for a step response with no overshoot emax is approximately the difference between the initial output and the setpoint.

$$\text{emax}_i = y_{ci} - y_{(o)i} \quad (3.3.3)$$

The EIDF is then found by minimizing the integral squared error of the output of the EIDF with respect to the output of the real nonlinearity.

$$\mathbf{e}_{DF}(t) = \mathbf{N}_{DF} \mathbf{u}_{DF}(t) - \mathbf{N}(t) \mathbf{u}_{DF}(t) \quad (3.3.4)$$

$$\mathbf{e}_{DF}(t) = \mathbf{N}_{DF} \mathbf{u}_{DF}(t) - \mathbf{y}_N(t) \quad (3.3.5)$$

$$\int_0^\infty \mathbf{e}_{DF}^2(t) dt = \mathbf{N}_{DF}^2 \int_0^\infty \mathbf{u}_{DF}^2(t) dt - 2\mathbf{N}_{DF} \int_0^\infty \mathbf{u}_{DF}(t) \mathbf{y}_N(t) dt + \int_0^\infty \mathbf{y}_N^2(t) dt \quad (3.3.6)$$

Minimizing by differentiating with respect to \mathbf{N}_{DF} and equating to zero yields the EIDF.

$$\mathbf{N}_{DF} = \frac{\int_0^\infty \mathbf{u}_{DF}(t) \mathbf{y}_N(t) dt}{\int_0^\infty \mathbf{u}_{DF}^2(t) dt} \quad (3.3.7)$$

Using the chosen nonlinear function from Equation (3.2.9)

$$\mathbf{y}_N(t) = (\mathbf{X}\mathbf{e}(t)^2 + \mathbf{Y})^2 \mathbf{E} \exp^{-t} \quad (3.3.8)$$

Therefore, the EIDF is given by

$$\mathbf{N}_{DF} = \frac{2}{5} \mathbf{E}^3 \mathbf{X}^2 + \frac{4}{3} \mathbf{E} \mathbf{X} \mathbf{Y} + \frac{2}{\mathbf{E}} \mathbf{Y}^2 \quad (3.3.9)$$

This results in quasi-linear gain that is dependent on the maximum amplitude of its input, E.

3.3.2 Average of the Nonlinearity over the Error Space

Another potential method of finding an approximate quasi-linear equivalent of the nonlinear VTR gain is to average the nonlinear function over the expected error space. In essence, this will determine the average gain from the initial error to the final error value.

The average of a function, $\mathbf{fn}(\mathbf{x})$, over the space $\mathbf{x}_{(0)}$ to $\mathbf{x}_{(f)}$ is given by

$$\mathbf{fn}_{\text{avg}} = \frac{1}{\mathbf{x}_{(f)} - \mathbf{x}_{(0)}} \int_{\mathbf{x}_{(0)}}^{\mathbf{x}_{(f)}} \mathbf{fn}(\mathbf{x}) \, d\mathbf{x} \quad (3.3.10)$$

For the nonlinear VTR gain the average, over the error space defined by Equation (3.3.3), is given by

$$\mathbf{N}_{\text{avg}} = \frac{1}{\mathbf{E} - 0} \int_0^{\mathbf{E}} (\mathbf{X}\mathbf{e}(t)^2 + \mathbf{Y})^2 \, d\mathbf{e} \quad (3.3.11)$$

$$\mathbf{N}_{\text{avg}} = \frac{1}{5} \mathbf{E}^4 \mathbf{X}^2 + \frac{2}{3} \mathbf{E}^2 \mathbf{X} \mathbf{Y} + \mathbf{Y}^2 \quad (3.3.12)$$

It can be seen by comparing this to N_{DF} that the two results are quite similar but the main difference is that N_{avg} has the dependence on E raised by a power.

3.4 Comparison of Approximate Linear Equivalents to Real Nonlinear Gain

A series of comparisons were undertaken in order to assess the accuracy of the two approximate methods in finding a quasi-linear approximation of the real VTR gain. Firstly, a decreasing exponential input with a maximum values equal to E was fed into the three gains; $N_{(t)}$, N_{DF} and N_{avg} and the resulting outputs compared. The values of X , Y and E were adjusted individually whilst holding the other two parameters constant to examine the sensitivity of the approximations to variations in these parameters.

3.4.1 Exponential Input, Variation of X

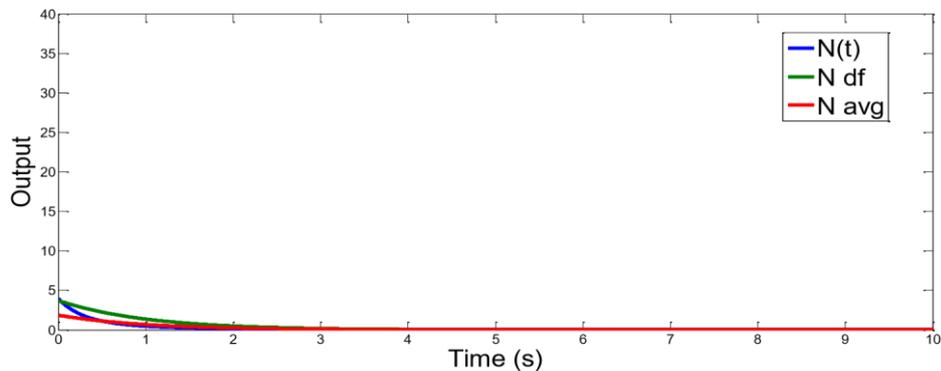


Figure 3.5 – Exponential Input: $X=1$, $Y=1$, $E=1$.

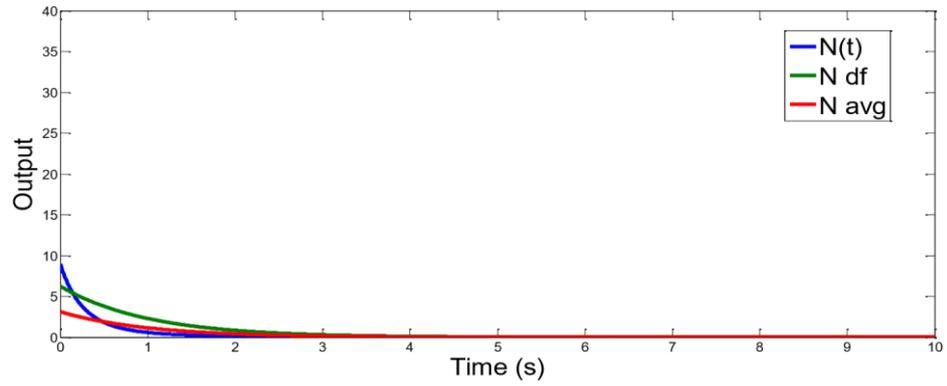


Figure 3.6 – Exponential Input: X=2, Y=1, E=1.

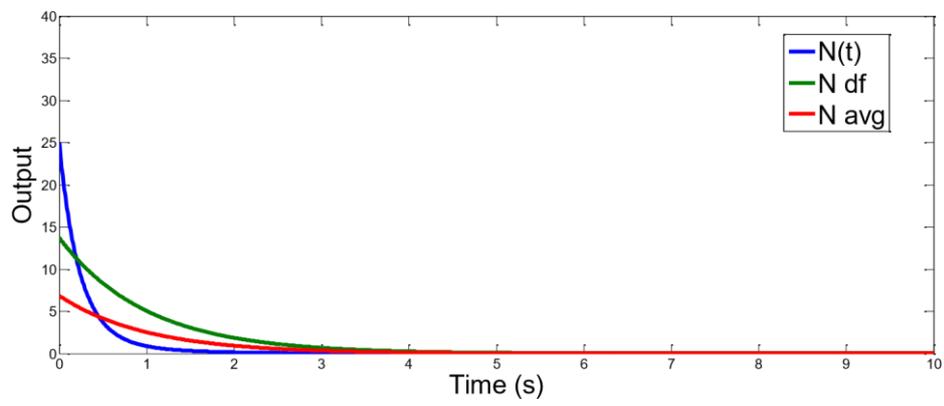


Figure 3.7 – Exponential Input: X=4, Y=1, E=1.

3.4.2 Exponential Input, Variation of Y

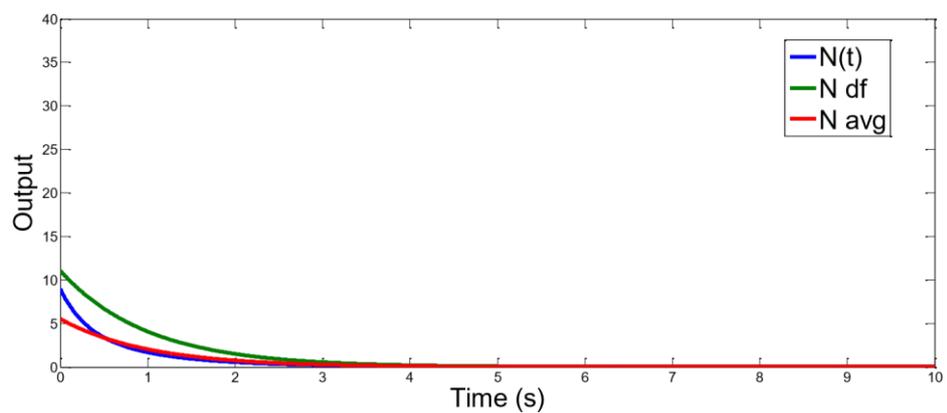


Figure 3.8 – Exponential Input: X=1, Y=2, E=1.

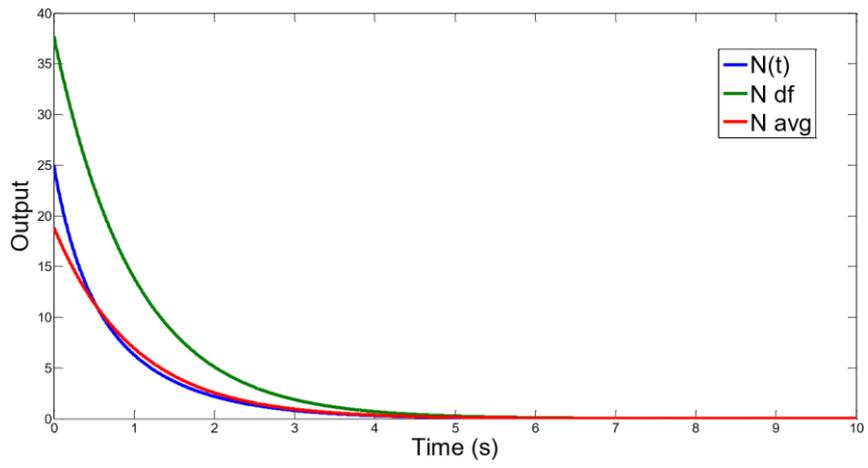


Figure 3.9 – Exponential Input: X=1, Y=4, E=1.

3.4.3 Exponential Input, Variation of E

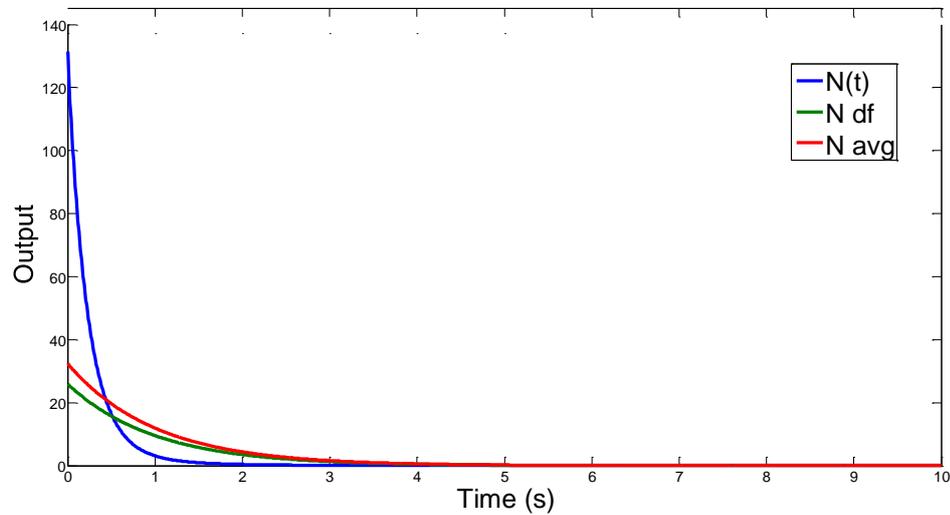


Figure 3.10 – Exponential Input: X=1, Y=1, E=2.5.

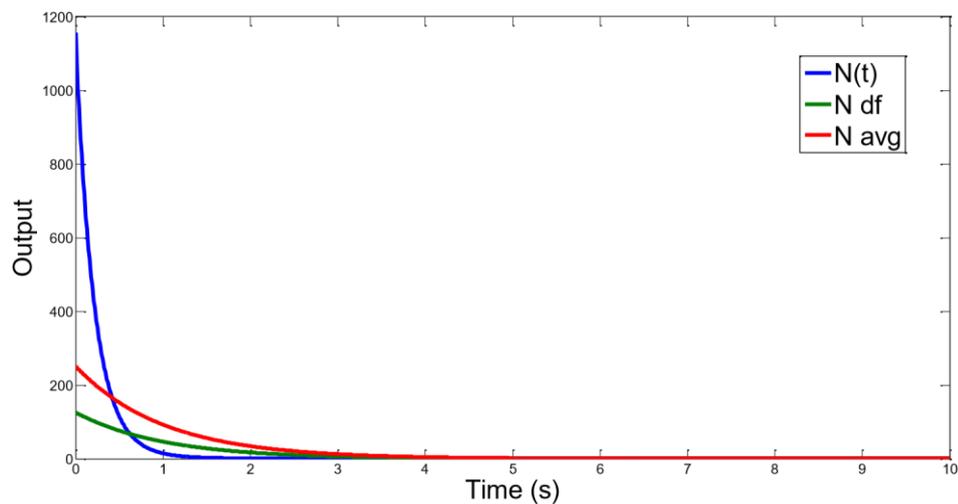


Figure 3.11 – Exponential Input: X=1, Y=1, E=5.

The comparison studies show that as the parameters are increased the accuracy of the approximations decreased, which is to be expected since increasing the parameters increases the effect of the nonlinearity in the original function. Varying X, as shown in Figures 3.5 to 3.7, resulted in good matches for both N_{DF} and N_{avg} . The describing function method provided a closer match for higher

output values whilst the average function method was a better match for lower output values. Varying Y , as shown in Figures 3.8 and 3.9, produced very close matches, especially for N_{avg} . The most sensitive parameter was clearly E . From Figures 3.10 and 3.11 it can be seen that a small change in its value resulted in a very large increase in the maximum output from the VTR nonlinearity. Matches for N_{DF} and N_{avg} were very similar and both became more inaccurate as E was increased. As they are both linear approximations either an underestimation or overestimation of the effect of the parameter E is to be expected. The results show that both methods significantly underestimate the effect of increasing the maximum error.

The second study was a comparison of the two approximation techniques when the quasi linear gains were substituted for the full nonlinear gain in a closed loop control system. The control structure used was that of RIDE but with no

Equivalent Control and a plant equal to $\frac{K}{s}$. The input was a step, with amplitude equal to E . The output, y , was plotted. It should be noted that this is not an investigation into the performance of the controller but merely a comparison of the ability of the two techniques to approximate the real nonlinear VTR response.

3.4.4 Closed Loop Response, Variation of X

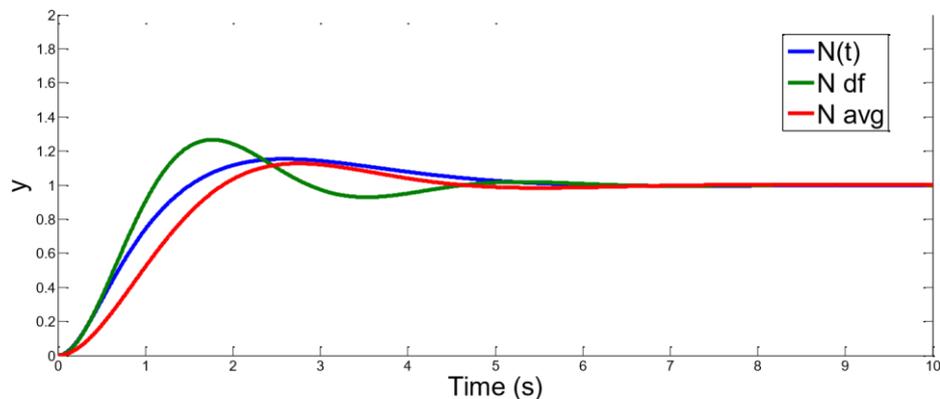


Figure 3.12 – Closed Loop Response: X=1, Y=1, E=1.

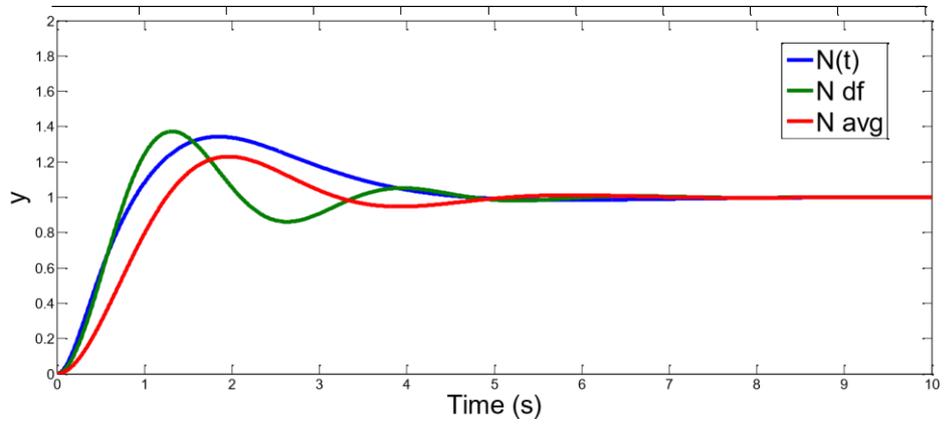


Figure 3.13 – Closed Loop Response: X=2, Y=1, E=1.

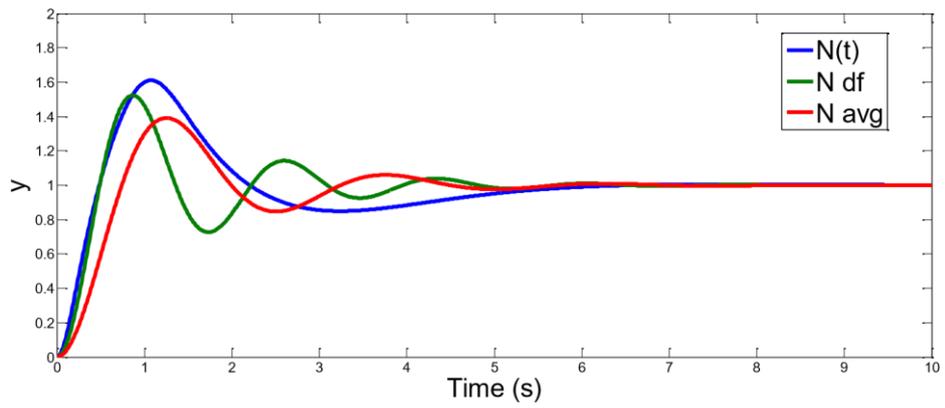


Figure 3.14 – Closed Loop Response: X=4, Y=1, E=1.

3.4.5 Closed Loop Response, Variation of Y

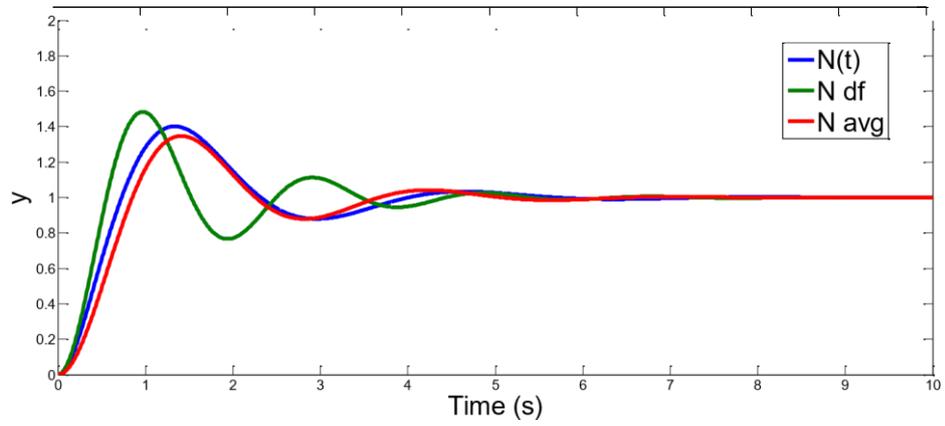


Figure 3.15 – Closed Loop Response: $X=1$, $Y=2$, $E=1$.

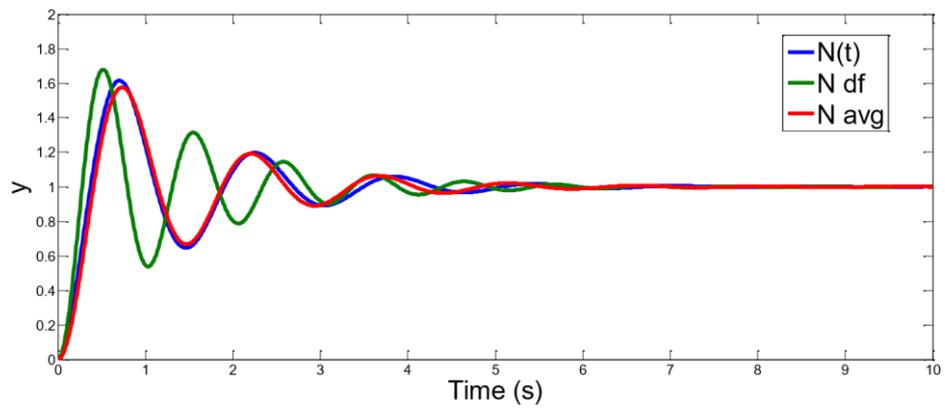


Figure 3.16 – Closed Loop Response: $X=1$, $Y=4$, $E=1$.

3.4.6 Closed Loop Response, Variation of E

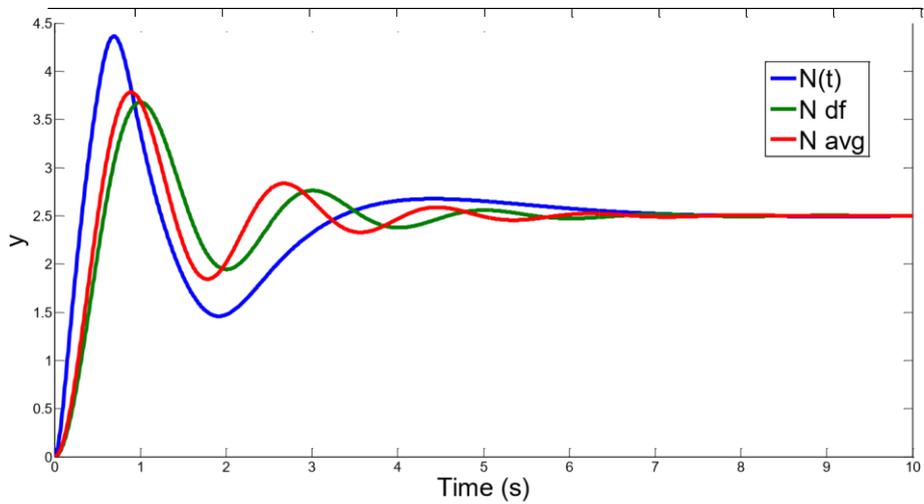


Figure 3.17 – Closed Loop Response: $X=1$, $Y=1$, $E=2.5$.

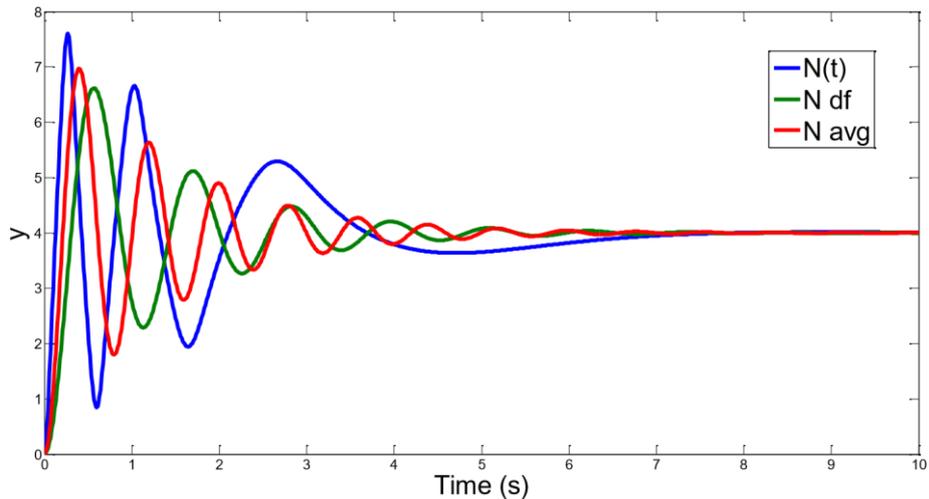


Figure 3.18 – Closed Loop Response: $X=1$, $Y=1$, $E=4$.

Figure 3.12 through to Figure 3.18 illustrate that both the exponential input describing function and the average of the function are able to find a good quasilinear approximation to the nonlinear response of the VTR controller. It appears that the EIDF method is better able to predict the response time of the

VTR controller when there are variations in X , whereas the average method more accurately predicts the damping characteristics. The ability to accurately approximate the response with variations in X is extremely important as the value of X determines the strength of VTR's nonlinearity. Figure 3.15 and Figure 3.16 demonstrate that a very accurate match can be achieved with the average function method whilst the fit using the EIDF is worse. Both methods are equally effective at providing a good approximation when the maximum error is varied.

In conclusion, based on this study, it is difficult to make a recommendation for one method over the other as they are both able to make adequate quasi-linear approximations of the nonlinear response of the VTR controller. The most appreciable difference between the methods is that the EIDF is better able to predict the response time or natural frequency characteristics of the VTR design; by contrast, the average function method results in a better approximation of the damping characteristics. The selection of the method will depend on whether the accurate design of the final natural frequency or damping ratio of the controller is more important.

3.5 VTR Controller Synthesis

Section 3.3 described the determination of quasi-linear gains that were equivalent to the nonlinear VTR gain, $N(\mathbf{t})$. If the nonlinear gain is substituted for the quasilinear gain N_{QL} , then the VTR controller can be designed in a similar fashion to the original RIDE controller.

Substituting N_{QL} into Equations (3.2.7) and (3.2.8) yields expressions for the overall equivalent natural frequency and damping ratio for the closed loop VTR controller.

$$\omega_{\text{VTR eq}} = \sqrt{\mathbf{K}_I \mathbf{C} \mathbf{B} \mathbf{N}_{\text{QL}}} \quad (3.5.1)$$

$$\mathbf{v}_{\text{VTR eq}} = \mathbf{K}_P \mathbf{C} \mathbf{B} (2\sqrt{\mathbf{K}_I \mathbf{C} \mathbf{B} \mathbf{N}_{\text{QL}}})^{-1} \quad (3.5.2)$$

These expressions can be simplified providing that the RIDE controller has been used, as the closed loop dynamics of the RIDE controller are specified as $\omega^2 = \mathbf{K}_I \mathbf{C} \mathbf{B}$ and $2\mathbf{v}\omega = \mathbf{K}_P \mathbf{C} \mathbf{B}$.

$$\omega_{\text{VTR eq}} = \omega \sqrt{\mathbf{N}_{\text{QL}}} \quad (3.5.3)$$

$$\mathbf{v}_{\text{VTR eq}} = \mathbf{v} (\sqrt{\mathbf{N}_{\text{QL}}})^{-1} \quad (3.5.4)$$

Using Equations 3.5.3 and 3.5.4 it is possible to relate the change in the overall transient response characteristics caused by the addition of VTR to the original RIDE transient characteristics. From a controller design point of view this gives the designer a simple description of the overall time domain behaviour that can then be used to design the VTR parameters X and Y.

The initial assumption would be to select desired values of $\omega_{\text{VTR eq}}$ and $\mathbf{v}_{\text{VTR eq}}$, then solve Equations 3.5.1 and 3.5.2 simultaneously to determine values of X and Y. This is, however, quite impractical as simultaneous solutions for many combinations of $\omega_{\text{VTR eq}}$ and $\mathbf{v}_{\text{VTR eq}}$ do not exist. Furthermore, this method does not align particularly well with the philosophy of VTR as there is no way of specifying the steady state gain and thus the steady state sensitivity to higher order un-modelled dynamics present in the actuator and sensor systems.

A more effective tuning strategy is to select a desired natural frequency and damping ratio for the RIDE controller and a value of $\mathbf{Y} < 1$ so that speed of

response is maintained but overshoot and steady state oscillations are reduced due to the reduced steady state integral gain. Since the goal is to preserve the natural frequency of the RIDE controller then $\omega_{VTR\ eq} = \omega$, hence

$$\sqrt{\mathbf{N}_{QL}} = \mathbf{I} \quad (3.5.5)$$

Where \mathbf{I} is the identity matrix. For \mathbf{N}_{DF}

$$\frac{2}{5}\mathbf{E}^3\mathbf{X}^2 + \frac{4}{3}\mathbf{E}\mathbf{X}\mathbf{Y} + 2\mathbf{E}^{-1}\mathbf{Y}^2 = \mathbf{I} \quad (3.5.6)$$

and for \mathbf{N}_{avg}

$$\frac{1}{5}\mathbf{E}^4\mathbf{X}^2 + \frac{2}{3}\mathbf{E}^2\mathbf{X}\mathbf{Y} + \mathbf{Y}^2 = \mathbf{I} \quad (3.5.7)$$

For a given value of \mathbf{Y} this can then be solved for \mathbf{X}

$$\mathbf{X}_{DF} = (-10\mathbf{E}^2\mathbf{Y} + \sqrt{10\sqrt{9\mathbf{E}^5 - 8\mathbf{E}^4\mathbf{Y}^2}})(6\mathbf{E}^4)^{-1} \quad (3.5.8)$$

$$\mathbf{X}_{avg} = (-5\mathbf{E}^2\mathbf{Y} + \sqrt{5\sqrt{9\mathbf{E}^4 - 4\mathbf{E}^4\mathbf{Y}^2}})(3\mathbf{E}^4)^{-1} \quad (3.5.9)$$

This now gives the value of \mathbf{X} required for any given value of \mathbf{Y} in order to preserve the overall natural frequency, and hence response time of the controller. The parameter \mathbf{Y} is effectively the steady state integral gain reduction factor and by choosing a corresponding value of \mathbf{X} using Equation (3.5.8) or (3.5.9) there will be no penalty paid in terms of speed of response to the setpoint.

In summary the following design procedure is proposed:

1. Design the RIDE control system to achieve desired closed loop natural frequency and damping with gains K_I and K_P . If the desired closed loop dynamics are not obtainable due to overshoot or oscillatory dynamics a VTR design is required.

2. Retune the affected channels of the RIDE controller with a reduced integral gain until oscillations or overshoot are no longer present. This is defined as K_{Icrit} .

3. Set $Y = \sqrt{\frac{K_{Icrit}}{K_I}}$. This will result in the steady state VTR integral gain being equal to K_{Icrit} . A safety factor may be applied to Y if extra robustness is required.

4. Use Equation (3.5.8) or (3.5.9) to determine the value of X so that the desired response time of the controller is maintained but with a lower steady state gain. Further manual tuning of X may be needed to achieve fully meet performance objectives.

3.5.1 Setpoint Dependence and Selection of e_{max}

The most important consideration when deciding if a VTR controller is suitable for control of a system is the setpoint that it is required to track. This is due to the algorithm being driven by the error, which is determined by the setpoint. A key controller design consideration for VTR is the expected maximum error.

When considering a step response it is apparent from Section 3.4 that the overall

transient response of the controller is dependent on the maximum amplitude of the error during a step change in setpoint. Figure 3.19 illustrates the change in overall transient behaviour as the setpoint (hence the maximum error amplitude) is altered for a fixed set of controller parameters. It is clear from this and Equations 3.5.3, 3.5.4, 3.3.11 and 3.3.8 that, for a controller tuned for a specific error amplitude, increasing the error amplitude will result in an under damped response and a decrease would result in an over damped response. Therefore, in order to maintain consistent performance across a range of error amplitudes it would be necessary to alter the parameters X and Y with the change in error amplitude. This is demonstrated by Figure 3.20 which plots Equation (3.5.8) and illustrates that X should be decreased as the maximum error amplitude is increased for a specified value of Y . This approach would require tuning the controller at multiple error amplitudes and accordingly scheduling the X and Y parameters. This may be a practical solution for a setpoint that varies in predefined step changes, as is typical in process and internal climate control. For systems with a continuously varying setpoint the notion of maximum error amplitude becomes very difficult to define. A properly functioning controller tracking a continuous setpoint will never completely track; instead, there will be a phase lag determined by the bandwidth of the controller. The constant phase lag results in a constant error and thus the VTR gain will remain constant. This may result in undesirable performance and, indeed, have the opposite effect to that intended by the introduction of VTR: i.e. reduced sensitivity to fast unmodelled dynamics. This is due to the integral gain increasing with error, so a large enough constant error signal will result in a permanently raised integral gain.

The ideal setpoint for a VTR controller would be a series of step changes where the controller is able to track before moving onto the next step. This is a common scenario in process control and HVAC control amongst others. As well as being

closer to the approximations made in the design process it allows for normalization of the setpoint and therefore elimination of the setpoint dependency on dynamic response.

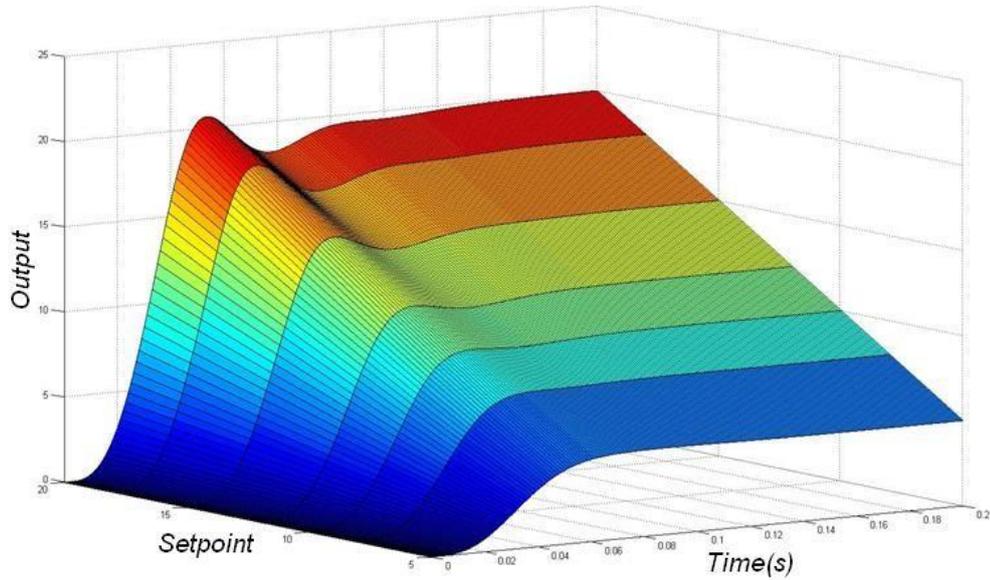


Figure 3.19 – Increase in overshoot as the setpoint is increased for fixed VTR parameters.

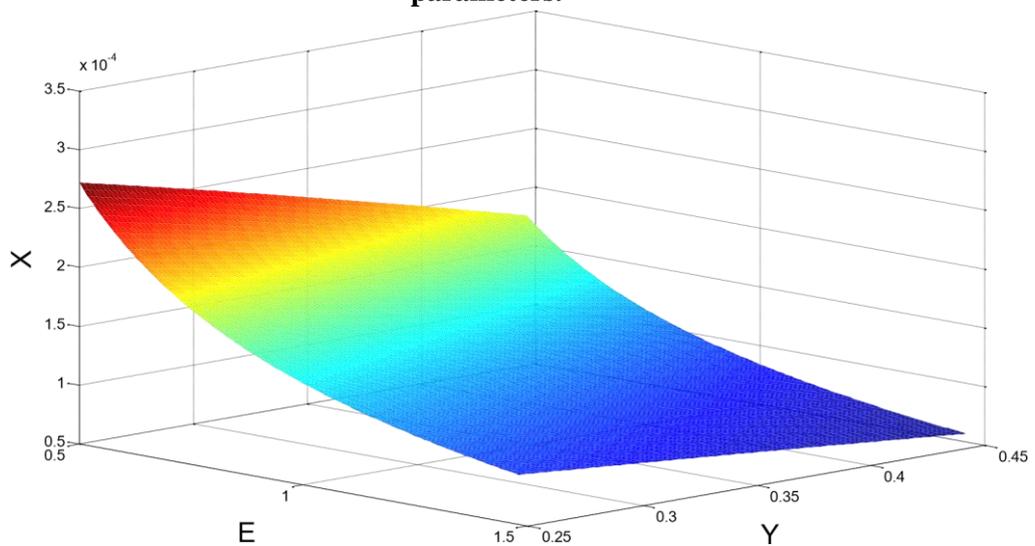


Figure 3.20 – Relationship between, X, Y and E for constant equivalent transient response.

3.5.2 Setpoint normalization

In simple terms, normalizing the setpoint means that the VTR algorithm always has the same maximum error amplitude and therefore performance is consistent with the same set of parameters, regardless of the actual setpoint. The drawback is that this requires easily identifiable steps in setpoint, but as already discussed, this is a prerequisite for VTR controller implementation. Importantly, it is not the setpoint that is actually normalized but e_{max} . Starting from an initial condition, e_{max} is defined as in Equation 3.3.2. If the setpoint is being tracked when the step change is made then e_{max} is simply the new setpoint subtracted from the previous setpoint. The goal of the normalization is to make this e_{max} the same no matter the change in setpoint. This first requires modifying the error that is fed into the VTR function. A new variable can be created called e_{VTR}

$$e_{VTR} = \frac{e}{\text{new setpoint} - \text{previous setpoint}} \quad (3.5.9)$$

If this is fed into the VTR function then e_{max} will always be equal to one.

Crucially, it does not affect the operation of the VTR function as it is still being driven by the error. The only change will be constancy of performance over a range of setpoints.

If this approach is adopted, the design process becomes simplified. E is now an identity matrix of dimension $n \times n$. This reduces the design equations to the following:

$$\mathbf{X}_{DF} = (-10\mathbf{Y} + \sqrt{10}\sqrt{9 - 8\mathbf{Y}^2})/6 \quad (3.5.10)$$

$$\mathbf{X}_{avg} = (-5\mathbf{Y} + \sqrt{5}\sqrt{9 - 4\mathbf{Y}^2})/3 \quad (3.5.11)$$

This results in a very simple equation for calculation of X for a given Y. It is then possible to produce a table, which can be referred to in the design process, of values of X for corresponding values of Y.

Table 3.2 – Setpoint normalized values of X for values of Y.

Y	X_{DF}	X_{AVG}
0.5	0.56	1.27
0.6	0.30	1.05
0.7	0.02	0.81
0.8	-0.30	0.56
0.9	-0.66	0.29

It can be seen from Table 3.2 that the predicted values for X vary considerably. The robustness of the predictions using the describing function method can be called into question as the X values for a Y of 0.8 and 0.9 are negative and so are not feasible in practice. This, combined with the results of the analysis in section 3.4, suggests that using a quasi-linearisation based on the average of the function is a more accurate and robust method of determining VTR controller parameters.

3.6 Design Process – Illustrative Example

The following illustrative example of the VTR controller design process is presented to demonstrate the proposed design steps and to investigate the characteristics of the theoretical tuning methods in a general manner.

Consider the following general and hypothetical MIMO linear time invariant system described in state space form:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} -5 & -3 \\ -2 & -2 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 3 & 1 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \quad (3.6.1)$$

$$\mathbf{y} = [x_1 \quad x_2]^T \quad (3.6.2)$$

The inputs are subject to second order actuator dynamics given by

$$\ddot{\mathbf{u}}(t) = -2\zeta_a\omega_a\dot{\mathbf{u}}(t) - \omega_a^2(\mathbf{u}_c(t) - \mathbf{u}(t)) \quad (3.6.3)$$

An Inverse Dynamics controller designed using RIDE methods assumes that the bandwidth of the actuator dynamics, ω_a , is sufficiently high so that they can be considered to be at a quasi steady state compared to the bandwidth of the closed loop system. Therefore, they are not included as part of the controller design state space system and are treated as un-modelled dynamics. Initially the actuator is given a natural frequency of 150 rad/s.

The RIDE controller is designed to achieve a closed loop bandwidth of 40 rad/s. This provides enough separation between the closed loop and actuator bandwidth so that the quasi-steady state assumption of the un-modelled dynamics is justified. Setpoints are constant values of 10 and 4 for y_1 and y_2 respectively. The controlled output is shown in Figure 3.21

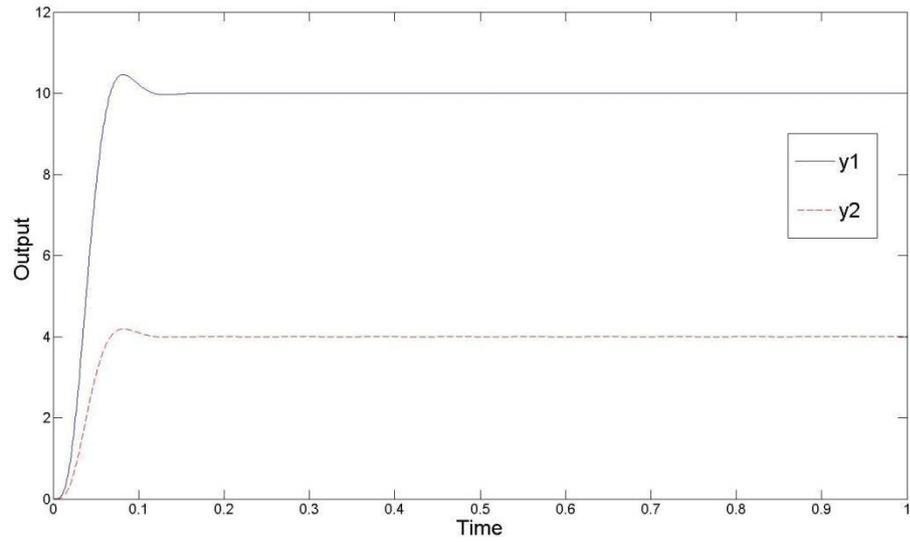


Figure 3.21 – RIDE controller response.

The validity of the assumption that the un-modelled dynamics are quasi-steady state is weakened if the actuator bandwidth is reduced to 70 rad/s. This causes unstable oscillatory behaviour to occur, as shown in Figure 3.22.

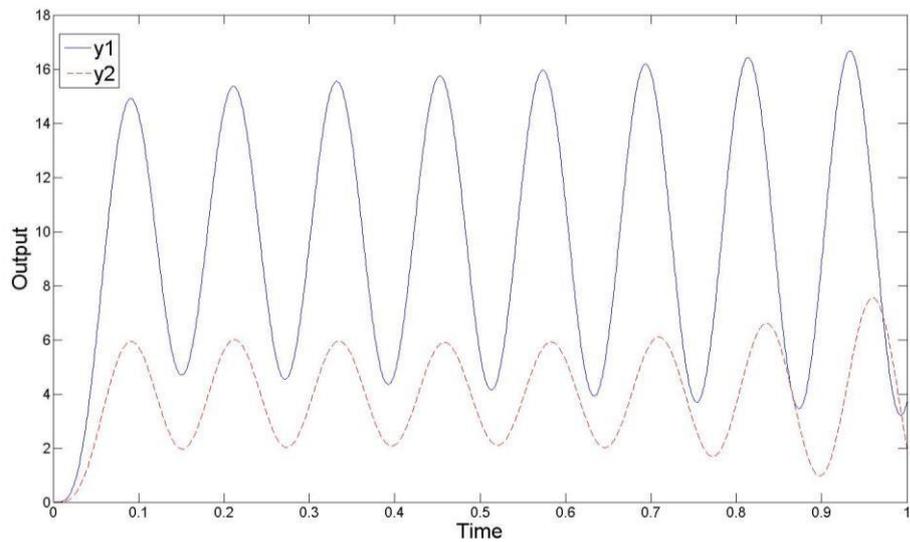


Figure 3.22 – RIDE controller response with large actuator inertia.

