Hanz Richter

Advanced Control of Turbofan Engines

Springer

Advanced Control of Turbofan Engines

Hanz Richter

Advanced Control of Turbofan Engines

Springer

Hanz Richter
Department of Mechanical Engineering
Cleveland State University
Euclid Avenue 2121
44115 Cleveland, Ohio
USA
h.richter@csuohio.edu

ISBN 978-1-4614-1170-3 e-ISBN 978-1-4614-1171-0
DOI 10.1007/978-1-4614-1171-0
Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2011936014

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

In memory of my father

Preface

Gas turbine engines – in particular, turbofan engines ubiquitously installed in commercial aircraft – must be operated by means of feedback control. In a broad sense, the objective of the control system is to achieve good thrust response qualities while maintaining critical engine outputs within safety limits. The design of controllers capable of delivering this objective represents a challenging problem, even when linear models with known parameters are considered for analysis. The fact that gas turbine engine dynamics are nonlinear and subject to uncertain parameter variations adds many layers of complexity to the problem.

Propulsion control systems installed in operating commercial aircraft, however, are ultimately based on classical, SISO linear compensation loops. Features have been added incrementally over the course of their development to address the exigencies of faster and more powerful, yet more reliable engine installations. Parameter variability from measurable sources – such as altitude and Mach number – has traditionally been accounted for by introducing gain scheduling, while engine safety limits have been addressed by override schemes. Both features still retain classical feedback compensation at their core.

Standard engine control systems have been in use for decades, without major conceptual changes. Concurrently, many new control theories – many of them with demonstrated industrial applications – have been developed. Much control systems research is devoted to the recurrent themes of parametric uncertainty, nonlinearity, and constraints in system variables. These themes are characteristic of the most challenging control problems and certainly arise in gas turbine engines.

The wide gap between the host of available advanced control technologies and the *ad-hoc* implementations of everyday practice may be explained by the simple fact that the latter have been so far sufficient. Recent research thrusts in aircraft control, however, require that the performance that can be extracted from the engines be maximized, given a set of allowable limits on critical variables. Specifically, fast thrust responses are critical in certain emergency flight maneuvers. As elaborated in the book, classical feedback is no longer suitable as a paradigm for the development of advanced propulsion control concepts.

One of the two chief motivations to write this book has been to bring advanced controls closer to the highly relevant application domain of gas turbine engines. Before advanced concepts can be introduced, standard engine controls must be reviewed and made precise. Presenting standard engine controls using a precise control systems framework has been the second major motivation. As a result, the book spans introductory topics such as the engine's principle of operation and dynamic model, followed by classical feedback compensation and ending with advanced research topics.

Book Audience

The book is aimed at readers falling between two ends of a broad spectrum: at one end are aerospace or mechanical engineers with basic knowledge in control systems; at the other are control engineers with little or no knowledge about gas turbines (the typical electrical engineer may fit this category). Of course, any "interpolated" engineer will also benefit from this book, since it includes detailed information on turbine engine systems and introduces advanced control topics.

The material should be accessible to first-year graduates in mechanical, electrical, or aerospace engineering. Readers are assumed to be familiar with classical control concepts such as stability, root locus design, and frequency response, as well as basic pole-placement design. Proficiency in Matlab/Simulink is required to follow the numerical examples and understand simulations.

The book contains numerous nonlinear engine simulations conducted using NASA's CMAPSS package, available to the public in the USA. This package is required only for readers seeking to reproduce the nonlinear engine simulations contained in the book. The linearized state-space matrices for the 90,000-lb and 40,000-lb engine have been included in the appendix for readers not using CMAPSS. The data are useful to follow many control design calculations, and may even be used to generate a custom simulation accounting for parametric changes.

Although the book does not contain end-of-chapter problems, it is written much in the style of a textbook. As such, it may be used as the basis or as reference material for a graduate course in aircraft engine controls. A course with emphasis in fundamental concepts could be designed using Chaps. 1–3, 5, and 7, while a course emphasizing research aspects could be designed using Chaps. 1, 2, 6–9.

Outline

Chapter 1: Introduction reviews the thermodynamic principles explaining the operation of a gas turbine, assuming minimal familiarity with thermodynamic variables. The components of the real gas turbine engine and its operation are described,

introducing key quantitative performance measures. Safety and operational limits are discussed, including mathematical descriptions of surge and stall phenomena.

Chapter 2: Engine Models and Simulation Tools offers a brief overview of engine dynamics, aiming at the extraction of linearized models that can be used as a basis for design. The Commercial Modular Aeropropulsion System Simulation (CMAPSS) package developed by the NASA Glenn Research Center is also described.

Chapter 3: Engine Control by Classical Methods reviews and applies classical SISO design techniques (root locus and frequency domain loopshaping) to the problem of fan speed control using fuel flow rate as control input. A model-matching method is also described that is used in CMAPSS as a design tool. The shortcomings associated with the use of fixed linear compensation are illustrated with simulation examples.

Chapter 4: Engine Control by Robust State Feedback reviews linear multivariable theory and introduces polytopic system descriptions of plant variability. This chapter also presents various methods for MIMO state-feedback synthesis, such as: LQR, $\mathcal{H}_2$, $\mathcal{H}_\infty$ and mixed-objective optimization with regional pole placement constraints. A simplified $\mathcal{H}_\infty$ compensator synthesis method is presented for SISO systems. Matlab code and simulations using the CMAPSS nonlinear engine model are included.

Chapter 5: Gain Scheduling and Adaptation introduces gain-scheduling and linear-parameter-varying techniques to address plant variability across the flight envelope. This chapter also introduces the concept of adaptive control and presents a basic model-reference adaptive control design. Matlab code and simulations using the CMAPSS nonlinear engine model are included.

Chapter 6: Sliding Mode Control of Turbofan Engines introduces the concept of sliding mode control and elaborates in its robustness properties and commonly-used tuning approaches. This chapter also presents MIMO versions of the sliding mode regulator and setpoint tracker, as well as a simplified SISO design. Linear and nonlinear engine simulations using CMAPSS are included.

Chapter 7: Engine Limit Management with Linear Regulators describes the min–max logic arrangement used in standard engine control systems to maintain critical variables within permissible bounds. A thorough analysis of this arrangement is conducted using the concept of positive invariance. The shortcomings of the min–max approach are made evident in simulations. A brief description of an acceleration-limiting approach is also included.

Chapter 8: Engine Limit Management with Sliding Modes develops a method to maintain critical engine variables within allowable limits, without the disadvantages associated with the standard min–max approach. Guidelines for the association of sliding mode regulators to logic max or min selectors are given, along with an $\mathcal{H}_2/\mathcal{H}_\infty$ sliding coefficient synthesis method. Simulations using the CMAPSS nonlinear engine model are included.

Chapter 9: Engine Limit Management with Model Predictive Control introduces the concept of model predictive control and develops basic prediction formulas based on linear state-space models. The constrained optimization problem

is formulated using compact matrix formulas suitable for incorporation in Matlab's quadratic program solver. Model predictive control is then applied to the engine control problem to address input and output constraints. The chapter also discusses computational complexity and approaches aimed at its reduction. Matlab code and simulations using the CMAPSS nonlinear engine model are included.

Topics Not Covered

Among the most relevant topics not covered are: (a) engine modeling through system identification, (b) observer-based control, (c) engine health estimation, and (d) robust output-feedback synthesis (except for the simplified method of Sect. 4.9).

Note that gas turbine engine control represents a rare (and fortunate) case where the state variables are available as real-time measurements. This explains the intentional omission of (b) and (d). As far as (a) and (c), these topics are vast and could each give rise to an entire volume. At the same time, they can be safely omitted from a controls-oriented treatment. References have been included throughout the book for readers interested in those topics.

Acknowledgments

I wish to express my deepest gratitude to Jonathan S. Litt and Sanjay Garg at NASA Glenn Research Center, Control and Dynamics Branch in Cleveland for their continued support of my research and their valuable guidance.

Special thanks to my cheerful wife,[1] Francisca, for being robust against, well, *me*, and for being tolerant to any disturbance I may have introduced as this book was being finished. Thanks to my three daughters, Micaela, Emilia, and Martina, for letting me install Linux and take possession of the home computer for 18 months.

Finally, thanks to Prof. William Atherton, Mechanical Engineering Department Chair, for granting me a semester of reduced teaching load so that I could concentrate on finishing the manuscript.

Department of Mechanical Engineering
Cleveland State University
Cleveland, OH

Hanz Richter

[1] *A cheerful wife is the joy of life – Goethe.*

Contents

Chapter 1
Introduction

Abstract This chapter reviews the thermodynamic principles explaining the operation of a gas turbine, assuming minimal familiarity with thermodynamic variables. The components of the real gas turbine engine and its operation are described, introducing key quantitative performance measures. Safety and operational limits are discussed, including mathematical descriptions of surge and stall phenomena.

The chapter introduces gas turbine engines (GTEs), describing their operating principles and constraints mostly in a qualitative fashion. Aerospace engineers and readers who are familiar with these topics can skip it altogether, or give it a quick look and start with the chapter on mathematical modeling.

The GTE is one of the most complex machines ever built, not only for its constructive intricacies, but also for the dynamic behaviors it displays and the sophisticated engineering required for its operation. Due to its high fuel efficiency, power ratings, and reliability, the GTE is used for various forms of transportation systems, including airplanes, helicopters, marine ships, and military tanks (the M1 Abrams, for instance). Ground-based electric power generation relies on GTEs for peak periods in many of the world's metropolitan areas.

This book focuses on control systems for turbofan engines, a GTE design optimized for commercial air transportation. The principles of operation of turbofan engines and their control methods are similar to those of other kinds of GTE. Novice readers who use this book as their primary reference source for aircraft engine controls will be comfortable when examining material specific to other kinds of GTE.

1.1 Principles of Operation

Useful power is extracted from a GTE by three means:

1. *Momentum exchange*: most of the useful power delivered by aeronautical GTEs results from the ejection of gas at high velocities through the exit nozzle.

H. Richter, *Advanced Control of Turbofan Engines*, DOI 10.1007/978-1-4614-1171-0_1,

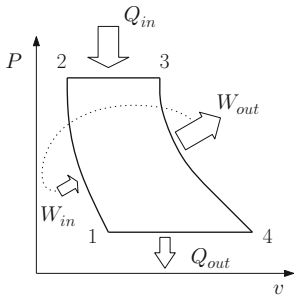

Fig. 1.1 The ideal Joule–Brayton cycle

The reaction thrust is proportional to both mass flow rate and velocity of the ejected gas relative to the engine.

2. *Direct mechanical connection of a load to an engine shaft*: accessories such as pumps and electric generators in aircraft engines derive their power this way. Ground-based engines used in power generation deliver most of their power through a shaft connected to a generator. Marine engines deliver power mechanically to propellers.
3. *Fluid transfer (bleed air)*: high-pressure, hot air may be extracted from aircraft GTE compressors for various purposes, for instance de-icing systems, pneumatic actuators, and cabin climate control.

1.1.1 The Joule–Brayton Cycle

GTEs are approximate practical implementations of the *Joule–Brayton thermodynamic cycle*. The cycle is devised so that heat energy (from combustion of fuel) can be converted into mechanical work with high efficiency. Since the energy content of a working fluid increases with pressure and temperature, mechanical compression (volume reduction) and heat addition through combustion are used. The fluid is then allowed to expand and cool, returning to its initial state. Mechanical power is extracted during the expansion stage by forcing the flow to impinge on a movable surface (turbine blade) or by momentum exchange as described above. The ideal Joule–Brayton cycle is shown in a pressure-volume diagram in Fig. 1.1. The working fluid is compressed from 1 to 2 without transferring heat to the surroundings or generating heat due to friction. This idealized process is known as *isentropic compression*. Heat is then added at constant pressure between 2 and 3. The energy required to bring the fluid from points 1 to 3 equals $Q_{\text{in}} + W_{\text{in}}$, that is, the heat energy added by the fuel and the mechanical work required to compress the fluid. A total mechanical power of W_{out} is extracted through isentropic expansion between 3 and 4. The dotted line in Fig. 1.1 indicates that part of W_{out} is diverted to drive the compressor, a feature inherent to all GTE designs. Thus, only Q_{in} is regarded as an energy "investment" in efficiency calculations. The cycle is

completed by constant-pressure cooling of the fluid. Cooling is achieved by heat transfer Q_{out} from the fluid to the surroundings. In the ideal Joule–Brayton cycle, this heat represents lost energy. The efficiency of the ideal cycle is obtained by dividing the net useful work by the energy "investment":

$$\eta_{\text{i}} = \frac{W_{\text{out}} - W_{\text{in}}}{Q_{\text{in}}}. \tag{1.1}$$

When the working fluid is an ideal gas with ratio of specific heats $\gamma = \frac{C_{\text{p}}}{C_{\text{v}}}$, the efficiency of the ideal Joule–Brayton cycle can be derived as a function of the cycle pressure ratio as [1]:

$$\eta_{\text{i}} = 1 - \frac{1}{(\frac{P_2}{P_1})^{\frac{\gamma-1}{\gamma}}}. \tag{1.2}$$

The ratio $\frac{P_2}{P_1}$ plays an important role in key aspects of engine modeling, control, and monitoring. It receives the name of *engine pressure ratio*, or *EPR*, a quantity analogous to the compression ratio of piston engines. Note that high ideal efficiencies correspond to high EPR. Engines installed in large commercial aircraft, such as General Electric's GE90, can have EPR values higher than 40. Several components are arranged in a GTE to perform the basic processes of the thermodynamic cycle:

1. A *turbocompressor*, or simply *compressor*: An arrangement of blades on a rotating disk, whose function is to force air to flow into the engine as it reduces its volume and increases its pressure. Torque is required to change the momentum of the working fluid, forcing it to follow the curved surface of the blades. The work required to drive the compressor comes from the engine itself (dotted line in Fig. 1.1), by means of a shaft connecting compressor and turbine. Thus, the compressor and the turbine rotate at the same angular speed. A GTE may have one or more compressors, typically attached to corresponding turbines in separate shafts.
2. A *combustion chamber*, or *combustor*: A cavity where fuel is added to compressed air, resulting in combustion. The amount of energy added to the gas as a result of combustion depends primarily on the heating value of the fuel and the rate of fuel injection. This rate is the main control input of the propulsion control system. To gain an idea of the orders of magnitude involved, consider that the engine used in Boeing 777 aircraft has a typical fuel consumption of about 0.5 l/s at takeoff [2]. The Jet A-1 fuel has a density of 0.8 kg/l and a heating value of 43 MJ/kg. Thus, the power contained in the fuel stream for a single engine is 17.2 MW. Naturally, not all the energy contained in the fuel is transferred to the working fluid and in turn, not all of the energy of the fluid can be converted in mechanical work propelling the aircraft.
3. A *turbine*: Conceptually, the opposite of the compressor, the turbine is an arrangement of blades on a disk, which rotates due to impingement of fluid. A turbine produces torque as a result of a momentum change of the fluid as it follows the curved surface of the blades. Turbines may be designed so that

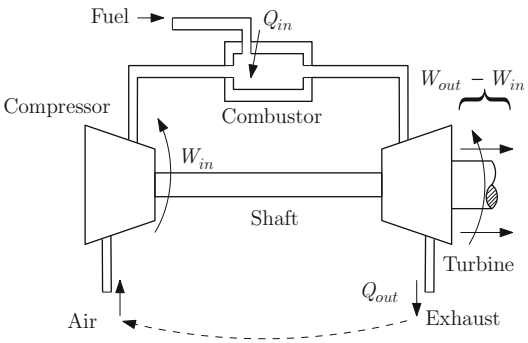

Fig. 1.2 Basic arrangement of components in a gas turbine engine

the exhaust gas has a large residual energy content. This is the case in jet engines, where most of the thrust used to propel the aircraft is obtained by high-velocity exhaust through a nozzle and only a small fraction of mechanical power is extracted by the turbine to drive the compressor, fan, and accessories. In contrast, a turbine may be designed for greater availability of mechanical power at the shaft. This is the case of turbopropeller engines, where the shaft drives the propellers, usually through a gearbox. Gas past the turbine is still directed to a nozzle, producing a small additional thrust.

Figure 1.2 illustrates the arrangement of these three basic components in a conceptual GTE. In terms of mechanical construction, GTEs can be roughly classified by the number of shafts, or *spools*. A single-spool engine has one turbine and one compressor which rotate as a unit, connected by one shaft. The two-spool configuration prevails in turbofan engines used in modern commercial aircraft as well as some military engines, for instance the SNECMA M88. In a two-spool configuration, two compressors and two turbines are linked by concentric shafts, which rotate independently.

1.2 The Bypass Turbofan Engine

The bypass turbofan GTE design was conceived to improve the fuel efficiency of commercial aircraft. An additional component – the fan – is installed at the inlet of the engine to increase the amount of air flowing through the engine. However, part of the inlet airflow is not directed toward the compressor, combustor, and turbine, but is rather *bypassed* through a *duct*, which ends in a nozzle. Since air leaves the nozzle at a speed which is higher than the intake velocity, thrust is produced by momentum exchange with the airframe. The flow which is not bypassed undergoes the processes studied in Sect. 1.1.1, producing part of the total thrust. Note that the thrust generated by the bypass flow does not require the burning of fuel, but some mechanical work must be derived from a turbine to drive the fan.