Due to the oscillations a VTR controller design is required to maintain speed of response without undesirable dynamics resulting. The first stage is to reduce the steady state integral gain by setting $Y_i < 1$ for each channel. A value of 0.5 was selected which provides adequate steady state damping as can be seen from Figure 3.23 where the VTR parameters X are set to zero so as to observe the effect of Y only.

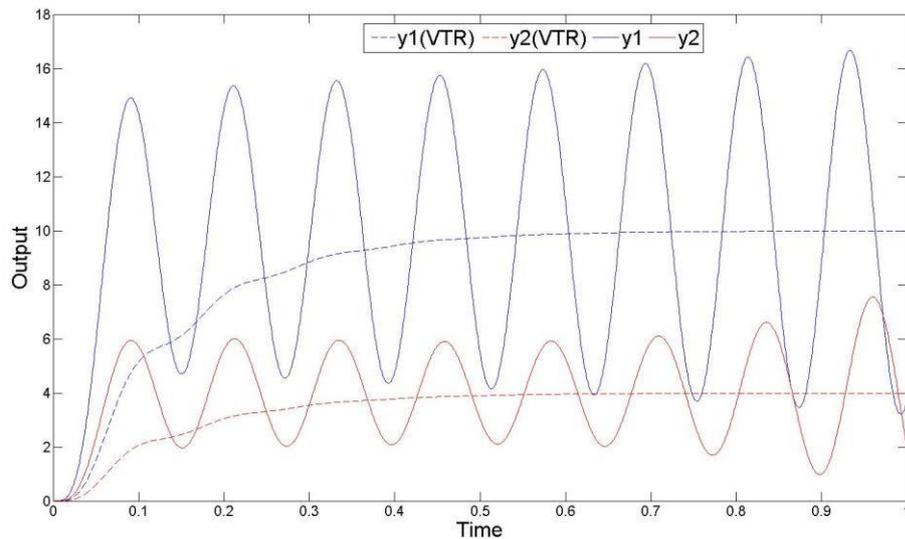


Figure 3.23 – Comparison of RIDE output with normal and reduced integral gain.

The purpose of the VTR controller is to preserve the speed of response of the original controller whilst damping the oscillations due to un-modelled dynamics. Therefore, the matrix X needs to be designed in order to achieve this. As can be seen from Equations 3.5.9 and 3.5.10 the correct selection of X depends only on the Y matrix and the maximum expected error. The Y matrix has been determined so the maximum error, E , needs to be decided upon. The plots of the controller's response without VTR give us a reasonable idea of the maximum error to be expected. Assuming that there will be initially a similar overshoot when using the VTR controller compared to the original a maximum error of 15 for the first channel and 6 for the second can be expected. We now have all the information required to calculate the X matrix based on the quasi linear

approximation method. Equation 3.5.10 gives the X matrix, with a predetermined Y and E matrix, for the Average Function method.

Inputting the Y and E matrices into this equation results in value of X

$$\mathbf{X} = \begin{bmatrix} 0.0065 & 0 \\ 0 & 0.0354 \end{bmatrix} \quad (3.6.4)$$

Figures 3.24 and 3.25 show the simulation results using the VTR controller design method using these parameters.

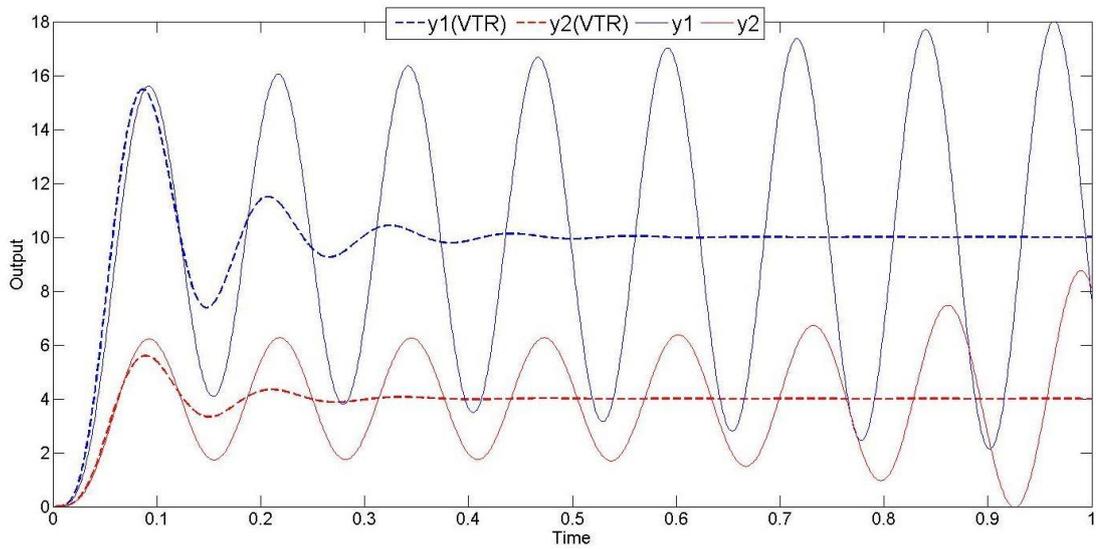


Figure 3.24 – VTR controller response in comparison to RIDE.

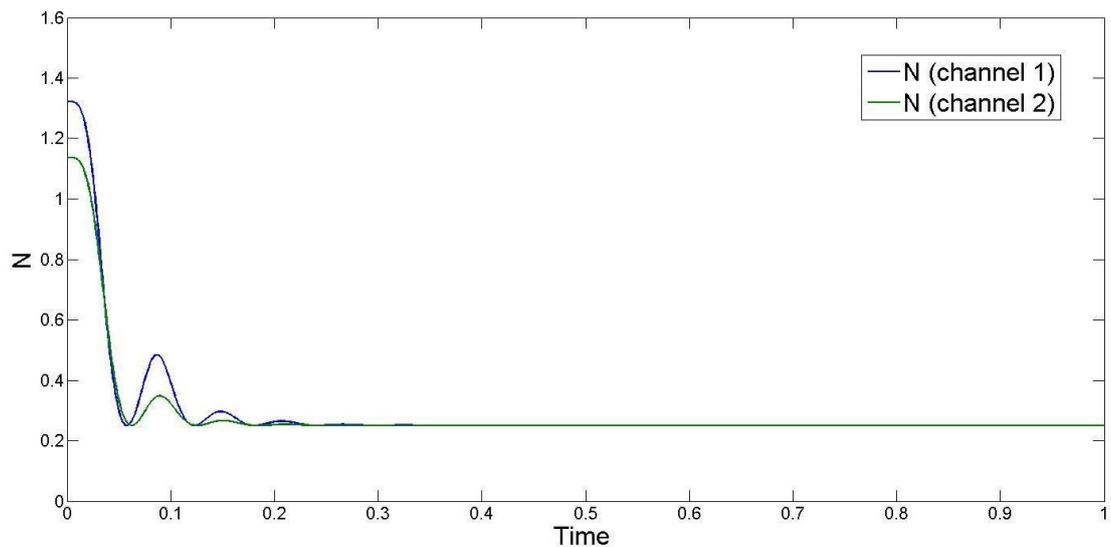


Figure 3.25 – Transient change in nonlinear function, N.

The plots in Figures 3.24 and 3.25 demonstrate that the estimated values for X and Y results in the design objectives being achieved. Namely, the response time of the controller is maintained and a smooth transition of the integral gain ensures that the oscillations about the setpoint are reduced.

This example is, of course, perfectly linear and makes a number of assumptions and approximations for simplicity and clarity. Real-world systems are never fully linear and as such the estimated X and Y values resulting from the controller design process may require further tuning in order to obtain best performance.

3.7 Conclusions

This chapter has presented a systematic method for designing a high performance control system with the aim of reducing controller oscillations when actuators or sensors with significant inertias are present in the system. This is achieved

through the use of a nonlinear function which reduces the systems natural frequency as the setpoint is approached. This in turn reduces the controller's sensitivity to exciting the higher order dynamics present in actuator and sensor systems, but without compromising response time.

The nonlinear function requires two parameters to be tuned, X and Y. This has proven a difficult challenge to overcome in similar controller designs and often trial and error is required. The VTR controller design uses a quasi-linear approximation of the system, which is then used in the specification of desired quasi-linear natural frequency and damping, from which best estimates of X and Y can be derived. This design process results in a simple expression from which, given a desired setpoint and steady-state gain reduction factor, a value of X can quickly be found for a corresponding value of Y. This makes the controller design process relatively simple for an initially complex problem.

Two methods were investigated of forming a quasi-linear approximation of the nonlinear VTR system; the first, relatively simple, method was to find an average of the function over the error space; the second, more complex, method was to invoke a time domain interpretation of the describing function method (the Exponential Input Describing Function). Perhaps surprisingly, better results were achieved with the more simple average function method. The effect of increasing the setpoint on the resulting overall transient response was underestimated by the EIDF method, resulting in estimated parameters that were too large.

The issue of setpoint magnitude on the performance of the VTR controller was investigated in detail. Increasing the setpoint increased controller overshoot, consequently requiring a smaller value of X to compensate. This was reflected in the quasi-linear approximations in the design process of the VTR parameters. However, there was a problem encountered when a range of setpoints were

encountered during the controller operation. Two solutions were proposed for this problem.

The first was to simply design the VTR parameters for a setpoint that is between the expected extremes. Therefore, for setpoints lower in magnitude than the designed setpoint the overall response will be over-damped and for setpoints larger than the designed the overall response will be under-damped. A compromise has to be reached that satisfies performance objectives across a range of setpoints.

The second, more complex, solution was to normalise the setpoint. Using this method performance can be kept consistent for each setpoint without any variation or scheduling of the VTR parameters. In fact, by normalising the setpoint, the design of the VTR parameter X reduces to a one dimensional problem: the solution is only dependent on the steady state gain reduction factor, Y . The extra complexity added to the controller design by normalising the setpoint is somewhat offset by the reduction in complexity of the parameter design problem.

Clearly, the complexity of the VTR design is heavily dependent on the range and type of setpoints that will be encountered during controller operation. Certain classes of systems will request types and ranges of setpoints making certain types of system more suitable than others for a VTR controller design. The “ideal” setpoint for VTR operation is a step change, where the step size is either constant or varies over a small range. Such setpoints are often typical in process control, manufacturing and HVAC control scenarios. A system which has a setpoint that results in a constant phase lag, such as a ramp or a sinusoidal wave, will not gain any benefit through the use of a VTR controller design. This is because the constant phase lag means the error is constant and therefore the VTR gain will be

constant. Therefore, systems that have continuously varying setpoints will not be particularly suitable for a VTR design as the benefits may not be significant.

The VTR design method is multivariable but the decoupling of the controlled system is achieved through the RIDE framework. A subtle but extremely important consideration for effective decoupling with the VTR controller is the placement of the nonlinear function, N , in the controller signal chain. It is clear that the nonlinear function must be placed before the RIDE decoupling gain K_I .

The disturbance rejection properties of the VTR controller design can be split into two categories: one for low amplitude disturbance and one for high amplitude disturbances. In the case of low amplitude disturbances the deviation from the setpoint will be minimal and so the dynamic function of the VTR controller will have minimal effect on the controller performance. Instead, the VTR controller will simply operate as a RIDE controller but with a reduced integral gain. Hence, the lower integral action will reduce the controller's disturbance rejection properties compared to a purely RIDE controller. A large amplitude disturbance will cause a significant deviation from the setpoint. This will result in a large error and consequently the VTR controller will be able to increase the integral action, bringing the output quickly back to the setpoint. Therefore, in the case of large amplitude disturbances the disturbance rejection of the VTR controller may be better than that of a purely RIDE controller.

4. Rate Actuated Inverse Dynamics

4.1 Introduction

Chapter 3 described a potential solution to the problem of achieving high performance, Inverse Dynamics based control when there are elements with slow dynamics, lags or large inertias present in the system. These elements are often the actuators as well as the sensors in the control system. There is, however, another factor that is common to all actuation systems that must be taken into account to achieve high performance control – the physical limits of the actuator. These can be split into amplitude limits and rate of change limits. Often these limits are imposed by the controller designer to be less than the full capabilities of the actuator to ensure that they are always achievable or to avoid over stressing the actuator and system to be controlled.

When reached, these limitations often cause instability if the system has not been designed to account for prolonged operation on these limits (Fielding and Flux, 2003). This has resulted in many control systems being designed so that the actuator limits are never reached. This conservative approach, whilst negating the problem of nonlinear stability, means that the performance potential of the system is never fully realised. Conversely, controllers can be designed in a normal fashion but with special design features, known as anti-windup designs, to modify the controller operation when actuator limits are reached.

There has been a great deal of research on anti-windup techniques for amplitude limits (a general overview of recent anti-windup advances is found in Tabouriech and Turner, 2009). However, there is significantly less research on anti-windup techniques for rate limits or rate and amplitude limits, despite rate limits being

present to some degree in almost all actuation systems. For instance, any actuator powered by an electric motor will have a rate of change of angle constraint governed primarily by the maximum voltage of the motor. If the rate of change of the control signal is unrestricted and the rates of change limits are severe enough, then catastrophic performance degradation can occur (Fielding and Flux, 2003). This is mainly due to the phase lag which occurs between the controller output and the actuator output when rate limits are reached (Yildiz et al., 2011). Therefore, input rate limits must be taken into account during the design of a high performance controller for it to be successfully implemented.

A traditional anti-windup compensator (Figure 4.1) design involves the construction of an extra feedback loop which reduces the control output when input limits are reached. Recent work in designing anti-windup controllers with rate saturation has used a derivative of the control signal as the controller output and anti-windup compensator gains designed using Linear Matrix Inequalities (LMI), L_2 and Linear Quadratic Regulator (LQR) techniques (Galeani et al., 2008), (Brieger et al., 2009), (Kahveci and Ioannou, 2008), (Forni et al., 2010), (Hu et al., 2008), (Biannic and Tarbouriech, 2009). Although deflection limit compensation has recently been developed for general Inverse Dynamics based controllers (Menon et al., 2008), there has been little research published on rate or rate and deflection limiting in Inverse Dynamics controller designs. Shin et al. (2008) addressed the problem of using neural networks with an actuator with rate and amplitude limits in an Inverse Dynamics design but the issue of anti-windup was not investigated.

Other recent methods for controller design with rate saturation involve phase compensation techniques to reduce the phase lag encountered when rate limits are reached (Yildiz et al., 2011).

Inverse Dynamics control systems designed using RIDE methods have demonstrated that prolonged operation on actuator amplitude limits can be achieved by using a simple conditioning methodology (Bennet et al., 1999), (Counsell et al., 2009). However, this feature of the RIDE design is only applicable to amplitude limits and so no provisions are made for input rate limits. The objective of this chapter is to present a nonlinear controller design based on Inverse Dynamics and RIDE methods which uses an anti-windup scheme to ensure that the control signal does not exceed either the amplitude or rate limits of the actuator. The controller design method is known as Rate Actuated Inverse Dynamics (RAID). The RAID controller design process will be described and the stability of the system will be analysed. Conditions for successful operation of the anti-windup scheme will be derived and conventional pole-zero stability analysis methods will also be investigated.

The RAID control system is a two-part design. When the actuator is not limited an Inverse Dynamics controller is used. When the actuator is limited a Variable Structure Control (VSC) law is implemented in order to limit the output of the control system to avoid overdriving the actuator. These are not two completely separate controllers as the VSC controller uses the Inverse Dynamics structure but with switching logic implemented on the error signal. Most anti-windup methods use an extra control loop to proportionally reduce the controller's output when there is a difference between the control output and the output of the actuator (i.e. an input limit has been reached). The literature on this type of controller design is extensive but the vast majority is focussed only on amplitude limiting problems. Recent examples of this type of anti-windup design include: Biannic and Tarbouriech 2009, Herrmann et al. 2008, Hermann et al. 2009, Hu et al. 2008, Kerr et al. 2008, Lan et al. 2006, Lavertsky et al. 2007, Marcos et al. 2007, Menon et al. 2007, and Turner et al. 2007. An alternative to the proportional antiwindup compensator is to use a VSC controller design to dynamically limit

the output of the controller by invoking sliding mode operation when the input limits are reached. The advantage of the VSC anti-windup method is that it is simpler to implement (especially using digital controllers) as it does not require design of the anti-windup loop gain. Scottedward and Hall (2001) and Lu (2008) have used this approached for input amplitude limits but there has been no adaption of the method for use with rate and amplitude limits.

The first principal of the RAID design is that the control system be transformed so that the output of the controller is a rate of change request to the actuation system, i.e. $u_c(t) \approx \dot{u}(t)$. Therefore, the amplitude limits of the control signal are equivalent to the rate limits of the actuator. Consequently, the control system designer can proceed with a rate limiting anti-windup design as if it were an amplitude limiting problem. This is advantageous as the extensive research on amplitude limiting designs can be utilised for a rate limiting design.

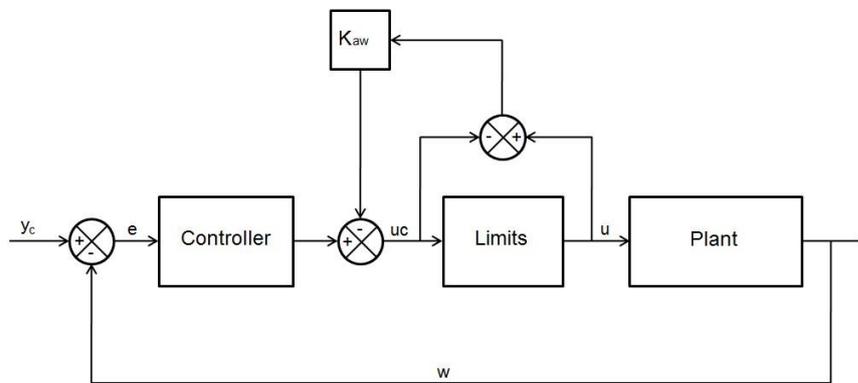


Figure 4.1 – Traditional anti-windup compensator with anti-windup gain K_{aw} .

4.2 State-Space Transform

The first step in the design of the RAID control system is to change the control input to a rate input, such that $u_c(t) \approx \dot{u}(t)$. This is accomplished through the use of a state space transformation. The transformed system is illustrated in Figure 4.2. Of course, a model for the plant could be derived that simply has a rate of

change as its input; however, this is quite rare as most plant models have amplitude as an input. It should also be noted that during the transformations the transformed plant will be expressed in terms of the original plant so that no modifications to the original plant model have to be undertaken.

Given the original plant model expressed in a general linear state space form:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{d}(t) \quad (4.2.1)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \quad (4.2.2)$$

Firstly, a new state vector needs to be specified. With reference to Figure 4.2:

$$\mathbf{x}^*(t) = \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{bmatrix} \quad (4.2.3)$$

$$\mathbf{u}^*(t) = \mathbf{u}(t) \quad (4.2.4)$$

$$\mathbf{A}^* = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ 0 & 0 \end{bmatrix} \quad (4.2.5)$$

$$\mathbf{B}^* = \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix} \quad (4.2.6)$$

The transformed plant can now be expressed in state space form

$$\dot{\mathbf{x}}^*(t) = \mathbf{A}^*\mathbf{x}^*(t) + \mathbf{B}^*\mathbf{u}^*(t) + \mathbf{d}^*(t) \quad (4.2.7)$$

$$\mathbf{w}^*(t) = \mathbf{w}(t) = \mathbf{M}\mathbf{x}(t) \quad (4.2.8)$$

$$\mathbf{y}^*(t) = \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \quad (4.2.9)$$

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{u}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix} \mathbf{u}^*(t) + \begin{bmatrix} \mathbf{d}(t) \\ 0 \end{bmatrix} \quad (4.2.10)$$

It can be seen from Equation (4.2.10) if feedback is taken in the form of $\mathbf{w} = \mathbf{y}$ then $\mathbf{M} = [\mathbf{C} \ 0]$ and \mathbf{MB}^* will be null. Therefore, extra measurements need to be taken in order to implement an Inverse Dynamics controller design. A new measurement vector is chosen, where \mathbf{K}_d is a diagonal gain matrix.

$$\mathbf{w}(t) = \mathbf{y}(t) + \mathbf{K}_d \dot{\mathbf{y}}(t) \quad (4.2.11)$$

$$\mathbf{w}(t) = [\mathbf{C} + \mathbf{K}_d \mathbf{C} \mathbf{A} \quad \mathbf{K}_d \mathbf{C} \mathbf{B}] \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{bmatrix} \quad (4.2.12)$$

$$\mathbf{M} = [\mathbf{C} + \mathbf{K}_d \mathbf{C} \mathbf{A} \quad \mathbf{K}_d \mathbf{C} \mathbf{B}] \quad (4.2.13)$$

The Equivalent Control, \mathbf{u}_{eq} , can now be designed based on the transformed state space matrices.

$$\mathbf{MB}^* = \mathbf{K}_d \mathbf{C} \mathbf{B} \quad (4.2.14)$$

$$\mathbf{u}_{eq}(t) = -(\mathbf{MB}^*)^{-1} \dot{\mathbf{w}}(t) + \dot{\mathbf{u}}^*(t) \quad (4.2.15)$$

This Equivalent Control is implemented within a RIDE control structure so that the RAID control algorithm is given by

$$\mathbf{u}_c(t) = \mathbf{z}(t) - \mathbf{K}_p \mathbf{w}(t) + \mathbf{u}_{eq}(t) \quad (4.2.16)$$

Where

$$\dot{\mathbf{z}}(t) = \mathbf{K}_I (\mathbf{y}_c(t) - \mathbf{w}(t)) \quad (4.2.17)$$

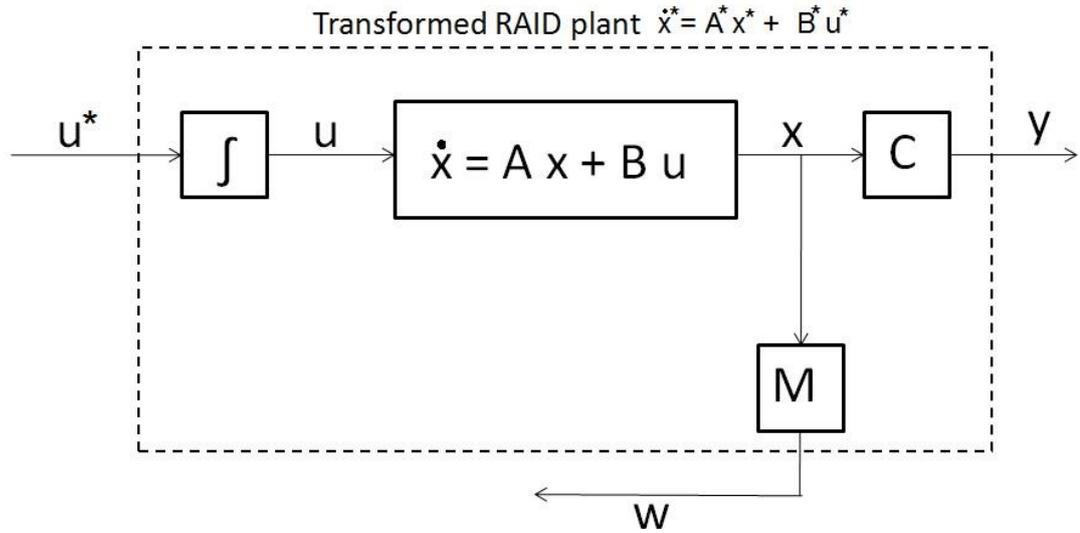


Figure 4.2 – Transformed RAID state space system.

4.3 Linear Stability Analysis

In order to determine the closed loop pole locations of the RAID system it is first necessary to perform a state-space transformation which will separate the slow and fast modes of the system. This is necessary as u_{eq} renders certain plant dynamics unobservable to the controller, thus poles associated with the unobservable dynamics are also unobservable. Since the Equivalent Control only affects the slow modes of the system, a slow/fast state decomposition will reveal all pole locations.

Initially the RAID closed loop state-space equation (Equation (4.2.10)) is partitioned into the generalised form shown in Equation 4.3.2. The only condition on the partitioning is that $\bar{\mathbf{B}}\bar{\mathbf{z}}$ be invertible.

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{u}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \mathbf{u}^*(t) + \begin{bmatrix} \mathbf{d}(t) \\ \mathbf{0} \end{bmatrix} \quad (4.3.1)$$

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} \bar{B}_1 \\ \bar{B}_2 \end{bmatrix} \bar{u}(t) + \begin{bmatrix} \bar{d}_1(t) \\ \bar{d}_2(t) \end{bmatrix} \quad (4.3.2)$$

Therefore

$$x_1 = x \quad (4.3.3)$$

$$x_2 = u \quad (4.3.4)$$

$$\bar{d}_1 = d \quad (4.3.5)$$

$$\bar{d}_2 = 0 \quad (4.3.6)$$

$$\bar{u} = u^* \quad (4.3.7)$$

$$\bar{A}_{11} = A \quad (4.3.8)$$

$$\bar{A}_{12} = B \quad (4.3.9)$$

$$\bar{A}_{21} = 0 \quad (4.3.10)$$

$$\bar{A}_{22} = 0 \quad (4.3.11)$$

$$\bar{B}_1 = 0 \quad (4.3.12)$$

$$\bar{B}_2 = I \quad (4.3.13)$$

The partitioned system can be transformed into fast, feedback states ($w(t)$) and slow states ($c(t)$).

$$\begin{bmatrix} \dot{\mathbf{c}}(t) \\ \dot{\mathbf{w}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{c}(t) \\ \mathbf{w}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{B}_2 \end{bmatrix} \mathbf{u}^*(t) + \begin{bmatrix} \mathbf{d}_1(t) \\ \mathbf{d}_2(t) \end{bmatrix} \quad (4.3.14)$$

Letting:

$$\mathbf{w}(t) = \mathbf{M}_1 \mathbf{x}_1(t) + \mathbf{M}_2 \mathbf{x}_2(t) \quad (4.3.15)$$

$$\mathbf{R} = \bar{\mathbf{B}}_1 \bar{\mathbf{B}}_2^{-1} \quad (4.3.16)$$

$$\mathbf{S} = (\mathbf{M}_2 - \mathbf{M}_1 \mathbf{R})^{-1} \quad (4.3.17)$$

Since $\bar{\mathbf{B}}_1$ is null hence \mathbf{R} is null and $\mathbf{S} = \mathbf{M}_2^{-1}$.

$$\begin{aligned} \mathbf{A}_{11} &= (\bar{\mathbf{A}}_{11} + \mathbf{R} \bar{\mathbf{A}}_{21})(\mathbf{I} + \mathbf{R} \mathbf{S} \mathbf{M}_1) - (\bar{\mathbf{A}}_{12} + \mathbf{R} \bar{\mathbf{A}}_{22}) \mathbf{S} \mathbf{M}_1 \\ &= \mathbf{A} - \mathbf{B} \mathbf{M}_1 \mathbf{M}_2^{-1} \end{aligned} \quad (4.3.18)$$

$$\mathbf{A}_{12} = -(\bar{\mathbf{A}}_{11} + \mathbf{R} \bar{\mathbf{A}}_{21}) + (\bar{\mathbf{A}}_{12} + \mathbf{R} \bar{\mathbf{A}}_{22}) \mathbf{S} = \mathbf{B} (\mathbf{M}_2)^{-1} \quad (4.3.19)$$

$$\begin{aligned} \mathbf{A}_{21} &= (\mathbf{M}_1 \bar{\mathbf{A}}_{11} + \mathbf{M}_2 \bar{\mathbf{A}}_{21})(\mathbf{I} + \mathbf{R} \mathbf{S} \mathbf{M}_1) - (\mathbf{M}_1 \bar{\mathbf{A}}_{12} + \mathbf{M}_2 \bar{\mathbf{A}}_{22}) \mathbf{S} \mathbf{M}_1 \\ &= \mathbf{M}_1 \mathbf{A} - \mathbf{M}_1^2 \mathbf{M}_2^{-1} \mathbf{B} \end{aligned} \quad (4.3.20)$$

$$\mathbf{A}_{22} = -(\mathbf{M}_1 \bar{\mathbf{A}}_{11} + \mathbf{M}_2 \bar{\mathbf{A}}_{21}) \mathbf{R} \mathbf{S} + (\mathbf{M}_1 \bar{\mathbf{A}}_{12} + \mathbf{M}_2 \bar{\mathbf{A}}_{22}) \mathbf{S} = \mathbf{M}_1 \mathbf{M}_2^{-1} \mathbf{B} \quad (4.3.21)$$

$$\mathbf{B}_2 = \mathbf{M}_1 \bar{\mathbf{B}}_1 + \mathbf{M}_2 \bar{\mathbf{B}}_2 = \mathbf{M}_2 \quad (4.3.22)$$

$$\mathbf{d}_1(t) = \mathbf{d}(t) \quad (4.3.23)$$

$$\mathbf{d}_2(t) = \mathbf{M}_1 \mathbf{d}(t) \quad (4.3.24)$$

The RAID control algorithm is defined as:

$$\mathbf{u}_c(t) = \mathbf{z}(t) - \mathbf{K}_p \mathbf{w}(t) + \mathbf{u}_{eq}(t) \quad (4.3.25)$$

In the fast and slow partitioned system the Equivalent Control is still defined as the control to set the rate of change of the feedback to zero and so is expressed as the following

$\mathbf{u}_{eq}(t) = -\mathbf{B}_2^{-1}\mathbf{A}_2\mathbf{c}(t) - \mathbf{B}_2^{-1}\mathbf{A}_2\mathbf{w}(t) - \mathbf{B}_2^{-1}\mathbf{d}_2(t)$ Therefore the closed loop state equation becomes: (4.3.26)

$$\begin{bmatrix} \dot{\mathbf{c}}(t) \\ \dot{\mathbf{w}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ 0 & -\mathbf{B}_2\mathbf{K}_p \end{bmatrix} \begin{bmatrix} \mathbf{c}(t) \\ \mathbf{w}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{B}_2 \end{bmatrix} \mathbf{z}(t) + \begin{bmatrix} \mathbf{d}_1(t) \\ 0 \end{bmatrix} \quad (4.3.27)$$

The closed loop poles are given by the following expressions, with p_1 coinciding with the transmission zeros relating the regulator to the measurement vector. This means that, in a similar fashion to the RIDE controller, the RAID controller places poles over the transmission zeros of the plant, thus rendering the slow modes of the system unobservable by the measurement vector. Clearly, if stability is to be ensured then the transmission zeros must not lie in the right half plane.

$$p_1 = |s\mathbf{I} - \mathbf{A}_{11}| = 0 \quad (4.3.28)$$

$$p_2 = |s\mathbf{I} + \mathbf{B}_2\mathbf{K}_p| = 0 \quad (4.3.29) \text{ For a measurement vector given by } \mathbf{w}(t) = \mathbf{y}(t) + \mathbf{K}_d\dot{\mathbf{y}}(t)$$

$$\mathbf{M}_1 = \mathbf{C}_1 + \mathbf{K}_d\mathbf{C}_1\mathbf{A} \quad (4.3.30)$$

$$\mathbf{M}_2 = \mathbf{K}_d\mathbf{C}_1\mathbf{B} \quad (4.3.31)$$

If $\mathbf{K}_d = \mu\mathbf{I}$, where μ is a scalar gain, then $\mathbf{A}_{11} = \frac{1}{\mu}\mathbf{I}$, thus transmission zeros are located at $-\frac{1}{\mu}\mathbf{I}$ and are always stable if μ is positive.

By introducing integral action on the regulator so that $\mathbf{z}(t) = \mathbf{K}_I(\mathbf{y}_c(t) - \mathbf{w}(t))$ a new closed loop state equation is formed.

$$\begin{bmatrix} \dot{\mathbf{c}}(t) \\ \dot{\mathbf{w}}(t) \\ \dot{\mathbf{z}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & 0 \\ 0 & -\mathbf{B}_2\mathbf{K}_p & \mathbf{B}_2 \\ 0 & -\mathbf{K}_I & 0 \end{bmatrix} \begin{bmatrix} \mathbf{c}(t) \\ \mathbf{w}(t) \\ \mathbf{z}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \mathbf{K}_I \end{bmatrix} \mathbf{y}_c(t) + \begin{bmatrix} \mathbf{d}_1(t) \\ 0 \\ 0 \end{bmatrix} \quad (4.3.32)$$

The closed loop poles are now given by:

$$\begin{vmatrix} s - \mathbf{A}_{11} & -\mathbf{A}_{12} & 0 \\ 0 & s + \mathbf{B}_2\mathbf{K}_P & -\mathbf{B}_2 \\ 0 & \mathbf{K}_I & s \end{vmatrix} = 0 \quad (4.3.33)$$

If $\mathbf{K}_P = \rho(\mu\mathbf{CB})^{-1}$ and $\mathbf{K}_I = \sigma(\mu\mathbf{CB})^{-1}$ then the closed loop poles are defined by the set p_3 .

$$p_3 = s^3 + s^2\left(\rho + \frac{1}{\mu}\right) + s\left(\frac{\rho}{\mu} - \sigma\right) + \frac{\sigma}{\mu} = 0 \quad (4.3.34)$$

The controller gain parameters σ , ρ and μ can then be chosen to meet closed loop transient response requirements and to ensure that all poles lie in the left half plane.

This stability requirement can be expressed by the Routh-Hurwitz stability criterion for third order polynomials (Franklin et al., 2001). If the system is to be stable then the following conditions must be met:

$$\rho + \frac{1}{\mu} > 0 \quad (4.3.35)$$

$$\frac{\rho}{\mu} - \sigma > 0 \quad (4.3.36)$$

$$\frac{\rho}{\mu} > 0 \quad (4.3.37)$$

$$\left(\rho + \frac{1}{\mu}\right)\left(\frac{\rho}{\mu} - \sigma\right) > \frac{\sigma}{\mu} \quad (4.3.38)$$

In summary, there are three design considerations for the linear stability and dynamics of the RAID controller; (i) the parameters must be selected to yield the desired closed loop transient response, (ii) the parameters must not invalidate any the criteria in Equation 4.3.35 – 4.3.38, (iii) there must be adequate bandwidth

separation between the closed loop and actuator and sensor dynamics, in order to prevent excitation of higher order dynamics.

4.4 Sliding Mode Control

4.4.1 Definition of the Switching Surfaces

Two independent switching surfaces need to be designed so that the control signal does not exceed its upper or lower bounds. In the case of a multivariable system with multiple inputs, each i^{th} channel of control signal should not exceed its corresponding limit, i.e.

$$LL_i \leq u_{c_i}(t) \leq UL_i \quad (4.4.1)$$

In order to achieve this two switching surfaces for each channel must be created; one corresponding to the upper limit and one corresponding to the lower limit.

$$\varepsilon u_i(t) = UL_i - u_{c_i}(t) \quad (4.4.2)$$

$$\varepsilon l_i(t) = LL_i - u_{c_i}(t) \quad (4.4.3)$$

The input limit of the system will have been reached when the trajectory of $u_c(t)$ results in either of these switching surfaces being equal to zero. The control input will not exceed the limit if the control enters a sliding mode. In total, there are $2n$ switching surfaces for a system with n inputs.

4.4.2 Existence of the Sliding Mode

In order for a sliding mode to be achieved on each surface, it is necessary to design maximum and minimum control inputs. These control inputs should be chosen so that the system states are always forced to $\varepsilon u_i = 0$ or $\varepsilon l_i = 0$.

Therefore the following conditions must be enforced.

$$\varepsilon \dot{u}_i(t) > 0 \text{ when } \varepsilon u_i(t) < 0 \quad (4.4.4)$$

$$\varepsilon \dot{u}_i(t) < 0 \text{ when } \varepsilon u_i(t) > 0 \quad (4.4.5)$$

$$\varepsilon \dot{l}_i(t) > 0 \text{ when } \varepsilon l_i(t) < 0 \quad (4.4.6)$$

$$\varepsilon \dot{l}_i(t) < 0 \text{ when } \varepsilon l_i(t) > 0 \quad (4.4.7)$$

Examining each of these conditions will provide insight into the conditions for maintaining sliding mode for chosen control actions.

When $uc_i(t)$ is in the region of $\varepsilon u_i(t)$ and $\varepsilon u_i(t) < 0$ (i.e. $uc_i(t) > UL_i$) then $\varepsilon \dot{u}_i(t) > 0$

$$\varepsilon \dot{u}_i(t) = -uc_i(t) \quad (4.4.8)$$

$$uc_i(t) < 0 \quad (4.4.9)$$

Therefore, it is necessary to choose a control action that will enforce this condition. A further consideration on selection of a control action is the presence of integrator windup when the actuator becomes limited. Ideally, a control action should be chosen that enforces the conditions in Equations 4.4.4 to 4.4.7 and prevents integrator windup. Previous work (Counsell et al., 2009) has used a form of regulator conditioning to achieve these goals. This conditioning can be formulated as a VSC commutation law

$$\dot{z}_i(t) = \begin{cases} 0 & \text{if } \varepsilon_{ui}(t) < 0 \text{ or } \varepsilon_{li}(t) > 0 \\ K_{I_i}(\mathbf{y}_c(t) - \mathbf{w}(t)) & \text{if } \varepsilon_{ui}(t) > 0 \text{ or } \varepsilon_{li}(t) < 0 \end{cases} \quad (4.4.10)$$

Where $K_{I_i} = [K_{I_{i1}} \ K_{I_{i2}} \ \dots \ K_{I_{im}}]$, $K_{P_i} = [K_{P_{i1}} \ K_{P_{i2}} \ \dots \ K_{P_{im}}]$ and $MB_i^* = [MB_{i1}^* \ MB_{i2}^* \ \dots \ MB_{im}^*]$

Since the rate of change of the regulator is set to zero when the limits of $u_i(t)$ are exceeded, any integrator windup will be prevented. This is also only a modification of the original control law and so conforms to the proposed antiwindup strategy. The following analysis will define the conditions under which Equations 4.4.4 to 4.4.7 are satisfied.

$$\text{When } \varepsilon_{ui}(t) < 0 \text{ then } \dot{u}_i(t) = -K_{P_i}\dot{\mathbf{w}}(t) + u\dot{e}q_i(t) < 0 \quad (4.4.11)$$

From Equation (4.2.15) $\dot{w}_i(t) = MB_i(\mathbf{u}(t) - \mathbf{ue}q(t))$ and from Equation (3.3.10) $K_P = \rho[MB]^{-1}$, therefore

$$\dot{u}_i(t) = -\rho(u_i(t) - ueq_i(t)) + u\dot{e}q_i(t) < 0 \quad (4.4.12)$$

Since in these circumstances $u_i \approx UL_i$

$$\dot{u}_i(t) = -\rho(UL_i - ueq_i(t)) + u\dot{e}q_i(t) < 0 \quad (4.4.13)$$

Providing that $\rho > 0$, rearranging gives

$$ueq_i(t) < UL_i - \frac{u\dot{e}q_i(t)}{\rho} \quad (4.4.14)$$

When $\varepsilon_{ui}(t) > 0$ (i.e. $u_i(t) < UL_i$) then $\varepsilon\dot{u}_i(t) < 0$ therefore

$$\dot{u}_i(t) > 0 \quad (4.4.15)$$

When $\varepsilon_{ui}(t) > 0$

$$\dot{u}c_i(t) = \dot{z}_i(t) - K_{P_i}\dot{\mathbf{w}}(t) + u\dot{e}q_i(t) > 0 \quad (4.4.16)$$

Noting that $\dot{z}_i(t) = K_{I_i}(\mathbf{y}_c(t) - \mathbf{w}(t))$ and $K_I = \sigma[\mathbf{MB}]^{-1}$

$$\dot{u}c_i(t) = \sigma[\mathbf{MB}_i]^{-1}(\mathbf{y}_c(t) - \mathbf{w}(t)) - \rho(u_i(t) - ueq_i(t)) + u\dot{e}q_i(t) > 0 \quad (4.4.17)$$

Providing that ρ and $\sigma > 0$ then rearranging

$$ueq_i(t) > UL_i - \frac{u\dot{e}q_i(t)}{\rho} - \frac{\sigma}{\rho}[\mathbf{MB}_i]^{-1}(\mathbf{y}_c(t) - \mathbf{w}(t)) \quad (4.4.18)$$

Combining Equations (4.4.14) and (4.4.18) yields a criterion which determines the conditions for a sliding mode to be maintained on the switching surface $\varepsilon u_i = 0$

$$UL_i - \frac{u\dot{e}q_i(t)}{\rho} - \frac{\sigma}{\rho}[\mathbf{MB}_i]^{-1}(\mathbf{y}_c(t) - \mathbf{w}(t)) < ueq_i(t) < UL_i - \frac{u\dot{e}q_i(t)}{\rho} \quad (4.4.19)$$

A similar analysis can be performed when uc_i is in the region of εl_i .

For sliding mode (SM), when $\varepsilon l_i(t) < 0$ (i.e. $uc_i(t) > LL_i$) $\varepsilon \dot{l}_i(t) > 0$

$$\dot{\varepsilon}l_i(t) = -\dot{u}c_i(t) > 0 \quad (4.4.20)$$

$$\dot{u}c_i(t) < 0 \quad (4.4.21)$$

With reference to the switching law in Equation (4.4.10) and the relation for uc_i is given by

$$\dot{u}c_i(t) = \sigma[\mathbf{MB}_i]^{-1}(\mathbf{y}_c(t)) - \rho(u_i(t) - ueq_i(t)) + u\dot{e}q_i(t) < 0 \quad (4.4.22)$$

Noting that when in SM $u_i \approx LL_i$ and providing that ρ and $\sigma > 0$ then rearranging gives

$$ueq_i(t) < LL_i - \frac{u\dot{e}q_i(t)}{\rho} - \frac{\sigma}{\rho}[\mathbf{MB}_i]^{-1}(\mathbf{y}_c(t) - \mathbf{w}(t)) \quad (4.4.23)$$

For the case when $\varepsilon l_i > 0$ (i.e. $uc_i(t) < LL_i$) $\varepsilon \dot{l}_i < 0$, therefore, $\dot{uc}_i(t) > 0$

And with reference to the switching law in Equation (4.4.10) for $\varepsilon l_i > 0$

$$\dot{uc}_i(t) = \rho(u_i(t) - ueq_i(t)) + u\dot{eq}_i(t) > 0 \quad (4.4.24)$$

Assuming that $u_i \approx LL_i$ and providing that ρ and $\sigma > 0$ then rearranging gives

$$ueq_i(t) > LL_i - \frac{u\dot{eq}_i(t)}{\rho} \quad (4.4.25)$$

Combining Equations (4.4.23) and (4.4.25) gives the conditions for sliding mode on the switching surface $\varepsilon l_i = 0$

$$LL_i - \frac{u\dot{eq}_i(t)}{\rho} - \frac{\sigma}{\rho} [MB_i]^{-1}(\mathbf{y}_c(t) - \mathbf{w}(t)) > ueq_i(t) > LL_i - \frac{u\dot{eq}_i(t)}{\rho} \quad (4.4.26)$$

The expressions given in Equations (4.4.19) and (4.4.26) results in the creation of two zones for each channel. If ueq passes through either of these zones it will signify that the controller has entered a sliding mode on the switching surfaces εu_i or εl_i . This is illustrated in Figure 4.3. More importantly, this also means that the control signal is being limited to UL_i or LL_i respectively.

The zone corresponding to sliding mode on the upper limit is defined by L1 and L2

$$L1(t) = UL_i - \frac{u\dot{eq}_i(t)}{\rho} \quad (4.4.27)$$

$$L2(t) = UL_i - \frac{u\dot{eq}_i(t)}{\rho} - \frac{\sigma}{\rho} [MB_i]^{-1}(\mathbf{y}_c(t) - \mathbf{w}(t)) \quad (4.4.28)$$

The zone corresponding to sliding mode on the lower limit is defined by L3 and L4

$$L3(t) = LL_i - \frac{u\dot{eq}_i(t)}{\rho} \quad (4.4.29)$$

$$L4(t) = LL_i - \frac{ueq_i(t)}{\rho} - \frac{\sigma}{\rho} [MB_i]^{-1} (y_c(t) - w(t)) \quad (4.4.30)$$

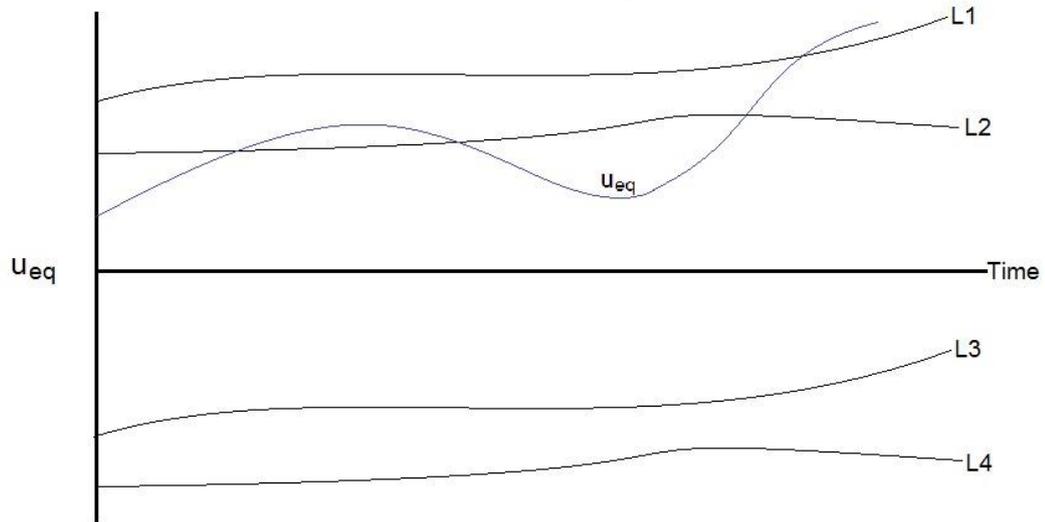


Figure 4.3 – Graphic illustration of u_{eq} criteria for sliding mode.

4.4.3 Breaking the Sliding Mode

There are two main scenarios that should be investigated by examining these criteria:

The sliding mode is left by uc_i moving further into the limit, which is an undesirable situation. This will occur when ueq_i exits the SM zone into the uncontrollable space.

The sliding mode is left by uc_i moving away from the limit and hence the controller resuming its normal mode of operation. This will occur when ueq_i exits the SM zone into the controllable space.

Before investigating scenario 1 it is necessary to explore the relationship between ueq and the input limits. The Equivalent Control, ueq , is the control required for the feedback to reach steady state at any given point in time. If ueq is greater than either the upper or lower input limits, there is not enough power available in the system to reach steady state. If a steady state cannot be reached (i.e. the steady state is unreachable), the system is uncontrollable. The limits on ueq_i which define when a steady state is reachable can be expressed as

$$LL_i < ueq_i(t) < UL_i \quad (4.4.31)$$

A similar criterion that defines when scenario 1 will not occur can be expressed as

$$-\frac{u\dot{e}q_i(t)}{\rho} + LL_i < ueq_i(t) < UL_i - \frac{u\dot{e}q_i(t)}{\rho} \quad (4.4.32)$$

Firstly, comparing the two expressions it can be seen that if ρ is sufficiently large then Equation (4.4.32) approaches Equation (4.4.31). Therefore, providing that the steady state is reachable, the control signal will remain on the sliding mode and not exceed its limit. When scenario 1 occurs it is clear that a steady state is not reachable and the control signal will exit the sliding mode into an uncontrollable space. The control signal will return to the sliding mode when a steady state becomes reachable again and the system is controllable.

Clearly the selection of ρ is extremely important to maximizing the ability of the control signal to remain on the sliding mode. The expression $\frac{u\dot{e}q_i}{\rho}$ will always cause the limits on ueq to become reduced when ueq is moving towards the limit. Therefore, decreasing ρ will effectively reduce the limits on ueq and hinder the ability of the controller to remain in sliding mode. Consequently, the selection of ρ will determine the effective size of the actuator limits. If ρ is large enough then

the only constraints on the ability of the control system to keep uc below its limit are the magnitudes of LL and UL.

Scenario 2 is also important as it is vital that the control signal does not become fixed on the limit. It is apparent that as $yc_i(t) - w_i(t)$ becomes smaller, the likelihood of the SM being maintained is diminished. This means that, as the controlled output approaches the desired setpoint, the SM will be left and uc_i will move away from the limit (providing that scenario 1 has not occurred). In fact, a condition for the existence of a SM is that

$$w_i(t) < yc_i(t) \tag{4.4.33}$$

This proves that SM is not possible if there is overshoot of the setpoint. Hence the control signal will not become fixed on a limit and will return to normal operation when the system trajectories determine that uc_i should reduce in magnitude.

4.5 Variable Structure Controller Design for Rate and Amplitude Limits

4.5.1 Rate Limits

The first considerations in designing the switching surface are the values of the limits UL LL and for each control channel. For a controller design that is only concerned with amplitude limits, such as control surface deflections, this is a relatively simple task. Considering any amplitude limits of an actuator it is perhaps best to limit the control signal to just below these limits in order to ensure that the actuator is not overdriven. In the case of the RAID controller design, where both amplitude and rate limits are to be considered, more thought

is required. Since the RAID control signal is a rate of change any amplitude limits on the control signal will result in rate limits on the actuator or input into the controlled system. Therefore, UL and LL should be designed to be slightly less than the rate limits of the actuator, such that

$$UL = sf\dot{u}_{\text{limit upper}} \quad (4.5.1)$$

$$LL = sf\dot{u}_{\text{limit lower}} \quad (4.5.2)$$

Where sf is a safety factor chosen to ensure that the designed limits UL and LL are never greater than the achievable rate limits of the actuator. Hence, the system's actuators will never be overdriven in rate.

4.5.2 Amplitude Limits

The amplitude limits of the actuator of must also be considered for the RAID controller. When an amplitude limit is reached the rate limits of the actuator will be modified.

$$\text{when } u_i(t) = u_{i \text{ limit}} \quad -\dot{u}_{i \text{ limit}} \leq \dot{u}_i(t) \leq 0 \quad (4.5.3)$$

$$\text{when } u_i(t) = -u_{i \text{ limit}} \quad 0 \leq \dot{u}_i(t) \leq \dot{u}_{i \text{ limit}} \quad (4.5.4)$$

The control signal limits UL and LL will need to be changed accordingly when either the upper or lower actuator amplitude limits are reached. UL and LL now have two possible values each and can be defined by the following logic:

$$\text{if } u_i(t) < u_{i \text{ limit upper}} \text{ then } UL = sf\dot{u}_{i \text{ limit upper}} \quad (4.5.5)$$

$$\text{if } u_i(t) \geq u_{i \text{ limit upper}} \text{ then } UL = 0 \quad (4.5.6)$$

$$\text{if } u_i(t) > u_{i \text{ limit lower}} \text{ then } LL = sf\dot{u}_{i \text{ limit lower}} \quad (4.5.7)$$

$$\text{if } u_i(t) \leq u_{i \text{ limit lower}} \text{ then } LL = 0 \quad (4.5.8)$$

From Equations 4.5.5 and 4.5.8 it can be seen that when the amplitude limit is reached the rate of change will be limited so that the control action will not further increase and overdrive the actuator.

4.5.3 Influence of controller gains

It is clear from Equation 4.4.27 to 4.4.30 that the value of the parameter ρ will have an influence on the behaviour of the sliding mode. When a trigger for initializing the sliding mode is reached (i.e. either LL or UL has been reached) then the control signal undergoes a transition from its current position to the sliding surface. This transition should ideally be as rapid as possible. Since this transition to the sliding mode occurs as soon as $uc > UL$ or $uc < LL$ the transition will be very brief as uc will already be extremely close to the switching surface of $uc = UL$ or $uc = LL$. The speed of transition to the switching surface will be determined by the control action which, during the transition, will be given by

$uc_i(t) = -\rho[MB]^{-1}w_i(t) + ueq_i(t)$. The control is essentially proportional and the larger the value of ρ the higher the gain of the control action. Thus, a larger value of ρ will result in a more rapid transition to the sliding surface. The transition to the sliding surface becomes a more important issue when an amplitude limit is reached. When this occurs the control signal limit will instantly change from its upper or lower value to zero. The control signal, uc , will then have to transition to the new sliding mode, which will be a substantial distance away. As the transition occurs in a finite, the value of the control signal will be non-zero, hence the rate of change of the input will be non-zero; therefore the actuator will be briefly overdriven. To remedy this problem an

extra limiting element needs to be introduced into the signal path before the actuator. The limiting element will limit the amplitude of the control signal (and hence the rate of the input) so that the rate of change of the input is zero, as soon as the control signal is detected to have caused the input to reach an amplitude limit. This can be described by the following logic.

$$\text{if } \int u c_i(t) dt \geq u_{i \text{ limit upper}} \text{ then } UL_{2i} = 0 \quad (4.5.9)$$

$$\text{if } \int u c_i(t) dt < u_{i \text{ limit upper}} \text{ then } UL_{2i} = UL_i \quad (4.5.10)$$

$$\text{if } \int u c_i(t) dt \leq u_{i \text{ limit lower}} \text{ then } LL_{2i} = 0 \quad (4.5.11)$$

$$\text{if } \int u c_i(t) dt > u_{i \text{ limit lower}} \text{ then } LL_{2i} = LL_i \quad (4.5.12)$$

Thus the actuator is prevented from being overdriven in amplitude as the final input immediately switches and its rate becomes zero. It is noteworthy that this limiting element does not interfere with the controller design as it effectively replaces the rate limits of the actuator.

4.5.4 Implementation of Switching Logic

In summary, the variable structure control actions are given by the following expression.

$$\dot{z}_i(t) = \begin{cases} 0 & \text{if } \varepsilon_{ui}(t) < 0 \text{ or } \varepsilon_{li}(t) > 0 \\ K_{I_i}(\mathbf{y}_e(t) - \mathbf{w}(t)) & \text{if } \varepsilon_{ui}(t) > 0 \text{ or } \varepsilon_{li}(t) < 0 \end{cases} \quad (4.5.13)$$

The implementation of this switching logic is quite straightforward for a singleinput single- output (SISO) system. In order to switch the rate of change of the regulator the error signal is simply switched between its normal definition of $y_c(t) - w(t)$ and zero in accordance with the logic in Equation 4.5.13. However,

for a multiple-input, multiple-output (MIMO) system other considerations have to be taken into account in the controller design. As shown in Equation 4.5.13 the control signal variable \dot{z}_i must be equal to zero when $\varepsilon_{ui} < 0$ or $\varepsilon_{li} > 0$. Since \dot{z}_i is a function of $e_{1...n}$ simply setting e_i equal to zero for the corresponding input channel will not ensure that \dot{z}_i is zero. Deconstructing the regulator vector illustrates this.

$$\dot{z}_i(t) = K_{I_{i1}}e_1(t) + K_{I_{i2}}e_2(t) + K_{I_{i3}}e_3(t) \dots K_{I_{in}}e_n(t) \quad (4.5.14)$$

One would assume that setting the entire e vector to zero would ensure that \dot{z}_i is zero and of course that would be the case. This is the method employed by Counsell, 1992. However, there is a major disadvantage to using this method. Since the error for every channel is now zero the switching logic defined in Equation 4.5.13 has not been followed as $\varepsilon_{ui} > 0$ or $\varepsilon_{li} < 0$ will not necessarily be true for all channels. Furthermore, the integral action for all channels will be disabled and knowledge of the setpoint is lost. This means that if a change in setpoint occurs during this period it will not be tracked, even if it would normally be possible as all the inputs may not be at their limits. This is of course extremely undesirable. A solution would be to firstly separate the calculation of the regulator into individual channels.

$$\begin{aligned} \dot{z}_1(t) &= K_{I_{11}}e_1(t) + K_{I_{12}}e_2(t) + K_{I_{13}}e_3(t) \dots K_{I_{1n}}e_n(t) \\ \dot{z}_2(t) &= K_{I_{21}}e_1(t) + K_{I_{22}}e_2(t) + K_{I_{23}}e_3(t) \dots K_{I_{2n}}e_n(t) \\ &\vdots \\ \dot{z}_n(t) &= K_{I_{n1}}e_1(t) + K_{I_{n2}}e_2(t) + K_{I_{n3}}e_3(t) \dots K_{I_{nn}}e_n(t) \end{aligned} \quad (4.5.15)$$

The whole regulator function can then be switched for each channel in accordance with Equation 4.5.13, ensuring that \dot{z}_i is equal to zero.

4.6 Robustness of the VSC Design

One of the most appealing properties of VSC control is its invariance in the face of parameter variations in the plant (Zinober, 1990). This inherent robustness means that the only information required to initiate sliding mode control is the controller output.

The main drawback of VSC sliding mode control is the “chatter” that can often occur on the control input during the sliding mode, as it cycles between the two VSC actions. In a typical VSC controller design the switching logic is usually implemented on the final control action. This can be problematic as high frequency chattering on the final control action can cause wear and degradation of the actuator. This is, however, less of a problem for the RAID VSC design as the switching does not actually occur on the final control action, rather, it occurs on the rate of change of the regulator. Therefore, chattering on the final control action is minimised, but there will still be some very small amplitude movement of the final control action during the sliding mode. The frequency of this action will be largely determined by the controller gain parameter ρ . A larger value of ρ will result in the sliding mode being reached more rapidly and, as has been demonstrated previously, will allow the control to remain on the sliding mode for longer. However a larger ρ will also cause the controller output to oscillate at a higher frequency on the sliding surface. The change in oscillation frequency when using an increased gain is illustrated by Figure 4.4.

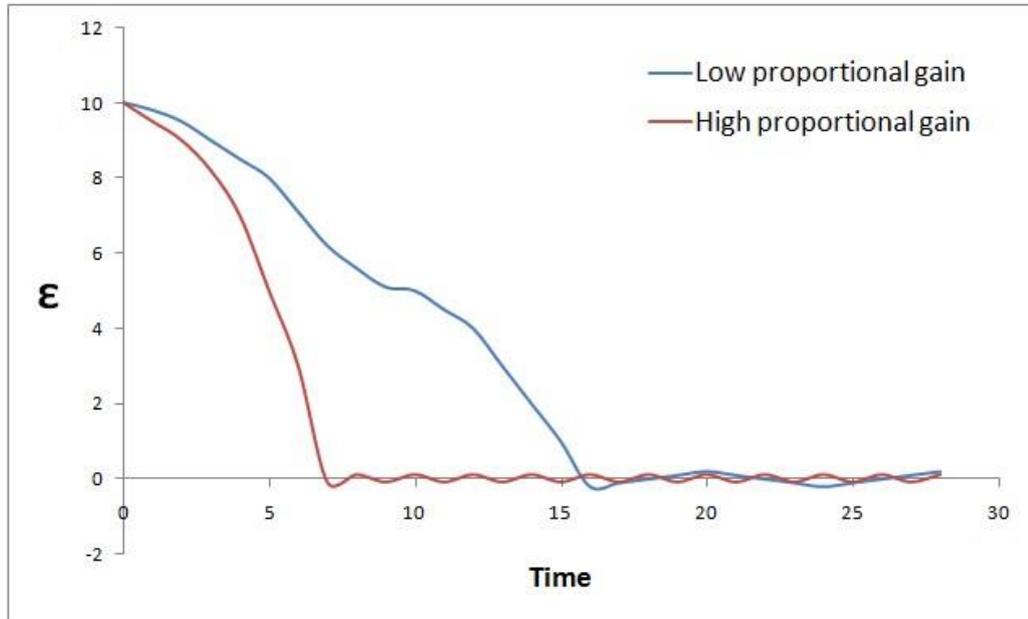


Figure 4.4 – Illustration of the influence of controller gain on sliding mode behavior.

4.7 Pole Placement

When the sliding mode is entered the linear stability must also be considered as described in section 4.3. Theoretically, during the sliding mode the controller is switching between two functions at an infinite frequency. Of course, in reality, the switching will not occur at an infinite frequency due to small lags in the system and the limits of the microprocessor's speed on which the control is implemented. Nonetheless, the fact that the control is switching rapidly between two values makes it difficult to perform the kind of analysis undertaken in section 4.3. A common method used in VSC analysis to resolve this problem is to consider an average of the control signal during the sliding mode. When the control system is in sliding mode the switching function can be equated to zero. This is then resolved for uc to give the average control signal. For the RAID system

$$u_i(t) = UL_i - uc_i(t) = 0 \quad (4.7.1)$$

$$\varepsilon l_i(t) = LL_i - uc_i(t) = 0 \quad (4.7.2)$$

Therefore, depending on which switching surface the controller is on, during sliding mode

$$uc_i(t) = UL_i \text{ or } uc_i(t) = LL_i \quad (4.7.3)$$

This confirms the intuitive assumption that uc is approximate to either the upper or lower limit. If the limits are not continuously changing, which would most often be the case, then the system is essentially open loop, where:

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{u}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix} \mathbf{UL} + \begin{bmatrix} \mathbf{d}(t) \\ 0 \end{bmatrix} \quad (4.7.4)$$

Or

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{u}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix} \mathbf{LL} + \begin{bmatrix} \mathbf{d}(t) \\ 0 \end{bmatrix} \quad (4.7.5)$$

Consequently, the poles of the system are the open loop poles, given by the following determinant solved for λ .

$$p_{ol} = \det \begin{vmatrix} \lambda \mathbf{I} - \mathbf{A} & \mathbf{B} \\ 0 & \lambda \mathbf{I} \end{vmatrix} = 0 \quad (4.7.6)$$

If the poles contain a positive real term it would not necessarily be problematic. The output would be expected to grow rapidly but, as shown by Equation 4.4.28 and 4.4.29, when the output approaches the setpoint the sliding mode will be left and the closed loop control will be reactivated, bring the poles to the designed locations. A problem could arise if the open loop poles contain significant imaginary parts. Depending on the magnitude of the imaginary parts, undesirable high frequency oscillatory behaviour could emerge when control limits are

reached. This problem would be compounded if the real parts were also positive. Therefore, it is advised that caution be taken if the system to be controlled is open loop oscillatory as the RAID controller design may not be suitable.

4.8 Conclusions

This chapter has presented a nonlinear controller design method which uses an Inverse Dynamics control law with a variable structure, sliding mode, antiwindup compensation scheme to limit the output of the controller; so that both input amplitude and rate limits are not exceeded.

The anti-windup design for rate limitations is made possible by transforming the Inverse Dynamics controller so that a rate of change is output instead of an amplitude. This enables rate limits to be treated in the same manner as amplitude limits in a conventional controller design. Importantly, information about the amplitude of the controller output is preserved, allowing for an anti-windup controller design method that will prevent the controller output from exceeding rate and amplitude limits.

The drawback of the transformation to a rate of change output is that the order of the closed loop system is increased by one. This means that for most systems, which are dominantly first order in nature, the closed loop system's order is increased from two to three. Closed loop dynamics are easier to assign for a second order system as the transient response characteristics of natural frequency and damping ratio are obtainable. For a third order system these terms do not adequately describe the transient response of the system and hence the tuning of a third order control system is more difficult than that of a second order system.

The conditions for successful operation of the RAID sliding mode anti-windup compensator were derived. These conditions can be visualised as limits to the magnitude of the Equivalent Control, u_{eq} . If u_{eq} remains within these limits then the controller output will not exceed either the actuator rate or amplitude limits.

If the RAID controller gain (k_{eq}) is sufficiently large then the limits to u_{eq} can be approximated by the limits of the actuator. In this case if u_{eq} exceeds its limits then there is simply not enough control authority available to the system to reach steady state. Therefore, the anti-windup compensator will only cease to function, and the controller output will exceed its limits, when it is impossible to reach a steady state at that point in time. In terms of the ability to access the maximum available energy, this can be considered an optimal solution, as the controller will operate effectively on its input limits until the absolute physical limitations of the system dictate that it is no longer possible to do so.

The choice of the controller gain parameter ρ determines how close the controller comes to realising this ideal. In effect the smaller the value of ρ the less the available control authority. This is expressed by Equation 4.4.28. Therefore, the best solution to maximise control authority would be to have a hypothetical infinite gain. It would appear from this that the hypothetical infinite gain controller provides an optimum solution for both the linear and nonlinear control problems. Of course, in reality, an infinite gain controller is not possible due to the inevitable excitation of higher order dynamics present in the system. Therefore, a balance needs to be struck between maximising available control authority and ensuring that the linear performance of the controller is robust and insensitive to higher order system dynamics.

5. Heating and Ventilation Control Case Study

5.1 Introduction

Heating Ventilation and Air-conditioning (HVAC) systems require a control system to regulate the temperature, humidity and/or CO₂ levels of an indoor environment. The performance of the control systems has a large impact on the overall performance of the HVAC system in terms of air quality and energy efficiency (Tashtoush et al., 2005). A poorly designed control system may be difficult to tune, result in overshoot of the setpoint or behave in an unstable and oscillatory manner – all of which can contribute to excessive energy usage, as well as reduced occupant comfort (Qiang et al., 2000).

Traditionally, controllers with relatively simple algorithms, such as Proportional Integrator Derivative (PID), are used for HVAC control, this is in spite of the complexity of the control problem. One of the most common problems in HVAC control is interaction between controlled variables (Counsell et al., 2011). Heating and ventilation systems are strongly coupled, in that the action of one has an effect on the other. Hence, the simultaneous control of temperature and humidity can be difficult to achieve without the two systems fighting each other and resulting in inefficient energy usage (Rentel-Gomez and Velez-Reyes 2001). Furthermore, HVAC systems are subject to many disturbances such as rapidly changing heat disturbances from occupants, appliances and external weather condition, as well as the nonlinear thermodynamics of the internal environment.

Advanced controller design methods have been investigated for use with HVAC systems using nonlinear, optimal control and auto-tuning methods:

(ArguelloSerrano and Velez-Reyes, 1999), (Qiang et al., 2000), (Komareji et al., 2009), (Rentel-Gomez and Velez-Reyes, 2001), (Bai and Zhang, 2007). The main barrier to the adoption of these methods has been their complexity. If an advanced controller design is to be adopted by the HVAC control industry it must be relatively simple and address the control problems directly.

These classes of control problems have been successfully addressed with a simple multivariable Inverse Dynamics based controller design methods in the field of flight control, resulting in robust high performance control (Fielding et al., 2002). However, achieving high performance control of HVAC systems using these methods is more problematic due to the limitations of the heat and ventilation delivery systems and sensors. As has previously been established the RIDE controller design requires a significant bandwidth separation between the slow closed loop dynamics and fast actuator and sensor dynamics. In HVAC systems the actuation and sensor systems have large inertias associated with them resulting in potentially very slow dynamics. Therefore, in order to guarantee satisfactory bandwidth separation the closed loop response has to be slowed to an often unacceptable degree. Failure to achieve satisfactory bandwidth can result in oscillatory behaviour occurring, which results in excessive energy usage and degradation of actuation systems.

The purpose of this case study is to demonstrate, through simulation, the potential of the VTR design method to enable practical implementation of a high performance Inverse Dynamics controller design for a modern heating and ventilation system. Simulations of HVAC system control for a modern office space are performed using a simplified multi-input multi-output thermodynamic model of the buildings physics and systems. The simulations compare the performance of a PI controller, a RIDE controller and a RIDE controller designed using VTR methods simultaneous tracking of internal air temperature and

relative humidity. The sensitivity of the controller performance to slow actuator dynamics and sensor delays is assessed and the resultant energy usage of each controller is estimated.

5.2 Nonlinear Building Energy System Mathematical Model

In order to test the controller designs a building thermodynamic model with particular requirements for controller design must be established. The controller design methods of RIDE / VTR require that the system to be controlled can be represented in a relatively low order state-space form. A high order, extremely detailed energy simulation such as ESP-r (Heim and Clarke, 2004) could be used to assess performance but would not be suitable for controller design, where stability must be investigated. Therefore, a simplified, low order model that still captures the essential dynamic properties of the building thermodynamics is required (Gouda et al., 2003). The resulting system is based on the models developed by Khalid and Murphy (Counsell et al., 2011), (Murphy and Counsell, 2011) but with the addition of a ventilation system and relative humidity model. The model has been validated by comparison studies against the SAP (Standard Assessment Procedure) model of building energy usage (Murphy et al., 2013).

The performance of the controllers was assessed with simulations using the same thermodynamic model. Whilst simplified, the model can still provide realistic estimates of energy usage and transient behaviour as demonstrated by Murphy.

5.2.1 Thermodynamics

There are four thermodynamic states in the building model; air temperature, T_a , internal structure temperature, T_{si} , external structure temperature, T_{se} and

internal thermal mass temperature, T_{tm} . The model assumes a single indoor air zone that is fully mixed and at constant pressure and density.

The energy and mass balance equations that govern the rates of change of temperature in the zone are

$$M_a C_a \dot{T}_a(t) = \dot{Q}_h(t) + \dot{Q}_{dist}(t) - \dot{Q}_{si}(t) - \dot{Q}_f(t) - \dot{Q}_r(t) - \dot{Q}_w(t) - \dot{Q}_{nv}(t) - \dot{Q}_{mv}(t) - \dot{Q}_{tm}(t) \quad (5.2.1)$$

$$M_{si} C_{si} \dot{T}_{si}(t) = \dot{Q}_{si}(t) - \dot{Q}_{wall}(t) \quad (5.2.2)$$

$$M_{se} C_{se} \dot{T}_{se}(t) = \dot{Q}_{wall}(t) - \dot{Q}_{se}(t) \quad (5.2.3)$$

$$M_{tm} C_{tm} \dot{T}_{tm}(t) = \dot{Q}_{tm}(t) \quad (5.2.4)$$

The roof, floor and glazing are assumed to be at constant temperature and as such there is negligible energy stored in them. The ground temperature is also assumed to be constant. Heat gains disturbing the system: solar radiation, occupants and devices, are lumped together and noted as Q_{dist} .

The heat losses through the floor windows and roof are given by the following

$$\dot{Q}_f(t) = U_f A_f (T_a(t) - T_g) \quad (5.2.5)$$

$$\dot{Q}_r(t) = U_r A_r (T_a(t) - T_{ex}(t)) \quad (5.2.6)$$

$$Q_w(t) = U_w A_w (T_a(t) - T_{ex}(t)) \quad (5.2.7)$$

The heat losses due to ventilation are expressed in equations 5.2.8 and 5.2.9. The mass flow rate due to natural ventilation is assumed to be constant.

$$Q_{nv}(t) = \dot{m}_{nv} C_a (T_a(t) - T_{ex}(t)) \quad (5.2.8)$$

$$Q_{mv}(t) = \dot{m}_{mv}(t)C_a(T_a(t) - T_{ex}(t)) \quad (5.2.9)$$

Heat is lost to thermal mass due to the temperature difference between the zone air and the thermal mass.

$$Q_{tm}(t) = U_{tm}A_{tm}(T_a(t) - T_{tm}(t)) \quad (5.2.10)$$

It is assumed that the walls of the zone are composite and are comprised of an internal surface, the wall structure and an external surface. Equation 5.2.11 represents the heat exchange between the zone air and the internal surface of the structure, Equation 5.2.12 describes the heat exchange through the wall between the internal and external surfaces and Equation 5.2.13 describes the heat exchange between the external surface and the outside air.

$$Q_{si}(t) = h_i A_s (T_a(t) - T_{si}(t)) \quad (5.2.11)$$

$$\dot{Q}_{wall}(t) = \frac{k}{wt} A_s (T_{si}(t) - T_{se}(t)) \quad (5.2.12)$$

$$Q_{se}(t) = h_e A_s (T_{se}(t) - T_{ex}(t)) \quad (5.2.13)$$

The rate of change of absolute humidity of the zone air is modelled as the difference between the moisture added and removed from the zone. In terms of occupant comfort relative humidity is a more practical measure of the quality of the internal air. An empirical relation between the absolute and relative humidity is derived from data from the Psychrometric chart (Rentel-Gomez and VelezReyes, 2001). This relationship is taken from an operating point of 23°C and an absolute humidity of 0.007 kg/kg.

$$M_a W_a(t) = \dot{m}_{mv}(t)(W_{ex} - W_a(t)) + \dot{m}_{nv}(W_{ex} - W_a(t)) + S_{occ}(t) \quad (5.2.14)$$

$$W_{ar}(t) = 5000W_a(t) - 1.388T\dot{a}(t) \quad (5.2.15)$$

5.2.2 Actuation Systems

The controllable inputs into the zone are the heat source, $Q_h(t)$, and the mechanical ventilation mass flow rate, \dot{m}_{mv} . These inputs are not able to deliver an instantaneous or unlimited amount of heat or fresh air due to the limits of their actuation systems. These actuator limitations are extremely important in controller design as they govern the maximum achievable controller performance, irrespective of the controller design. It is the controller design that determines how close actual performance is to the maximum.

The actuation systems in this study are approximated as having first order dynamics and amplitude limits. The first order dynamics have a time constant which represents the inertia present in the actuation system between the commanded controller output and the actual output achieved by the actuator (Qiang et al., 2000). Equation 5.2.16 describes a simplified relationship for the heat transfer rate for a typical actuator system and can be used to represent many heat delivery systems, such as convectors, radiators, under-floor heating and even storage heaters where the controller demands a certain power from the heater. The amplitude limits represent the power limitations of the heating system. The ventilation system is more straightforward as the mass flow rate is directly proportional to the fan speed, thus the time constant is an approximation of the delay caused by the power limitations of the fan motor and the fan inertia. The amplitude limits of the ventilation system are approximate and directly proportional to the maximum fan speed. Values of the heating and ventilation

time constants and amplitude limitations are indicative of a commercial HVAC system with a direct convector. uc_{Qh} and uc_{mv} are the heater and mechanical ventilation control signals, respectively.

$$\ddot{Q}_h(t) = \frac{1}{\tau_h} (uc_{Qh}(t) - \dot{Q}_h(t)) \quad (5.2.16)$$

$$LL_{Qh} \leq \dot{Q}_h(t) \leq UL_{Qh} \quad (5.2.17)$$

$$\ddot{m}_{mv}(t) = \frac{1}{\tau_v} (uc_{mv}(t) - \dot{m}_{mv}(t)) \quad (5.2.18)$$

$$LL_{mv} \leq \dot{m}_v(t) \leq UL_{mv} \quad (5.2.19)$$

5.2.3 State-Space Representation

Modern controller design methods require that the system to be controlled is represented in linear generalised state-space form.

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{D}\mathbf{d}(t) \quad (5.2.20)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \quad (5.2.21)$$

Where $\mathbf{x}(t)$ is the state vector, $\mathbf{u}(t)$ is the input vector, $\mathbf{d}(t)$ is the disturbance vector and $\mathbf{y}(t)$ is the output to be controlled. In this study the controller is required to simultaneously track internal air temperature, T_a and relative humidity, W_{ar} by varying the power of the heat source, $\dot{Q}_h(t)$ and the mechanical ventilation rate, $\dot{m}_{mv}(t)$. The inputs into the system that are not controllable are the heat disturbances and the external temperature. Therefore, in this case, the state-space vectors are as follows

$$\mathbf{x}(t) = \begin{bmatrix} T_a(t) \\ T_{si}(t) \\ T_{se}(t) \\ T_{tm}(t) \\ W_a(t) \\ W_{ar}(t) \end{bmatrix} \quad (5.2.22)$$

$$\mathbf{u}(t) = \begin{bmatrix} \dot{Q}_h(t) \\ \dot{m}_{mv}(t) \end{bmatrix} \quad (5.2.23)$$

$$\mathbf{d}(t) = \begin{bmatrix} \dot{Q}_{dist}(t) \\ T_{ex}(t) \\ S_{occ} \\ W_{ex} \\ T_g \end{bmatrix} \quad (5.2.24)$$

$$\mathbf{y}(t) = \begin{bmatrix} T_a(t) \\ W_{ar}(t) \end{bmatrix} \quad (5.2.25)$$

It can be seen from Equations 5.2.1 to 5.2.15 that the B matrix is a function of the state and disturbance vectors. Therefore, a generalised nonlinear form of statespace representation is more appropriate and will enable a nonlinear controller design to be used.

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}_{nl}(\mathbf{x}(t), \mathbf{d}(t))\mathbf{u}(t) + \mathbf{D}\mathbf{d}(t) \quad (5.2.26)$$

Equation 5.2.27 shows the zone thermodynamics represented in a generalised nonlinear state-space form.

$$\begin{bmatrix} \dot{T}_a(t) \\ \dot{T}_{si}(t) \\ \dot{T}_{se}(t) \\ \dot{T}_{tm}(t) \\ \dot{W}_a(t) \\ \dot{W}_{ar}(t) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 & a_{14} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 0 & 0 \\ 0 & a_{32} & a_{33} & 0 & 0 & 0 \\ a_{41} & 0 & 0 & a_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{55} & 0 \\ a_{61} & a_{62} & 0 & a_{64} & a_{65} & 0 \end{bmatrix} \begin{bmatrix} T_a(t) \\ T_{si}(t) \\ T_{se}(t) \\ T_{tm}(t) \\ W_a(t) \\ W_{ar}(t) \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12}(x, d) \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & b_{52}(x, d) \\ b_{61} & b_{62}(x, d) \end{bmatrix} \begin{bmatrix} \dot{Q}_h(t) \\ \dot{m}_{mv}(t) \end{bmatrix} \\ + \begin{bmatrix} d_{11} & d_{12} & 0 & 0 & d_{15} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & d_{32} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & d_{53} & d_{54} & 0 \\ d_{61} & d_{62} & d_{63} & d_{64} & d_{65} \end{bmatrix} \begin{bmatrix} \dot{Q}_{dist}(t) \\ T_{ex}(t) \\ S_{occ} \\ W_{ex} \\ T_g \end{bmatrix} \quad (5.2.27)$$

$$a_{11} = \frac{1}{M_a C_a} (-U_f A_f - U_r A_r - U_w A_w - \dot{m}_{nv} C_a - U_{tm} A_{tm} - h_i A_s) \quad (5.2.28)$$

$$a_{12} = \frac{1}{M_a C_a} h_i A_s \quad (5.2.29)$$

$$a_{14} = \frac{1}{M_a C_a} U_{tm} A_{tm} \quad (5.2.30)$$

$$a_{21} = \frac{1}{M_{si} C_{si}} h_i A_s \quad (5.2.31)$$

$$a_{22} = \frac{1}{M_{si} C_{si}} \left(-h_i A_s - \frac{k}{wt} A_s \right) \quad (5.2.32)$$

$$a_{23} = \frac{1}{M_{si} C_{si}} \frac{k}{wt} A_s \quad (5.2.33)$$

$$a_{32} = \frac{1}{M_{se} C_{se}} \frac{k}{wt} A_s \quad (5.2.34)$$

$$a_{33} = \frac{1}{M_{se} C_{se}} \left(-\frac{k}{wt} A_s - h_e A_s \right) \quad (5.2.35)$$

$$a_{41} = \frac{1}{M_{tm} C_{tm}} U_{tm} A_{tm} \quad (5.2.36)$$

$$a_{44} = \frac{-1}{M_{tm} C_{tm}} U_{tm} A_{tm} \quad (5.2.37)$$

$$a_{55} = -\frac{1}{M_a} \dot{m}_{nv} \quad (5.2.38)$$

$$a_{61} = \frac{1.388}{M_a C_a} (U_f A_f + U_r A_r + U_w A_w + \dot{m}_{nv} C_a + U_{tm} A_{tm} + h_i A_s) \quad (5.2.39)$$

$$a_{62} = -\frac{1.388}{M_a C_a} h_i A_s \quad (5.2.40)$$

$$a_{64} = -\frac{1.388}{M_a C_a} U_{tm} A_{tm} \quad (5.2.41)$$

$$a_{65} = -\frac{5000}{M_a} \dot{m}_{nv} \quad (5.2.42)$$

$$b_{11} = \frac{1}{M_a C_a} \quad (5.2.43)$$

$$b_{12}(x, d) = -\frac{1}{M_a} (T_a(t) - T_{ex}(t)) \quad (5.2.44)$$

$$b_{52}(x, d) = \frac{1}{M_a} (W_{ex} - W_a(t)) \quad (5.2.45)$$

$$b_{61} = -\frac{1.388}{M_a C_a} \quad (5.2.46)$$

$$b_{62}(x, d) = \frac{1.388}{M_a} (T_a(t) - T_{ex}(t)) + \frac{5000}{M_a} (W_{ex} - W_a(t)) \quad (5.2.47)$$

$$d_{11} = \frac{1}{M_a C_a} \quad (5.2.48) \quad d_{12} = \frac{1}{M_a C_a} (U_r A_r + U_w A_w + \dot{m}_{nv} C_a) \quad (5.2.49)$$

$$d_{15} = \frac{1}{M_a C_a} (U_f A_f) \quad (5.2.50)$$

$$d_{32} = \frac{1}{M_{se} C_{se}} (h_e A_s) \quad (5.2.51)$$

$$d_{53} = \frac{1}{M_a} \quad (5.2.52)$$

$$d_{54} = \frac{1}{M_a} \dot{m}_{nv} \quad (5.2.53)$$

$$d_{61} = -\frac{1.388}{M_a C_a} \quad (5.2.54)$$

$$d_{62} = -\frac{1.388}{M_a C_a} (U_r A_r + U_w A_w + \dot{m}_{nv} C_a) \quad (5.2.55)$$

$$d_{63} = \frac{5000}{M_a} \quad (5.2.56)$$

$$d_{64} = \frac{5000}{M_a} \dot{m}_{nv} \quad (5.2.57)$$

$$d_{65} = -\frac{1.388}{M_a C_a} (U_f A_f) \quad (5.2.58)$$

5.3 Linearised Building Energy System Mathematical Model

The controller designs presented in this thesis requires a linear model of the system to be controlled in order to check transmission zeros, pole locations, design gains etc. The building thermodynamic model described in section 5.2 is nonlinear due to the time varying elements in the B matrix, therefore, it needs to be linearised so that a fully linear model is available for the controller design process.

A linear state-space system can be derived from a nonlinear system by perturbing the nonlinear system about a steady-state operating condition. The linear system perturbed by δ is expressed as follows

$$\delta \dot{\mathbf{x}}(t) = \hat{\mathbf{A}}(\mathbf{x}_0, \mathbf{u}_0, \mathbf{d}_0) \delta \mathbf{x}(t) + \hat{\mathbf{B}}(\mathbf{x}_0, \mathbf{u}_0, \mathbf{d}_0) \delta \mathbf{u}(t) + \hat{\mathbf{D}}(\mathbf{x}_0, \mathbf{u}_0, \mathbf{d}_0) \delta \mathbf{d}(t) \quad (5.3.1)$$

Where x_0, u_0, d_0 is the operating point and $\hat{\mathbf{A}}, \hat{\mathbf{B}}$ and $\hat{\mathbf{D}}$ are the Jacobians of $f\mathbf{n}(\mathbf{x}_o, \mathbf{u}_o, \mathbf{d}_o)$ in the nonlinear system $\dot{\mathbf{x}}(t) = f\mathbf{n}(\mathbf{x}, \mathbf{u}, \mathbf{d})$ with respect to \mathbf{x} , \mathbf{u} and \mathbf{d} .

$$\hat{\mathbf{A}}(\mathbf{x}_0, \mathbf{u}_0, \mathbf{d}_0) = \begin{bmatrix} a_{11} - \frac{1}{M_a} \dot{m}_{mv(0)} & a_{12} & 0 & a_{14} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 0 & 0 \\ 0 & a_{32} & a_{33} & 0 & 0 & 0 \\ a_{41} & 0 & 0 & 0 & a_{44} & 0 \\ 0 & 0 & 0 & 0 & a_{55} - \frac{1}{M_a} \dot{m}_{mv(0)} & 0 \\ a_{61} + \frac{1.388}{M_a} \dot{m}_{mv(0)} & a_{62} & 0 & a_{64} & a_{65} - \frac{5000}{M_a} \dot{m}_{mv(0)} & 0 \end{bmatrix} \quad (5.3.2)$$

$$\hat{\mathbf{B}}(\mathbf{x}_0, \mathbf{u}_0, \mathbf{d}_0) = \begin{bmatrix} b_{11} & \frac{1}{M_a} T_{\text{ex}(0)} - \frac{1}{M_a} T_{\text{a}(0)} & & & \\ 0 & 0 & & & \\ 0 & 0 & & & \\ 0 & 0 & & & \\ 0 & \frac{1}{M_a} W_{\text{ex}(0)} - \frac{1}{M_a} W_{\text{a}(0)} & & & \\ b_{61} & \frac{1}{M_a} (1.388T_{\text{a}(0)} - 1.388T_{\text{ex}(0)} + 5000W_{\text{ex}(0)} - 5000W_{\text{a}(0)}) & & & \end{bmatrix} \quad (5.3.3)$$

$$\hat{\mathbf{D}}(\mathbf{x}_0, \mathbf{u}_0, \mathbf{d}_0) = \begin{bmatrix} d_{11} & d_{12} + \frac{1}{M_a} \dot{m}_{\text{mv}(0)} & 0 & 0 & d_{15} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & d_{32} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & d_{53} & d_{54} + \frac{1}{M_a} \dot{m}_{\text{mv}(0)} & 0 \\ d_{61} & d_{62} - \frac{1.388}{M_a} \dot{m}_{\text{mv}(0)} & d_{63} & d_{64} + \frac{5000}{M_a} \dot{m}_{\text{mv}(0)} & d_{65} \end{bmatrix} \quad (5.3.4)$$

Therefore, the linearisation does not alter the structure of the state-space system; however, terms are modified in the A, B and D matrices.

5.4 Modern Office Case Study

In this study, an HVAC system providing heat and ventilation to a modern office space is simulated. The office space is a single zone with a floor area of 12m by 12m and a ceiling height of 4m. The control system varies the rate of heat output by the heater and the mass flow rate of the ventilation system in order to regulate the indoor air temperature and relative humidity at the desired setpoints.

Large open spaces or poorly positioned sensors can lead to a sensor lag developing in the control system. This occurs when there is a physical separation between the area to be controlled and the sensor measuring the controlled variable, resulting in a pure time delay in the closed loop system. This time delay can cause instability and may require the controller's responsiveness to be reduced (Setiawa

et al., 2000). The simulations will investigate each controller design's sensitivity and robustness with varying degrees of sensor lag.

5.4.1 Simulated zone properties

Zone model properties, shown in Tables 5.1 – 5.7, were chosen to be indicative of a modern office building, with dimensions typical of an open-plan space. Values for h_e and h_i were calculated using the method detailed in Murphy et al., 2013.

Table 5.1 – Zone mass.

$M_a = 691 \text{ kg}$
$M_{si} = 7000 \text{ kg}$
$M_{se} = 7000 \text{ kg}$
$M_{tm} = 8000 \text{ kg}$

Table 5.3 – Zone area.

$A_s = 192 \text{ m}^2$
$A_f = 144 \text{ m}^2$
$A_r = 144 \text{ m}^2$
$A_w = 51 \text{ m}^2$
$A_{tm} = 138 \text{ m}^2$

Table 5.5 – Overall heat transfer coefficients.

$U_f = 0.2 \text{ W/m}^2\text{K}$
$U_r = 0.13 \text{ W/m}^2\text{K}$
$U_w = 1.5 \text{ W/m}^2\text{K}$
$U_{tm} = 2 \text{ W/m}^2\text{K}$

Table 5.2 – Zone thermal properties.

$k = 0.1 \text{ W/mK}$
$wt = 0.5 \text{ m}$
$h_e = 0.12 \text{ W/m}^2\text{K}$
$h_i = 0.11 \text{ W/m}^2\text{K}$

Table 5.4 – Occupant and environment properties.

$m_{nv} = 0.02 \text{ kg/s}$
$W_{ex} = 0.01$
$S_{occ} = 0.02 \text{ g/s}$
$T_g = 10 \text{ }^\circ\text{C}$

Table 5.6 – Zone specific heat capacities.

$$C_a = 1012 \text{ J/kgK}$$

$$C_{si} = 1000 \text{ J/kgK}$$

$$C_{se} = 1000 \text{ J/kgK}$$

$$C_{tm} = 900 \text{ J/kgK}$$

Table 5.7 – Building heating and ventilation properties.

$$LL_{Qh} = 0 \text{ W}$$

$$UL_{Qh} = 6000 \text{ W}$$

$$LL_{mv} = 0 \text{ kg/s}$$

$$UL_{mv} = 0.35 \text{ kg/s}$$

$$\tau_h = 120 \text{ s}$$

$$\tau_v = 60 \text{ s}$$

5.5 Controller Design

5.5.1 Benchmark – Proportional Integral Control

The PI controller design is single input single output (SISO), meaning that it can only control one variable at any one time. Therefore, it is necessary to design two individual control loops; one for temperature control and another for relative humidity control. The major drawback is that any coupling between the heating and ventilation systems is not accounted for explicitly in the controller designs and will consequently lead to performance degradation.