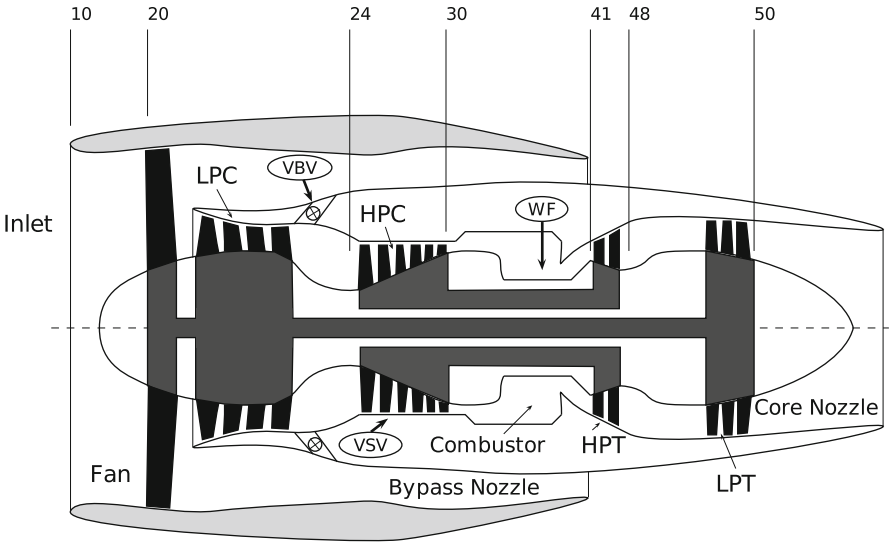

Fig. 1.3 Two-spool turbofan engine design

Although extra fuel must be burned to generate the power required to drive the fan, more would have to be burned to achieve the same total thrust in the absence of a bypass flow.

A more detailed explanation of the superior thrust efficiency of high bypass versus low bypass designs is offered in Sect. 1.2.1. Figure 1.3 illustrates the two-spool turbofan engine design upon which most of this book is based. A cutaway model of a single-spool turbofan design can be seen in Fig. 1.4. The fan, low-pressure compressor, (LPC) and low-pressure turbine (LPT) are connected by one shaft and thus rotate synchronously. The high-pressure compressor (HPC) and high-pressure turbine (HPT) are next to the combustor and are connected by a separate shaft, usually concentric with the LP shaft. The arrangement of HPC, combustor, HPT, and core nozzle is referred to as *core engine*. The symbols WF, VBV, and VSV enclosed in ovals in Fig. 1.3 correspond to the main actuators used in GTE control systems. WF corresponds to the *fuel flow* delivered by a pump, usually expressed in pounds per second (pps). VBV denotes the *variable bleed valve*. The VBV is used to extract high-pressure gas from the core, injecting it into the bypass flow, which has a lower static pressure. This is done to prevent the compressor from entering an unstable region of operation known as *surge*, which is discussed in Sect. 1.3.1. VBV action is usually measured as a percentage or fraction of the fully-open position. VSV represents the *variable stator vanes*. A compressor is built with several stages of rotating blades separated by a stationary vane arrangements called *stators*. The orientation of stator vanes may be changed with the engine in operation to control the characteristics of the flow reaching the rotating blades. Specifically, VSV are

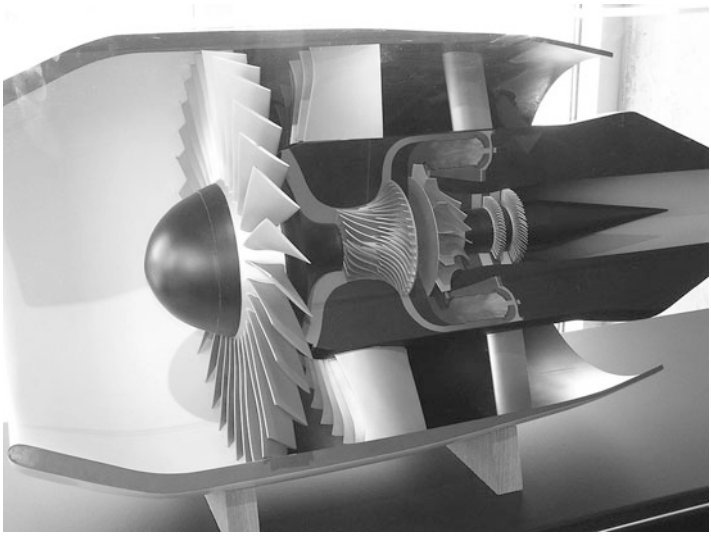

Fig. 1.4 Cutaway model of a single-spool turbofan engine (photo courtesy Great Lakes Science Center, Cleveland)

manipulated to reduce flow separation and subsequent *stall* in compressor blades. VSV action is measured in degrees. More details about compressor and turbine operability will be given in Sect. 1.3. Figure 1.3 includes part of the numbering used to designate engine stations in standard practice [3]. For instance, numbers 20–24 are used to designate points between fan inlet and LPC outlet, while the range 41–48 is used for the HPT.

Turbofan engines may also be classified by their *bypass ratio*, that is, the ratio of bypass to core mass flows. High bypass turbofans may have ratios as high as 11, as is the case of the Rolls–Royce Trent 1000 engines installed in Boeing 787 aircraft. Low bypass engines may use ratios as low as 0.3; the SNECMA M88 used in Dassault Rafale aircraft is one example. The bypass ratio value is chosen to prioritize either fuel efficiency or aircraft speed. Indeed, the core exhaust has a significantly higher velocity than the bypass exhaust due to the addition of energy through fuel. When the bypass ratio is high, a large, slow-moving mass of air is ejected through the bypass nozzle, creating thrust mainly due to high mass flow rather than high exhaust velocity. If the bypass mass flow is reduced by certain factor and the thrust is to be constant, the velocity of the bypass flow must be increased by the same factor, which can be achieved by properly designing the duct and the nozzle. In both scenarios, no fuel is burned in the bypass flow and the thrust is the same, giving the impression that there is no difference in terms of fuel efficiency. However, in the low bypass ratio case, more fuel has to be burned, since the core flow is higher and the proportion of air to fuel has to be maintained in a narrow range.

1.2.1 Performance and Efficiency of Real Engines

A number of measures besides the ideal Joule–Brayton cycle efficiency are defined to characterize the overall operation of a GTE. When the actual thrust produced by the engine is F_n, the *effective jet velocity* v_{eff} is defined through (1.3) below:

$$F_n = \dot{m}(v_{eff} - v_0), \tag{1.3}$$

where v_0 is the velocity of the air entering the engine relative to the moving aircraft. When there is no wind, v_0 is numerically equal to the aircraft's ground speed. The *thrust power* P_{th} is defined as the rate of work exerted by the thrust in moving the aircraft at constant speed, that is,

$$P_{th} = F_n v_0 = \dot{m} v_0 (v_{eff} - v_0). \tag{1.4}$$

In contrast, the *propulsion power* P_{pr} is the rate at which kinetic energy is added to air as it flows at constant speed through the engine:

$$P_{pr} = \frac{1}{2}\dot{m}(v_{eff}^2 - v_0^2). \tag{1.5}$$

The ratio of useful (thrust) power to invested (propulsion) power is known as *propulsion efficiency*, or *Froude efficiency*:

$$\eta_{pr} = \frac{P_{th}}{P_{pr}} = \frac{2}{1 + \frac{v_{eff}}{v_0}}. \tag{1.6}$$

Equation (1.6) elucidates the question of whether a design should favor a high value of $\dot{m}$ and a low value of v_{eff} in (1.3), given a constant thrust level and some fixed v_0. For higher propulsion efficiency, it is clearly preferable to use a low v_{eff} and a high $\dot{m}$.