The SISO (Single Input Single Output) PI control law is given as follows:

$$u_c(t) = z(t) + K_p e(t) \quad (5.5.1)$$

$$\dot{z}(t) = K_I e(t) \quad (5.5.2)$$

This control law was applied to control, individually, the heating and ventilation systems. The gains for each of the systems were iteratively tuned to provide the best performance without an oscillatory response occurring in the controller.

Common to both control loops is the need for some form of integrator antiwindup. Reaching limits of the actuators leads to the integrator term rapidly building (“winding up”) whilst the controller is not able to provide any more useful output. This leads to the control signal increasing in magnitude causing the actuator to remain fixed on its limit, often resulting in overshoot or limit cycles. The PI controller is implemented with a digital form of anti-windup, where the integrator part of the PI controller is turned off when an actuator limit is reached.

As the controller is switched off at the end of each working day it is necessary to ensure that the controller is properly initialised when it is switched back on. When the controller is switched on the output should be zero, therefore, the regulator term, $z(t)$, must be initialized so that this is the case. Thus, for each channel during initialisation the regulator is reset as follows:

$$z(t) = -\frac{K_P}{K_I} e(t) \quad (5.5.3)$$

5.5.2 RIDE Controller Design

The first stage in the design of the RIDE control system is to establish the **MB** matrix for Equivalent Control vector. The measurement can be set as the output and since the outputs to be controlled are T_a and W_{ar} the M matrix is

$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ (5.5.4) Referring to Equation 5.2.28 the nonlinear \mathbf{B}_{nl} matrix is

$$\mathbf{B}_{nl}(\mathbf{x}, \mathbf{d}) = \begin{bmatrix} \frac{1}{M_a C_a} & -\frac{1}{M_a} (T_a(t) - T_{ex}(t)) \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{M_a} (W_{ex} - W_a(t)) \\ -\frac{1.388}{M_a C_a} & \frac{1.388}{M_a} (T_a(t) - T_{ex}(t)) + \frac{5000}{M_a} (W_{ex} - W_a(t)) \end{bmatrix} \quad (5.5.5)$$

Therefore

$$\mathbf{M}\mathbf{B}_{nl}(\mathbf{x}, \mathbf{d}) = \begin{bmatrix} \frac{1}{M_a C_a} & -\frac{1}{M_a} (T_a(t) - T_{ex}(t)) \\ -\frac{1.388}{M_a C_a} & \frac{1.388}{M_a} (T_a(t) - T_{ex}(t)) + \frac{5000}{M_a} (W_{ex} - W_a(t)) \end{bmatrix} \quad (5.5.6)$$

The Equivalent Control vector and controller gain matrices can now be constructed from Equations 3.2.6, 3.2.10 and 3.2.11. This is a remarkable result because in order to implement a multivariable Inverse Dynamics controller the only information about the structure required are the mass and specific heat capacity of the air. The sensory requirements are simply the external and internal air temperatures, the internal relative humidity and internal and external absolute humidity, which are all practically obtainable. From this it can be seen that an advanced HVAC controller design can be relatively simple to implement and would only require a few extra sensors (external temperature and humidity sensors) compared to a traditional PI design.

Having established that a nonlinear dynamic inverse design is feasible it is then necessary to ensure that the transmission zeros of the measurement vector are stable. The dynamic inverse places closed loop poles over the transmission zeros so it is clear that unstable measurement vector transmission zeros will result in unstable closed loop dynamics.

Transmission zeros are a linear construct so it is necessary to use the linearised model of the building zone derived in Equations 5.3.1 – 5.3.4. The transmission zeros are the values of s for which the rank of matrix, given by Equation 5.5.7, drops.

$$\begin{bmatrix} s\mathbf{I} - \widehat{\mathbf{A}} & -\widehat{\mathbf{B}} \\ \mathbf{C} & 0 \end{bmatrix} \quad (5.5.7)$$

The operating point $\mathbf{x}_0, \mathbf{u}_0, \mathbf{d}_0$ is chosen to be typical of steady state operation. Only the values of $\mathbf{x}_0, \mathbf{u}_0, \mathbf{d}_0$ that appear in the linearised state matrices are shown.

$$\mathbf{x}_0 = \begin{bmatrix} T_{a(0)} \\ W_{a(0)} \end{bmatrix} = \begin{bmatrix} 304 \text{ K} \\ 0.03 \end{bmatrix} \quad (5.5.8)$$

$$\mathbf{u}_0 = [\dot{m}_{mv(0)}] = [0.15 \text{ kg/s}] \quad (5.5.9)$$

$$\mathbf{d}_0 = \begin{bmatrix} \dot{m}_{nv(0)} \\ T_{ex(0)} \end{bmatrix} = \begin{bmatrix} 0.03 \text{ kg/s} \\ 2836 \text{ K} \end{bmatrix} \quad (5.5.10)$$

At the operating point the solution to Equation 5.5.7 gives the location of the transmission zeros.

$$\mathbf{T}_z = \begin{bmatrix} -3.833 \times 10^{-5} \\ -2.859 \times 10^{-5} \\ -2.95 \times 10^{-6} \\ 0 \end{bmatrix} \quad (5.5.11)$$

The location of the transmission zeros are all in the left half plane (real negative) and so are all stable. Therefore, it is possible to move to the next stage of the controller design; specifying closed loop transient dynamics.

The closed loop transient characteristics of the RIDE controller are ultimately set by the values of the gain matrices \mathbf{K}_P and \mathbf{K}_I . As described in Chapter 2

$$\mathbf{K}_P = 2v_w(\mathbf{M}\widehat{\mathbf{B}})^{-1} \quad (5.5.12)$$

$$\mathbf{K}_I = w^2(\mathbf{M}\hat{\mathbf{B}})^{-1} \quad (5.5.13)$$

Where ν is the specified closed loop damping ratio and w is the specified closed loop natural frequency. As has been extensively discussed, choosing the gain is a trade-off between the closed loop response and the stability of the controller with regards to the actuator and sensor dynamics. It is desired that the commanded temperature and humidity be tracked as quickly as possible but this has to be balanced with ensuring that there is enough separation between the bandwidth of the sensors or actuators and the closed loop bandwidth. As a rule of thumb the closed loop bandwidth should be no more than one third of the smallest actuator or sensor bandwidth (Counsell, 1992).

Assuming that there is minimal sensor lag for the temperature sensor and considering that the actuators for the heater have a time constant of two minutes and that $\omega_n \approx \frac{\text{rise time}}{1.6}$ (Franklin et al. , 2001), the values shown in Table 5.8 provide an acceptable response time with a sufficient bandwidth separation between the closed loop and actuator/sensor dynamics.

Table 5.8 – Closed loop tuning parameters for RIDE controller.

ω 0.00175 ν
0.8

The RIDE controller also needs to be initialized at start-up, as described in the PI design. Therefore, upon starting the controller the regulator is set to the following:

$$\mathbf{z}(t) = \mathbf{K}_I^{-1} (\mathbf{K}_P \mathbf{w}(t) - \mathbf{u}_{eq}(t)) \quad (5.5.14)$$

In order to prevent the controller output from exceeding the actuator limits the

VSC design of the RIDE controller is employed as described by Muir and Bradshaw. The regulator is switched by turning on or off the error signal based on the heating and ventilation system output. The controller limits LL_1 , LL_2 , UL_1 and UL_2 are set equal to the lower and upper heating power and ventilation mass flow rate limits respectively.

5.6 Simulation

Firstly, the performance of the RIDE and PI controllers with a small and large sensor delay were investigated. The small sensor delay was set as 2 minutes, with the large sensor delay set at 7 minutes. Two sets of simulations with the PI controller were performed: one with anti-windup on the integrator and one without. The purpose of this was to investigate the influence and importance of the anti-windup scheme in reducing setpoint overshoot and overall energy usage.

Secondly, the performance of the VTR controller, with the large sensor delay present, was investigated. The intention of the VTR controller design was to reduce controller oscillations, which would otherwise be induced in the RIDE or PI controllers by the large sensor lag. The overall heater energy usage and ventilation system mass flow rate was compared when using each of the controller designs.

5.6.1 Simulation Parameters

The simulations were performed by numerically integrating using a 4th order Runge-Kutta algorithm with a time step of 8 seconds. Results are presented over a 100 hour period from days 80 to 84 (early March), with external air temperature and solar gains taken from data for Glasgow, Scotland. This period and location was chosen so as to provide challenging conditions for temperature and humidity regulation, due to the changeable spring weather conditions and need for significant heating during office hours. The setpoints for temperature and humidity were set at 21 °C and 55% respectively. The control system is active between the typical office hours of 7am and 7pm, out of these hours the HVAC system is switched off.

5.6.2 Simulation Results – PI and RIDE

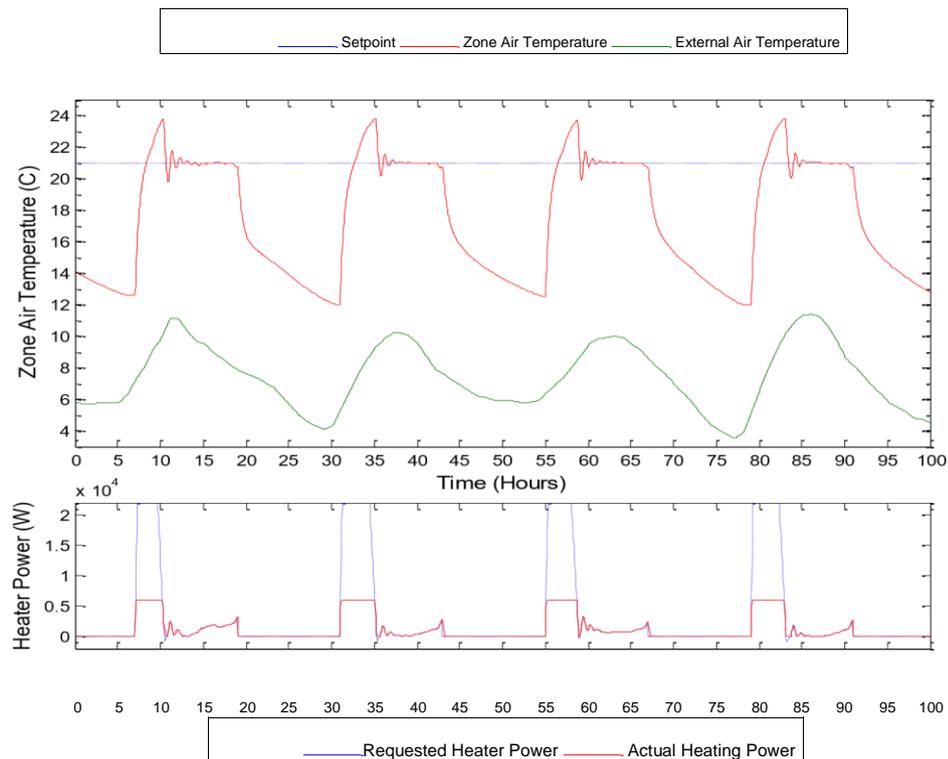


Figure 5.1 – Air temperature control, PI (no anti-windup), 2 min lag.

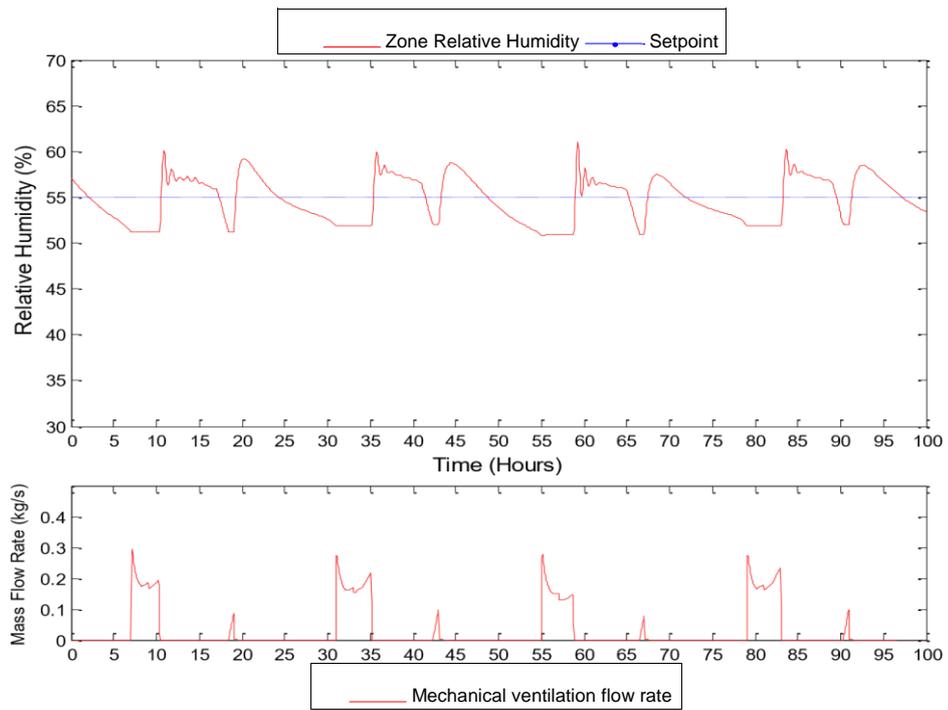


Figure 5.2 – Relative humidity control, PI (no anti-windup), 2 min lag.

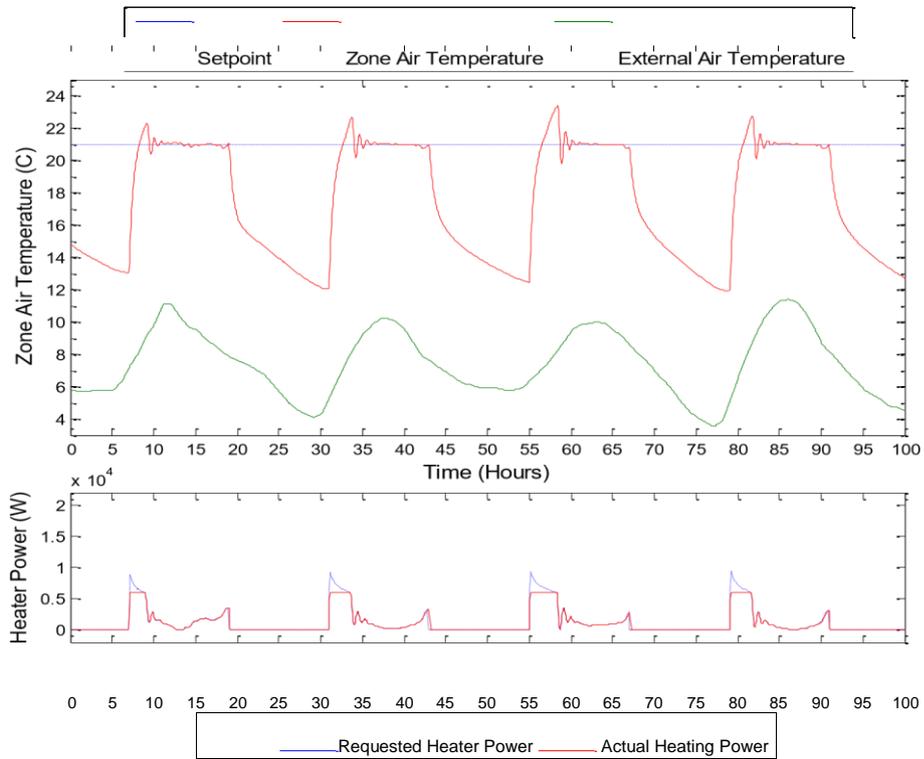


Figure 5.3 – Air temperature control, PI, 2 min lag.

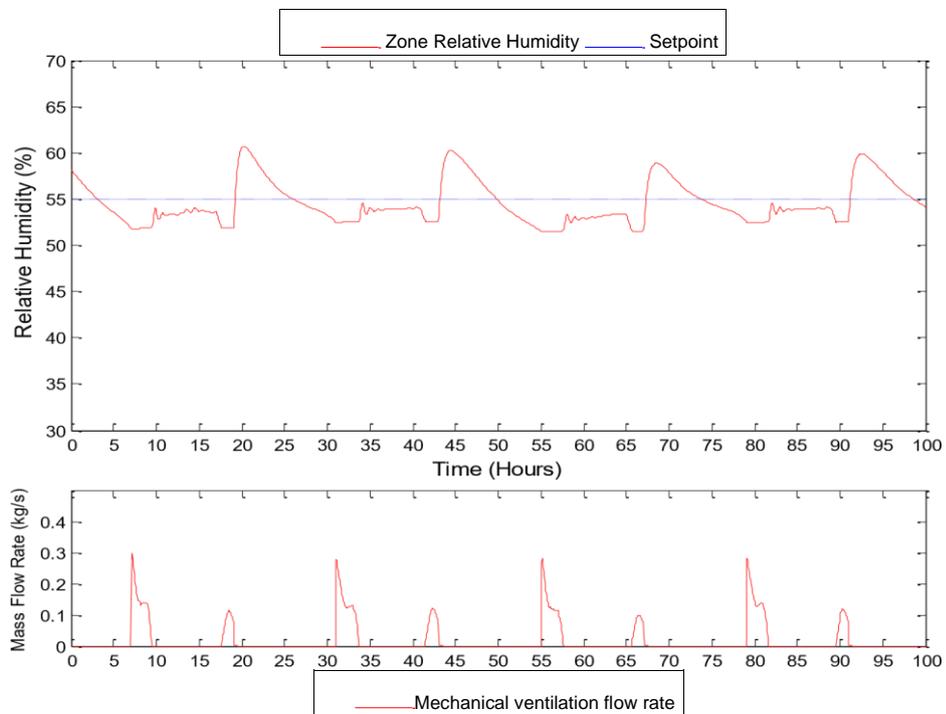


Figure 5.4 –

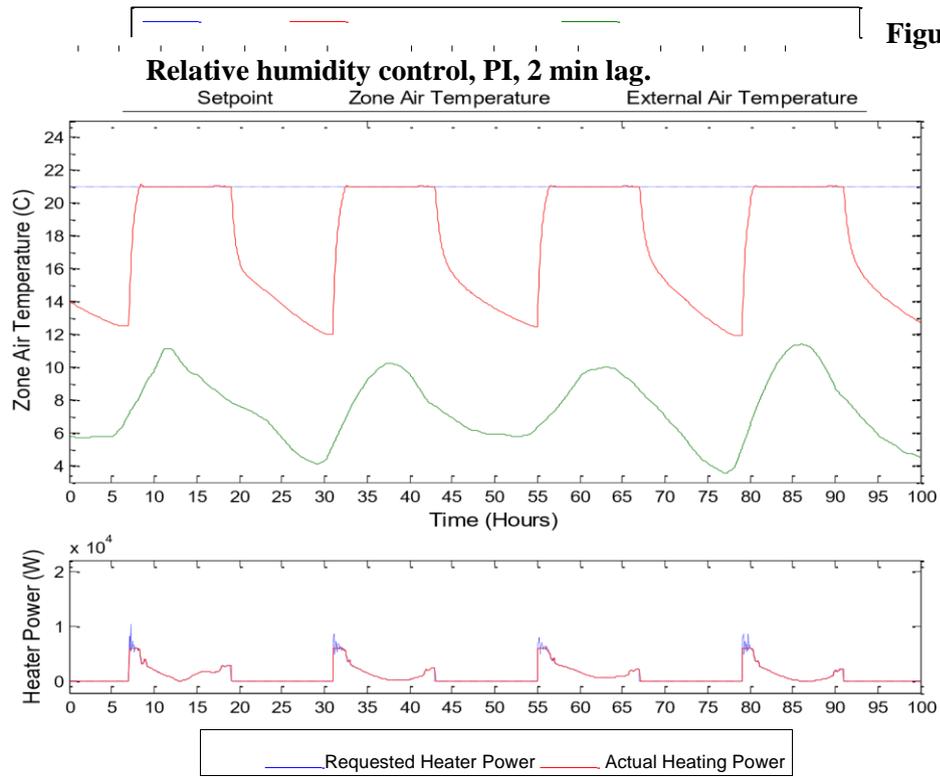
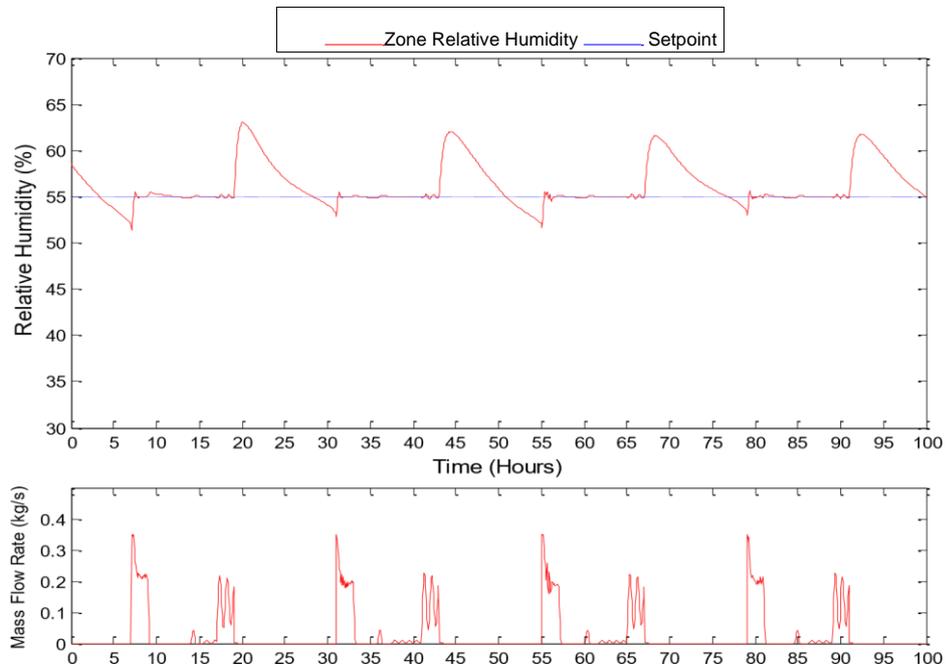


Figure 5.5 – Air temperature control, RIDE, 2 min lag.



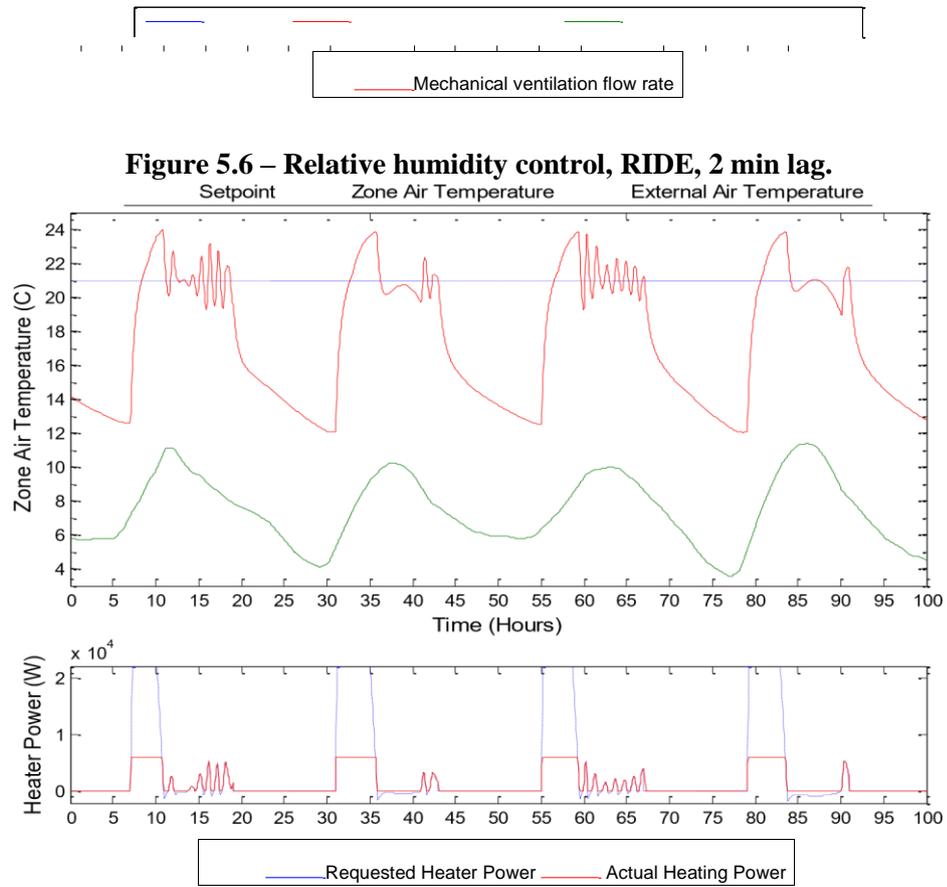


Figure 5.7 – Air temperature control, PI (no anti-windup), 7 min lag.



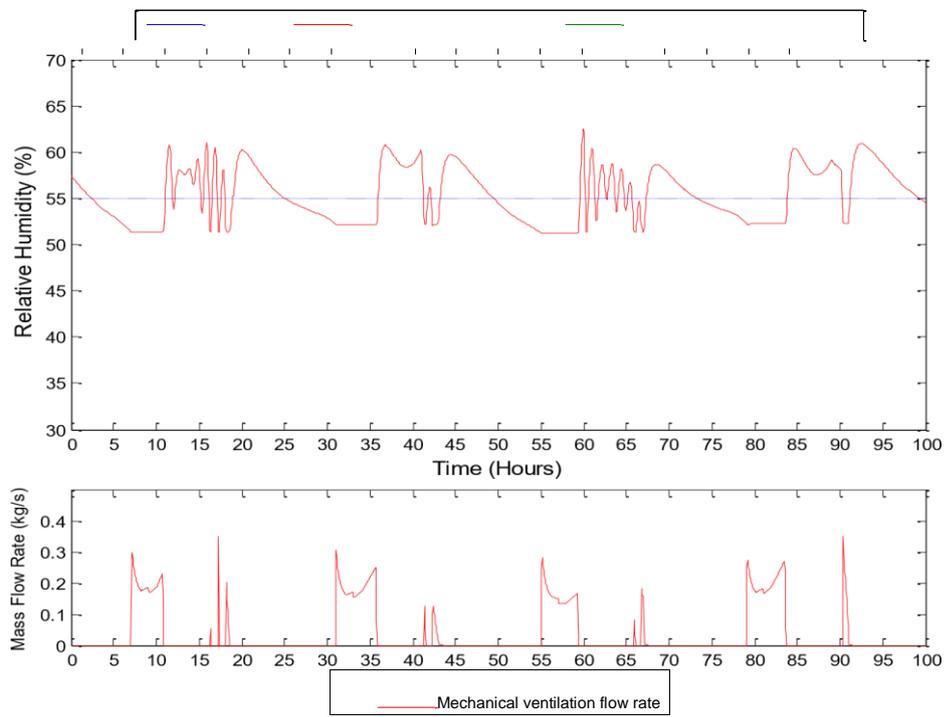


Figure 5.8 – Relative humidity control, PI (no anti-windup), 7 min lag.

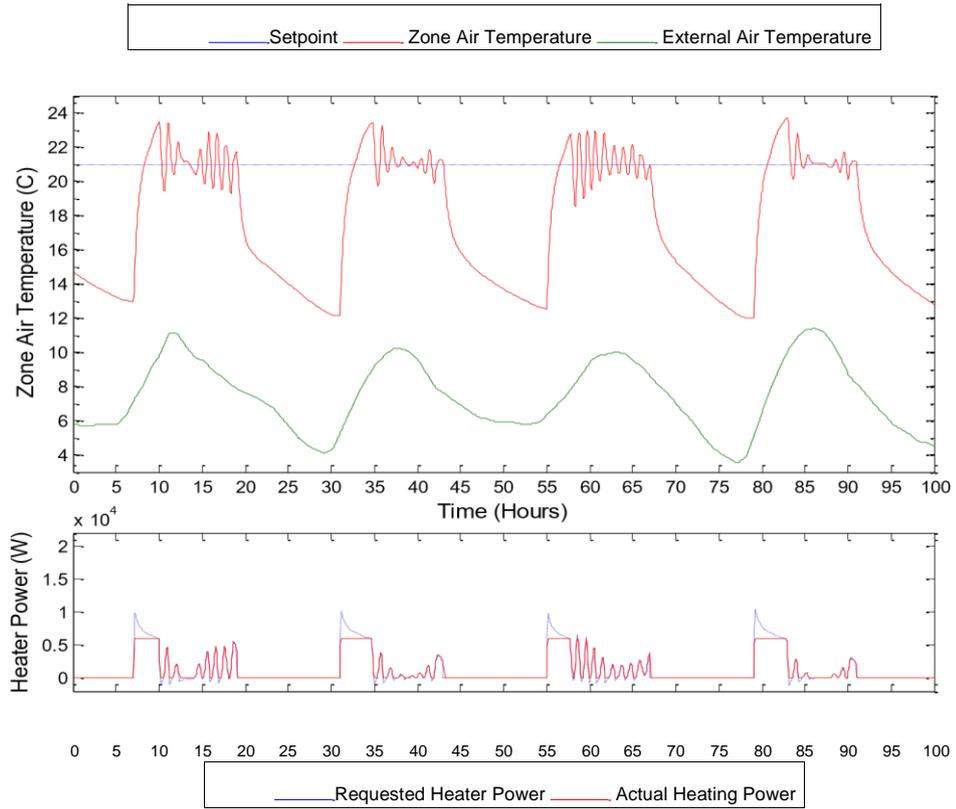


Figure 5.9 – Air temperature control, PI, 7 min lag.

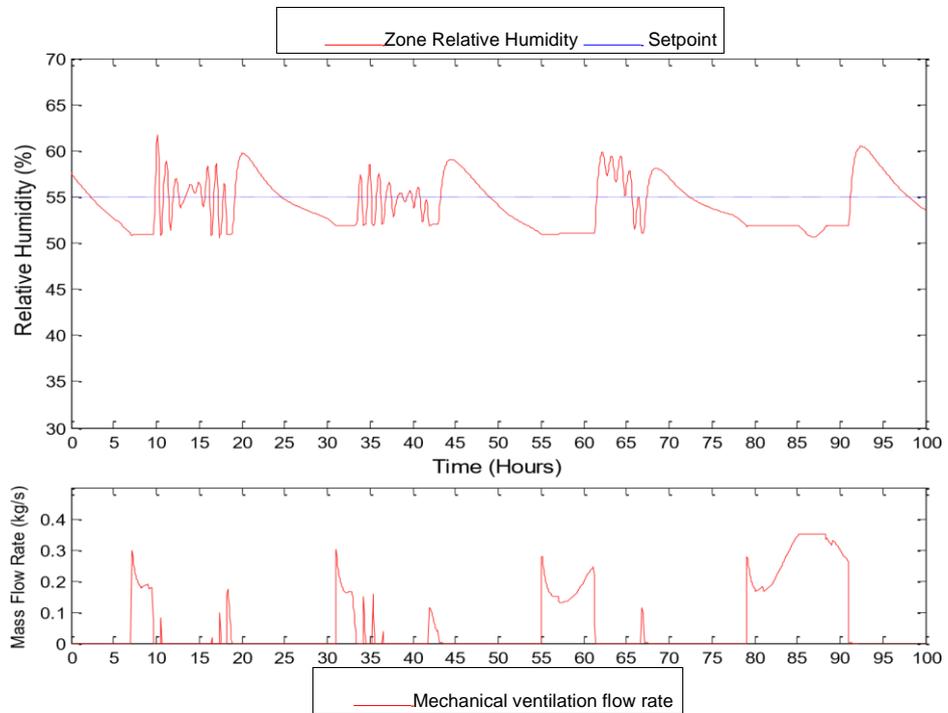


Figure 5.10 – Relative humidity control, PI, 7 min lag.

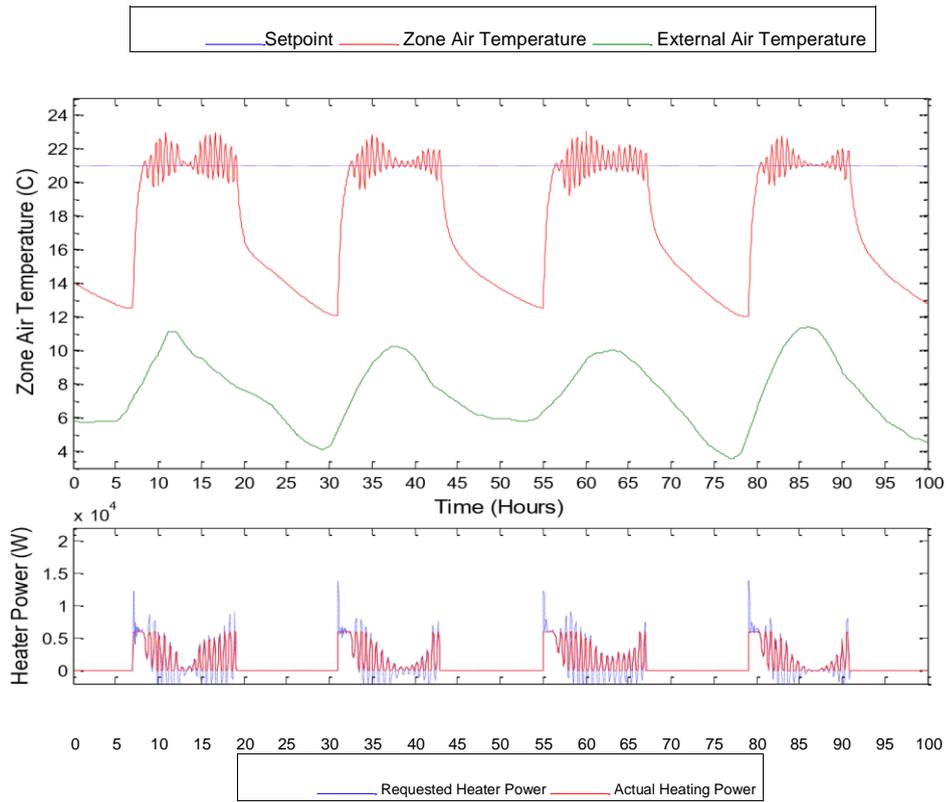


Figure 5.11 – Air temperature control, RIDE, 7 min lag.

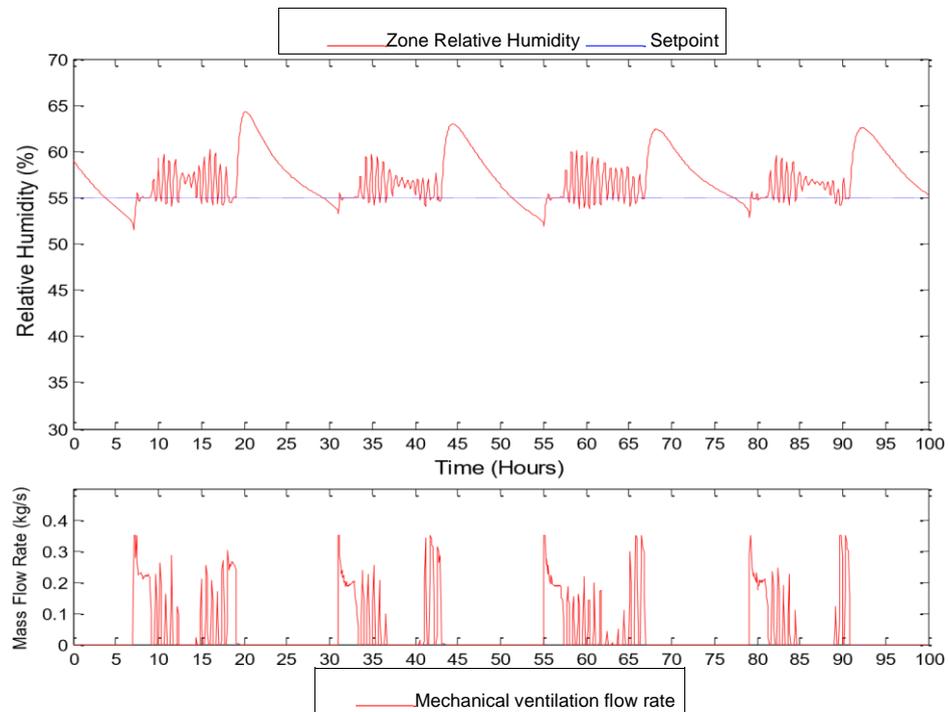


Figure 5.12 – Relative humidity control, RIDE, 7 min lag.

5.6.3 VTR Controller Design

With a large sensor lag the transient response characteristic of no oscillations is not attainable with either a PI or a RIDE design; this is demonstrated by Figures 5.1 – 5.6. Therefore a VTR design is needed to meet the transient response requirements.

The first stage in the VTR controller design is to select the parameter Y which will determine the steady state gain of the controller when the setpoint is being tracked. This parameter should be selected so that the gain is low enough not to cause any oscillations or unstable behaviour caused by a high gain. Hence, the value of ω was reduced so that the integral gain for both control channels was reduced until no steady state oscillations were observed (Figures 5.9 and 5.10). The gain needed to be reduced by 70% which corresponds to a value of 0.55 for Y . The designer must be careful not to reduce the integral gain so much that the tracking and disturbance rejection properties of the controller are compromised.

The second stage is to select the parameter X which will determine how the gain changes with error – the larger the value of X , the stronger the effect of the dynamic gain change. The design objective for the VTR controller is to maintain the response time of the standard RIDE controller but with the elimination of steady state disturbances, therefore, the value of X needs to be designed to yield the same response time as the RIDE controller. Chapter 3 describes a method to estimate the value of X for a corresponding maximum expected error and value of Y . The final method is shown again below in Equation 5.6.1

$$\mathbf{X}_{\text{avg}} = (-5\mathbf{E}^2\mathbf{Y} + \sqrt{5\sqrt{9\mathbf{E}^4 - 4\mathbf{E}^4\mathbf{Y}^2}})(3\mathbf{E}^4)^{-1} \quad (5.6.1)$$

A value of Y was chosen to provide adequate reduction in gain at steady-state and reduction in sensitivity to un-modelled dynamics.

$$\mathbf{Y} = \begin{bmatrix} 0.55 & 0 \\ 0 & 0.55 \end{bmatrix} \quad (5.6.2)$$

The value of the maximum error (emax) is largely determined by the setpoint. An HVAC system is an ideal system for a VTR controller design as the setpoint is very consistent (normally set at 21°C and 50% relative humidity) and a step input. Therefore, the value of emax is given by the difference between the steady state value of the controlled output when the controller is off and the value of the setpoint. This is perhaps the most important design consideration as this will not be a fixed value as even if the setpoint is fixed the steady state value of the output when the controller is off will vary with weather conditions. A compromise needs to be made between underestimating the off period output (resulting in an over damped response) and overestimating the off period output (resulting in an under damped response). In this case an over damped response is probably preferable to an under damped one so it is best to allow some “safety factor” in the estimate of emax. From Figure 5.11 it can be seen that the minimum temperature is 12°C, therefore, given that the setpoint is at 21°C, and allowing for some safety factor, emax was set at 10 for the temperature control. For the humidity control channel the minimum humidity was observed at 40% so for a setpoint of 50% and for simplicity of the design the emax for the humidity control was set at 10 as well. Inputting the parameters into equation 5.6.1 results in the following estimate for X:

$$\mathbf{X}_{\text{avg}} = \begin{bmatrix} 0.0116 & 0 \\ 0 & 0.0116 \end{bmatrix} \quad (5.6.3)$$

These are estimates and, as with all real nonlinear systems, need to be fine-tuned to produce the best possible controller performance. It was noted in Chapter 3 that the average function method tended to produce an over-damped response. Therefore, using the estimated values as a starting point, the best value could be expected to be slightly larger than the estimate. After manual tuning it was found that the optimum value for

\mathbf{X} was:

$$\mathbf{X} = \begin{bmatrix} 0.022 & 0 \\ 0 & 0.022 \end{bmatrix} \quad (5.6.4)$$

5.6.4 Simulation Results - VTR

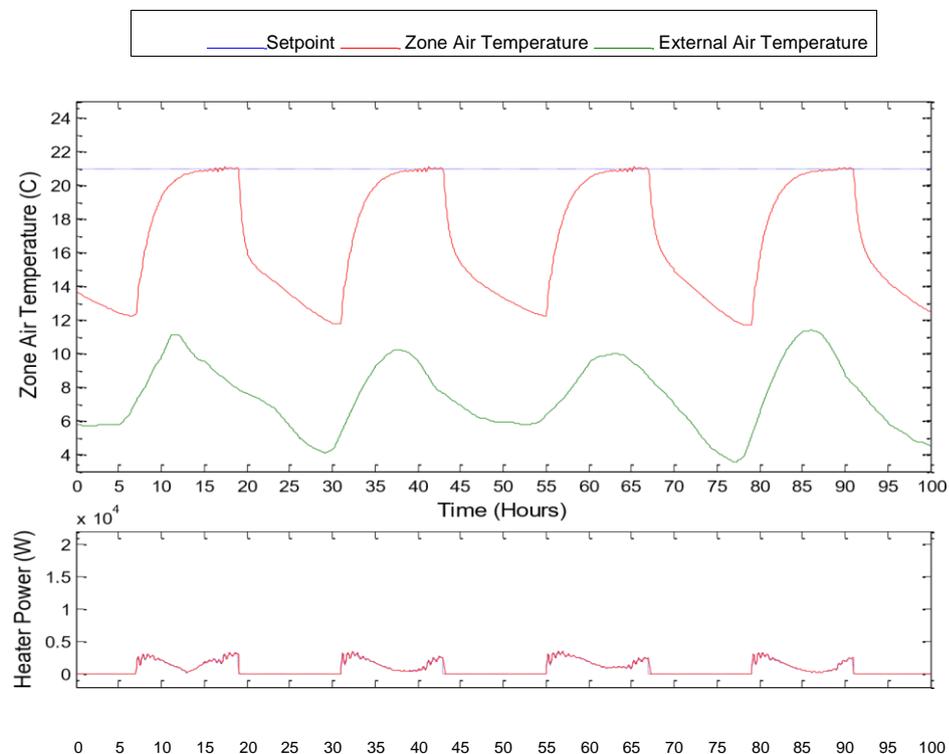




Figure 5.13 – Air temperature control, VTR ($K_I = K_{I\text{crit}}$), 7 min lag.

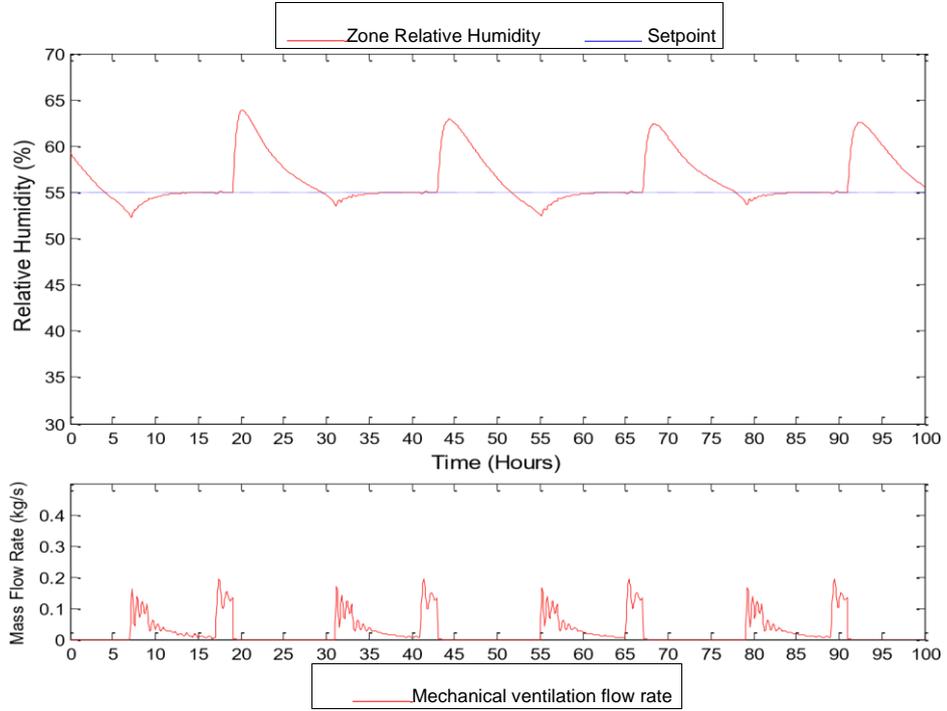


Figure 5.14 – Relative humidity control, VTR ($K_I = K_{I\text{crit}}$), 7 min lag.