The *thermal efficiency* η_{th} is defined as the ratio of propulsion power to the power introduced through fuel burning:

$$\eta_{th} = \frac{P_{pr}}{\dot{m}_f H_f}, \tag{1.7}$$

where $\dot{m}_f$ is the rate of fuel mass consumption and H_f is the heating value of the fuel. The *total efficiency* is defined as the ratio of thrust power to the power introduced through the burning of fuel:

$$\eta_{tot} = \frac{P_{th}}{\dot{m}_f H_f} = \eta_{pr}\eta_{th}. \tag{1.8}$$

The *specific thrust* F_s is defined as the engine thrust per unit of air mass flowrate:

$$F_s = \frac{F_n}{\dot{m}}. \tag{1.9}$$

Table 1.1 Overall performance figures of high-bypass and low-bypass GTEs (adapted from [4] and [5])

	SNECMA M88-2	GE-90
Max. thrust, kN	50	513
Engine pressure ratio	25:1	42:1
Air mass flow, kg/s	65	1,350
Bypass ratio	0.3:1	8.4:1
Thrust specific fuel consumption, kg/N-h	0.8	0.03

Finally, the *thrust specific fuel consumption* $\dot{m}_{\mathrm{ns}}$ is a measure of fuel economy in relation to thrust generation:

$$\dot{m}_{\mathrm{ns}} = \frac{\dot{m}_{\mathrm{f}}}{F_{\mathrm{n}}} = \frac{v_0}{\eta_{\mathrm{tot}} H_{\mathrm{f}}}. \tag{1.10}$$

1.2.1.1 High-Bypass vs. Low-Bypass Engines

A high-bypass engine is designed to enhance $\dot{m}_{\mathrm{f}}$ in (1.3) while maintaining a low jet exhaust velocity v_{eff}. This is accomplished by the large-diameter fan mounted at the inlet and a proper bypass nozzle design. From (1.6) and (1.8), we see that for equal values of thrust, a high-bypass engine will have greater propulsion and total efficiencies than a low-bypass engine. Since $\dot{m}$ is larger in a high-bypass engine, the specific thrust will be lower, and most importantly, the thrust specific fuel consumption will also be smaller, as judged from (1.10). This reasoning explains why high-bypass turbofans are a *de facto* choice for modern commercial aircraft.

Note that v_0 is not constant, but increases as the aircraft reaches cruise conditions. When v_0 increases, thrust decreases, as seen in (1.3). Such *thrust lapse* is more serious in high-bypass engines than in low-bypass engines, since v_{eff} is relatively small. Designers must oversize engines to maintain the necessary thrust at high airspeeds. As a beneficial consequence, high-bypass engines have very large take-off thrust values, allowing them to lift heavy aircraft with relatively short take-off distances. In contrast, low-bypass turbofans operate well at very high airspeeds, explaining their choice for military aircraft, where fuel economy is not the biggest priority. A comparison of the characteristics of a high-bypass and a low-bypass engine is shown in Table 1.1.

1.3 Operability Limits and Component Maps

The efficiency figures introduced in Sect. 1.2.1 are formulated on the basis of the engine as a whole. Individual engine components have their own efficiencies, which collectively determine overall engine efficiency. Thus, it is important to describe the

efficiencies of individual components as well as safe ranges of operation. Note that, in some cases, highly efficient operating conditions may not necessarily correspond to safe conditions. This occurs in fans and compressors, where stall limits overlap regions of high adiabatic efficiency.

Efficiency and operability limits are described for fans, compressors, and turbines by means of a *component map*. An accurate component map is constructed by operating the component at various combinations of outlet/inlet pressure ratio and mass flow rate under steady-state conditions. Although it is customary to use nondimensional and corrected forms of pressure and mass flow rate, the plot essentially describes the relationship between mass flow rate (horizontal axis) and the corresponding pressure ratio (vertical axis). The component may also be operated across several equilibrium points as a parameter or an input is varied. For instance, when a throttle valve is installed at the outlet of a compressor, and valve opening is varied slowly, a locus of points called *throttle line* may also be represented on the compressor map.

1.3.1 Compressor and Fan Maps

The coordinates of a point in the compressor map correspond to steady-state operation at some mass flow rate and corresponding pressure ratio. Unique values of rotor speed and efficiency exist for a given set of compressor map coordinates. Efficiency and speed data are represented in the map in the form of constant-efficiency and constant-speed lines, as shown in Fig. 1.5. Specifically, the *adiabatic efficiency* of compression is used, given by the ratio of the work needed to raise the pressure under isentropic conditions to the actual work required to produce the same pressure ratio.

1.3.1.1 Surge and Stall

The controls-minded reader may now find it useful to interpret the isolated compressor as a nonlinear dynamic system where mass flow rate and pressure rise are states. The compressor map is drawn in phase space, where outputs of interest (rotor speed, efficiency) are represented as constant-value contours. Compressors interacting with other engine elements such as a combustor or higher-pressure compressor stages exhibit complex nonlinear behavior, some of which is regarded as destructive and highly undesirable.

Rotating stall is an operating regime in which groups of compressor blades are unable to move air in the flow direction due to localized off-design flow conditions. Pockets of stagnant air, called *stalled cells*, which span a portion of the circumference of the compressor disk are formed, reducing compression efficiency. Stall is a wave-like phenomenon: stalled cells travel around the circumference of the compressor at speeds near the mechanical rotational speed, hence the designation

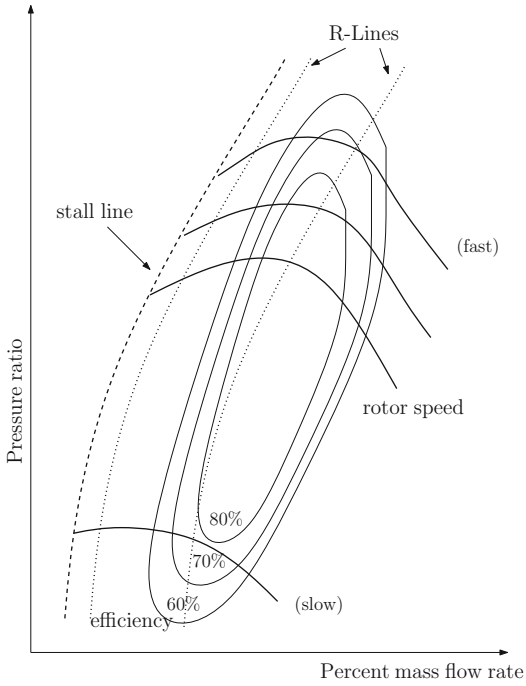

Fig. 1.5 Typical features of a compressor map

rotating stall. Stall may arise as a consequence of off-design engine operation or damage to the blades caused by foreign object ingestion. The notorious US Airways flight 1549 Hudson river landing of January 15, 2009 was caused by stall-induced power loss in both engines. The Airbus A320 had flown through a flock of birds a few minutes prior to the crash [6].

Surge is a more nefarious occurrence, where air not only fails to flow in the desired direction, but also may flow explosively toward the inlet of the compressor. Unlike rotating stall, surge is axisymmetric, spanning the whole compressor circumference. When reverse flow occurs, upstream pressure is somewhat relieved, tending to suppress reverse flow. If boundary conditions are maintained, however, upstream pressure builds up again and the process is repeated in a limit cycle. The mechanical vibrations and thermal stresses arising from a surge cycle can have consequences ranging from accelerated engine wear to damage and total engine destruction.

The operating conditions which induce surge or stall in compressors and fans are provided by engine manufacturers and represented in component maps in the form of a boundary, which is equivalently referred to as *stall line* or *surge line*. This information is supplied to the controls engineer with the basic premise that controller designs must not generate transients causing the component to operate close to the surge line.

Although it is standard practice to design controllers to maintain a safe distance from the surge line, researchers have given ample consideration to compressor

operation across this line, in an effort to produce mathematical models that can be used to design surge recovery and surge suppression controllers. These controllers could be made active upon the detection of unforeseen damage to an engine component. The model by Moore and Greitzer [7] was widely adopted as a basis for nonlinear analysis and control design. This model considers a compression system formed by a ducted compressor discharging to a chamber, whose exhaust is controlled by a throttle valve. First principles of fluid dynamics are applied to generate a system of partial differential equations (PDE), which is then converted to a system of three ordinary differential equations (ODE) by Galerkin approximation. The system of equations is formulated in terms of pressure rise, flow coefficient, and amplitude of oscillation of the first harmonic of angular flow disturbance.

Moore and Greitzer used the PDE model to establish that pure rotating stall and pure surge are eigenmodes of the same coupled phenomena, and that they appear combined in general cases. The third-order ODE model was later used by Baillieul, Dahlgren, and Lehman to show that stall inception occurs through *equilibrium point bifurcation* [8], a nonlinear effect whereby abrupt changes in the number or nature of equilibrium points are brought about by a continuous change of a parameter or input. As described earlier, compressor stall is characterized by a self-sustaining oscillation, or limit cycle, around an equilibrium point. Away from the stall boundary, the equilibrium point is stable. Baillieul and co-workers showed that feedforward injection of air at compressor inlet could be used to enlarge the region in which the equilibrium point is stable, an effective means of suppressing stall.