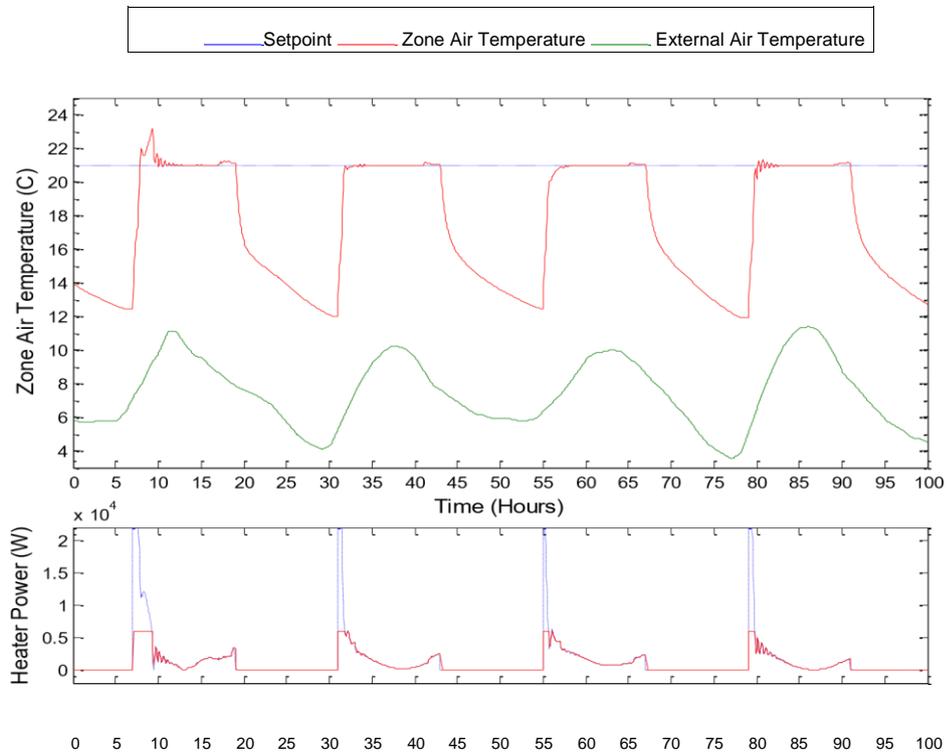




Figure 5.15 – Air temperature control, VTR, 7 min lag.

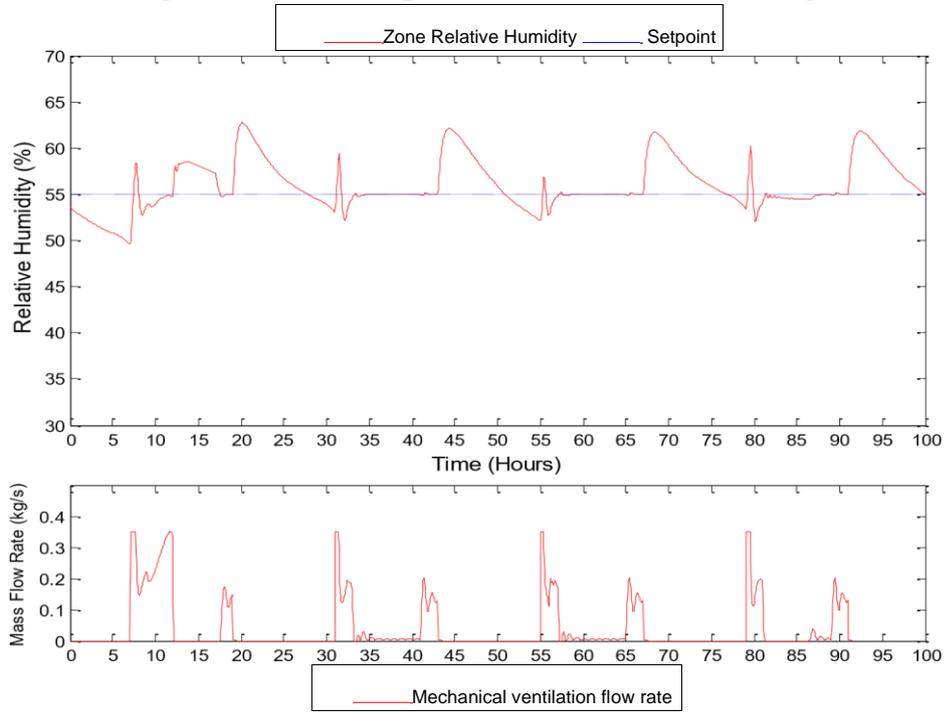


Figure 5.16 – Relative humidity control, VTR, 7 min lag.

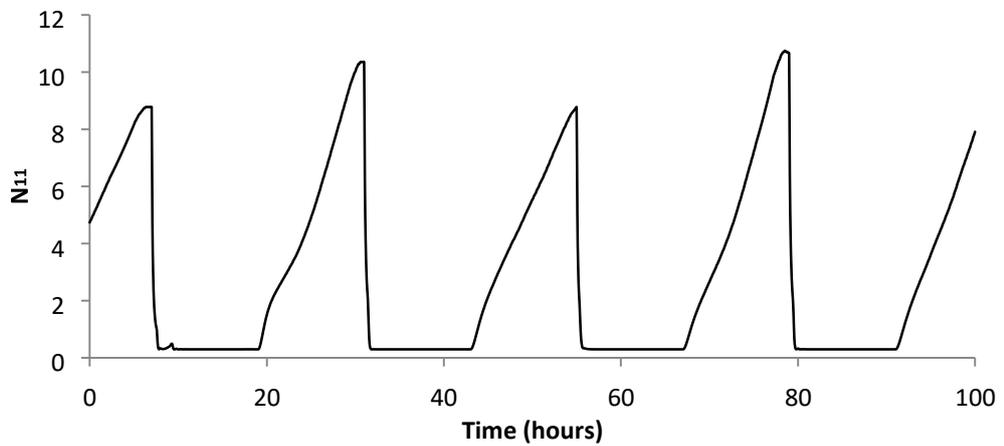


Figure 5.17 – Evolution of VTR gain N_{11} with temperature error.

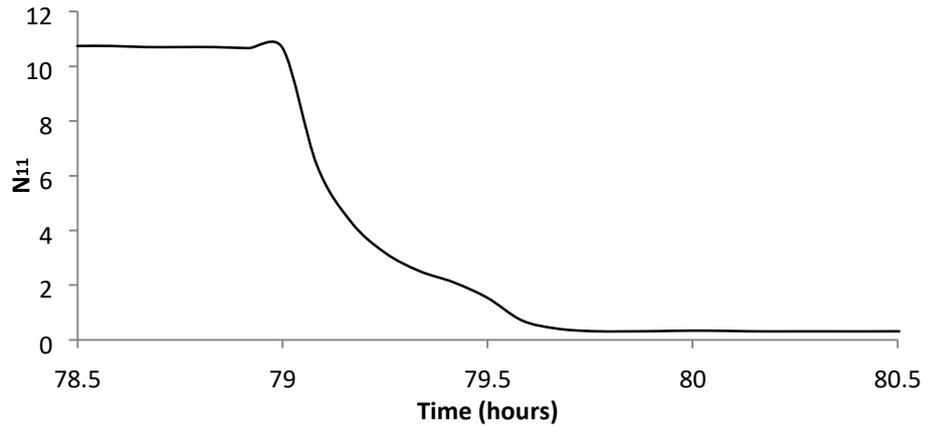


Figure 5.18 – Evolution of VTR gain N_{11} with temperature error, detailed view.

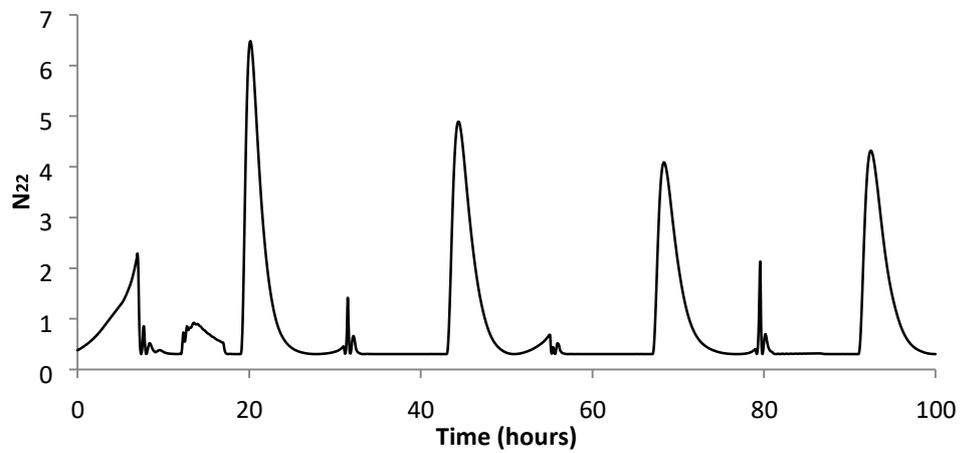


Figure 5.19 – Evolution of VTR gain N_{22} with humidity error.

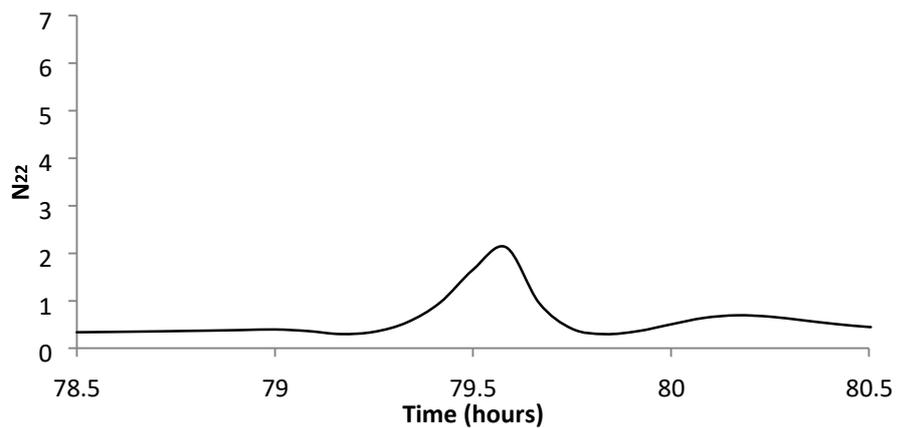
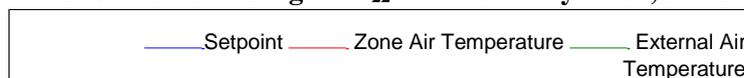


Figure 5.20 – Evolution of VTR gain N_{22} with humidity error, detailed view.



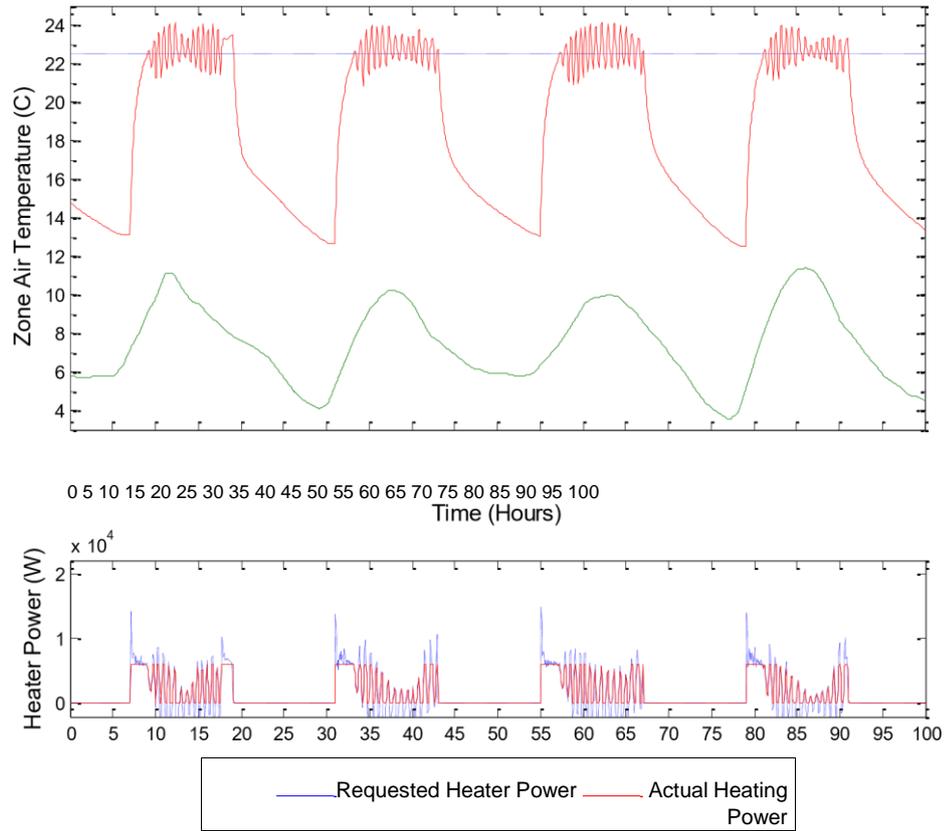


Figure 5.21 – Air temperature control at temp. setpoint of 22.5 °C, RIDE, 7 min lag.

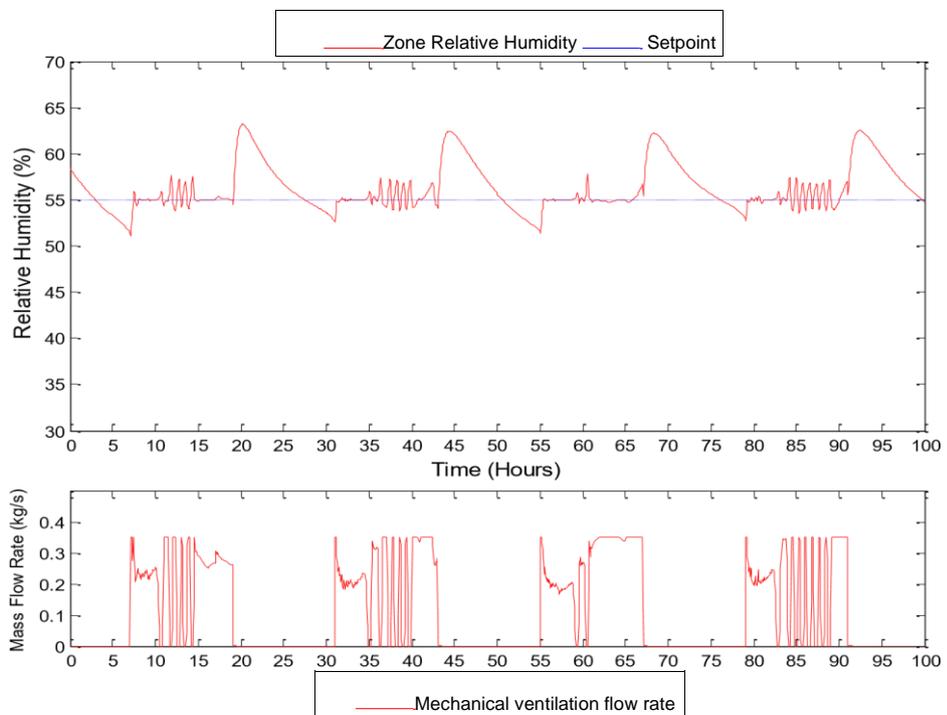


Figure 5.22 – Relative humidity control at temp. setpoint of 22.5 °C, RIDE, 7 min lag.
Table 5.9 – Total heating energy usage.

	Heating 2min lag (kWh)	Heating 7min lag (kWh)	Heating 7min lag at 22.5C (kWh)
PI	106.71	118.70	—
PI (no AW)	119.58	130.52	—
RIDE	97.10	105.18	142.25
VTR	—	95.06	—

Total ventilation usage.

Table 5.10 –

	Ventilation 2min lag (kg)	Ventilation 7min lag (kg)	Ventilation 7min lag at 22.5C (kg)
PI	7145	20189	—
PI (no AW)	10410	10380	—
RIDE	10989	15094	39658
VTR	—	13443	—

5.6.5 Results Discussion

The first set of simulation results, those of Figures 5.1 through to 5.12, compare the performance of a PI controller, a PI controller with no integrator anti-windup and a RIDE controller, with a small, 2 minute sensor delay. In summary, there was a significant difference between the controllers in the control of air temperature

and humidity; most notably in first overshoot of the temperature setpoint and the cross-coupling between the heating and ventilation systems.

Figures 5.1 and 5.2 show simulation results for air temperature and humidity control using a PI algorithm with no anti-windup. The most immediately apparent problem using this controller is the large, 2.5 °C, overshoot of the 21 °C temperature setpoint. This is due to absence of anti-windup on the integrator. The windup of the integrator can be observed between hours 7-10, 31-35, 55-58 and 79-83 where the heating actuator has reached the power limit but the control signal continues to increase to over 3x this limit. This causes the heater to remain at its maximum output for longer than is necessary to reach the setpoint, thus causing a large overshoot of the temperature setpoint. After the initial overshoot there is a period of small amplitude (1-2 °C peak to peak) oscillation for between 2 and 3 hours, after which the tracking of the setpoint is good, with less than 0.5 °C deviation from the setpoint. Figure 5.2 shows control of the indoor relative humidity using the PI controller with no anti-windup. There is a strong coupling between the humidity and air temperature controllers – as the air temperature is raised by the heating system the relative humidity of the air will decrease, therefore the ventilation system must bring in more humid external air to compensate. If the temperature of the external air is below that of indoor the air, the indoor air temperature will then decrease, requiring more heating. In non-multivariable controllers, such as single-input-single-output PI, this coupling can cause oscillations to occur as the two systems fight each other, alternatively, one system can dominate to the detriment of the other system. Figure 5.2 shows that the humidity is between 4-5% below the setpoint for approximately 4 hours after the ventilation system starts; this is caused by the large temperature overshoot in the heating system acting as a disturbance to the ventilation system. Following this, there is a humidity offset of between 3-4% above the setpoint, which the ventilation controller is unable to respond to. This demonstrates that,

in this case, the heating system dominates. The absence of anti-windup on the ventilation controller does not affect the performance as the ventilation system as maximum flow rate is not reached.

Figures 5.3 and 5.4 display the simulation results for the PI controller with antiwindup active. The results are similar to those of Figures 5.1 and 5.2, with the most significant difference being the reduced initial temperature overshoot. With the integrator anti-wind up active the control signal briefly exceeds the heating limit but then quickly returns to the maximum actuator value, as can be seen in Figure 5.3. This results in a reduced initial overshoot of approximately 50% compared to the PI controller without anti-windup (Figure 5.1). The effect that this has on the total energy used to heat the zone can be examined by calculating the kWh usage of the heating system over the 100 hour period of the simulation. The results are displayed in Table 5.9. The results show that the absence of antiwindup in the PI algorithm causes the energy usage to be increased by 12% - a significant increase, underlining the importance of adequate anti-windup in the heating controller design. The introduction of anti-windup also has an effect on the performance of the humidity controller. The initial negative offset from the humidity setpoint is reduced slightly and the subsequent positive offset is significantly reduced, now below the setpoint by 1-2%. The controller is still unable to accurately track the setpoint, however, the deviation from the setpoint is reduced. As the overall zone temperature is lower, compared to the PI controller with no anti-windup, it would be expected that the ventilation system would not have to be as active in order to maintain desired humidity. This is confirmed by Table 5.10 which shows that the use of anti-windup reduces the total mass of air moved over the 100 hour period by 46%.

Figures 5.5 and 5.6 display the simulation results of the multivariable RIDE controller with a small sensor lag. The overshoot of the air temperature setpoint,

that is present with the PI controllers, is completely eliminated by using the RIDE controller. Referring to Figure 5.5 it can be seen that the control signal exceeds the heating limit by a small amount but almost instantly returns to the upper limit. Thus, the heater remains at its maximum level for only as long as is necessary to reach the setpoint, and any overshoot is avoided. This demonstrates that the integrator anti-wind present in the RIDE controller is superior when compared to that implemented in the PI controller. The tracking of the temperature setpoint is excellent with no oscillation or deviation. The elimination of the temperature overshoot means that the overall heater energy usage is reduced by 10% compared to the PI controller with anti-windup and 23% compared to the PI controller without anti-windup. The results for the humidity control simulation, shown in Figure 5.6, show improved performance when using the RIDE controller when compared to PI. The RIDE controller is able to accurately track the humidity setpoint to within less than 0.5%. This is due to the RIDE controller being multivariable in structure, which means that coupling from the heating system is directly accounted for in the controller design. This results in the controller being able to compensate for the interactions and track the setpoint. However, this does mean that the controller is more active in its use of ventilation to achieve this, as is displayed in Table 5.10, where the RIDE controller moves 54% more air compared to the PI controller. This could be considered acceptable as the PI controller fails to perform its function of accurately regulating the humidity in the zone, whilst using significantly more heating energy.

The response time to the setpoint is same for all controllers, as the algorithms force the heating system to its limit until the setpoint is reached.

The next set of simulation results compare the performance of the PI controller (with and without anti-windup) and the RIDE controller with an increased

sensor lag of 7 minutes. Figures 5.7 to 5.12 show that the increased sensor lag induces oscillations in all of the controllers. Referring to the temperature plots in Figures 5.7, 5.9 and 5.11, it can be observed that, on average, the oscillations are offset slightly above the setpoint, suggesting that more energy is used compared to no oscillations being present. This is confirmed by Table 5.9 which shows an increase in total heater energy for all of the controllers when subjected to the increased sensor lag. The PI controller without anti-windup has a heater energy increase of 9%, the PI controller with anti-windup has an increase of 11% and the RIDE controller has an increase of 8%. The response times for all the controllers are the same due to initial saturation of the heating actuator, as was the case with the previous set of results.

Looking at the results for ventilation (Figures 5.8, 5.10 and 5.12), it can be seen that there is oscillation in the relative humidity for all controllers. However, inspecting controller outputs for the PI controllers shows very little oscillation in the mass flow rates. This demonstrates that the oscillations in this case are due to the open-loop coupling of air temperature and relative humidity. Conversely, the RIDE controller does have oscillation in the controller output, leading to an increase of 37% in total moved air compared to the RIDE controller with a 2 minute sensor lag.

The final set of results are for air temperature and humidity control using the VTR controller with a large (7 minute) sensor lag. The VTR controller attempts to reduce oscillations when tracking the setpoint, whilst maintaining the previously attainable speed of response. Figures 5.13 and 5.14 show temperature and humidity results with K_I reduced to the critical value and VTR parameters X and Y set to 0 and 1 respectively. This is significant as it demonstrates the reduction in gain and thus responsiveness necessary in a standard controller design in order to eliminate oscillations. It is immediately apparent that the time to reach the setpoint

is increased for both temperature and humidity – increasing from approximately 1 hour to 7 hours for temperature and from 30 minutes to 4 hours for humidity. This is obviously very significant, and could be unacceptable in practice.

Figures 5.15 and 5.16 show the results for temperature and humidity control using the VTR controller with the designed parameters. It is worth comparing these figures to the results for RIDE with a 2 minute and 7 minute lag (Figures 5.5/5.6 and 5.11/5.12), as the VTR design aims to eliminate the oscillations found in 5.11/5.12, effectively becoming equivalent to 5.5/5.6. It is clear that this aim is close to being fully met; using the VTR design the oscillations are greatly reduced in the temperature control. There is a small overshoot of the setpoint (1.5°C) on one of the four days, but any oscillations are quickly damped and the setpoint is tracked within 2.5 hours. The remaining three days have no oscillation above 0.5°C amplitude. Results for humidity control are slightly less impressive. The sustained oscillations have been eliminated, however, there remains an initial overshoot then undershoot of the setpoint. After this, the setpoint tracking is good (apart from a period of un-reachability during the first day) with no oscillations present. This would suggest that the closed loop de-coupling of the humidity and temperature control systems is preserved when adding VTR design methods to the RIDE algorithm.

The second objective of the VTR design, no-increase in response time, is also well met. The VTR design process assumes that the time constant remains the same when the design is used with the RIDE algorithm. Therefore, the response time is defined as being equivalent to the time constant, which is the time to reach approximately 60% of the setpoint. Comparing to the results with the RIDE design (Figures 5.5/5.6), it can be seen that there is no significant difference in response times using the VTR controller. There is a slight increase in temperature settling time with the VTR design, especially on days 1 and 3. However, considering that

an increase of 6 hours settling time was required to remove oscillations using the RIDE design, the increase in settling time with VTR is relatively small: there is less than 1 hour increase on day 3 and, due to overshoot, 3 hours increase on day 1. The humidity control response time is almost identical between the RIDE and VTR controllers, although due to overshoot, the settling time is slightly increased.

It is apparent from the heating power plots that, when approaching the setpoint, the VTR design forces the control signal to exceed its limits. The anti-windup design does ensure that it quickly returns to the limit, however, the initial exceeding of the heating limit it is attributable to the temperature overshoot on the first day. This issue should be investigated in further work.

The nonlinear variation in the nonlinear gain matrix N that, makes it possible to simultaneously reduce oscillations and maintain response time, is highlighted in Figures 5.17 to 5.20. The change in the value of with time of element N_{11} , which relates to the temperature control, is plotted in Figures 5.17 and 5.18, and N_{22} , which relates to humidity control, is plotted in Figures 5.19 and 5.20. The plots demonstrate the decreasing magnitude of the gain as the sensor signal moves closer to the setpoint. When the setpoint is reached the gain is steady at its lowest value. It is immediately apparent that the maximum value of the gain is many times greater than that of the steady, setpoint value. In this case study, this results in the required controlled response, however, for applications that are more safety critical then it may be wise to limit the maximum gain through a saturation element. Another design consideration, highlighted by these plots, is that although the setpoint remains constant, the maximum error changes each day.

This results in slightly different responses, as can be seen in days 3 and 4, with day 3 having a smaller maximum error, resulting in a slightly slower time to the setpoint. This could be rectified by recording the error immediately before the controller is initialised, and then using this to normalise the VTR error (as

described in Chapter 3). This adds an extra degree of complication into the controller design and so is perhaps more suited to applications where speed of settling time is absolutely critical.

The VTR algorithm is particularly suited to an NDI based controller design due to the NDI (Nonlinear Dynamic Inverse) controller not requiring a high gain at steady state in order to reject disturbances. Traditional designs use a high gain combined with feedback to reduce the sensitivity of the controller to disturbances and provide better tracking of the setpoint. NDI based designs use the Inverse Dynamics knowledge to effectively replace the need for a high gain meaning that the VTR design can be used without the lower steady state gain resulting in reduced tracking accuracy. The results for the VTR simulations reinforce this reasoning as the setpoint tracking is as good as the PI controllers, which use a much higher integral gain when the setpoint is reached. Compared to the RIDE controller, the reduced integral gain at the setpoint does not appear to significantly affect disturbance rejection, with deviations only marginally greater in magnitude.

The oscillation effects of the 7 minute sensor lag are reduced by the VTR design, hence it should be expected that the energy usage should be similar to that of the RIDE controller with the 2 minute sensor lag. The results in Tables 5.9 and 5.10 demonstrate this to a degree. The overall heating energy usage is actually lower than that of the RIDE controller with the 2 minute lag - a small reduction of 2%. This could be attributable to the very small increase in settling time when using the VTR design. The total air moved by the ventilation system is higher than the RIDE design with the 2 min lag - an increase of 22%. This is caused by the first overshoot of the humidity setpoint, resulting in greater actuator usage. However, it is significant that the VTR design reduces the mass of air moved when compared with the RIDE design and a 7 minute sensor lag - a reduction of 12% is achieved. There is a similar outcome when looking at the heating energy usage, where a 10%

reduction is achieved when switching from the RIDE to the VTR design with the 7 minute lag.

An additional set of simulations were undertaken supposing a scenario where, in order to meet a minimum temperature of 21 °C , the setpoint was raised by 1.5 °C. This increase was set so that the low points of the temperature oscillations did not go below the original setpoint of 21 °C. The results, using the RIDE design, are plotted in Figures 5.21 and 5.22. The resulting energy usage increase from this small change in setpoint is dramatic. There is a 35% increase in heating energy, and due to the open-loop coupling between temperature and humidity, a 162% increase in ventilation usage. These results demonstrate that even a small change in temperature setpoint can have a very significant impact on the energy consumption of the HVAC system. Therefore, being able to accurately and robustly track a specified setpoint is of great importance.

5.7 Conclusions

This case study has presented a controller design to enable high performance control of a HVAC system. The controller design used a new type of Nonlinear Dynamic Inverse (NDI) by combining RIDE and VTR techniques to simultaneously control internal air temperature and relative humidity. The benefits of using the NDI design compared to a traditional Proportional Integral PI based controller are significantly reduced interaction between the heating and ventilation systems as well as reduced sensitivity to disturbances. It was demonstrated that, in order to implement a NDI design, very little extra sensory information and knowledge of the building structure is required compared to a PI design. The only structure information needed is the mass and specific heat capacity

of the airspace to be controlled and the extra sensory information is the external air temperature and humidity.

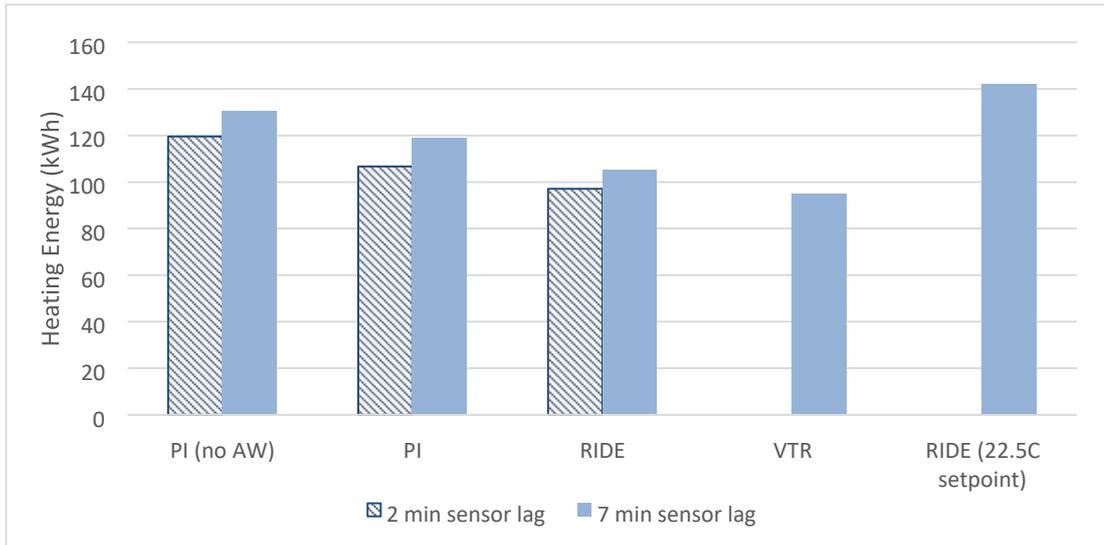


Figure 5.23 – Summary of heating energy usage.

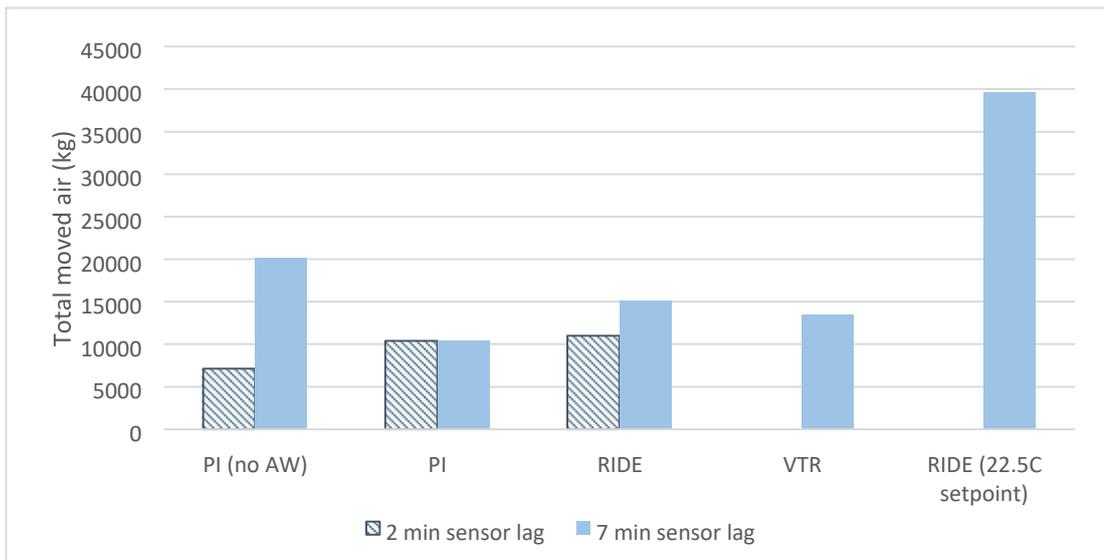


Figure 5.24 – Summary of ventilation system usage.

The robustness qualities of the NDI controller were greatly improved by introducing the nonlinear, dynamic gain changing design of Variable Transient Response (VTR). The addition of VTR significantly reduced the occurrence of oscillations with large sensor lags by reducing the steady state gain, whilst preserving the speed of response to the setpoint.

The results of the HVAC simulation demonstrated the performance and energy saving gains that could be made possible by using an advanced NDI controller design and the added robustness that was made possible by using VTR in conjunction with a NDI controller. A summary of the energy and ventilation usage of all the controller designs is presented in Figures 5.23 and 5.24. These figures highlight the reduction in heating energy when moving from the simple PI controller to the more advanced RIDE design. Also clear, was the increase in heating energy when the sensor lag was increased. Most importantly, for this thesis, was the reduction in energy usage when the VTR design was used with the large sensor lag. The results for ventilation usage were less clear as more air was moved by the advanced controller designs in more accurately tracking the setpoint. However, it is apparent that more air was required to be moved when the sensor lag was increased. Similarly to the heating energy results, the VTR design resulted in less air being moved when a large sensor lag was present.

6. Flight Control Case Study

6.1 Introduction

Missile flight control demands the highest possible performance from the control system. The limitations on the performance of the missile are governed by the

limitations of the control surface actuators; these limitations are the deflection limits and rate of deflection limits of the control surfaces. The deflection limits are imposed to prevent the air-flow separating at high angles of attack and due to space considerations. The rate of deflection limit is due to a limit on the available power of the actuation device. If a controller can operate on these limits for as long as possible then maximum performance of the system can be realised.

Rate limits are particularly significant in missile control surface actuation systems. This is because a smaller and lighter missile is a more manoeuvrable missile and the more manoeuvrable the missile the better chance it has of intercepting its target. Weight and space can be minimised through the shape and material of the airframe and the reduction in weight and size of the missile systems, of which the control surface actuation devices are a significant part. Typically the control surfaces are actuated by cold-gas hydraulic actuators, or as is more common recently, high performance electric D.C. motors. Reducing the weight of a DC motor will usually reduce the maximum voltage and available torque – thus reducing the maximum rate of deflection of the control surface, whereas the amplitude limit is unaffected. Therefore, the smaller and lighter the actuation system the more significant the rate limits become. Hence, it is apparent that if smaller and lighter actuators are to be used then a controller design that is able to operate for long periods on rate limits is required if maximum performance is to be achieved.

This case study aims to demonstrate that the Rate Actuated Inverse Dynamics controller design method is able to maintain satisfactory performance when the input is heavily rate limited and that there is a benefit to using a rate and amplitude limiting anti-windup design over a purely amplitude limiting antiwindup scheme. The RAID method is compared to the amplitude limiting

only design of RIDE which has been used as a flight control benchmark (Fielding et al., 2002).

An assessment of the RIDE and RAID designs is provided by a simulation of a body-rate autopilot with the nonlinear missile model. The main purpose of the simulations is to compare the performance of a controller with amplitude and rate limiting to the performance of a similar controller with a purely amplitude limiting anti-windup design.

6.2 Missile Mathematical Model

A mathematical model needs to be established in order to proceed with a RAID controller design as well as to provide a method of simulating controller performance. The model will encompass the aerodynamics of the missile, the actuator and sensor dynamics and the limitations of the control surfaces. There are a number of essential behaviours of missile aerodynamics that should be captured by the mathematical model; the most important are:

Coupling and interaction between the three body-rates (pitch, roll and yaw) of the missile

The effect of missile angle of incidence on the effectiveness of the control surfaces and the degree of coupling.

6.2.1 Aerodynamics

The nonlinear aerodynamic equations of motion (Counsell, 1992), (Brindley, 2012) for the missile are shown below in Equations 6.2.1 to 6.2.16. There are six aerodynamic states (x); pitch rate (q), roll rate (p), yaw rate (r), vertical velocity

(v_w), sideslip velocity (v_v) and forward velocity (v_u). These six states describe the motion of the missile in the body axis reference frame, illustrated in Figure 6.1. The inputs are elevator deflection, η , rudder deflection, ζ , and aileron deflection, ξ . A number of assumptions were made in the development of the model; the missile forward velocity is constant and the aerodynamics are invariant with Mach number (hence the model is only valid for small changes in altitude), the flexible body dynamics of the missile are not modeled and as such the missile is assumed to be completely rigid and the missile is assumed to have a constant mass, i.e. the effects of fuel being consumed are neglected.

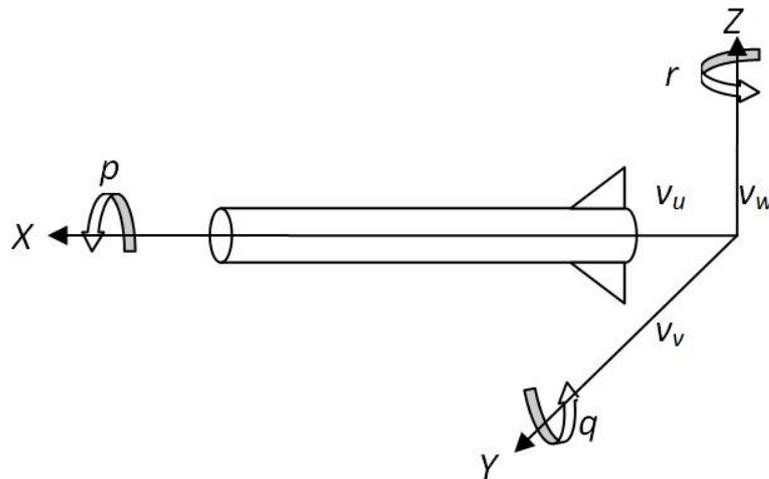


Figure 6.1 – Missile body axis.

$$\dot{p}(t) = \frac{L(t)}{I_{xx}} \quad (6.2.1)$$

$$\dot{q}(t) = \frac{Q(t) - (I_{xx} - I_{zz})r(t)p(t)}{I_{yy}} \quad (6.2.2)$$

$$\dot{r}(t) = \frac{N(t) - (I_{yy} - I_{xx})p(t)q(t)}{I_{zz}} \quad (6.2.3)$$

$$\dot{v}_u(t) = 0 \quad (6.2.4)$$

$$\dot{v}_v(t) = \frac{F_Y(t)}{m} + p(t)v_w(t) - r(t)v_u(t) \quad (6.2.5)$$

$$\dot{v}_w(t) = \frac{F_Z(t)}{m} + p(t)v_w(t) - r(t)v_u(t) \quad (6.2.6)$$

$$L(t) = P_d Swh \left\{ \begin{array}{l} 0.009\phi(t)^{0.5}(\sin(\lambda(t)) - \sin(4\lambda(t))) \\ -(0.035 - 0.000275\phi(t))(1 - 0.125\cos(4\lambda(t)))\xi(t) \\ +0.0004\phi(t)(1 + 0.18\cos(4\lambda(t)))\cos(\lambda(t))\zeta(t) \\ -0.0004\phi(t)(1 + 0.18\cos(4\lambda(t)))\sin(\lambda(t))\eta(t) \end{array} \right\} \quad (6.2.7)$$

$$Q(t) = P_d Swh \left\{ \begin{array}{l} \cos(\lambda(t))\{-0.2\phi(t) - (0.2 + 0.06\phi(t))\cos(1.5\lambda(t)) \\ -X_d\phi(t)^2(0.0005\cos(\lambda(t)) - 0.01)\} \\ +\sin(\lambda(t))\{-0.1333\phi(t)\sin(\lambda(t)) \\ +X_d(-0.02\phi(t))\sin(\lambda(t))\} \\ -\left(0.215 + 0.00425\phi(t) - 0.002875\phi(t)\cos(4\lambda(t))\right)\eta(t) \end{array} \right\} \quad (6.2.8)$$

$$N(t) = P_d Swh \left\{ \begin{array}{l} -\sin(\lambda(t))\{-0.2\phi(t) - (0.2 + 0.06\phi(t))\cos(1.5\lambda(t)) \\ -X_d\phi(t)^2(0.0005\cos(\lambda(t)) - 0.01)\} \\ +\cos(\lambda(t))\{-0.1333\phi(t)\sin(\lambda(t)) \\ +X_d(-0.02\phi(t))\sin(\lambda(t))\} \\ +\{X_d(0.032 + 0.0006\phi(t)\cos(4\lambda(t))) \\ -0.2 - 0.004\phi(t)\cos(4\lambda(t))\}\zeta(t) \end{array} \right\} \quad (6.2.9)$$

$$F_Y(t) = P_d Sw \left\{ \begin{array}{l} -\cos(\lambda(t))0.02\phi(t)\sin(\lambda(t)) \\ +\sin(\lambda(t))(0.0005\cos(\lambda(t)) - 0.01)\phi(t)^2 \\ +\left(0.032 + 0.0006\phi(t)\cos(4\lambda(t))\right)\zeta(t) \end{array} \right\} \quad (6.2.10)$$

$$F_Z(t) = P_d Sw \left\{ \begin{array}{l} \left(\sin(\lambda(t))\right)^2 0.02\phi(t) \\ +\cos(\lambda(t))(0.0005\cos(\lambda(t)) - 0.01)\phi(t)^2 \\ +\left(0.00033\phi(t)\cos(4\lambda(t)) - 0.001\phi(t) - 0.03\right)\eta(t) \end{array} \right\} \quad (6.2.11)$$

$$\lambda(t) = \sin^{-1} \frac{v_v(t)}{v_v(t)^2 + v_w(t)^2} \quad (6.2.12)$$

$$\phi(t) = \frac{360}{2\pi} \sin^{-1} \left((v_v(t)^2 + v_w(t)^2) V_m^{-2} \right) \quad (6.2.13)$$

$$V_m = Ma \sqrt{\gamma R_a T_a} \quad (6.2.14)$$

$$S_w = \frac{\pi h^2}{4} \quad (6.2.15)$$

$$P_d = \frac{\gamma M a^2 P_a}{2} \quad (6.2.16)$$

6.2.2 Control Surface Actuation

Whilst the aerodynamics contains continuous nonlinearity, it is the discontinuous nonlinearity present in the actuator model which is the focus of this case study. Specifically, the deflection and rate of deflection limit of the elevator which is caused by the limitations of the electric motor actuating the control surface. For simplicity the electric motor is not modelled directly. However, the elevator is modelled as having dynamics, deflection and rate limits which are typical of a controlled high performance DC electric motor. The RIDE controller design requests a deflection from the actuator whereas the RAID controller design requests a velocity, therefore, two different models are used. For RIDE, second order dynamics are assigned to the input (Equation 6.2.19) which approximates an electric motor under position control and for RAID a first order model (Equation 6.2.18) is used that approximates an electric motor under speed control. For the purpose of the simulation studies, two rate limits are used: a higher limit (17.5 rad/s) and a lower limit (4.5 rad/s) which are representative of a larger and smaller motor respectively. The amplitude limits for both motors are the same at ± 0.3 radians.

$$\mathbf{u}(t) = \begin{bmatrix} \eta(t) \\ \zeta(t) \\ \xi(t) \end{bmatrix} \quad (6.2.17)$$

$$\text{Motor under speed control: } \dot{\mathbf{u}}(t) = \frac{1}{\tau} (\mathbf{uc}(t) - \dot{\mathbf{u}}(t)) \quad (6.2.18)$$

$$\text{Motor under position control: } \ddot{\mathbf{u}}(t) = \omega_a^2 (\mathbf{uc}(t) - \mathbf{u}(t)) - 2\omega_a v_a \dot{\mathbf{u}}(t) \quad (6.2.19)$$

6.3 Missile Flight Control Case Study

6.3.1 System Context

A typical missile autopilot is made up of two components; a navigation system and a flight control system. The navigation system is comprised of a seeker head which acquires and tracks a target and sends position information to the guidance algorithm which then calculates the required lateral accelerations if the target is to be intercepted. It is the task of the flight control system to achieve these lateral accelerations.

The flight control system also typically consists of two stages. The first stage, known as the LATAX controller, calculates the angular velocities of the missile required to achieve the commanded lateral accelerations from the navigation system.

The final stage is the body rate controller, which calculates control surface deflections in order to attain the angular velocities commanded by the LATAX controller. The body rates are composed of the pitch, roll and yaw rates which are the rates of rotation about the missile's X, Y, and Z axes respectively. The controller designs in this case study are restricted to the body rates controller, therefore, the objective is the control of the pitch, roll and yaw rates. The holistic missile autopilot system is shown in Figure 6.2, with the modelled and simulated area highlighted in red.

An assessment of the RIDE and RAID designs is provided by a simulation of a body-rate autopilot with the nonlinear missile model. The main purpose of the simulations is to compare the performance of a controller with amplitude and

rate limiting to the performance of a similar controller with purely amplitude limiting. The context of the body-rate controller within a complete autopilot design is shown in Figures 6.2 and 6.3.

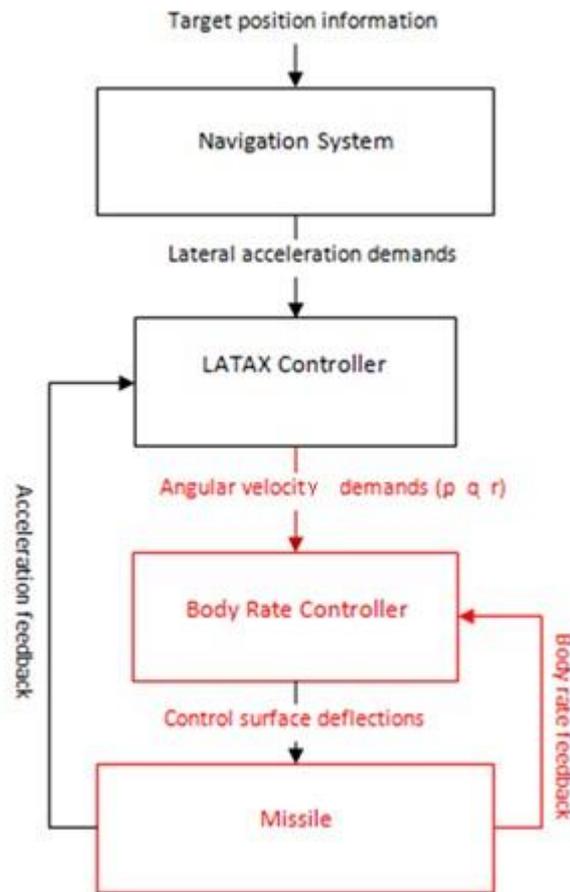


Figure 6.2 – Complete flight control system (modelled system in red).

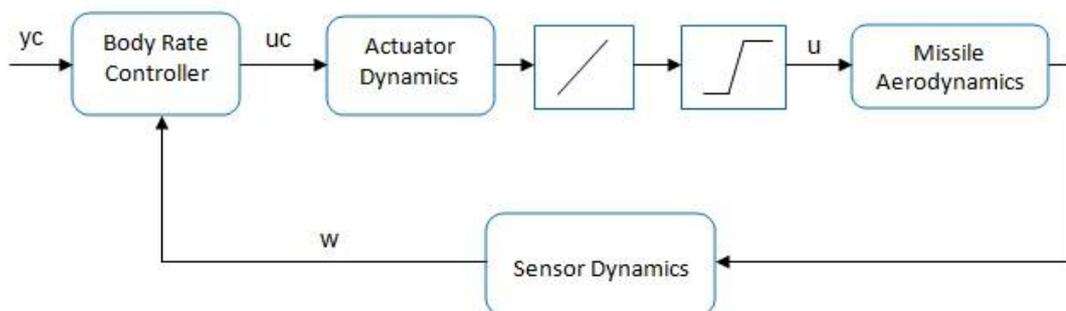


Figure 6.3 – Expanded body-rate control system.

6.4 Controller Design

The body rate controller is required to track yaw, roll and pitch rate with little or no overshoot, accurate tracking of the setpoint and minimised coupling between the body rates – especially between roll and yaw. The controllers were tested with two actuator models: firstly, one with a high rate of deflection limit which, simulates a large high performance electric motor and secondly, one with a smaller rate limit to simulate performance with a much smaller motor. The simulations were run with a Mach number of 2.0, an initial angle of incidence of 10 degrees and an initial roll angle of 57 degrees.

6.4.1 RIDE Controller

The initial step in the design of the RIDE control system is to ensure that the transmission zeros are all stable for the chosen feedback. This requires the construction of a linearised model of the missile aerodynamics.

The aerodynamics as presented in Equations 6.2.1 to 6.2.16 can be represented in a general nonlinear state-space form:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}) + \mathbf{B}_{nl}(\mathbf{x})\mathbf{u}(t) \quad (6.4.1)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \quad (6.4.2)$$

$$\mathbf{w}(t) = \mathbf{M}\mathbf{x}(t) = \mathbf{C}\mathbf{x}(t) = \mathbf{y}(t) \quad (6.4.3)$$

Where

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} v_u(t) \\ v_v(t) \\ v_w(t) \\ p(t) \\ q(t) \\ r(t) \end{bmatrix} \quad (6.4.4)$$

$$\mathbf{u}(t) = \begin{bmatrix} \eta(t) \\ \zeta(t) \\ \xi(t) \end{bmatrix} \quad (6.4.5)$$

$$\mathbf{M} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.4.6)$$

The nonlinear state-space system can be linearised about the trimmed operating point $(\mathbf{x}_{(0)}, \mathbf{u}_{(0)})$ and represented in a general linear state-space from:

$$\delta\dot{\mathbf{x}}(t) = \mathbf{A}\delta\mathbf{x}(t) + \mathbf{B}\delta\mathbf{u}(t) \quad (6.4.7)$$

The transmission zeros are then obtained by solving the determinant in Equation 6.4.8.

$$\begin{vmatrix} s\mathbf{I} - \mathbf{A} & -\mathbf{B} \\ \mathbf{C} & 0 \end{vmatrix} = 0 \quad (6.4.8)$$

$$\mathbf{T}_z = \begin{bmatrix} 0 \\ -0.6252 \\ -1.0328 \end{bmatrix} \quad (6.4.9)$$

The transmission zeros are all stable so it is then possible to proceed to the next stage of the controller design – the design of the controller closed loop response.

The closed loop dynamics were specified with a natural frequency, ω , of 40 rad/s and a damping ratio, ν , of 0.8 for each channel. These specifications are used to design the two gain matrices, K_I and K_P as follows

$$\mathbf{K}_I = \omega^2[\mathbf{CB}]^{-1} \quad (6.4.10)$$

$$\mathbf{K}_P = 2v\omega[\mathbf{CB}]^{-1} \quad (6.4.11)$$

The RIDE design uses an error conditioning law to prevent integrator windup and keep the control signal below the specified amplitude limits (Muir and Bradshaw, 1996). This is implemented to maintain controller performance when the amplitude limits of the actuator are reached. The VSC conditioning law is summarised by the following

$$\mathbf{e}(t) = \begin{cases} \mathbf{y}_c - \mathbf{w}(t) & \text{if } \mathbf{LL} < \mathbf{u}_c(t) < \mathbf{UL} \\ 0 & \text{if } \mathbf{LL} \geq \mathbf{u}_c(t) \geq \mathbf{UL} \end{cases} \quad (6.4.12)$$

Where UL = Upper actuator amplitude limit and LL = Lower actuator amplitude limit.

6.4.2 RAID Controller

Like the RIDE controller design the RAID design also requires that the transmission zeros for the measurement vector be stable. This is more straightforward for a RAID controller design as the zeros are always stable providing that the controller tuning parameter μ is positive. Therefore, the transmission zeros will be described after this parameter is set.

The closed loop dynamics of the RAID controller are set through the controller gain matrices K_I , K_P and K_d . In order to make a fair comparison between the RAID and RIDE controllers the RAID controller gains are set to yield an equivalent natural frequency and damping to the RIDE controller. The closed loop dynamics of the RAID controller will be dominantly third order and so it is not simply a case of specifying a natural frequency and damping ratio as with the RIDE

controller. Instead, the parameters ρ , σ and μ have to be set to yield the desired pole locations and transient response.

$$\mathbf{K}_P = \rho[\mu\mathbf{CB}]^{-1} \quad (6.4.13)$$

$$\mathbf{K}_I = \sigma[\mu\mathbf{CB}]^{-1} \quad (6.4.14)$$

$$\mathbf{K}_D = \mu\mathbf{I} \quad (6.4.15)$$

$$\rho = 90 \quad (6.4.16)$$

$$\mu = 0.015 \quad (6.4.17)$$

$$\sigma = 3600 \quad (6.4.18)$$

With these parameters the closed loop pole locations (Figure 6.4) can be determined from Equation 4.3.34.

$$\text{Closed loop poles} = \begin{bmatrix} -151.3 \\ -2.69 + 39.7i \\ -2.69 - 39.7i \end{bmatrix} \quad (6.4.19)$$

Having set the controller parameters the transmission zeros corresponding to the measurement vector can then be determined.

$$\mathbf{T}_Z = -\frac{1}{\mu}\mathbf{I} \quad (6.4.20)$$

$$\mathbf{T}_Z = \begin{bmatrix} -66.67 \\ -66.67 \\ -66.67 \end{bmatrix} \quad (6.4.21)$$

The closed loop poles and transmission zeros are plotted in Figure 6.4.

The VSC control laws are implemented as described in Chapter 4 to maintain controller performance when the actuator rate and deflection limits are reached. The upper and lower rate of change limits and upper and lower amplitude limits for the VSC design are therefore set equal to the upper and lower rate and amplitude limits of the actuator respectively. More detail on how the VSC control laws are implemented in the simulation is provided in the Appendix.

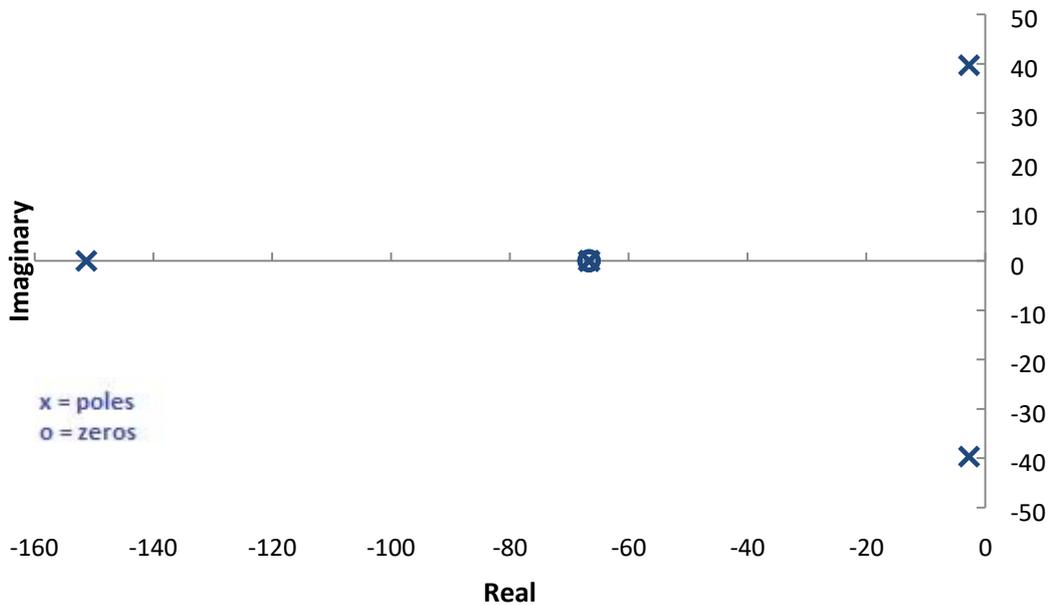


Figure 6.4 – RAID pole and zero locations.

6.4.3 Simulation Parameters and Integration Algorithm

The multiple discontinuities present in the RAID controller design merit special attention when the simulation of the missile flight control system is considered. Multiple discontinuities can occur simultaneously as both the missile's actuators and the control system itself are nonlinear. This poses a

significant simulation challenge. Specifically, a discontinuity occurring within an integration step will invalidate the Taylor series representation of the step and thus any integration algorithms used. Accurate simulation of these discontinuities is of paramount importance as the RAID conditioning logic will not work effectively if the switching is not performed in the correct order and at the right time. The simulations presented in this case study were performed using a solver which addresses these problems. The solver uses a specified integration algorithm (e.g. 4th/5th order variable step) but when a discontinuity is detected an integration discontinuity control mechanism is initiated that ensures the discontinuity does not occur within the step. The solver arranges it to occur after the end of one step and before the beginning of the next, that is, between steps. This would normally lead to a gross time error, however at the end of each step a check is made to see if a discontinuity should have occurred in the step. If this was the case the last step may be repeated with a shorter step-length based on an interpolation of the discontinuity function (the relational expression describing the discontinuity). The interpolation process is repeated until the step-end occurs just after the point of discontinuity, that is, within specified error bounds. The change to a modelling parameter may then be made, between steps, before proceeding with the simulation of the new state of the system. When multiple discontinuities occur within the same step the discontinuity treatment mechanism is used as before, however, a check is made to see if this has triggered any consequential discontinuities. The process is then repeated until all discontinuities occurring within the step have been processed in the correct sequence.

A more in-depth description of the modelling methods used in this case study can be found in the Appendix.

6.5 Simulation Results

The simulations of the body rate autopilot were performed in two stages – Firstly, the RIDE and RAID controllers were compared with an actuator with large, 17.5 rad/s, rate limits. Secondly, the RIDE and RAID controllers were compared with a smaller, less powerful actuator with reduced, 4.5 rad/s, rate limits. A step demand for yaw, roll and pitch rate was simulated and the resulting body rates and actuator responses were plotted.

The controller output was also plotted to observe instances when it exceeds the actuator output. For the RIDE design the controller output was compared to the achieved actuator deflection and for the RAID design the controller output was compared to the achieved actuator rate of change of deflection.

The criteria for sliding mode when the RAID controller output reaches actuation limits, as defined by equations 4.4.19 and 4.4.26, is also plotted. When the Equivalent Control enters these regions (highlighted in yellow) then the sliding mode is entered and the control output will remain below the actuator limits. Exiting these regions will indicate that the sliding mode has been broken, indicating that either the system cannot obtain steady state or that the control has re-entered its unlimited mode of operation.

6.5.1 Large amplitude *Pitch Rate* responses, Large actuators

 = Yaw / Rudder

= Pitch / Elevator

= Roll / Aileron

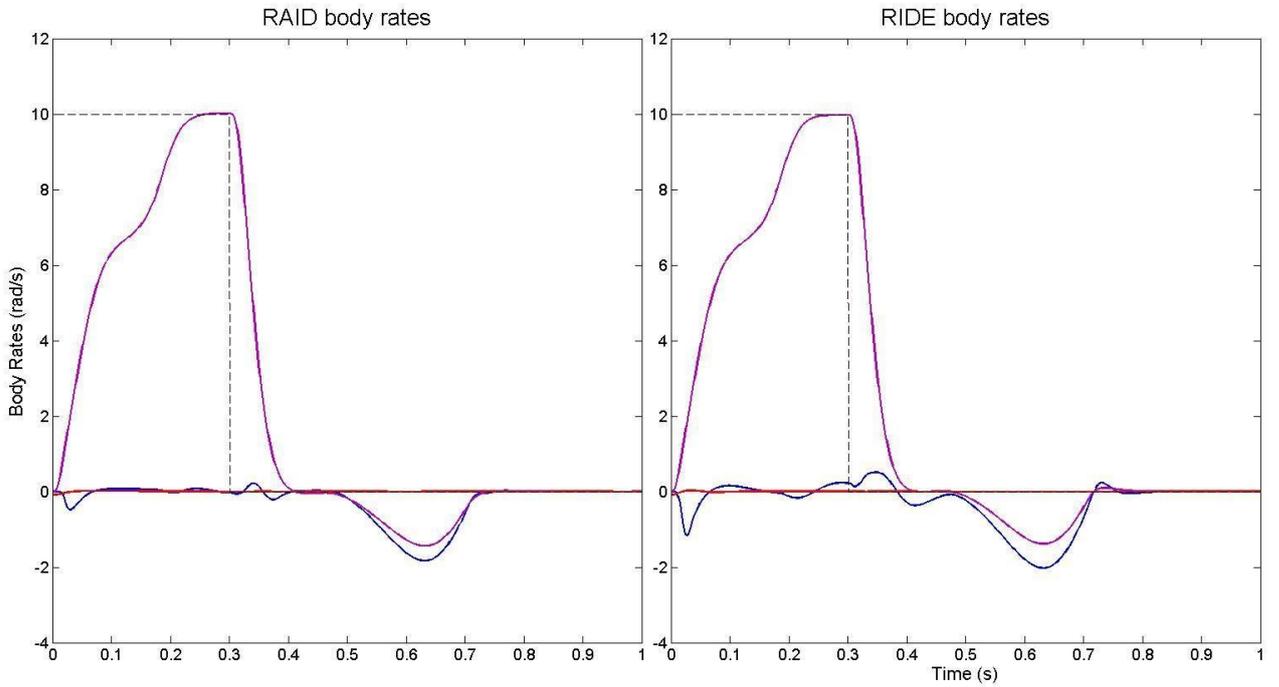


Figure 6.5 – Body Rates / Large Pitch Rate Response / Large Actuator.

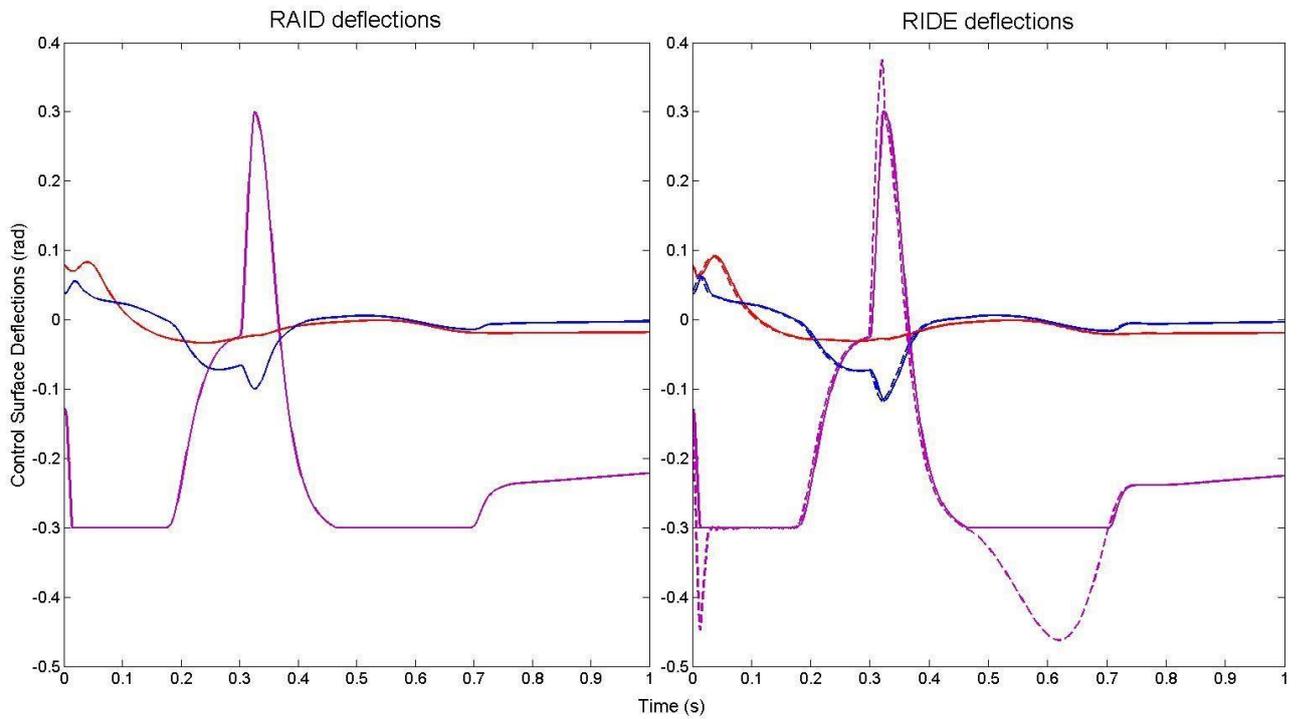


Figure 6.6 – Deflections / Large *Pitch Rate* Responses / Large Actuator.

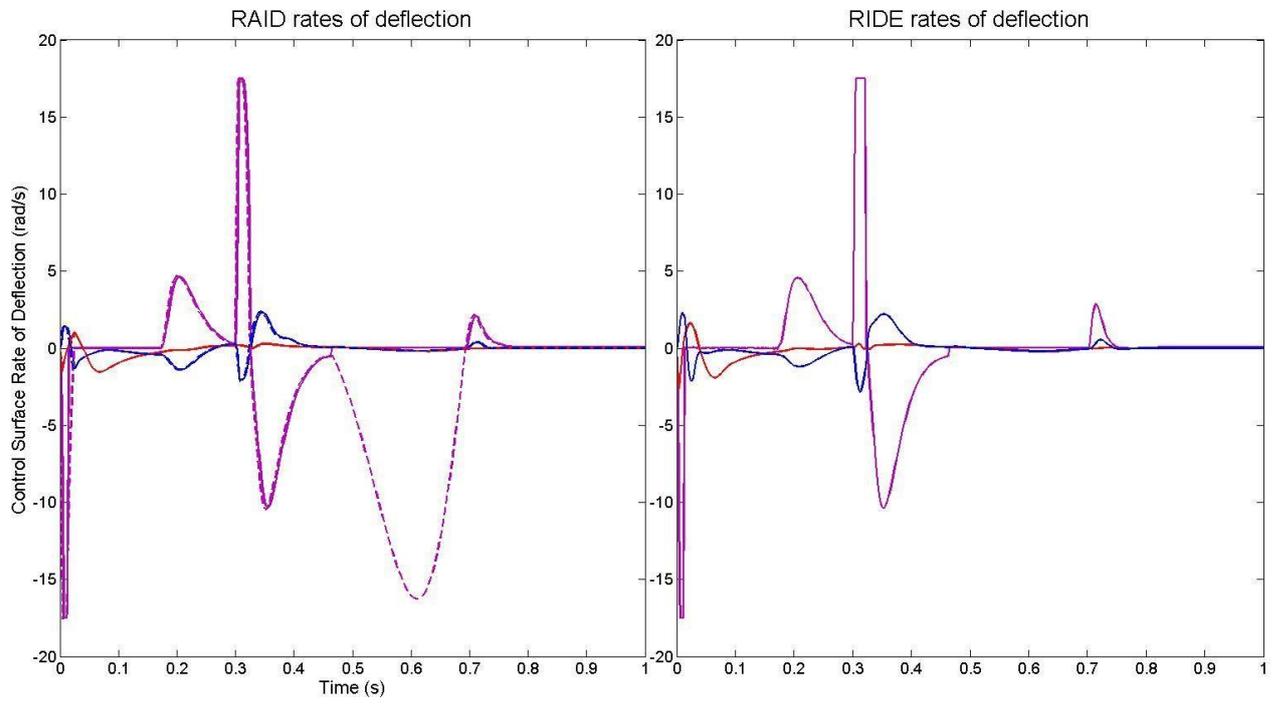


Figure 6.7 – Rate of Deflections / Large *Pitch Rate* Responses / Large Actuator.

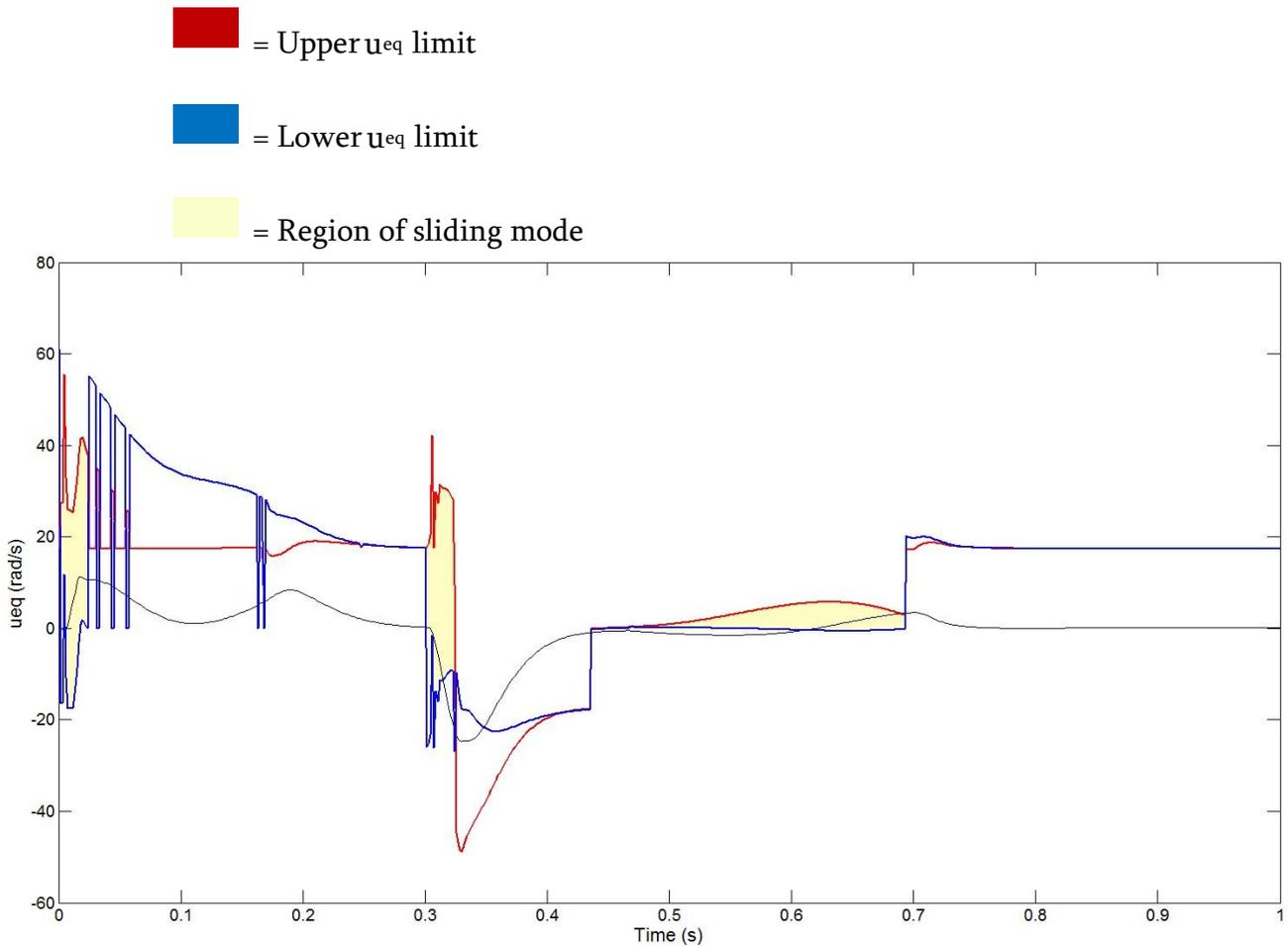


Figure 6.8 – u_{eq} sliding mode criteria / Large *Pitch Rate* Responses / Large Actuator.

The results for a pitch rate demand with a large actuator indicate similar performance between the RAID and RIDE controller designs. Both controllers briefly reach rate limits followed by deflection limits for an extended period, rate limits again and finally another extended period on deflection limits. The RAID controller enters sliding mode to limit the control output so that rate limits are not exceeded. This is observed from Figure 6.7 at 0 and 0.3 seconds and confirmed from Figure 6.8 where u_{eq} enters sliding mode zones, in yellow.

Both designs are able to keep the controller output below the actuator deflection limit for the first period of limitation. It can be observed from Figure 6.8 that the RAID controller enters brief periods of sliding mode from 0.2 – 0.6 seconds to keep the rate of change at zero and prevent the actuator from being overdriven.

Both designs are unable to keep the control output from exceeding the actuator amplitude limits during the second period of limitation. This is due to the steady state being unreachable for a small period as confirmed by Figure 6.8 where the

u_{eq} exits the lower u_{eq} limit. Figure 6.8 also illustrates that the steady state then becomes reachable again at 0.6 and the control output returns to the sliding surface as seen in Figure 6.7.

6.5.2 Large amplitude *Yaw Rate* responses – Large actuator

■ = Yaw / Rudder

■ = Pitch / Elevator

■ = Roll / Aileron

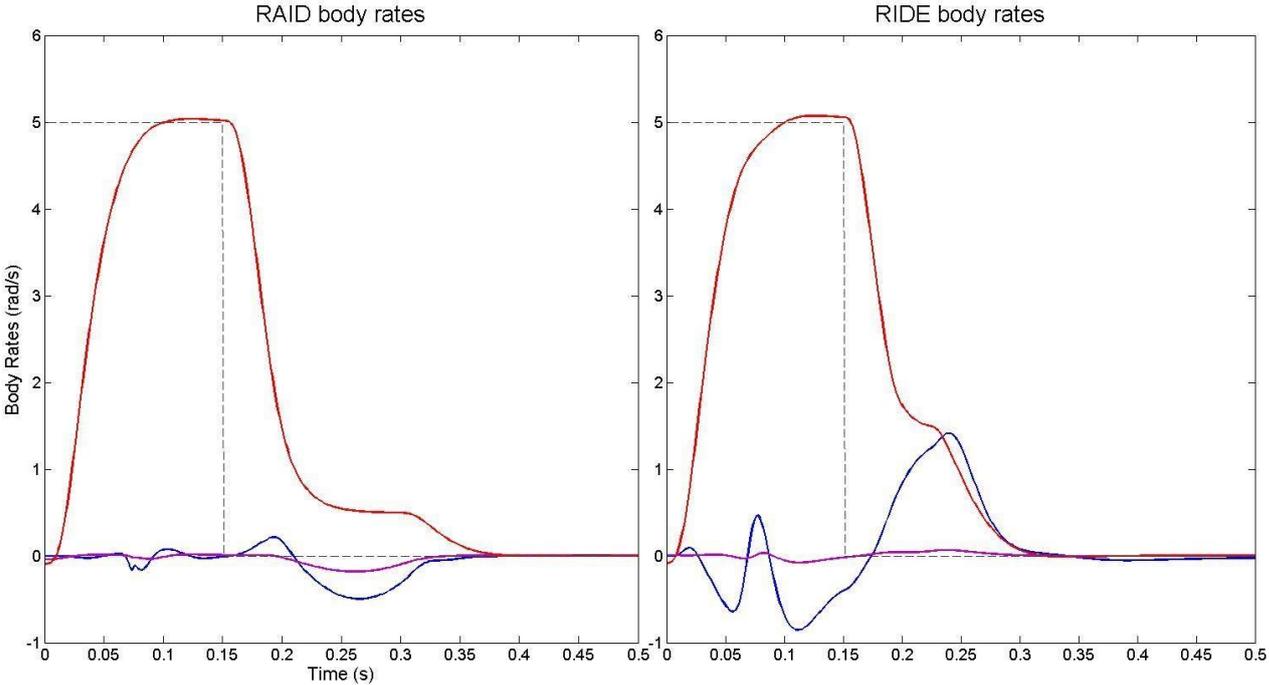


Figure 6.9 – Body Rates / Large *Yaw Rate* Response / Large Actuator.

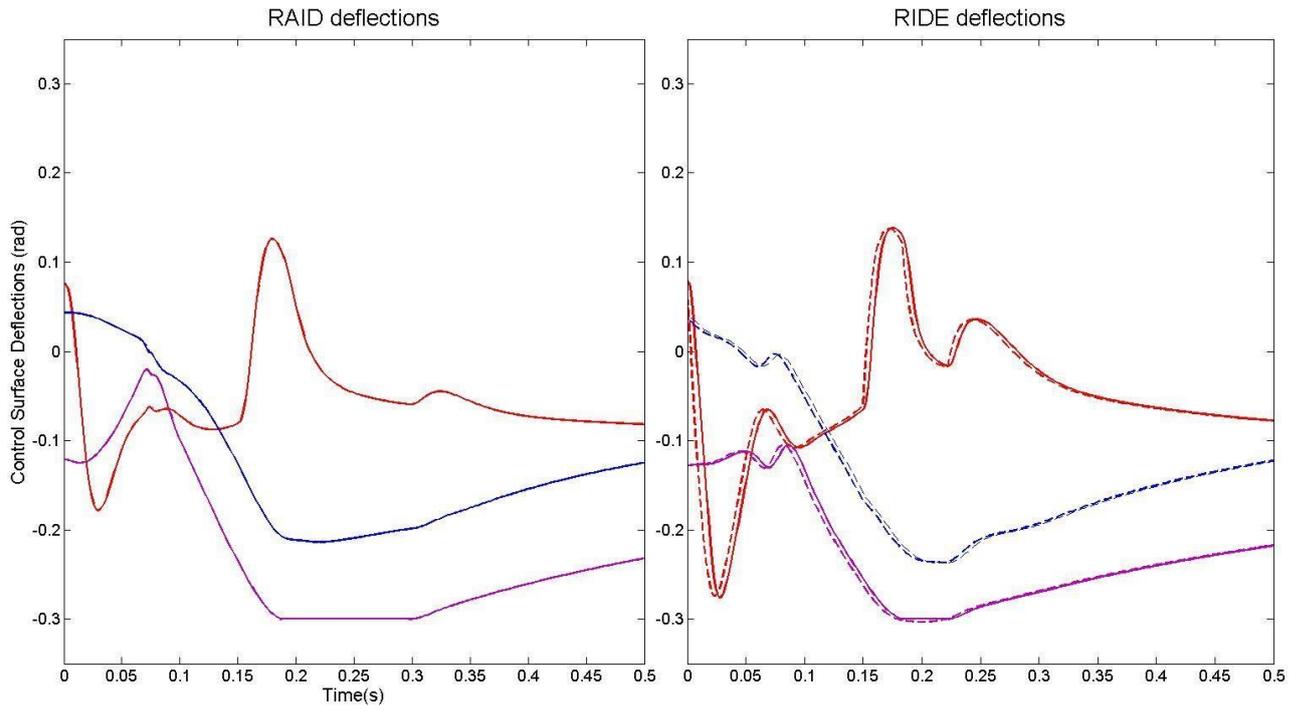


Figure 6.10 – Deflections / Large Yaw Rate Responses / Large Actuator.

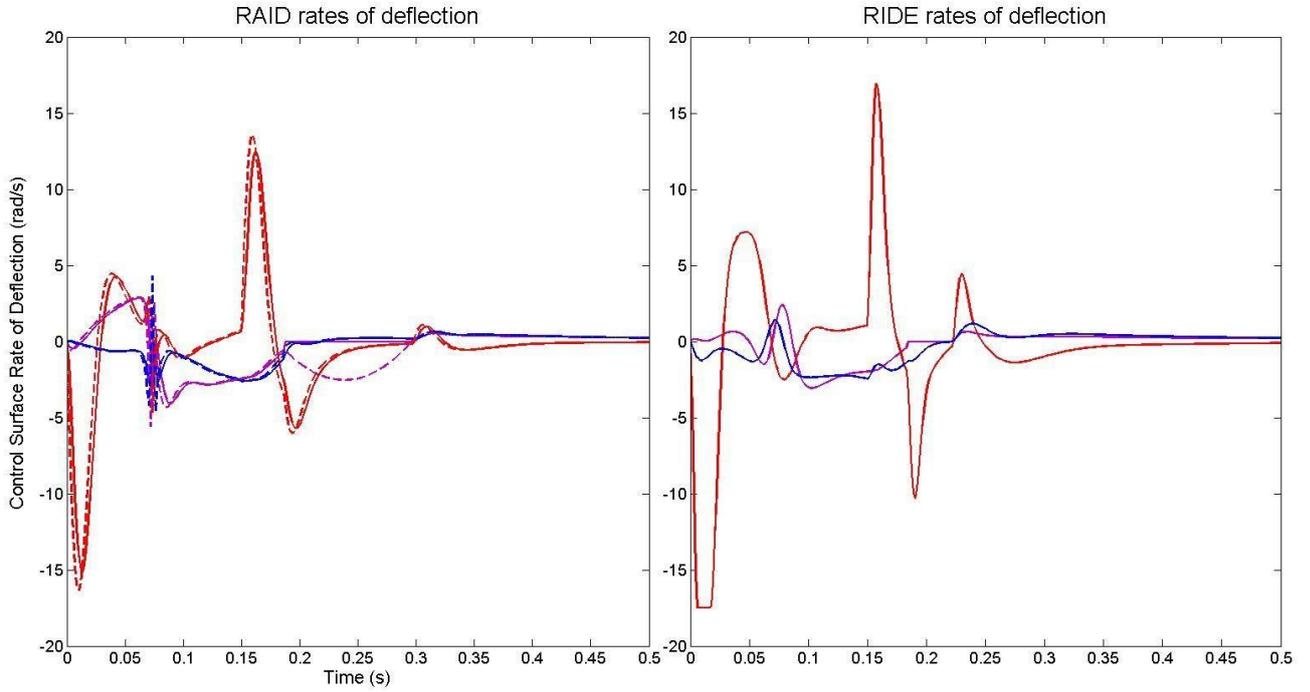


Figure 6.11 – Rate of Deflections / Large Yaw Rate Responses / Large Actuator.

- = Upper u_{eq} limit
- = Lower u_{eq} limit
- = Region of sliding mode

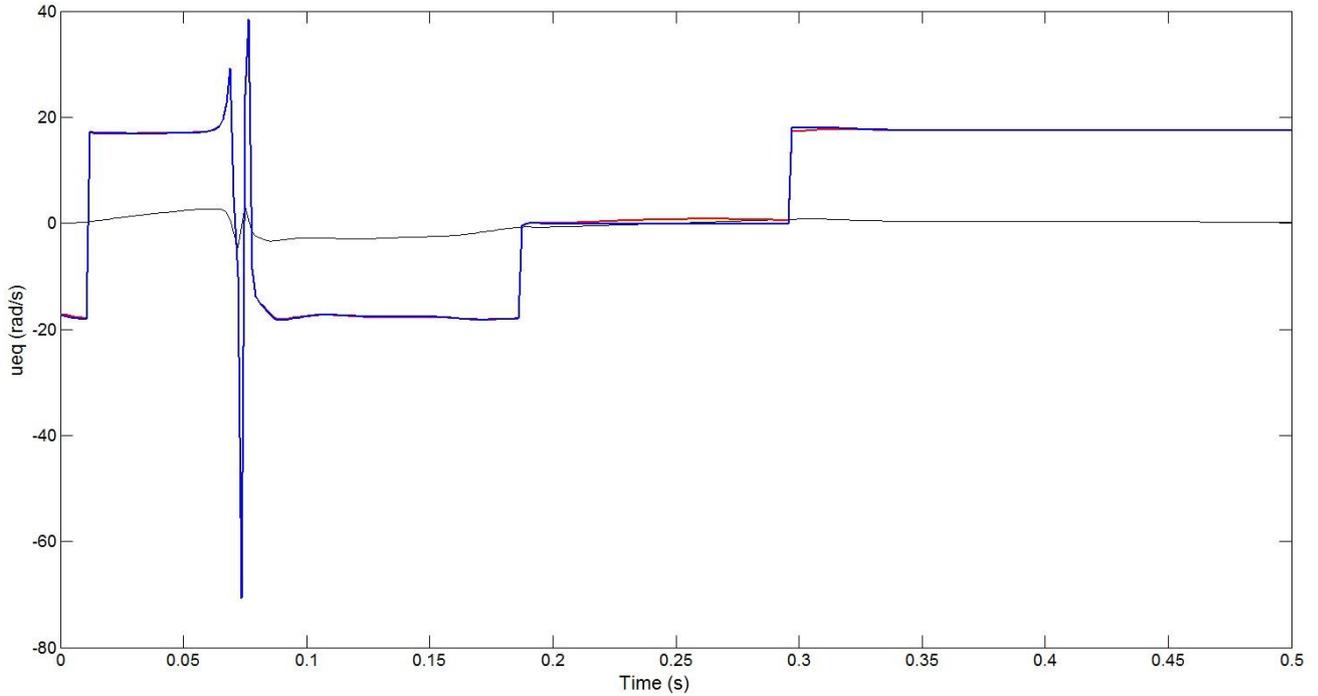


Figure 6.12 – u_{eq} sliding mode criteria / Large Yaw Rate Responses / Large Actuator.

The results for a yaw rate demand with a large actuator indicate a slight performance increase for the RAID controller compared to the RIDE design. Both controllers are unable to prevent the control output from exceeding the amplitude limit during a period of limitation. This is unavoidable as the steady state is not achievable as indicated by Figure 6.12 where u_{eq} exceeds the lower limit before re-entering the limit when the steady state becomes achievable again.

6.5.3 Large amplitude *Roll Rate* responses – Large Actuators

- = Yaw / Rudder
- = Pitch / Elevator
- = Roll / Aileron

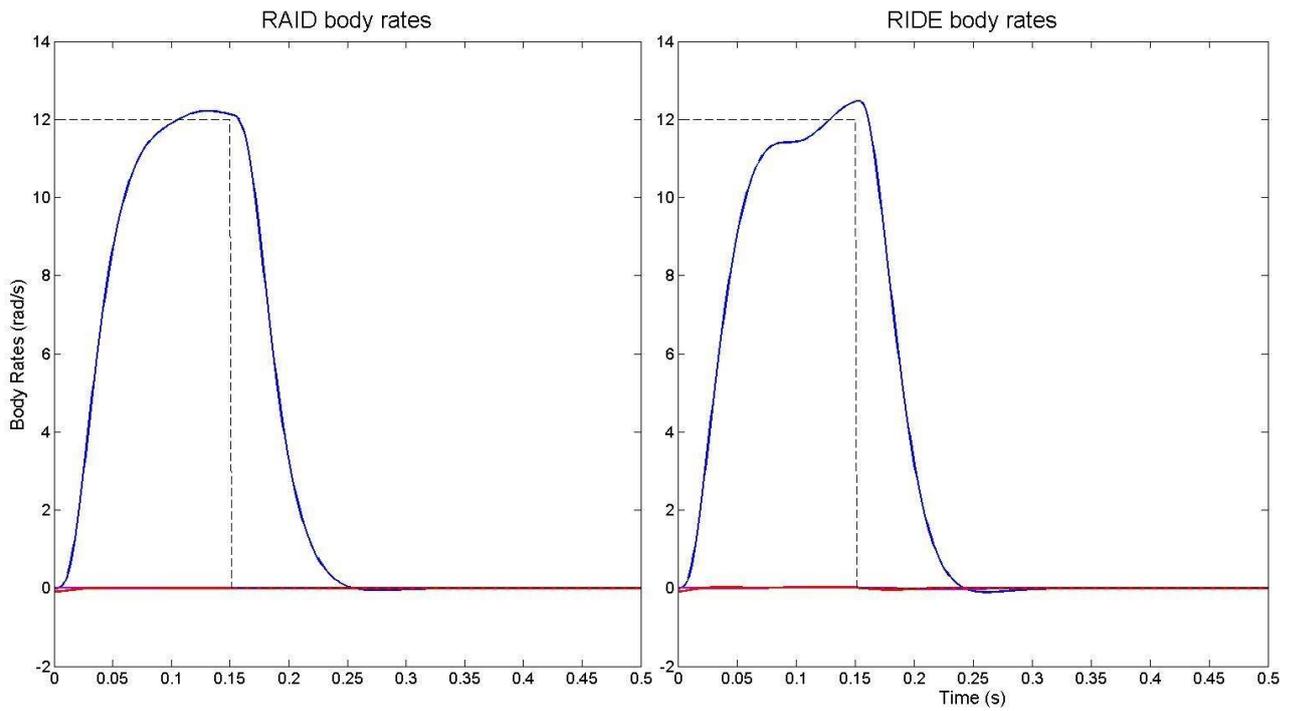


Figure 6.13 – Body Rates / Large *Roll Rate* Response / Large Actuator.