A simplified second-order model, due to Greitzer [9] was later used by Simon and Valavani [10] to design a feedback controller using Lyapunov methods. The controller enables compressor operation beyond the nominal surge line. Here, we only present the second-order model and a few simulations to provide further insight into surge behavior. Assume that a flow resistance (i.e., a valve), called *load* is installed at the compressor outlet and let P denote the pressure rise between compressor inlet and outlet and f the flow rate through the system. The simplified model equations are given by

$$\dot{f} = B(\Psi(f) - P), \tag{1.11}$$

$$\dot{P} = \frac{1}{B}(f - \Phi(P)), \tag{1.12}$$

where B is a constant called the *Greitzer parameter* and $\Psi(f)$ and $\Phi(P)$ are the static pressure-flow characteristics of the compressor and load, respectively. Note that the function $\Psi(f)$ is the compressor's surge line, which may be determined experimentally. Consider $B = 0.5$ and $\Phi^{-1}(f) = \gamma f^3$. The first simulation is conducted with $\gamma = 1$ and $\Psi(f) = -3.5f^3 + 8.8f^2 - 4f + 0.8$. There is a unique equilibrium point obtained by simultaneously solving $P = \Psi(f)$ and $f = \Phi(P)$. Setting the initial conditions to unity for both variables, a stable transition to the equilibrium point is observed in Fig. 1.6. For the second simulation, the load characteristic is changed to use $\gamma = 3$, so that intersection with the surge line

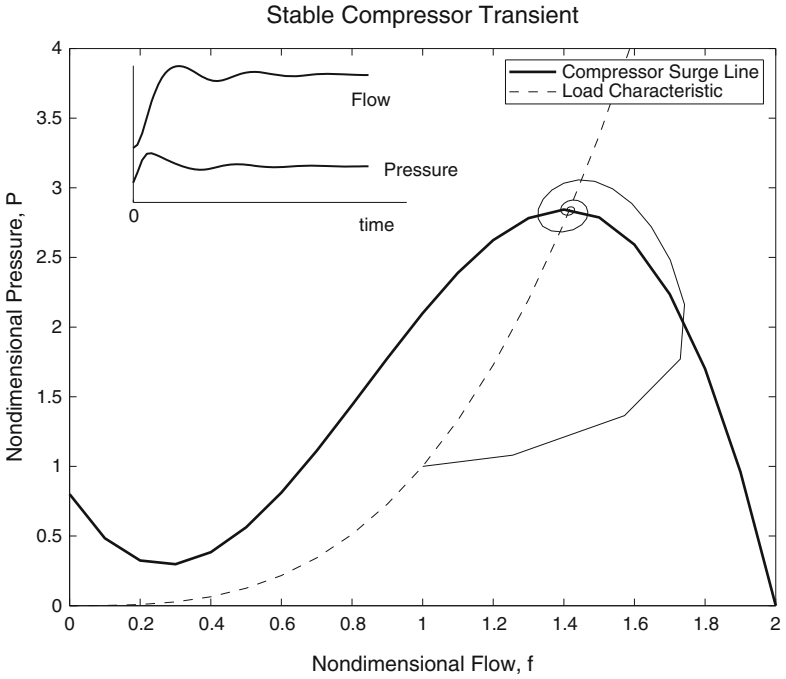

Fig. 1.6 Stable compressor transient across surge line

occurs where its slope is positive. The same initial conditions are used, however a surge cycle with intervals of reverse flow is observed in Fig. 1.7.

Although fans are generally limited to one or two stages and their blades have a somewhat different geometry, they share the same stalling and surge characteristics of compressors. Therefore, a fan map including a surge line is also furnished to the controls engineer as part of the design requirements.

1.3.2 Turbine Maps

A multistage turbine is composed of stationary inlet guide vanes followed by rotating disks. Gas flow through a turbine resembles the flow through a convergent–divergent nozzle. Thus, pressure ratio is a monotonically increasing function of flow for a range of flow values constituting the *non-choked regime*, corresponding to subsonic flow throughout the nozzle. Given a constant inlet pressure, flow increases as outlet pressure is reduced in the nonchoked regime. When the fluid at the narrowest nozzle section reaches sonic speed (unity Mach number for adiabatic, ideal gas flow), flow attains a maximum value and becomes independent of outlet pressure. This regime is said to be *choked*. Higher values of flow are obtainable only

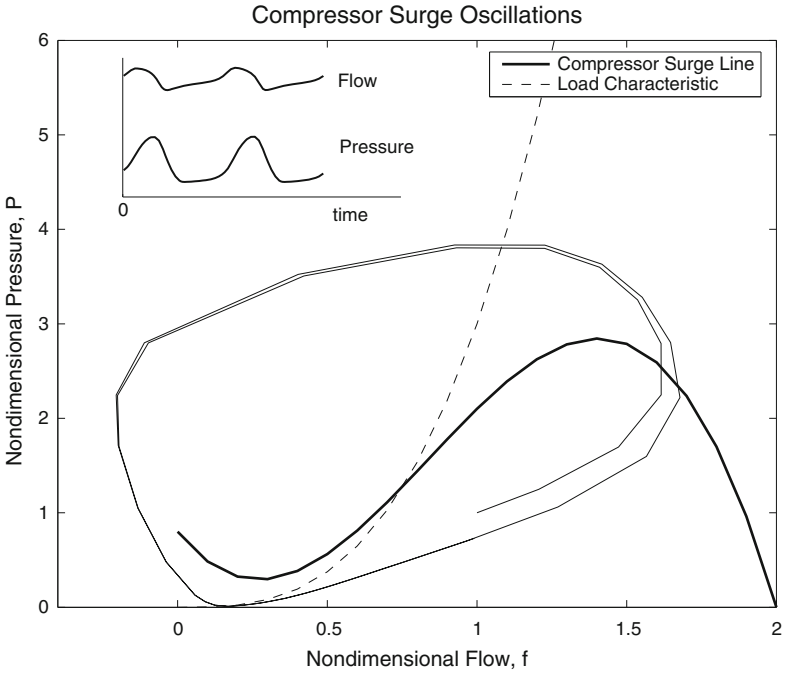

Fig. 1.7 Compressor surge cycle with reverse flow

by increasing the inlet pressure or temperature. In a GTE, turbines typically operate in the choked regime, so that maximum flow is attained (recall that engine thrust is proportional to mass flow rate). Besides, the mechanical work extracted from the turbine increases with pressure ratio, making it convenient to operate the turbine in the vertical portion of the map lines shown in Fig. 1.8.

Turbines are not prone to stall or surge, but place limits on engine operation according to their allowable blade temperatures. Although temperature limits are representable in a turbine map, these boundaries are obtained at steady-state conditions. As it will be seen later, turbine temperature transients feature pronounced overshoots which place tighter constraints on control system design than steady limits.

1.3.3 Ray Lines and Stall Margin

As evident from Fig. 1.5, if mass flow rate and pressure ratio are independently given, unique values exist for rotor speed and adiabatic efficiency. Numerical solvers used in simulation packages rely on rotor speed as one of the independent variables, however. This arises from the fact that the mechanical inertia associated

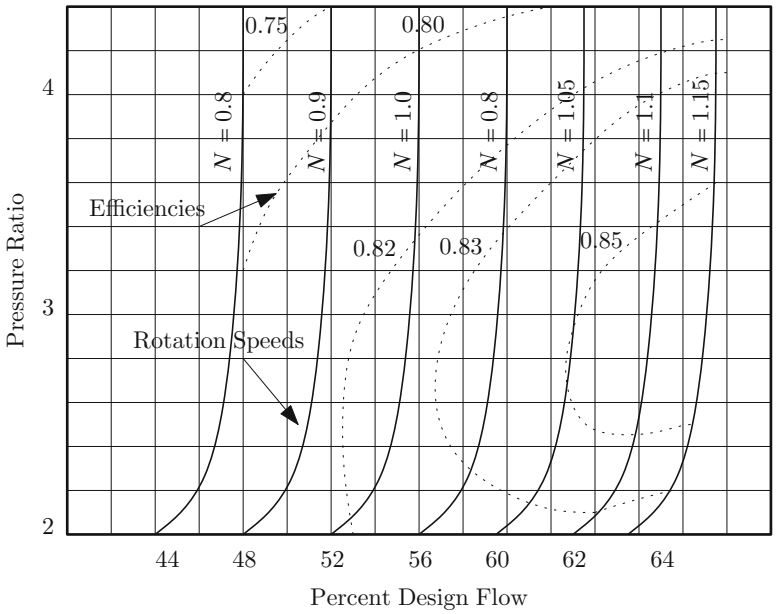

Fig. 1.8 Typical turbine map

with rotating masses constitutes the dominant mode of engine dynamics. Newton's law is readily applied to core and fan shafts, making their speeds a natural choice of state variables. Rotor speed, however, cannot be paired with mass flow rate, pressure ratio or efficiency to form a set of independent variables. For instance, Fig. 1.5 shows that more than one value of mass flow rate may correspond to a choice of pressure ratio and rotor speed. Nonuniqueness can also be observed when rotor speed is paired with efficiency. Mass flow rate cannot be determined without pressure ratio information. To circumvent these problems, a family of mass flow rate-pressure ratio loci are chosen so that a single intersection exists with each rotor speed line. Each locus is termed *ray line* or *R-line*, and is chosen arbitrarily, under the single-intersection restriction. Straight lines have been used historically, which justifies the "ray" terminology [11]. In any case, a scalar "R-function" of mass flow rate and pressure ratio is defined, its contour lines corresponding to R-lines in the compressor map. Engine simulators such as NASA's CMAPSS define the R-function so that one of its contours is the stall line itself. R-function value and rotor speed are used in solver routines to determine the remaining variables uniquely.

Given a point on the compressor or fan map defined by a pressure ratio PR and a mass flow rate $\dot{m}$, the *stall margin* is defined as

$$SM = \frac{PR_{\text{stall}}(\dot{m}) - PR}{PR}, \tag{1.13}$$

where $PR_{\text{stall}}(\dot{m})$ is the function defining the stall line. Thus, points of the stall line have $SM = 0$.

1.3.4 Combustor Instabilities and Blowout

Heat release due to combustion and acoustic wave propagation in a cavity are dynamically coupled phenomena. In GTEs, the characteristics of this interaction are modulated by the fuel flow rate imposed by the pump. Under normal operating conditions, variables such as combustor pressure and temperature exhibit stable responses to changes in fuel flow rate. Other regimes exist, however, that induce potentially destructive pressure oscillations. As described empirically by Rayleigh in 1845 [12], if perturbations are introduced to a steady flow condition so that pressure and heat release are in phase, resonant oscillations are induced.

Lean combustor blowout, or flame extinction, occurs when the air-to-fuel ratio is very small. This condition may be encountered upon sharp decelerations, when the control system commands a decrease in fuel flow rate. Aircraft descent involves engine operation near idle speeds, which corresponds to minimal fuel flow rates. The risk of engine blowout is increased during these operations, especially under inclement weather. Modern aircraft control systems allow safe engine operation near blowout limits, with improved fuel economy and reduced emissions. Much controls-oriented research has been done in support of achieving stable combustor operation in a wide range of operating conditions [13].

Each point in a compressor map corresponds to a combination of mass flow rate and pressure ratio. When considering the whole engine, a fuel flow rate is also associated with the point. The locus of compressor map points having a fuel flow rate equal to the lean blowout can be represented as an additional operating boundary to be taken into account when designing engine controllers. A simple approach often adopted in practice is to introduce absolute limits on both rate and magnitude of the commanded fuel flow rate [2].

1.3.5 Structural and Thermal Limits

Aside from limits introduced by specific engine components, mechanical and thermal limits on engine materials are imposed to reduce the rate at which wear occurs. These limits are more restrictive than those which would cause immediate component failure due to overstress or overtemperature. Vibration, which is directly linked to rotation speed, is associated with fatigue and component life. Similarly, turbine blade wear is strongly affected by temperature. Like fatigue, thermal damage is accumulative. That is, any periods of time spent by the blades at high temperature will reduce remaining life.

The locus of compressor map points having a steady turbine temperature equal to the imposed limit may also be represented as a design boundary.

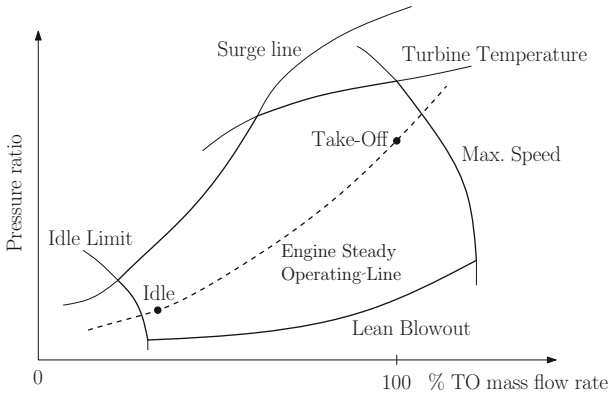

Fig. 1.9 Engine line and operating limits on compressor map

1.3.6 Overall Engine Operating Limits

Compatibility of pressure in adjacent components and core flow equality throughout the engine restrict the possible steady operating points and transient trajectories observed in a component map. Compressor maps are typically used as the common platform on which all constraints are represented. The *engine operating line* is the locus of compressor pressure ratio and mass flow rate points obtained at steady-state conditions as an engine input is varied. In single-input control systems, fuel flow rate is changed by adjusting a throttle setting. The resulting compressor pressure ratio and air mass flow rate are plotted on the compressor map once a steady-state has been reached. Figure 1.9 shows such engine operating line and various constraints represented on a compressor map.

1.4 A Note on Total and Corrected Quantities

The pressure and temperature attained by a gas when its velocity is reduced to zero following an isentropic process are called *total pressure* and *total temperature*. The point at which the fluid has zero velocity is called *stagnation point*. Thus, any sensor designed to measure pressure or temperature at stagnation will indicate total quantities. If an ideal gas has velocity V, pressure P and temperature T, there exist unique values for the corresponding total quantities. The Mach number M is defined as the ratio of flow velocity to the speed of sound at the same temperature:

$$M = \frac{V}{\sqrt{\gamma R T}}, \tag{1.14}$$

where $\gamma = \frac{C_p}{C_v}$ is the ratio of specific heats at constant pressure and constant volume ($\gamma = 1.4$ for air) and R is the universal gas constant ($R = 286$ J/kg-° K).

The ratios of total to absolute quantities are derived using conservation laws and basic thermodynamic identities [1, 14]:

$$\frac{T_\mathrm{t}}{T} = 1 + \frac{\gamma - 1}{2} M^2, \tag{1.15}$$

$$\frac{P_\mathrm{t}}{P} = \left(1 + \frac{\gamma - 1}{2} M^2\right)^{\frac{1-\gamma}{\gamma}}. \tag{1.16}$$

In addition to their relevance to sensing installations in aircraft, total quantities appear repeatedly in many thermodynamic formulas and are frequently used to define dimensionless parameters. The mass flow through a duct of area A is given by

$$\dot{m} = \rho A V, \tag{1.17}$$

where ρ is the density. Using the ideal gas law $\rho R T = P$ to eliminate density from (1.17) and introducing total quantities from (1.15) and (1.16), the following expression is obtained for $\dot{m}$:

$$\dot{m} = \psi A \frac{\sqrt{T_\mathrm{t}}}{P_\mathrm{t}}, \tag{1.18}$$

where the *mass flow parameter* ψ has been defined as

$$\psi = \sqrt{\frac{\gamma}{R}} M \left(1 + \frac{\gamma - 1}{2} M^2\right)^{-\frac{\gamma+1}{2(\gamma-1)}}. \tag{1.19}$$

The usefulness of the mass flow parameter definition is clear from its dependence of Mach number only. Equation (1.19) shows that, given fixed P_t and T_t, the maximum mass flow rate per unit area is obtained at $M = 1$ (choked flow condition).

Nondimensional forms of the total pressure and temperature at an engine station are usually obtained by normalization relative to the standard atmospheric properties ($P_\mathrm{std} = 101.3$ kPa and $T_\mathrm{std} = 288.15°$ K):

$$\delta = \frac{P_\mathrm{t}}{P_\mathrm{std}}, \tag{1.20}$$

$$\theta = \frac{T_\mathrm{t}}{T_\mathrm{std}}. \tag{1.21}$$

A normalized – albeit dimensional – form of the mass flow rate of air through the engine frequently used in the aerospace industry is the *corrected mass flow rate*, defined as

$$\dot{m}_\mathrm{cr} = \dot{m} \frac{\sqrt{\theta}}{\delta} = \frac{P_\mathrm{std} A}{\sqrt{T_\mathrm{std}}} \psi. \tag{1.22}$$

Therefore, $\dot{m}_{\text{cr}}$ is the mass flow rate that would be observed if an engine initially operating with Mach number M and atmospheric conditions given by P and T is now operated at sea-level with the same value of M.

The *corrected speed* for an engine component rotating at angular rate N is defined as

$$N_{\text{cr}} = \frac{N}{\sqrt{\theta}}. \tag{1.23}$$

The motivation behind this definition is that the temperature change across a fan or compressor is proportional to the square of the ideal gas velocity. It can be shown [14] that the ratio of a component's rotational speed to the total nondimensional temperature at its inlet is essentially constant hence, N_{cr} is constant and allows to find values of N at any flight condition defined by θ, once as reference has been established.

1.4.1 Summary

Various constraints associated with safe and efficient engine operation have been introduced in this chapter. A significant portion of this book is devoted to constraint handling using the tools of control systems theory. All of the variables which are subject to a constraint will be presented as outputs of a state-space model. In a linearized model, for instance, turbine temperature, compressor stall margin, fan speed, and other variables will correspond to transfer functions from fuel flow rate, all stemming from the same A and B matrices of a state-space system. Established approaches, such as the industry-standard fan speed regulation with max–min limit protection logic will be examined under this framework. Recent advances, including model predictive control and sliding mode control applied to aircraft engines, will also be introduced and evaluated under the constrained-output framework.