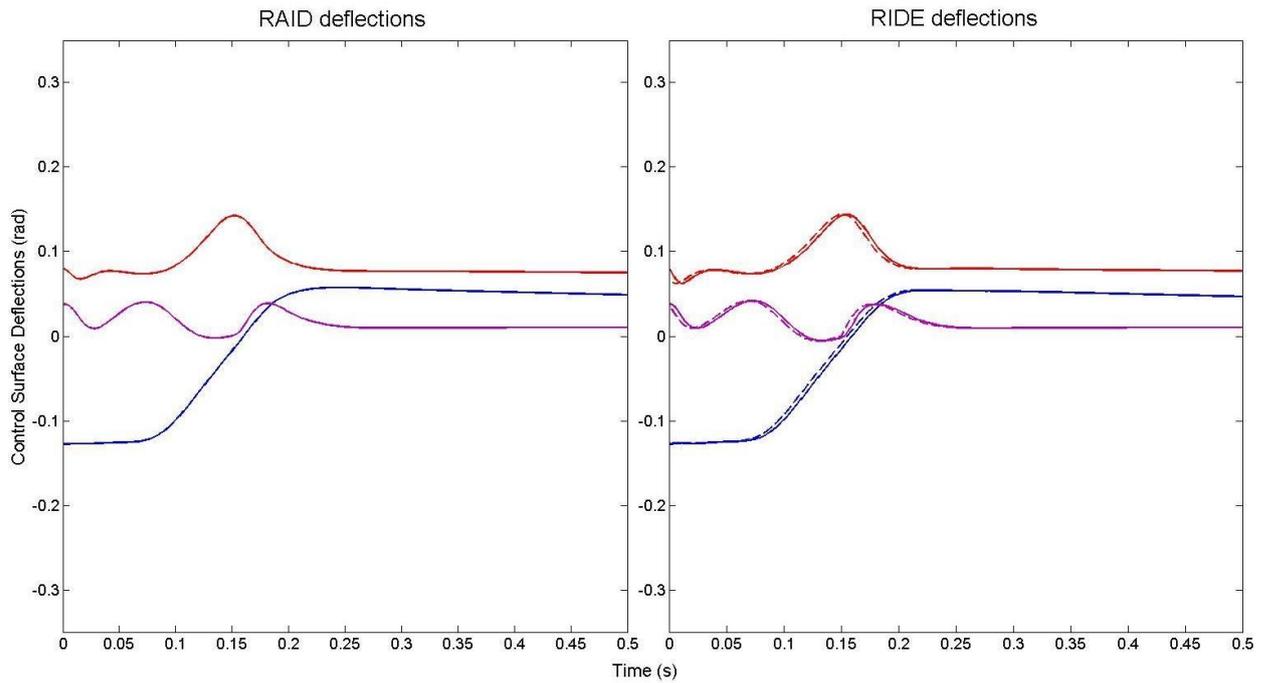


Figure 6.14 – Deflections / Large Roll Rate Responses / Large Actuator.

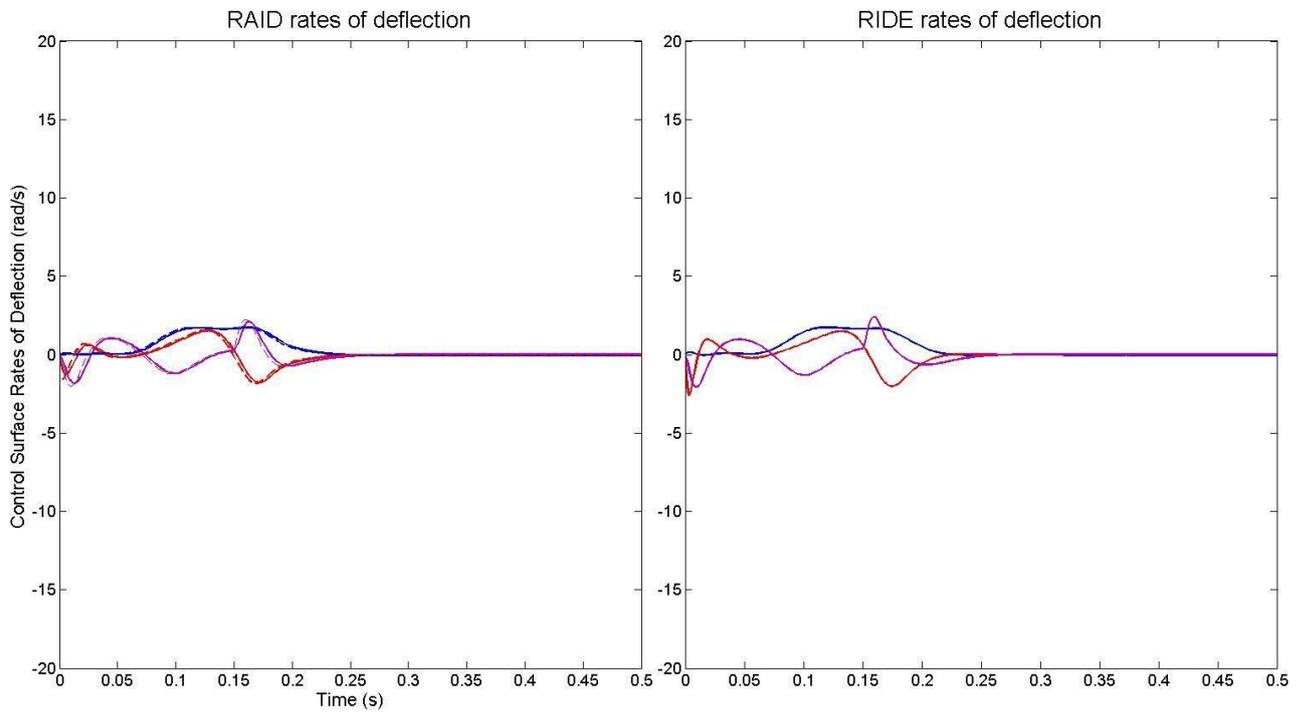


Figure 6.15 – Rate of Deflections / Large Roll Rate Responses / Large Actuator.

The simulation results for a roll rate demand with a large actuator indicate similar performance between the two controller designs. The moment of inertia of the missile is very small for roll so very little control effort is required to achieve the requested response. Hence, the actuators are never saturated. The RAID controller does track the setpoint a little better than the RIDE controller. This is perhaps due to the faster response of the speed controller for the actuator in the RAID system compared to the slower response of the position controller for the RIDE actuator.

6.5.4 Large amplitude *Yaw Rate* responses – Small actuators

- = Yaw / Rudder
- = Pitch / Elevator
- = Roll / Aileron

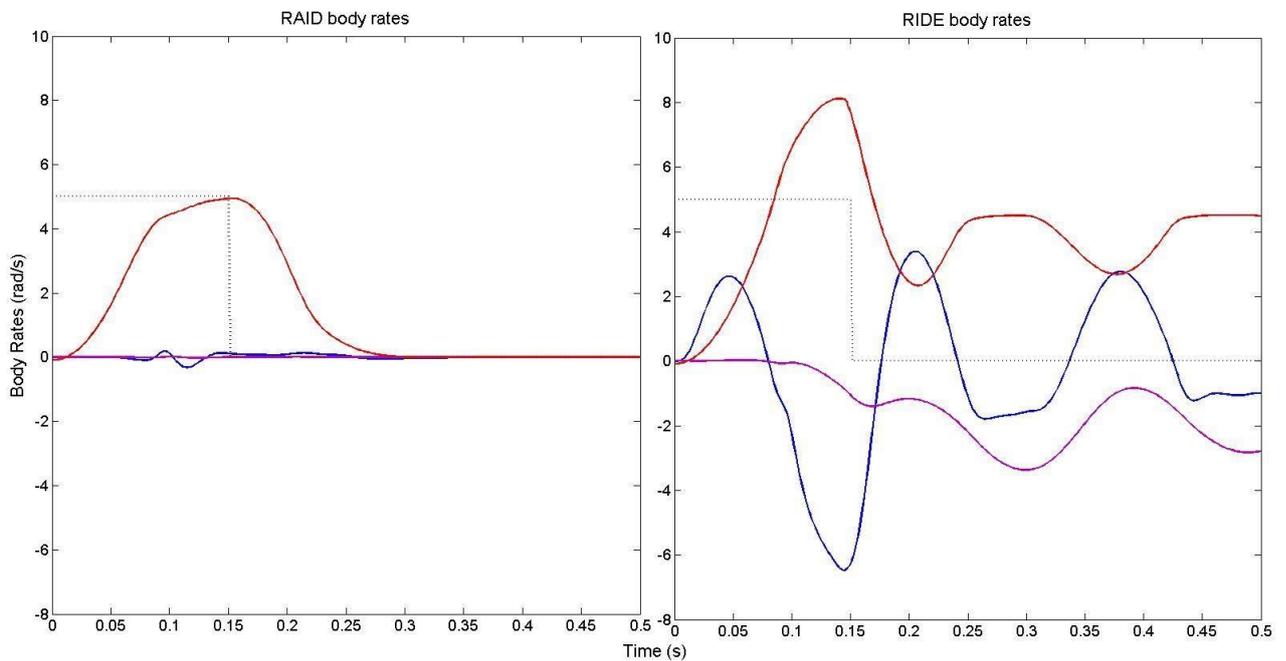


Figure 6.16 – Body Rates / Large *Yaw Rate* Response / Small Actuator.

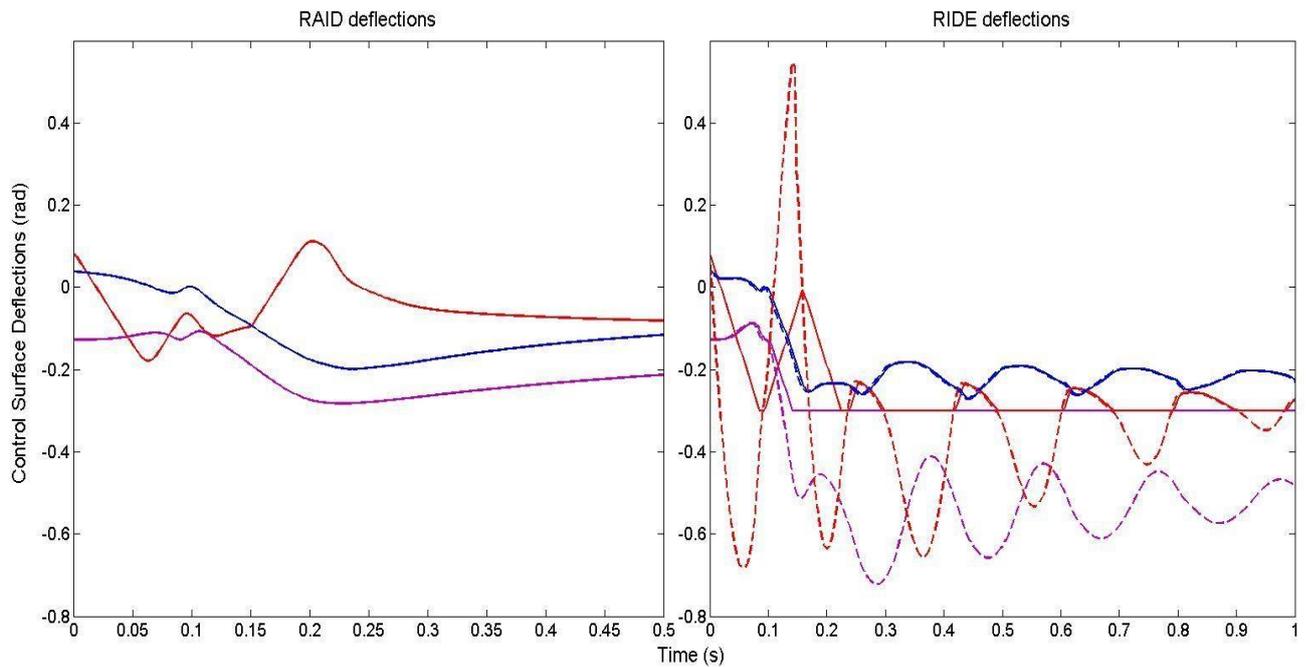


Figure 6.17 – Deflections / Large Yaw Rate Responses / Small Actuator.

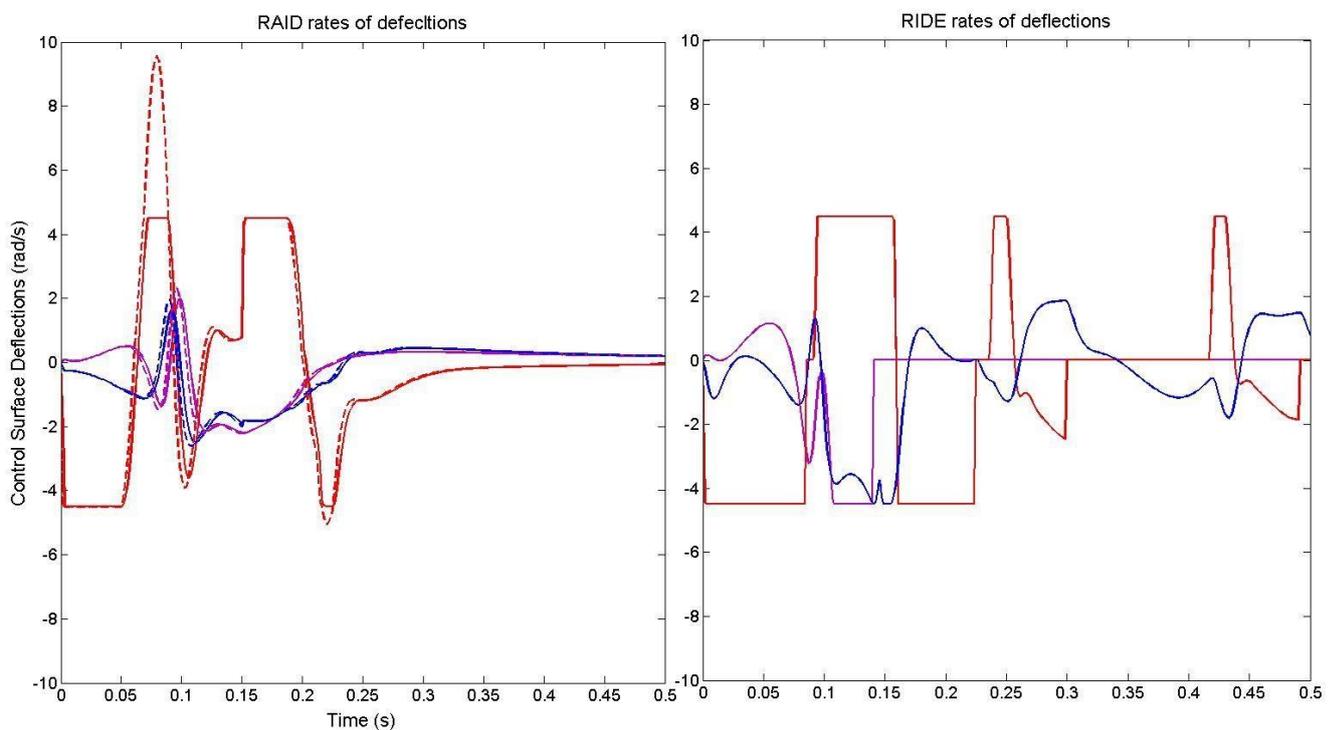


Figure 6.18 – Rate of Deflections / Large Yaw Rate Responses / Small Actuator.

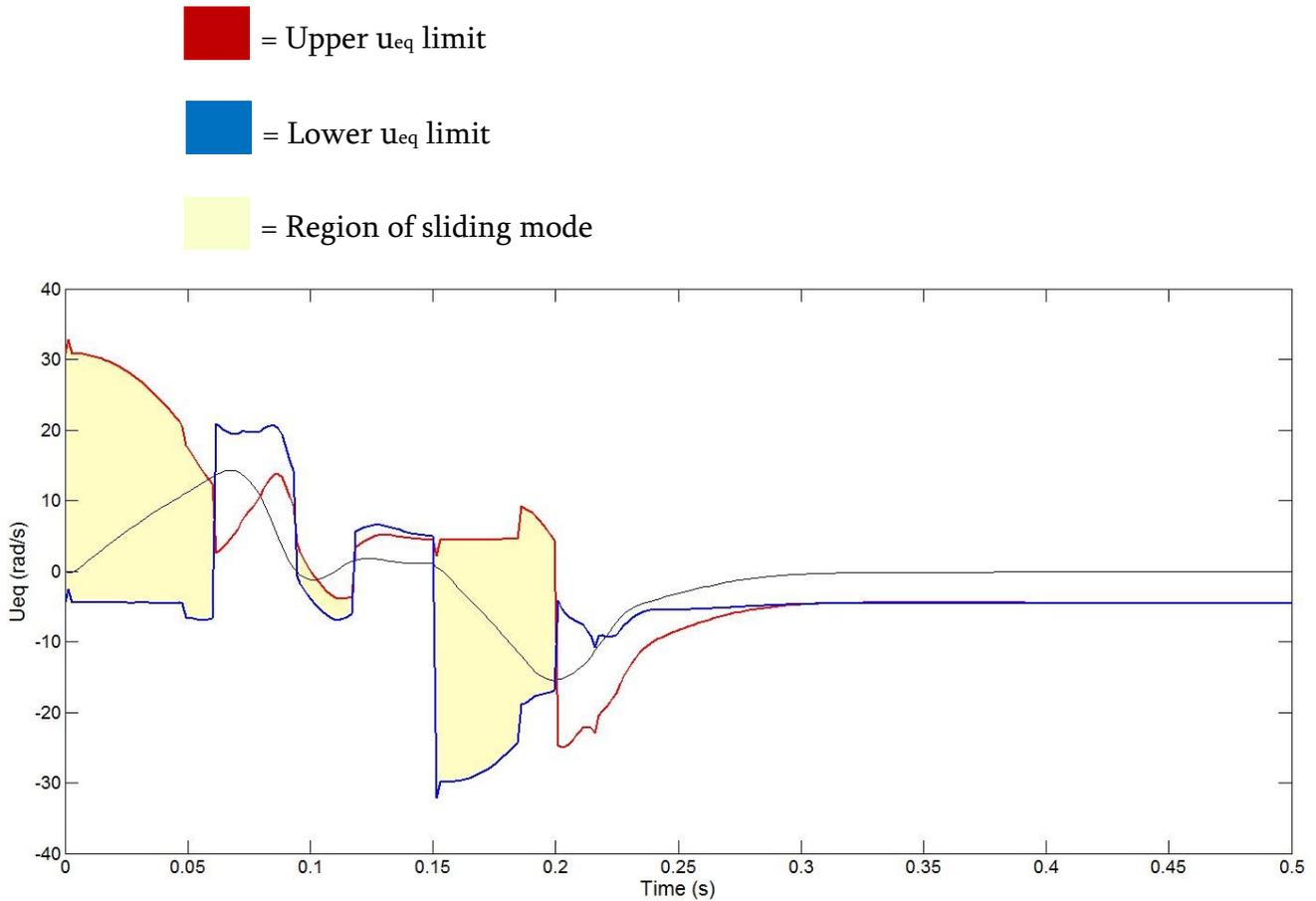


Figure 6.19 – u_{eq} sliding mode criteria / Large Yaw Rate Responses / Small Actuator.

The results for the yaw rate demand for the smaller actuator show a marked difference in response between the RAID and RIDE controllers. This is clearly due to the reduced rate limits of the actuator. Figure 6.17 illustrates that there is a very large phase lag between the controller and actuator output due to the fact that there is no anti-windup compensation in the RIDE design for rate of change limits. This phase lag results in a limit cycle occurring and potential failure of the system. The response of the RAID controller is almost unaffected by the reduced rate limits. The only significant difference between the response with the small and large actuators is the increased time to reach the setpoint.

6.5.5 Large amplitude *Pitch Rate* responses – Small actuators

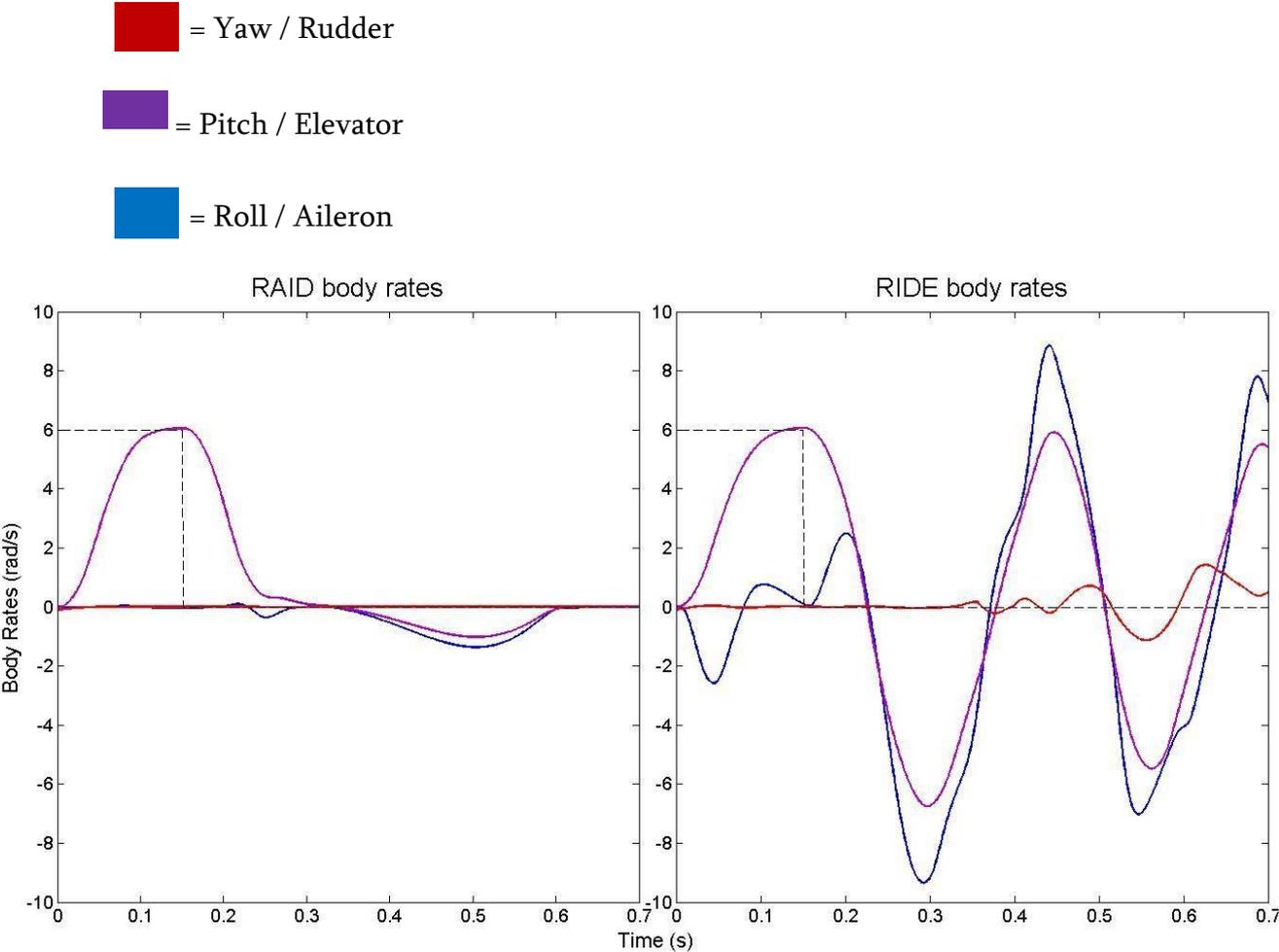


Figure 6.20 – Body Rates / Large *Pitch Rate* Response / Small Actuator.

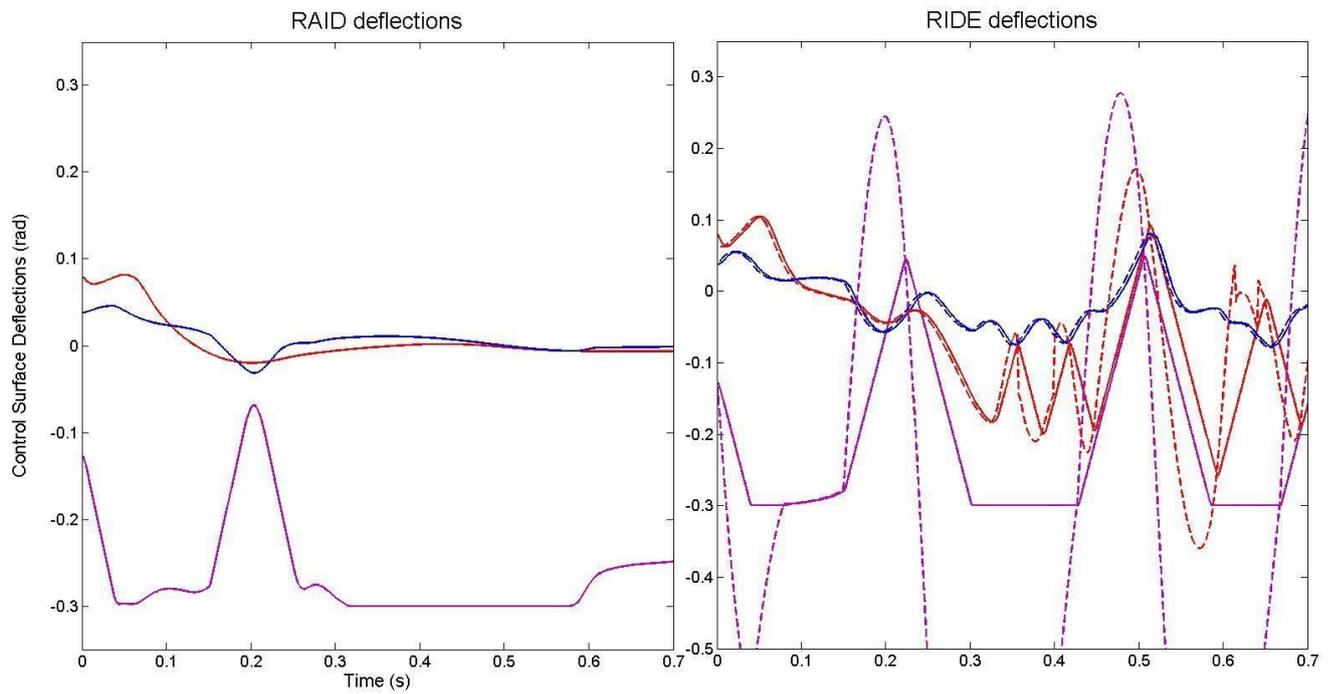


Figure 6.21 – Deflections / Large *Pitch Rate* Responses / Small Actuator.

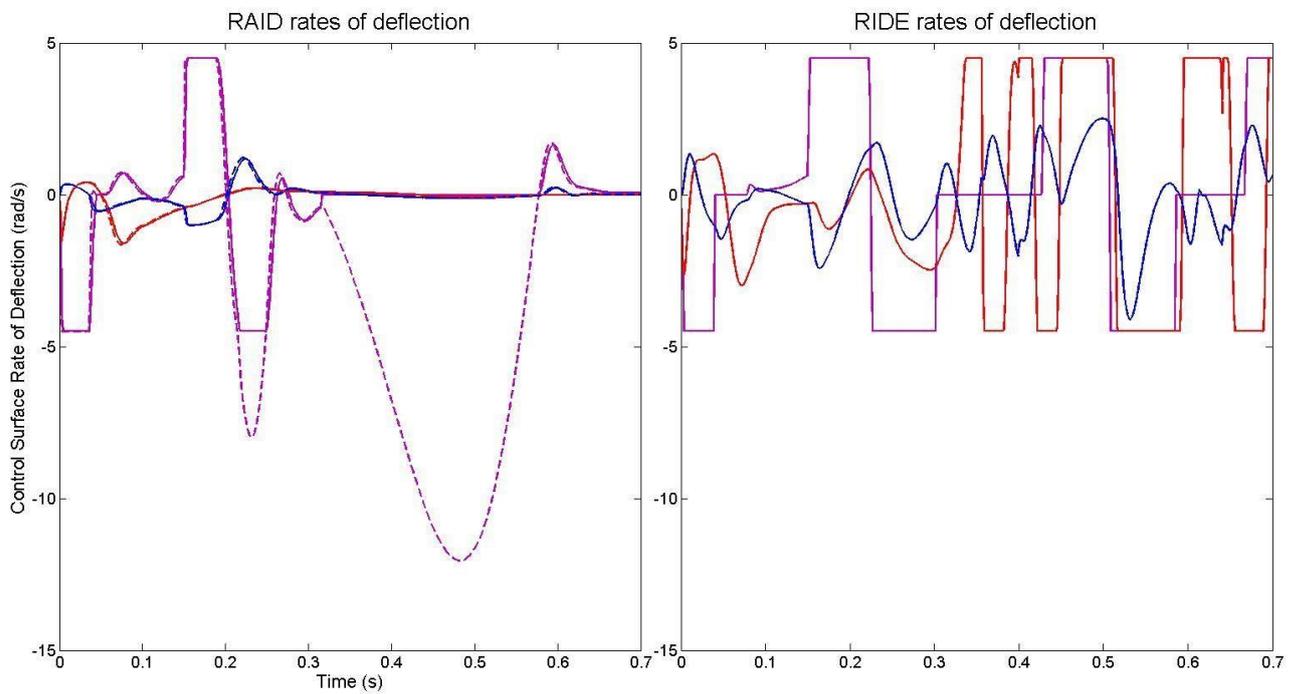


Figure 6.22 – Rate of Deflections / Large *Pitch Rate* Responses / Small Actuator.

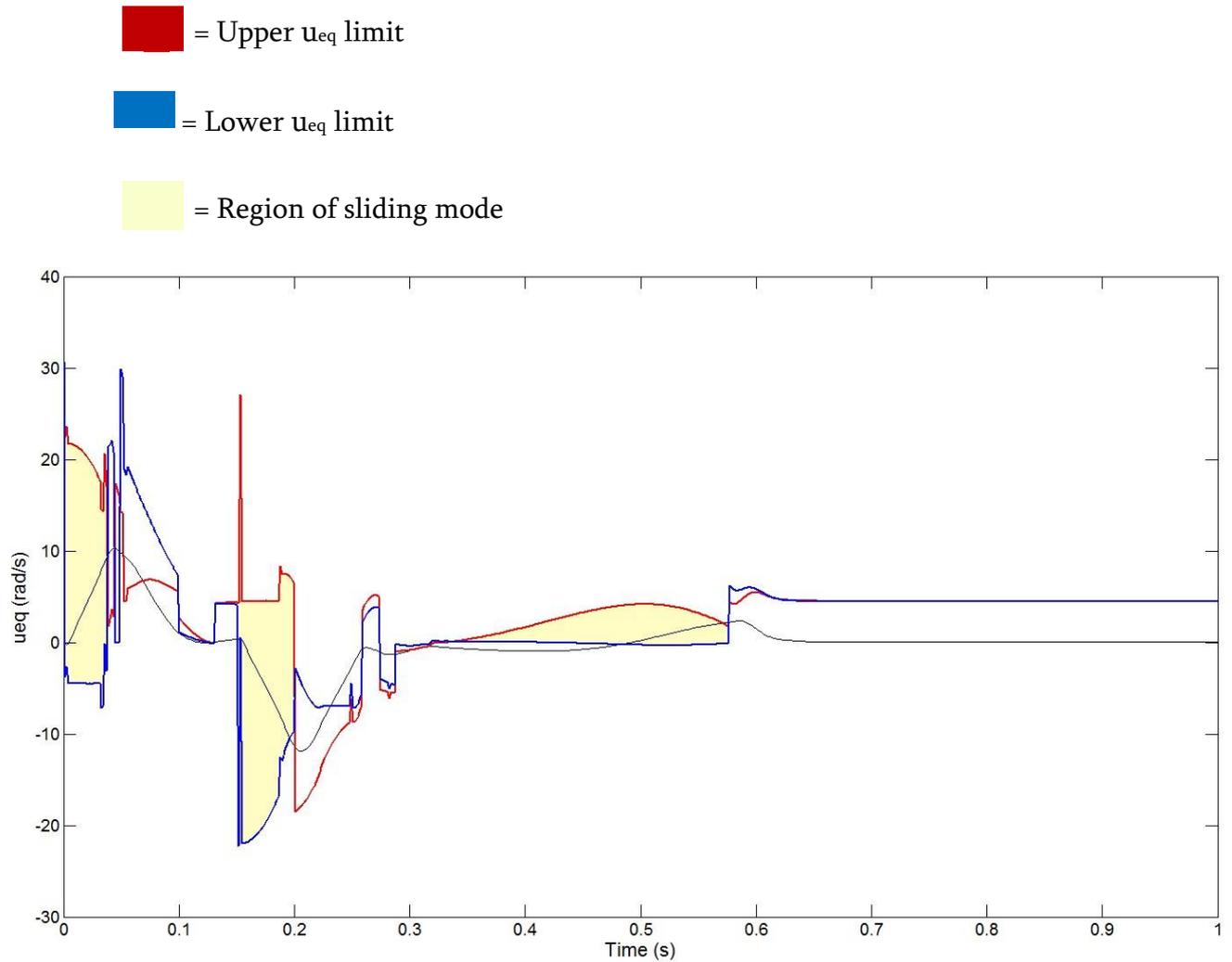


Figure 6.23 – u_{eq} sliding mode criteria / Large Pitch Rate Responses / Small Actuator.

In a similar manner to the yaw rate responses, the pitch rate response for the small actuator shows a very significant difference in performance between the two tested controller designs. The RIDE design becomes unstable, limit cycling between both rate and amplitude limits. The resulting response cycles between limits and does not reach steady state. Conversely, the RAID design performs far better. The controller spends large periods of time in sliding mode operation to keep the controller output below limits as can be seen from Figure 6.21 to 6.23.

The steady state is not achievable for a period of time between 0.2-0.225 seconds and 0.3-0.5 seconds. After 0.5 seconds the steady state becomes achievable again and the controller output returns to the sliding mode. Clearly, the missile is operating on the very edge of its performance limits.

6.5.6 Large amplitude *Roll Rate* response – Small actuators

■ = Yaw / Rudder

■ = Pitch / Elevator

■ = Roll / Aileron

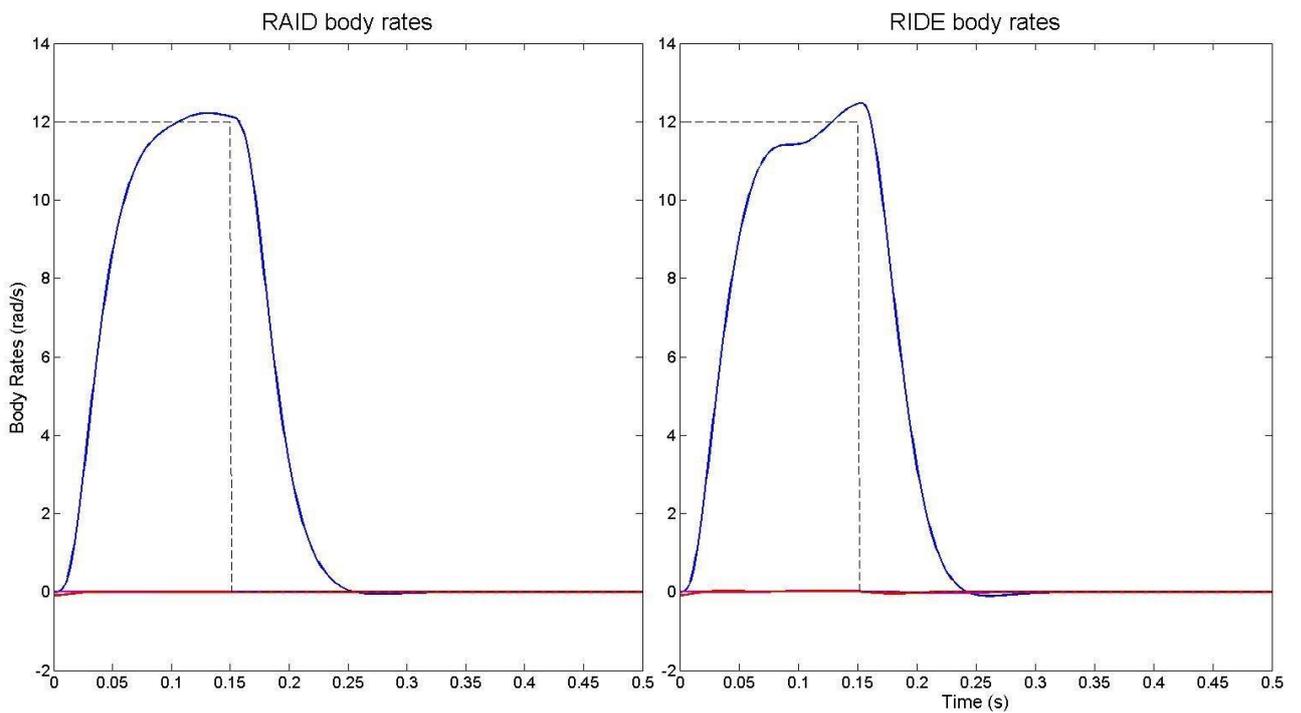


Figure 6.24 – Body Rates / Large *Roll Rate* Response / Small Actuator.

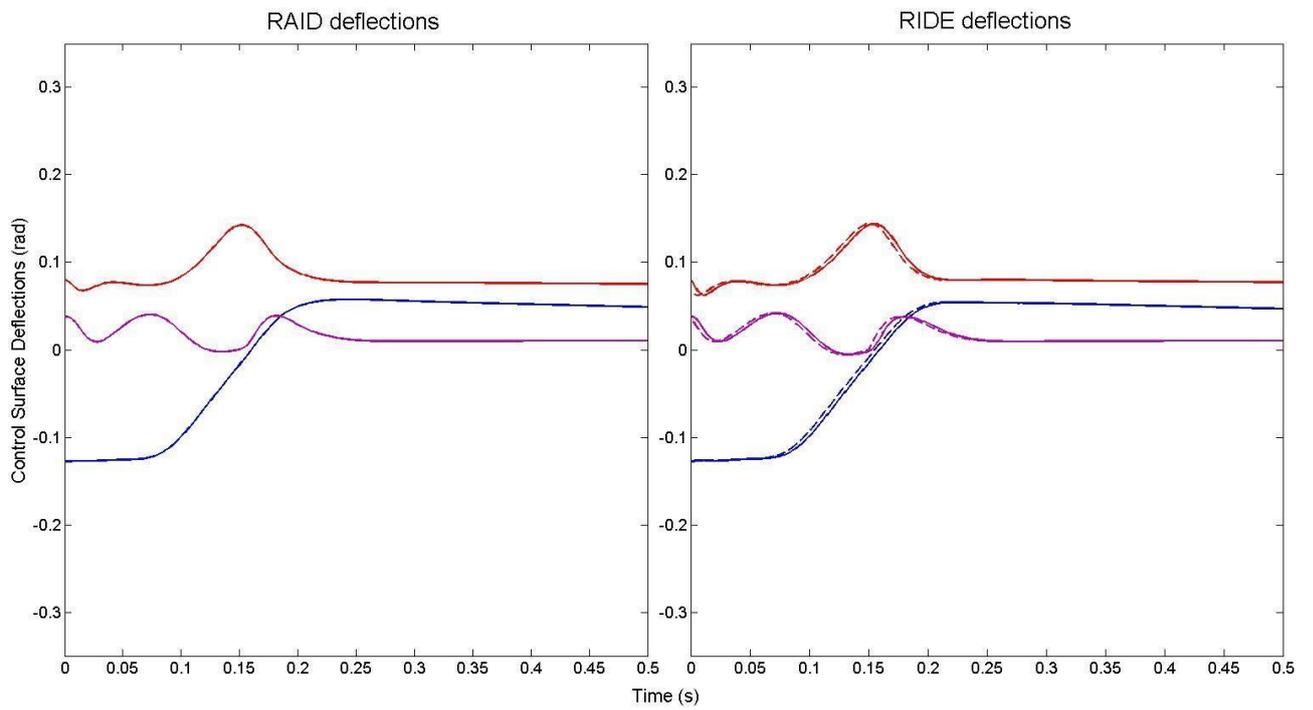


Figure 6.25 – Deflections / Large *Roll Rate* Response / Small Actuator.

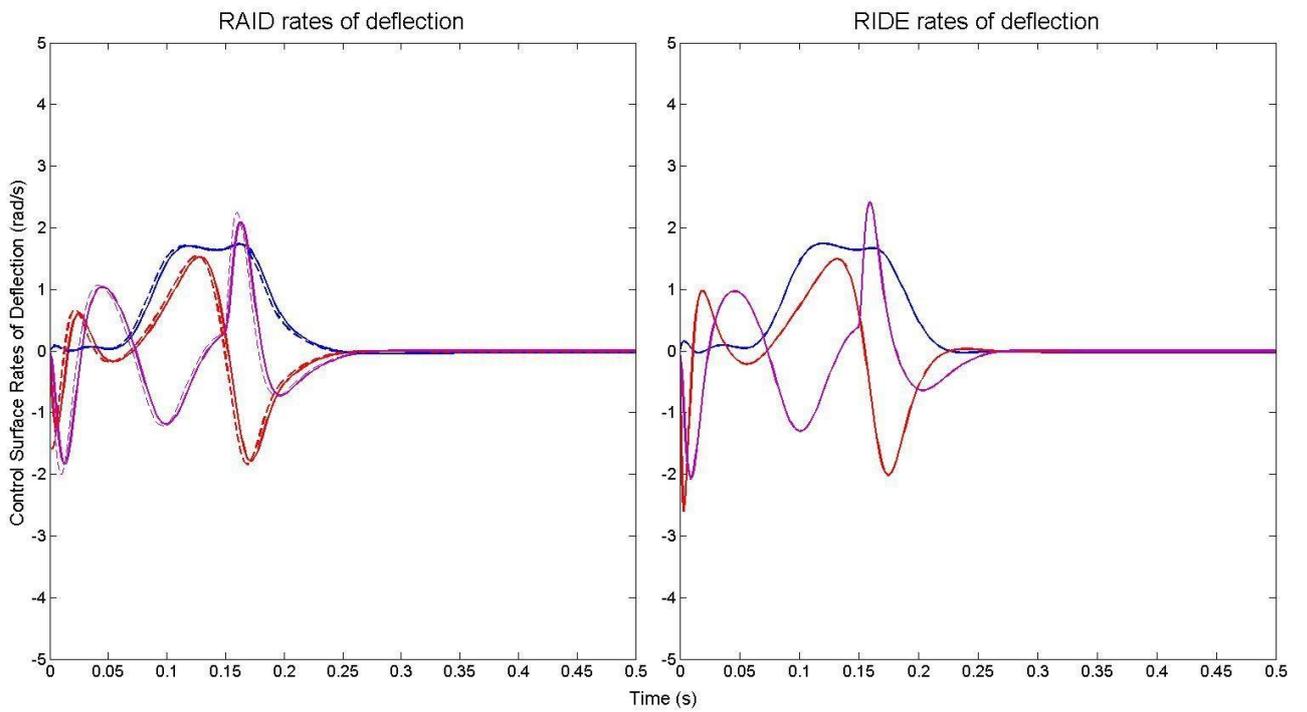


Figure 6.26 – Rates of deflection / Large *Roll Rate* Response / Small Actuator.

6.6 Results Summary

The simulation results in section 6.5.2 – 6.5.4 are those corresponding with the large, high performance actuator. Both controller designs demonstrate good performance in tracking the demanded yaw rate. However, the RAID controller is able to better de-couple the system. This is due to the fact that the RAID design uses a motor speed control system which can operate at a higher bandwidth compared to the position control system used by the RIDE design. Therefore the RAID actuator has faster dynamics and the design assumption that the actuator dynamics are fast compared to the closed loop dynamics is better realised. Overall the effect of the rate limit of the actuator on the controller responses is marginal, this is not surprising as the rate limit is relatively large and reached only for brief periods. Both designs reach the amplitude limit of the actuator, where the antiwindup design of both controllers ensures that the response remains stable.

The simulation results in section 6.5.5 – 6.5.7 were performed with an actuator model with a significantly reduced rate limit to approximate a smaller, less powerful motor. Here, there is a great disparity in performance between the RIDE and RAID designs. For both controller designs the reduced rate limits are reached for extended periods and multiple times. The absence of any conditioning to deal with this, results in the RIDE controller limit cycling between actuator limits. The use of the smaller actuator is not possible with the RIDE controller design.