Chapter 2
Engine Models and Simulation Tools

Abstract This chapter offers a brief overview of engine dynamics, aiming at the extraction of linearized models that can be used as a basis for design. The Commercial Modular Aeropropulsion System Simulation (CMAPSS) package developed by the NASA Glenn Research Center is also described.

This chapter outlines the methodologies commonly followed to arrive at dynamic models, which are suitable for control design. A public-domain simulation package developed by NASA is also described, which will be used for many simulation studies contained in this book.

2.1 Two-Spool Shaft Dynamics

Mechanical system dynamics due to rotating inertias constitute the most important contribution to engine transient behavior. In fact, shaft speeds are directly linked with mass flow through the engine and thrust, which is the main output to be manipulated by the propulsion control system. Additional dynamics due to gas mass storage and heat transfer between gas and metal are present, but their use is reserved to high-fidelity, detailed models. Likewise, the dynamics of actuator systems are significantly faster than those associated with rotating masses, and are usually left unmodeled. As it will be seen next, *outputs of interest such as turbine temperatures, pressure ratios and stall margins can be regarded as outputs of a dynamic model whose states are shaft speeds*.

In what follows, we assume the engine to be a two-spool turbofan, where N_f denotes the angular speed of the assembly formed by fan, LPC, and LPT, and N_c denotes the angular speed of the HPC-HPT shaft assembly (see Fig. 1.3). For consistency with the standard terminology and simulation software, N_f and N_c will be denoted *fan speed* and *core speed*, respectively. Newton's law for rotating masses is applied to each shaft as follows:

$$\dot{N}_\mathrm{f} = f_1(N_\mathrm{f}, N_\mathrm{c}, u, w), \tag{2.1}$$

$$\dot{N}_\mathrm{c} = f_2(N_\mathrm{f}, N_\mathrm{c}, u, w). \tag{2.2}$$

H. Richter, *Advanced Control of Turbofan Engines*, DOI 10.1007/978-1-4614-1171-0_2,

Here, f_1 and f_2 are the net torques delivered by the LPT and HPT, respectively, normalized by the mass moment of inertia of the shaft assemblies. Vector u contains the control input components. In a single-input control system, u is given by the fuel flow rate W_F. Additional control effectors may be enabled as actuators; for instance, VBV and VSV may be included as components of u in addition to W_F. These additional inputs may be changed in a closed-loop or in an open-loop fashion. In the latter case, VBV and VSV are scheduled, that is, calculated as functions of engine variables other than states or inputs. Special attention must be placed on whether these effectors are held constant – thus becoming parameters –, scheduled, or used as control inputs. These choices have important implications in subsequent linearization procedures and controller designs. Vector w is used to capture the effects of uncertain inputs such as disturbances. An important interpretation of w is given by the *health parameter inputs*, a set of quantities representing engine deterioration and faults. Health parameters are discussed in Sect. 2.1.3.

Engine dynamics arise from complex, interacting phenomena: gas-flow behavior in the compressor and turbine (affected by air inlet as well as engine conditions), shaft inertias and losses, fuel flow transport delay, combustion and the thermal behavior of the engine and its surroundings. Due to the intricate geometry of the engine components and the complexity of gas flow, algebraic expressions for f_1 and f_2 are unavailable. Even the more sophisticated model generation and simulation tools [2] resort to linearization of (2.1) and (2.2) as a basis to include information about f_1 and f_2 through their partial derivatives. Note that these functions are highly dependent on external variables such as aircraft speed and atmospheric conditions, which act as their parameters.

When constant values of u and w are applied, along with a fixed set of parameters for f_1 and f_2, the engine reaches a steady–state operating point, with corresponding constant values of core and shaft speeds. Small-signal linearization is performed at these conditions, yielding the model:

$$\begin{aligned}\Delta\dot{N}_f = &\left.\frac{\partial f_1}{\partial N_f}\right|_o \Delta N_f + \left.\frac{\partial f_1}{\partial N_c}\right|_o \Delta N_c + \left.\frac{\partial f_1}{\partial u_1}\right|_o \Delta u_1 + \left.\frac{\partial f_1}{\partial u_2}\right|_o \Delta u_2 + ... \\ &.... + \left.\frac{\partial f_1}{\partial w_1}\right|_o \Delta w_1 + \left.\frac{\partial f_1}{\partial w_2}\right|_o \Delta w_2 + ...,\end{aligned} \tag{2.3}$$

$$\begin{aligned}\Delta\dot{N}_c = &\left.\frac{\partial f_2}{\partial N_f}\right|_o \Delta N_f + \left.\frac{\partial f_2}{\partial N_c}\right|_o \Delta N_c + \left.\frac{\partial f_2}{\partial u_1}\right|_o \Delta u_1 + \left.\frac{\partial f_2}{\partial u_2}\right|_o \Delta u_2 + ... \\ &.... + \left.\frac{\partial f_2}{\partial w_1}\right|_o \Delta w_1 + \left.\frac{\partial f_2}{\partial w_2}\right|_o \Delta w_2 + ...,\end{aligned} \tag{2.4}$$

where the subscript $_o$ has been used to indicate that the partial derivatives are evaluated at steady-state conditions. The steady values of N_f and N_c corresponding to constant u, w and parameters are found by iterative procedures attempting to equate the right-hand sides of (2.1) and (2.2) to zero. The process of finding the steady

speeds and outputs corresponding to a set of steady inputs and parameters is carried out by an *engine model balancer*, or *steady-state solver*. The implicit relationship between constant inputs and steady outputs is called a *steady engine map*.

2.1.1 Model Construction from Cycle Deck Data

The partial derivatives of f_f and f_2 are obtained from a perturbation method based on experimental data. Engine manufacturers are able to operate the machine in a test stand, simulating multiple combinations of atmospheric conditions and airspeeds, in addition to any desired input values. Data is collected from a host of sensors, from which relevant quantities such as adiabatic efficiencies and stall margins are computed. To obtain an approximate value of a partial derivative, the dependent variable is slightly perturbed from its steady value, and the corresponding increment in net torque recorded. Such sensitivity of the net torque to perturbations in the dependent variable is the desired value for the partial derivative.

By repeating this procedure in a grid of values for u and function parameters, local gradient information for f_1 and f_2 is collected. While a controls-oriented model can be as simple as a single linear model corresponding to a point in the grid, a high-fidelity simulation model uses partial derivative information obtained from a fine grid covering the entire flight envelope. Such models "piece together" this massive information, enabling numerical integration of trajectories across a wide range of operating conditions.

Outputs of the form $y_i = y_i(N_\mathrm{f}, N_\mathrm{c}, u, w)$ are also linearized to yield

$$\Delta y_i = \left.\frac{\partial y_i}{\partial N_\mathrm{f}}\right|_o \Delta N_\mathrm{f} + \left.\frac{\partial y_i}{\partial N_\mathrm{c}}\right|_o \Delta N_\mathrm{c} + \left.\frac{\partial y_i}{\partial u_1}\right|_o \Delta u_1 + \left.\frac{\partial y_i}{\partial u_2}\right|_o \Delta u_2 + ... \\ .... + \left.\frac{\partial y_i}{\partial w_1}\right|_o \Delta w_1 + \left.\frac{\partial y_i}{\partial w_2}\right|_o \Delta w_2 + ... \tag{2.5}$$

It is worth emphasizing that engine dynamic models are available to the controls engineer only as a collection of linear models, each one valid in a neighborhood of an equilibrium point and corresponding to fixed set of parameters reflecting flight conditions.