The performance of the RAID controller design with the smaller actuator is excellent; the stability of the system is not compromised by the reduced rate limit. The actuator is still reaching its rate limit but is not overdriven resulting in a very stable output response. The controller undergoes periods where the steady state is not achievable but regains control when the steady state becomes achievable

again. This demonstrates that the missile is operating on the limit of its performance potential.

6.7 Conclusions

A controller design, given the name of Rate Actuated Inverse Dynamics (RAID), is able to achieve stable control when the actuator is severely limited in rate through the application of a VSC based anti-windup design. The resulting controller design is relatively simple but allows for safe control when the actuator of a missile is saturated in both rate and deflection. It should be noted that the RAID controller is best suited to systems which are first order in nature (such as the body rates of a missile). This is because the transformation used in the RAID design process increases the order of the controlled system by one. If the original controlled system is first order then this is not a problem as second order systems are relatively easy to control. However, if the original system is second order then the use of RAID gives a third order system. This may become more difficult to control and may require a cascade control structure.

The RAID controller design was simulated (along with RIDE as a benchmark) with a nonlinear missile model for body rate flight control. The results demonstrated that RAID was able to achieve excellent control with an underpowered, heavily rate limited motor. Conversely, the benchmark RIDE controller displayed unacceptable performance; limit cycling in both rate and amplitude. Furthermore, the RAID design showed a performance improvement over RIDE when the actuator was not saturated. This was due to the RAID design being able to use speed control for the actuator motor (compared with position control for RIDE), resulting in comparatively faster actuator dynamics.

In summary, by employing the RAID controller design to deal with actuator rate limits, high performance flight control with significantly smaller motors was possible.

7. Conclusions and Further Work

7.1 Thesis summary – aims and achievements

Chapter 1 introduced the concept of high performance control and the challenges that actuator and sensor limitations present to achieving high performance control.

Importantly, high performance control was generalised and defined in terms of energy utilisation, relating the different control challenges that various classes of actuator and sensor limitations provide. Two classes of actuator and sensor limitations were identified as posing some of the most serious high performance control challenges: (i) input rate and amplitude limits and (ii) actuator and sensor inertia.

Chapter 2 provided background information on various feedback controller structures and methods. The aim of this chapter was to introduce the framework on which the solutions to the control problems identified in Chapter 1 were to be built. The controller design method of Nonlinear Inverse Dynamics was selected as the basis for the controller designs presented in the proceeding chapters. A robust and practical form of Nonlinear Inverse Dynamics known as Robust Inverse Dynamics Estimation (RIDE) was described in detail and formed the framework for the controller design methods presented in this thesis.

Chapter 3 described the controller design method of Variable Transient Response (VTR). The aim of the VTR controller design was to provide a simple but effective method of achieving high performance control with significant actuator or sensor inertia. This was achieved through the use of a nonlinear gain within the RIDE control structure to reduce the controller responsiveness when the setpoint is reached. The resulting nonlinear transient response characteristics of the system

were analysed and a simple method of designing the nonlinear gain was constructed.

Chapter 4 described the controller design method of Rate Actuated Inverse Dynamics (RAID). The aim of this chapter was to develop a control method that directly addressed the controller performance problems that occur when the input is both amplitude and rate limited. The RAID controller design method overcame this problem by dynamically limiting the controller output to not exceed specified rate of change or amplitude limits; this ensured that the controller output remained on its limits for only as long as was necessary to achieve the desired transient response. This was achieved by the design and implementation of a Sliding Mode, Variable Structure, anti-windup control law, within an Inverse Dynamics framework. The Inverse Dynamics controller was designed to provide a rate of change as the control input, thus allowing the antiwindup controller to operate with rate limits. The sliding mode control law was analysed and its limitations of operation were defined, leading to the creation of a set of criteria for appropriate operation of the RAID controller.

Chapter 5 presented a building heating and ventilation control case study in which the performance of the VTR controller design was assessed. A simulation of a modern office building with heating and ventilation control was constructed and three controller designs were modelled. The ability of the VTR controller to simultaneously control indoor air temperature and humidity was compared to an industry standard PI controller and a more advanced nonlinear Inverse Dynamics controller.

Chapter 6 presented a flight control case study in which the performance of the RAID controller design method was investigated. A missile system containing aerodynamic, actuation, sensor and control elements was modelled and simulated.

Two types of actuator were modelled; (i) a large actuator with large rate limits and (ii) a smaller, less powerful actuator with significantly reduced rate limits. The performance of a benchmark RIDE controller was compared with that of the RAID controller design in order to assess the relative merits of each controller with actuator rate and amplitude limitations. The simulation results demonstrated that controller performance between the two methods was similar for the larger actuator when rate limits were reached less often. With the smaller actuator, control was not achievable using the RIDE design because the reduced rate limits caused controller instability. The RAID design was able to overcome the rate limit constraints and maintain satisfactory controller performance.

7.2 Summary of Outcomes– Variable Transient Response

Controller design methods using nonlinear functions or gains to improve the transient response of control systems have almost exclusively focused on reducing the response time without causing overshoot or oscillations. It was demonstrated in Chapter 3 that the controller design process can be simplified if the control problem is re-cast as a reduction in controller oscillations but achieved with the same response time. This reduces the parameter selection from a two degree problem to a single-degree problem; i.e. only one parameter per control channel needs to be tuned for a specified controller response.

This parameter can then be estimated by using a quasi-linear approximation of the real nonlinear system dynamics. The VTR case study demonstrated that the estimated parameter still required further manual tuning to reach an optimum value but the use of an initial estimate would significantly reduce controller design time.

Perhaps more importantly than simply providing an estimate of controller parameters the quasi-linear analysis provided insight into the performance of the controller with changes in setpoint. The analysis demonstrated that as the setpoint (and hence maximum error) was increased the system became very quickly under-damped, necessitating a change in parameters. Two solutions were proposed to remedy this problem: normalising the setpoint or simply choosing a set of parameters that found a compromise across a range of setpoints. Normalising the setpoint further reduced the complexity of the controller parameter design problem at the expense of an increase in implementation difficulty.

It was observed that certain types of system are more suitable for a VTR design than others, mainly determined by the nature of the setpoint that is fed into the controller. The VTR controller design does not offer any benefit for ramp or sinusoidal setpoints as there would be a constant phase lag between the controlled response and the setpoint, hence constant error. Generally, continuously varying setpoints reduced the effectiveness of a VTR or VTR type controller design as the error space over which the nonlinear gain can work is reduced. The ideal setpoint would therefore be a large, consistent step change.

Such setpoints are found in building heating and ventilation control and, combined with systems often significant actuator and sensor inertias, made for an ideal case study to investigate VTR performance. Remarkably, it was found the implementation of a multivariable, nonlinear Inverse Dynamics controller design (upon which VTR is based) for HVAC control required very little extra knowledge of the system compared to a conventional PI controller design. The only extra information needed was the mass of the zone air and the external air temperature and humidity and all are easily obtainable.

The simulation results demonstrated that the quality of sensor feedback had a significant impact on controller performance. With a small sensor lag the Inverse

Dynamics controller demonstrated improved performance over the simpler PI controller. With a large sensor lag the performance of the Inverse Dynamics controller was negated. The use of the VTR controller design demonstrated the ability to restore the performance advantage of the Inverse Dynamics controller with large sensor lags.

7.3 Summary of Outcomes– Rate Actuated Inverse Dynamics

The RAID controller design used a rate of change control output in order to use a rate and amplitude limiting anti-windup design. This has a drawback in that the order of the closed loop system is increased by one, complicating the tuning process. However, the flight control case study demonstrated that there were possible benefits to having a rate of change as a controller output, other than those related to the anti-windup design. This was due to the actuator control system only requiring speed control for the RAID design, rather than the slower operating position control used for conventional controller designs. Therefore, there is a greater separation between the actuator and closed loop dynamics when using a RAID design. This allows for a higher controller gain to be used and a subsequent improvement in the non-limited transient response.

The controller gain parameter ρ , which determines the magnitude of the proportional gain in the feedback loop, has an important effect on the performance of the RAID anti-windup design. Increasing the gain effectively increased the control authority allowing the anti-windup design to limit the controller output for longer periods. At an infinite gain, the anti-windup design performance was only limited by the magnitude of the actuator rate or amplitude limits. Since an infinite gain, or even a very large gain, is not practically possible a

balance must be struck between the actuator limited and non-limited operation of the controller.

The flight control case study demonstrated the extent of effect the actuator rate limits can have on the overall system stability. With a severely rate limited system the rate limits are almost constantly being reached and so are encountered much more often than amplitude limits. An amplitude limiting anti-windup design will then struggle to maintain system stability as a very large phase lag begins to occur between the controller and actuator output. In these situations a rate-limiting anti-windup design is essential to minimise the phase lag. If an effective rate limiting anti-windup design is implemented then the performance degradation caused by the reduced rate limit is then surprisingly limited.

7.4 Further Work

The immediately obvious area for further work would be to combine the VTR and RAID controller design methods. Whilst technically this would not be a particularly challenging proposition, since they share the same control architecture, the merits of such a combination are questionable. The VTR controller design relies on having the freedom to alter the controller output. If the system is heavily rate limited then the controller output is restricted for significant periods of the controller's operation, reducing the effectiveness of the VTR design. During periods of actuator saturation the VTR design will have very limited effect as the integral gain, which VTR influences, is negated.

There is potential for improving the performance of the VTR controller by investigating different target response functions which may lead to more application specific solutions. The HVAC case study for VTR could be extended to use more detailed building physics simulation tools such as ESP-r ; this extension

could provide clearer insight into the potential energy savings obtained from using the VTR design over more conventional PI based HVAC controllers.

It has been demonstrated that the VTR parameters X and Y can be estimated to provide a good initial value for the parameter tuning problem. However, further manual tuning is needed to find optimum values due to the limitations of the quasi-linear approximations used in the estimation process. Therefore, an interesting area for investigation would be to apply auto-tuning techniques to find optimum values for the X and Y parameters. This is could be particularly beneficial in the area of HVAC control as minimising controller design complexity is an important factor to industrial acceptance. Removing the need to manually tune controller parameters would certainly reduce the controller design workload. Studies have been conducted that apply auto-tuning to PI HVAC controllers (Qiang et al., 2000) and RIDE based controllers (Counsell et al., 2010). These auto-tuning techniques could be readily applied to the VTR parameter tuning problem.

Perhaps the most interesting areas for further exploration are the steady state reachability constraints of the RAID controller. Areas of operation during which the controller cannot obtain steady state could be avoided by using the u_{eq} limit criteria to modify the controller's setpoint. An additional VSC sliding mode compensator could be set up, in a similar fashion to the actuator limit compensation employed by RAID, to prevent the Equivalent Control from exceeding the specified limits by reducing the setpoint. In effect, the setpoint would become the control action. This could provide a globally stable solution to the rate and magnitude limiting anti-windup problem.

The flight control case study could be extended to include an outer, lateral accelerations (LATAX), control loop. In doing so, a better understanding of the

influence of actuator rate limits on the complete autopilot design could be achieved. This could be especially interesting if combined with the previously mentioned u_{eq} limiting VSC design. The LATAX controller will provide a setpoint to the body rate controller. The setpoint will then be modified by the u_{eq} limiting VSC compensator to ensure that an unreachable steady state does not occur. In effect the missile will be operating on the very edge of its performance capability.

7.5 Overall Conclusions

The single most significant factor in determining the limits of performance of a controller design are the inherent limitations present in the system's actuators and sensors. A controller design that is better able to perform despite these limitations is better able to realise the ultimate performance potential of the system. To this end, the VTR and RAID controller design methods were developed to improve the performance of a control system with significant actuator or sensor inertias and actuator rate and amplitude limits respectively.

The development of the VTR controller design also provided new insight into the problem of designing control systems with nonlinear, error dependant, gains - obtained through the investigation of the transient response. This in turn provided greater understanding of the tuning of such systems.

The RAID design process expanded the knowledge of anti-windup designs, improving the understanding of such designs with actuator rate limits and their use with Inverse Dynamics controllers. Insight was gained into the ultimate performance limits of anti-windup schemes and performance trade-off that exists between the actuator saturated and unsaturated controller performance.

Through the development of two, nonlinear controller design methods this work has contributed to the knowledge of how to achieve better control of systems with the most significant real-world imperfections. The solutions presented were, simple to implement and addressed the causes of ultimate controller performance limitation directly.

References

ABDELNOUR G, CHEUNG JY, CHANG C, TINETTI G, Application of describing functions in the transient response analysis of a three-term fuzzy controller, *IEEE Transactions on Systems, Man and Cybernetics*, 1993, Vol 23, 603-606

AISWAILEM SI, Application of robust control in unmanned vehicle flight control system design, 2004, *PhD Thesis, Cranfield University*

ARGUELLO-SERRANO B, VELEZ-REYES M, Nonlinear control of a heating, ventilation and air conditioning system with thermal load estimation, *IEEE Transactions on Control Systems Technology*, 1999, Vol 7, 56-63

BAI J, ZHANG X, A new adaptive PI controller and its application in HVAC systems, *Energy Conversion and Management*, 2007, Vol 48, 2043-1054

BAYO E, PAPADOPOULOS P, STUBBE J, SERNA MA, Inverse Dynamics and Kinematics of Multi-Link Elastic Robots: An Iterative Frequency Domain Approach, *The International Journal of Robotics Research*, 1989, Vol 8, 49-62

BENNET D, BURGE S.E, BRADSHAW A, Design of a controller for a highly coupled V/STOL aircraft, 1999, *Trans of the Institute of Measurement and Control*, Vol 21, 63-75

BIANNIC JM, TARBOURIECH S, Optimization and Implementation of a dynamic anti-windup compensator with multiple saturations in flight control systems, *Control Engineering Practice*, 2009, Vol 17, 703-713

BRADSHAW A, COUNSELL JM, Design of autopilots for high performance missiles, *Proceedings of the Institute of Mechanical Engineers*, 1992, Vol 206, 75-84

BRINDLEY J, COUNSELL JM, ZAHER OS, PEARCE JG, Design and simulation of a nonlinear, discontinuous, flight control system using rate actuated Inverse Dynamics, *Proc. IMechE Part G: Journal of Aerospace Engineering*, 2012, doi:0954410012438829

BRIEGER O, KERR M, LEISSLING D, POSTLETHWAITE I, SOFRONY J, TURNER MC, Flight testing of a rate saturation compensation scheme on the ATTAS aircraft, *Aerospace Science and Technology*, 2009, Vol 13, 92-104

BROWN RG, HWANG PYC, Introduction to Random Signals and Applied Kalman Filtering, *Wiley*, 1996

BURGE SE, BARLOW MI, The influence of actuator nonlinearities in advanced flight control system design, *Proceedings of the International Conference on Control*, 1991, Vol 1, 243-248

CHEN HB, ZHANG SG, Robust dynamic inversion flight control law design, *Proceedings of the 2nd International Systems and Control in Aerospace and Astronautics*, 2008, Shenzhen China, 1-6

CHEN P-C, Robust output feedback control for saturated linear systems with magnitude and rate constraints, *Proceedings of the 2010 SICE Annual Conference*, 18-21 Aug, 2010, 2405-2410

COUNSELL JM, Optimum and safe control algorithm for modern missile autopilot design, PhD Thesis Mechanical Engineering Department, *University of Lancaster*, 1992

COUNSELL JM, BRINDLEY J, MACDONALD M, Nonlinear autopilot design using the Philosophy of Variable Transient Response, *Proceedings of the AIAA Guidance Navigation and Control Conference*, Aug 2009, Chicago US

COUNSELL JM, KHALID YA, BRINDLEY J, Controllability of buildings: A multiinput multi-output stability assessment method for buildings with slow acting heating systems, *Simulation Modelling Practice and Theory*, 2011

COUNSELL JM, ZAHER OS, BRINDLEY J, Autotuning for high performance autopilot design using robust Inverse Dynamics estimation, *Proceedings of the 2010 International Multi-Conference on Computing in the Global Information Technology*, 2025 Sept 2010, Valencia, 10-17

DING L, BRADSHAW A, TAYLOR CJ, Design of discrete time RIDE control systems, 2006, *Proceedings of the UKACC International Conference on Control*, Glasgow UK

FIELDING C, FLUX PK, Nonlinearities in flight control systems, *Royal Aeronautical Society Journal*, 2003, Vol 107, 673-686

FIELDING C, VARGA A, BENNANI S, SELIER M, Advanced techniques for clearance of flight control laws, *Springer Verlag Publishing*, New York, 2002

FORNI F, GALEANI S, ZACCARIAN L, An almost anti-windup scheme for plants with magnitude and rate and curvature saturation, *American Control Conference*, 2010, Baltimore US, 6769-6774

FRANKLIN GF, POWELL JD, EMAMI-NAEINI A, Feedback control of dynamic systems, *Prentice Hall*, 2001

FREEMAN RA, KOKOTOVIC PV, Robust nonlinear controller design, *Birkhauser Boston*, 1996

GALEANI S, ONORI S, TEEL AR, ZACCARIAN L, A magnitude and rate saturation model and its use in the solution of a static anti-windup problem, *Systems and Control Letters*, 2008, Vol 57, 1-9

GATLEY SL, TURNER MC, POSTLETHWAITE I et al., A comparison of rate-limit compensation schemes for pilot-induced-oscillation avoidance, *Aerospace Science and Technology*, 2006, Vol 10, 37-47

GELB A, VANDER VELDE WE, Multiple input describing functions and nonlinear system design, *McGraw-Hill*, 1968

GOODHART SG, BURNHAM KJ, JAMES DJ, Bilinear self-tuning control of a high temperature heat treatment plant, *IEE Proceedings Control Theory Applications*, 1994, Vol 141, 12-18

GOUDA MM, UNDERWOOD CP, DANAHER S, Modelling the robustness properties of HVAC plant under feedback control, *Building Services Engineering and Technology*, 2003

HANG CC, CAO L, Improvement of Transient Response by Means of Variable Set Point Weighting, *IEEE Transactions on Industrial Electronics*, 1996, Vol 43, 477-484

HERRMANN G, HREDZAK B, TURNER MC et al., Discrete robust anti-windup to improve novel dual-stage large-span track-seek/following method, *IEEE Transactions on Control Systems Technology*, 2008, Vol 16, 1342-1351

HAMZA A, ALI A, MORSY G, Energy efficiency and indoor thermal perception: a comparison between radiant panel heaters and portable convective heaters, *Energy Efficiency*, 2012, Vol 3, 283-301

HATCHER RP, SHINNERS SM, Adaptive control techniques for reducing settling time, *American Control Conference*, June 1982, 658-659

HANG CC, CAO L, Improvement of Transient Response by Means of Variable Set Point Weighting, 1996, *IEEE Transactions on Industrial Electronics*, Vol 43, 477-484

HE Y, CHEN BM, WU C, Composite nonlinear control with state and measurement feedback for general multivariable systems with input saturation, *Systems and Control Letters*, 2005, Vol 54, 455-469

HEIM D, CLARKE JA, Numerical modeling and thermal simulation of PCM-gypsum composites with ESP-r, *Energy and Buildings*, 2008, Vol 36, 795-805

HERMANN G, MENON P, TURNER M, BATES D, POSTLETHWAITE I, Anti-windup synthesis for nonlinear dynamic inversion control schemes, *International Journal of Robust and Nonlinear Control*, 2010, Vol 20, 1465-1482

HU T, TEEL AR, ZACCARIAN L, Anti-windup synthesis for linear control systems with input saturation achieving regional nonlinear performance, *Automatica*, 2008, Vol 44, 512-519

JANABI TH, GRAY JO, Method for the control of a special class of nonlinear systems, *International Journal of Control*, 1991, Vol 54, 215-239

JIN G-Y, CAI W-J, WANG Y-W, YAO Y, A simple dynamic model of a cooling coil unit, *Energy Conversion and Management*, 2006, Vol 47, Issues 15-16, 2659-2672

KAHVECI NE, IOANNOU PA, Indirect adaptive control for systems with input rate saturation, *American Control Conference*, 2008, Seattle US, 3396-3401

KARLSSON JF, MOSHFEGH B, Energy demand and indoor climate in a low energy building – changed control strategies and boundary conditions, *Energy and Buildings*, 2006, Vol 38, 315-326

KERR ML, TURNER MC, POSTLETHWAITE I, Practical approaches to low-order anti-windup compensator design: a flight control comparison, *Proceedings of the 17th World Congress of the International Federation of Automatic Control*, July 6-11, 2008, Seoul Korea

KERR ML, TURNER MC, POSTLETHWAITE I, Stability criteria for rate limited Lur'e systems with connections to magnitude limits and anti-windup, *7th IFAC Symposium on Nonlinear Control Systems*, 2007, Johannesburg SA

KOMAREJI M, STOUSTRUP J, RASMUSSEN H et al., Simplified optimal control in HVAC systems, *Proceedings of the IEEE Conference on Control Applications and Intelligent Control*, 8-10 July 2009, 1033-1038

LANE SH, STENGEL RF, Flight control using nonlinear Inverse Dynamics, *Automatica*, 1988, Vol 24, 471-483

LAN W, THUM CK, CHEN BM, A hard-disk-drive servo system design using composite nonlinear-feedback control with optimal nonlinear gain tuning methods, *IEEE Transactions on Industrial Electronics*, 2010, Vol 57, 1735-1745

LAN W, CHEN BM, HE Y, On improvement of transient performance in tracking control for a class of nonlinear systems with input saturation, *Systems and Control Letters*, 2006, Vol 55, 132-138

LAVERTSKY E, HOVAKIMYAN N, Stable adaption in the presence of actuator constraints with flight control applications, *Journal of Guidance Control and Dynamics*, 2007, Vol 30, 337-345

LEE D, JIN KIM H, SASTRY S, Feedback linearisation vs. adaptive control for a quadrotor helicopter, *International Journal of Control, Automation and Systems*, 2009, Vol 7, 419-428

LEWIS FL, SYRMOS VL, Optimal Control 2nd Edition, 1995, *Wiley-Interscience*

LIN Z, PACHTER M, BANDA S, Towards improvement of tracking performance – nonlinear feedback for linear systems, *International Journal of Control*, 1998, Vol 70, 111

LIU C, PENG H, Inverse-dynamics based state and disturbance observers for linear time-invariant systems, *Journal of Dynamic Systems, Measurement and Control*, 2002, Vol 124, 375-381

LU YS, Integral variable-structure control with variable-structure sliding dynamics for antireset windup, *Journal of Systems and Control Engineering*, 2008, Vol 222, 209-216

MAGNI JF, BENNANI S, TERLOUW J, Robust flight control: A Design challenge, *Springer Verlag Publishing*, 1997

MARCOS A, TURNER MC, POSTLETHWAITE I, An architecture for design and analysis of high-performance robust anti-windup compensators, *IEEE Transactions on Automatic Control*, 2007, Vol 52, 1672-1679

MENON PP, BATES DG, POSTLETHWAITE I, Nonlinear robustness analysis of flight control laws for highly augmented aircraft, *Control Engineering Practice*, 2007, Vol 15, 655-662

MENNON PP, HERRMANN G, TURNER MC, et al., Nonlinear dynamic inversion based anti-windup – an aerospace application, *Proceedings of the 17th IFAC World Congress*, 2008, Seoul City, South Korea

MENON PP, TURNER MC, HERRMANN G, et al., Dynamic wind tunnel rig implementation of nonlinear dynamic inversion based anti-windup scheme, 2008 *AIAA Guidance Navigation and Control Conference*, 18-21 Aug 2008, Hawaii,

MUIR E, BRADSHAW A, Control law design for a thrust vectoring fighter aircraft using robust Inverse Dynamics estimation (RIDE), 1996, *Proc Institute of Mechanical Engineers*, Vol 210, 333-343

MURPHY GB, COUNSELL JM, Symbolic energy estimation model with optimum start algorithm, *CIBSE Technical Symposium*, 2011, Leicester UK

MURPHY GB, COUNSELL J, ALLISON J, Calibrating a combined energy systems analysis and controller design method with empirical data, *Energy*, 2013, Vol 57, 484494

NASSIF N, MOUJAES S, ZAHEERUDDIN M, Self tuning dynamic models of HVAC system components, *Energy and Buildings*, 2008, Vol 40, 1709-1720

PAPAGEORGIU C, GLOVER K, Robustness Analysis of Nonlinear Flight Controllers, *Journal of Guidance Control and Dynamics*, 2005, Vol 26

PATHAK K, FRANCH J, AGRAWAL SK, Velocity and position control of an inverted pendulum by partial feedback linearisation, *IEEE Transactions on Robotics*, 2005, Vol 21, 505-513

PENG K, CAI G, CHEN BM, DONG M, LEE TH, Comprehensive Modeling and

Control of the Yaw Dynamics of a UAV Helicopter, *Proceedings of the 25th Chinese Control Conference*, Aug 2006, Harbin China, 2087-2092

PERNG JW, WU BF, CHIN H, LEE T, Gain-phase margin analysis of dynamic fuzzy control systems, *IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics*, 2004, Vol 34, 2133-2139

PHELAN RM, Automatic control systems, *Cornell University Publishing*, 1977

PLETT GL, Adaptive inverse control of linear and nonlinear systems using dynamic neural networks, *IEEE Transactions on Neural Networks*, 2003, Vol 14, 360-376

PORTER B, BRADSHAW A, Asymptotic properties of linear multivariable continuous-time tracking systems incorporating high-gain error-actuated controllers, *International Journal of Systems Science*, 1979, Vol 10, 1433-1444

QIAN W, STENGEL RF, Robust nonlinear flight control of a high performance aircraft, *IEEE Transactions of Control Systems Technology*, 2005, Vol 13, 15-26

QIANG B, CAI W-J, WANG Q-G, HANG C-C, et al., Advanced controller autotuning and its application in HVAC systems, *Control Engineering Practice*, 2000, Vol 8, 633-644

REHAN M, AHMED A, IQBAL N, Design and implementation of full order antiwindup with actuator amplitude-rate limiter for an AC motor speed control system, *Journal of the Chinese Institute of Engineers*, 2010, Vol 33

RENTEL-GOMEZ C, VELEZ-REYES M, Decoupled control of temperature and relative humidity using a variable-air-volume HVAC system and non-interacting control, *Proceedings of the 2001 IEEE International Control Conference on Control Applications*, 2001, Mexico City

RIEDERER P, MARCHIO D, VISIER JC, HUSAUNDEE A, et al., Room thermal modeling adapted to the test of HVAC control systems, *Building and Environment*, 2002, Vol 37, 777-790

ROHRS C, VALVANI L, ATHANS M, STEIN G, Robustness of continuous-time adaptive control algorithms in the presence of un-modelled dynamics, *IEEE Trans. On Automatic Control*, 1985, Vol 30, 881-889

RUGH WJ, SHAMMA JS, Research on gain scheduling, *Automatica*, 2000, Vol 36,

1401-1425

SCOTTEDWARD H, HALL CE, Variable-structure PID control to prevent integrator windup, *IEEE Transactions on Industrial Electronics*, 2001, Vol 48, 442-451

SETIAWAN A, ALBRIGHT LD, PHELAN RM, Application of pseudo-derivative-feedback algorithm in greenhouse air temperature control, *Computers and Electronics in Agriculture*, 2000, Vol 26, 283-302

SIEBERLING S, CHU Q, MULDER J, Robust flight control using incremental nonlinear Inverse Dynamics and angular acceleration prediction, *Journal of Guidance Control and Dynamics*, 2010, Vol 33

SHIN Y, CALISE AJ, JOHNSON MD, Adaptive control of a advanced fighter aircraft in nonlinear flight regimes, *Journal of Guidance Control and Dynamics*, 2008, Vol 31, 1464-1477

SHTESSEL Y, BUFFINGTON J, BANDA S, Tailless aircraft flight control using multiple time scale reconfigurable sliding modes, *IEEE Transactions on Control Systems Technology*, 2002, Vol 10, 288-296

STATE C, IORGA I, Absolute stability for the lateral-directional BWB model with rate limited actuator, *INCAS Bulletin*, 2012, Vol 4, 111-123

STEER AJ, Application of NDI flight control to a second generation supersonic transport aircraft, *Computing and Control Engineering Journal*, 2001 **TABOURIECH S, TURNER M**, Anti-windup design: An overview of some recent advances and problems, *Control Theory and Applications*, 2009, Vol 3, 1-19

TASHTOUSH B, MOLHIM M, AL-ROUSAN M, Dynamic model of a HVAC system for control analysis, *Energy*, 2005

TSACHOURIDIS VA, POSTLETHWAITE I, Pseudo-linear controllers for systems with saturated actuators, *IEE Proceedings – Control Theory and Applications*, 2004, Vol 151, 783-789,

TURNER MC, HERRMANN G, POSTLETHWAITE I, Incorporating robustness requirements into antiwindup design, *IEEE Transactions on Automatic Control*, 2007, Vol 52, 1842-1855

UTKIN V, Chattering Problem in Sliding Mode Control Systems, *2006 International Workshop on Variable Structure Systems*, 5-7 June 2006, 346-350

UTKIN V, CHANG H-C, Sliding mode control on electro-mechanical systems, *Mathematical Problems in Engineering*, 2002, Vol 8, 451-473

VISOLI A, Modified anti-windup scheme for PID controllers, *IEE Proceedings – Control Theory and Applications*, 2003, Vol 150, 49-54

WANG Q, STENGEL RF, Robust nonlinear flight control of a high performance aircraft, *IEEE Transactions on Control Systems Technology*, 2005, Vol 13, 15-26

YILDIZ Y, KOLMANOVSKY IV, ACOSTA D, A control allocation system for automatic detection and compensation of the phase shift due to actuator rate limiting, *American Control Conference 2011*, San Francisco USA, 444-449

YOKOYAMA M, KIM G-N, TSUCHIYA M, Integral sliding mode control with antiwindup compensation and its application to a power assist system, *Journal of Vibration and Control*, 2010, Vol 16, 503-512

YOUNG KD, KOKOTOVIC P, UTKIN V, Singular perturbation analysis of high-gain feedback control systems, *IEEE*, 1977, 931-938

YUAN R, YI J, YU W, FAN G, Adaptive control for a class of uncertain nonlinear system with input magnitude and rate saturation, *Proceedings of the 2010 International Conference on Intelligent Computing and Integrated Systems*, 22-24 Oct 2010, 456-459

ZHENGXIAN X, HUAGUANG Z, Nonlinear decoupled controller design for wind power generator under ride-through, *2007 IPC International Power Engineering Conference*, 3-6 Dec 2007, Singapore, 988-993

ZHOU K, DOYLE JC, GLOVER K, Robust and Optimal Control, 1995, *Prentice Hall*

ZINOBER ASI, Deterministic control of uncertain systems, *Short Run Press*, Exeter, 1990

Appendix – Simulation and Modelling

The modelling and simulation of systems with multiple discontinuous elements is a challenging task and is itself an active area of research. The case study presented in Chapter 6 involved the modelling and simulation of a RAID control system with a missile flight control system. Within this system were many discontinuous elements, namely, the variable structure switching laws of the RAID controller and the rate and amplitude limitations for each of the missile's control surface actuators.

Whenever discontinuities are present within a system to be modelled and simulated it is important to pay close attention to the integration algorithm in order to ensure that simulation errors do not occur. ESL features an advanced discontinuity detected system to ensure that the integration solver does not lose accuracy when discontinuities occur. This is especially useful when many discontinuities occur simultaneously, as is the case for the RAID case studies when multiple actuator limits can be reached triggering corresponding discontinuities in the control algorithm.

The following section presents excerpts of ESL modelling code for the flight control case study and accompanying annotations of how the various discontinuities and system elements were modelled. The case study in chapter 5 – HVAC control – was also modelled and simulated in ESL but, due to the fewer number of discontinuous or nonlinear elements, is not described here.

ESL has a systematic modelling structure. This is illustrated by Figure A1. The fundamental elements of the ESL modelling environment are “Submodels”. Each of the Submodels are separately coded entities with input and output variables.

The various submodels are linked together and communicate through a central modelling entity known as the “Study”. The Study sets simulation parameters such as simulation time, integration step length, solver and user inputs such as requested setpoint. Within the Study is a central section of modelling code known as the “Model”, this is where all the submodels can interact. Data and parameters can be set within each model or submodel or if the data set is very large they can be included as an external “Package”.

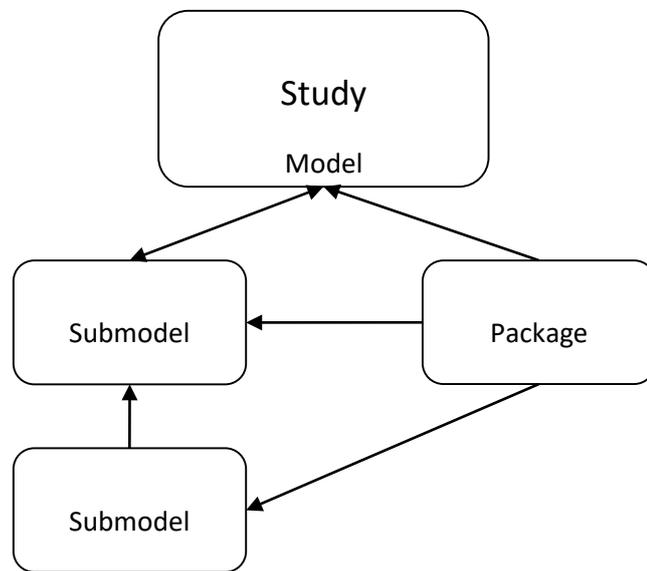


Figure A.1 – ESL modelling structure.

It is immediately clear to see that this type of structure is well suited to a flight control simulation with many systems and subsystems. Below is a diagram of the flight control case study system as it was modelled in ESL:

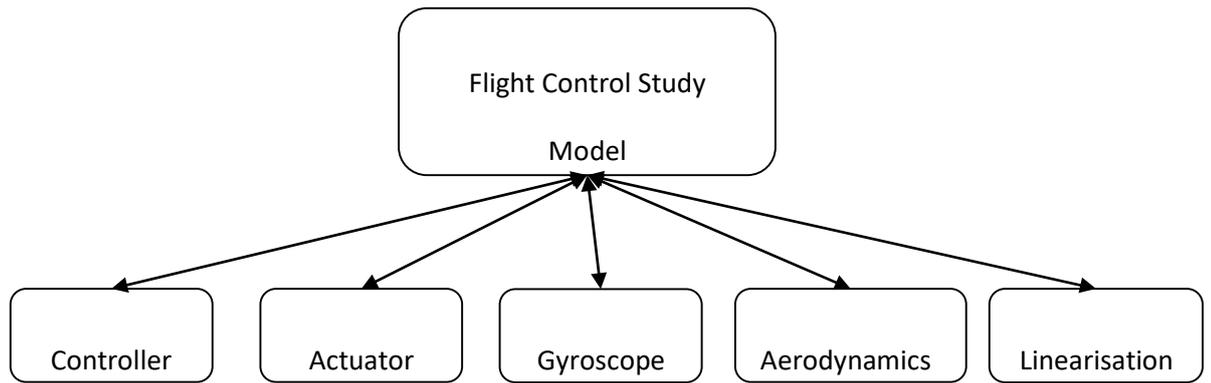


Figure A.2 – Flight control case study modelled system.

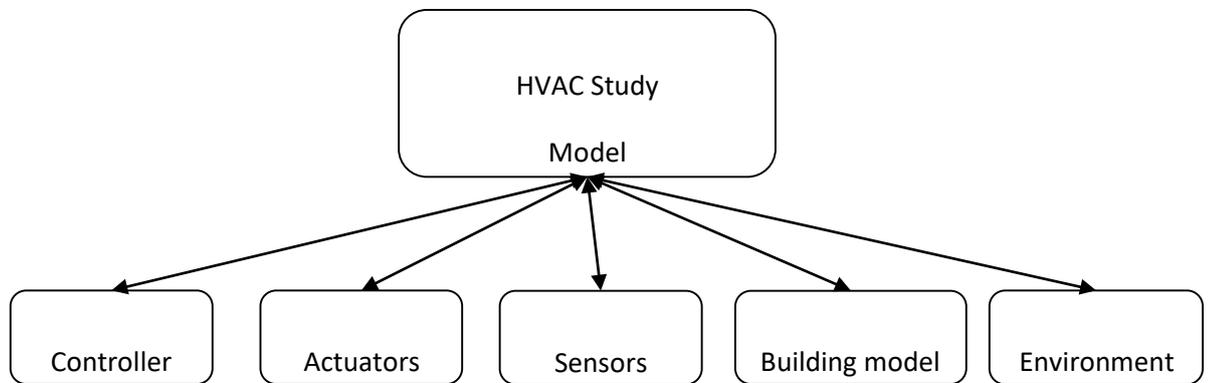


Figure A.3 – HVAC case study modelled system.

Each model or submodel has a strict structure containing separated regions.

This is as follows:

Initial

Variables are defined

Parameters and initial conditions are set

Arrays are defined

Dynamic

Dynamic modelling code

Submodels can be called here

Step

Executes code such as plotting or tabulation commands at each integration step

Communication

Executes commands at a specified communication interval

The following code excerpts are from the Dynamic region as this region pertains to the dynamic modelling code which is of interest.

RAID Controller Submodel

```
Dynamic
```

```
% Feedback vector is defined %
```

```
w := fb + (KD*dy);
```

%%%%%%%%%

% Actuator vector and feedback rates of changed defined %

```
PROCEDURAL(u1,u2,u3 := uvector);  
u1 := uvector(1); u2 := uvector(2); u3  
:= uvector(3);  
END_PROCEDURAL;
```

```
udot1:=DERIV(0.0,u1); udot2:=DERIV(0.0,u2);  
udot3:=DERIV(0.0,u3);
```

```
PROCEDURAL(udot := udot1,udot2,udot3);  
udot(1):=udot1; udot(2):=udot2;  
udot(3):=udot3;  
END_PROCEDURAL;
```

```
wdot1 := DERIV(0.0,w(1)); wdot2  
:= DERIV(0.0,w(2)); wdot3 :=  
DERIV(0.0,w(3));
```

```
PROCEDURAL(wdot := wdot1,wdot2,wdot3);  
wdot(1):=wdot1; wdot(2):=wdot2;  
wdot(3):=wdot3;  
END_PROCEDURAL;
```

%%%%%%%%%

& Ueq estimate is defined %

```
ueq:= -(inv(KD*cb))*wdot + urate;
```

%%%%%%%%%

% Regulator vector and regulator switching is defined

--Define regulator

zd := KI*error;

zd1s:= if abs(uc(1)) > RL then 0.0 else_if
uvector(1) > AL and uc(1) > 0.0 then 0.0 else_if
uvector(1) < -AL and uc(1) < 0.0 then 0.0
 else zd(1);

zd2s:=if abs(uc(2)) > RL then 0.0 else_if
uvector(2) > AL and uc(2) > 0.0 else_if
uvector(2) < -AL and uc(2) < 0.0 then 0.0
 else zd(2);

zd3s:=if abs(uc(3)) > RL then 0.0 else_if
uvector(3) > AL and uc(3) > 0.0 else_if
uvector(3) < -AL and uc(3) < 0.0 then 0.0
 else zd(3);

PROCEDURAL(zds := zd1s,zd2s,zd3s);
zds(1):=zd1s; zds(2):=zd2s;
zds(3):=zd3s;
END_PROCEDURAL;

z' := zds;

%%%%%%%%%

% Define RAID control algorithm

--
uc:= z - KP*w + ueq;

%%%%%%%%%

% Control signal vector is deconstructed for actuators %

```
con_eta:=uc(1);
con_zeta:=uc(2); con_xi:=uc(3);
ceta:=con_eta; czeta:=con_zeta;
cxi:=con_xi;
```

%%%%%%%%%

Actuator submodel for RAID controller

```
SUBMODEL sm_actuatorraid2(REAL:eta,zeta,xi,eta_rate,zeta_rate,xi_rate:=
REAL:ceta,czeta,cxi,etat,zetat,xit);
```

```
REAL: accln_eta,accln_zeta,accln_xi;
CONSTANT REAL: LVL/-17.5/, UVL/17.5/, LL/-0.3/, UL/0.3/;
CONSTANT REAL: tau/0.003/;
```

```
--
INITIAL
-- eta_rate :=
0.0; zeta_rate :=
0.0;
xi_rate := 0.0;
```

```
--
DYNAMIC
--
accln_eta := (1.0/tau)*(ceta - eta_rate); accln_zeta
:= (1.0/tau)*(czeta - zeta_rate);
accln_xi := (1.0/tau)*(cxi - xi_rate);
```

```
--
--
eta := LIMINT(etat,LL,UL,eta_rate); zeta
:= LIMINT(zetat,LL,UL,zeta_rate);
xi := LIMINT(xit,LL,UL,xi_rate);
```

```
--
--
STEP
--
```

```

-- digitally integrate accln using a simple euler integration and calculate real rate when on limits
--
eta_rate := eta_rate + (CINT/NSTEP)*accln_eta; zeta_rate
:= zeta_rate + (CINT/NSTEP)*accln_zeta; xi_rate :=
xi_rate + (CINT/NSTEP)*accln_xi;
--
-- rate limits
--
if eta_rate > UVL then
eta_rate := UVL; else_if
eta_rate < LVL then
eta_rate := LVL;
end_if;

if zeta_rate > UVL then
zeta_rate := UVL; else_if
zeta_rate < LVL then
zeta_rate := LVL;
end_if;

if xi_rate > UVL then
xi_rate := UVL; else_if
xi_rate < LVL then
xi_rate := LVL; end_if;

--
-- correct rate to ensure it is zero on deflection limits
if eta >= UL and eta_rate > 0.0 then eta_rate := 0.0;
else_if eta <= LL and eta_rate < 0.0 then
eta_rate := 0.0;
end_if;

if zeta >= UL and zeta_rate > 0.0 then
zeta_rate := 0.0; else_if zeta <= LL and
zeta_rate < 0.0 then zeta_rate := 0.0;
end_if;

if xi >= UL and xi_rate > 0.0 then
xi_rate := 0.0; else_if xi <= LL and
xi_rate < 0.0 then
xi_rate := 0.0;
end_if;

END sm_actuatorraid2;

```

Actuator submodel for RIDE controller

```

SUBMODEL                                sm_actuator(REAL:eta,zeta,xi,eta_rate,zeta_rate,xi_rate:=
REAL:ceta,czeta,cxi,etat,zetat,xit);
--
--
REAL: accln_eta,accln_zeta,accln_xi;
CONSTANT REAL: LVL/-17.5/, UVL/17.5/, LL/-0.3/, UL/0.3/;
CONSTANT REAL: adamp/0.7/,awn/350.0/;
--
INITIAL
-- eta_rate :=
0.0; zeta_rate :=
0.0;
xi_rate := 0.0;
--
DYNAMIC
--
accln_eta := (awn*awn*(ceta-eta)) - (2.0*adamp*awn*eta_rate); accln_zeta
:= (awn*awn*(czeta-zeta)) - (2.0*adamp*awn*zeta_rate); accln_xi :=
(awn*awn*(cxi-xi)) - (2.0*adamp*awn*xi_rate);
--
--
eta := LIMINT(etat,LL,UL,eta_rate); zeta
:= LIMINT(zetat,LL,UL,zeta_rate);
xi := LIMINT(xit,LL,UL,xi_rate);
--
--
STEP
--
-- digitally integrate accln using a simple euler integration and calculate real rate when on limits
--
eta_rate := eta_rate + (CINT/NSTEP)*accln_eta; zeta_rate
:= zeta_rate + (CINT/NSTEP)*accln_zeta;
xi_rate := xi_rate + (CINT/NSTEP)*accln_xi;
--
-- rate limits
--
if eta_rate > UVL then
eta_rate := UVL; else_if
eta_rate < LVL then
eta_rate := LVL; end_if;
--

```

```

if zeta_rate > UVL then
zeta_rate := UVL; else_if
zeta_rate < LVL then
zeta_rate := LVL;
end_if;
--
if xi_rate > UVL then
xi_rate := UVL; else_if
xi_rate < LVL then
xi_rate := LVL;
end_if;
--
-- correct rate to ensure it is zero on deflection limits
if eta >= UL and eta_rate > 0.0 then eta_rate
:= 0.0;
else_if eta <= LL and eta_rate < 0.0 then
eta_rate := 0.0; end_if;
--
if zeta >= UL and zeta_rate > 0.0 then zeta_rate
:= 0.0;
Else if zeta <= LL and zeta rate < 0.0 then
Zeta rate: = 0.0;
Endive;
--
If xi >= UL and irate > 0.0 then
Irate: = 0.0;
Else if xi <= LL and irate < 0.0 then
Irate: = 0.0;
Endive;
--
--
END sm_actuator;

```