These models are expressible in the standard state-space form:

$$\dot{x} = Ax + Bu + \Gamma w, \tag{2.6}$$

$$y = Cx + Du + \Lambda w, \tag{2.7}$$

where $x^\mathrm{T} = [\Delta N_\mathrm{f}\ \Delta N_\mathrm{c}]$, $y^\mathrm{T} = [\Delta y_1\ \Delta y_2....]$ and matrices A, B, C, D, Γ, and Λ contain the partial derivatives as follows:

$$A = \begin{bmatrix} \left.\frac{\partial f_1}{\partial N_f}\right|_o & \left.\frac{\partial f_1}{\partial N_c}\right|_o \\ \left.\frac{\partial f_2}{\partial N_f}\right|_o & \left.\frac{\partial f_2}{\partial N_c}\right|_o \end{bmatrix}, \qquad B = \begin{bmatrix} \left.\frac{\partial f_1}{\partial u_1}\right|_o & \left.\frac{\partial f_1}{\partial u_2}\right|_o & \cdots \\ \left.\frac{\partial f_2}{\partial u_1}\right|_o & \left.\frac{\partial f_2}{\partial u_2}\right|_o & \cdots \end{bmatrix},$$

$$C = \begin{bmatrix} \left.\frac{\partial y_1}{\partial N_f}\right|_o & \left.\frac{\partial y_1}{\partial N_c}\right|_o & \cdots \\ \left.\frac{\partial y_2}{\partial N_f}\right|_o & \left.\frac{\partial y_2}{\partial N_c}\right|_o & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix}, \qquad D = \begin{bmatrix} \left.\frac{\partial y_1}{\partial u_1}\right|_o & \left.\frac{\partial y_1}{\partial u_2}\right|_o & \cdots \\ \left.\frac{\partial y_2}{\partial u_1}\right|_o & \left.\frac{\partial y_2}{\partial u_2}\right|_o & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix},$$

$$\Gamma = \begin{bmatrix} \left.\frac{\partial f_1}{\partial w_1}\right|_o & \left.\frac{\partial f_1}{\partial w_2}\right|_o & \cdots \\ \left.\frac{\partial f_2}{\partial w_1}\right|_o & \left.\frac{\partial f_2}{\partial w_2}\right|_o & \cdots \end{bmatrix}, \qquad \Lambda = \begin{bmatrix} \left.\frac{\partial y_1}{\partial w_1}\right|_o & \left.\frac{\partial y_1}{\partial w_2}\right|_o & \cdots \\ \left.\frac{\partial y_2}{\partial w_1}\right|_o & \left.\frac{\partial y_2}{\partial w_2}\right|_o & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix}.$$

2.1.2 Models from System Identification

System identification techniques – parametric and spectral – are also applicable to GTE model generation. These techniques are more formal and systematic than the empirical partial derivative evaluations described above. Also, measurement noise and its impact on model quality are addressed explicitly. In a parametric system identification approach, a model structure is first selected where system order and representation (state-space, transfer function, etc.) are defined. Then parameters are estimated from experimental data using a numerical optimization algorithm. In a nonparametric (spectral) model, the objective is to obtain numerical frequency response traces (Bode plots) using input/output experimental information. A parametric model may also be fitted to the frequency domain traces by least squares or any other suitable curve fitting procedure.

System identification techniques have several drawbacks, including the need for time-consuming trial-and-error procedures to arrive at the proper model structure, the influence of the chosen test input on the results and the need for long data records [15]. For comprehensive information on system identification, the reader is referred to classical works such as Ljung [16] or Sage and Melsa [17]. For a detailed treatment of these methods as applicable to GTEs, refer to Kulikov and Thompson [18].

2.1.3 Engine Aging and Deterioration Modeling

The response of engine states and outputs to actuator inputs changes with time, according to the engine's "age", typically measured in hours of operation. Mechanical

wear due to normal use is reflected as changes in internal flow characteristics and component efficiencies. Component damage and engine-to-engine manufacturing variations also result in significant changes to engine response. Although aging and deterioration effects are distributed through the engine, a finite set of *engine health parameters* is chosen that describe changes in specific engine components. For instance, a normalized health parameter defined for the turbine efficiency may vary between 0 and 1, with 0 representing a new turbine and 1 a "fully deteriorated" one. The maximum level of deterioration is chosen by the modeler to indicate that the component is no longer usable and that an engine overhaul is necessary. Health parameter changes are represented as a bounded disturbance input vector w in the linearized engine model of (2.6) and (2.7). Note that the same set of health parameter inputs may be used to represent both slow aging and sudden fault, the distinction being the particular function of time used in simulation studies.

Health parameters are used in several ways, ranging from condition monitoring and fault detection to active control. In the latter case, designers may resort to robustness properties or may rely on disturbance estimation techniques. Condition monitoring and fault detection techniques require reliable parameter estimates during engine operation. Health parameter estimation has been the subject of much research and continues to be an active area of interest. The interested reader is referred to the comparative study by Volponi et al. [19] covering Kalman filter and soft computing estimation techniques. Further details on related problems and specific applications of these techniques are found, for instance, in [20–22] and references therein.

This book does not cover health parameter estimation techniques or their use in fault detection and accommodation. However, control strategies will be studied in detail which strive to preserve engine performance in the face of such uncertain inputs. Some robust strategies incorporate a disturbance estimator, while others rely entirely on robustness and disturbance bounds. Accounting for engine model variations is crucial in propulsion control system design. Besides the need for consistent thrust response throughout engine life, health parameter changes cause the safety limits described in Sect. 1.3 to shift, reducing operability margins. For instance, compressor deterioration – as captured by a corresponding parameter – shifts the stall line in a direction that reduces the stall margin.

2.2 Commercial Modular Aero-Propulsion System Simulation

Most simulations presented in this book were generated using the state-of-the-art Commercial Modular Aeropropulsion System Simulation, or Commercial Modular Aero-Propulsion System Simulation (CMAPSS). This package was developed at the NASA Glenn Research Center and is intended for public distribution [2]. CMAPSS is a Simulink port and a database with a user-friendly graphical user interface (GUI) allowing the user to perform model extraction, elementary control design, and simulations without much effort. One version of CMAPSS contains a model

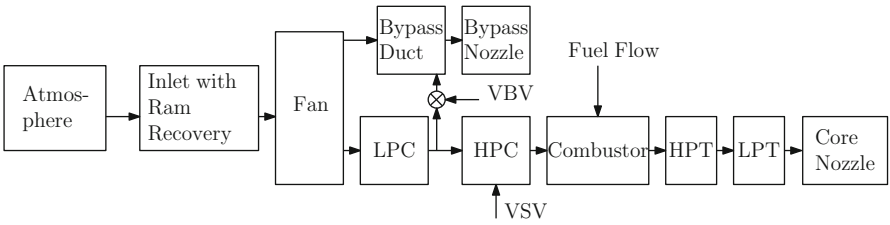

Fig. 2.1 CMAPSS software modules

corresponding to a large, high-bypass ratio turbofan engine similar to the GE90. Indeed, the CMAPSS-1 engine produces about 90,000 lbs of thrust at take-off conditions. A separate version, CMAPSS-40k [23], contains a model representing an engine in the 40,000-lb thrust class.

In this section, the main features of CMAPSS are described, along with a representative session including model linearization, basic controller specification, and simulation functions.

2.2.1 CMAPSS Main Features

The engine model is composed of several cascading modules, as shown in Fig. 2.1. The atmospheric model comprises air properties spanning altitudes from 0 to 40,000 ft above sea-level, Mach numbers from 0 to 0.9, and sea-level temperatures between −60 and 103°F.

2.2.1.1 Model Inputs

Model inputs are divided into two categories: control inputs and health parameter inputs. All versions of CMAPSS allow direct manipulation of fuel flow, with total freedom for the implementation of feedback laws. Health parameter inputs may only be specified as step functions of time. Simple modifications to the supplied Simulink model can be introduced to allow for more general health parameter variations. Depending on version, VSV and VBV may be accessible as control inputs or may be subjected to scheduling functions hidden from the user. CMAPSS-1 includes 14 inputs, fuel flow rate being the first. The remaining 13 inputs constitute the set of health parameters associated with pressure, flow, and efficiency characteristics of the fan, LPC, LPT, HPC, and HPT. VSV and VBV are not available as control inputs in this version. CMAPSS-40k includes 16 inputs: W_F, VSV, VBV, and 13 health parameters. The complete list of inputs in CMAPSS is shown in Table 2.1. Each time linearization is performed in CMAPSS-1, a B matrix with 2 rows and 14 columns is obtained. In CMAPSS-40k, B has 2 rows and 16 columns.

Table 2.1 CMAPSS engine model inputs (adapted from [2, 23])

Index	Input
1	Fuel flow (pps)
2	Variable stator vane (degrees)
3	Variable bleed valve (degrees)
4	Fan efficiency modifier
5	Fan flow modifier
6	Fan pressure-ratio modifier
7	LPC efficiency modifier
8	LPC flow modifier
9	LPC pressure-ratio modifier
10	HPC efficiency modifier
11	HPC flow modifier
12	HPC pressure-ratio modifier
13	HPT efficiency modifier
14	HPT flow modifier
15	LPT efficiency modifier
16	LPT flow modifier

2.2.1.2 Pilot Commands: TRA and PLA

The *throttle resolver angle*, or TRA, is the angular deflection of the pilot's power lever, having a range from 0 to 100%. In closed-loop operation, it is desirable to obtain an approximately linear steady-state relationship between TRA setting and net thrust across a wide range of inlet conditions. As elaborated in Sect. 3.1, thrust control is achieved indirectly by controlling fan speed. This is because no real-time thrust sensing is available in aircraft systems. For this reason, TRA is mapped into a corrected fan speed demand through a static function. Actual fan speed demand is then calculated using the inlet conditions reflected in θ. In control systems terminology, the TRA-to-$N_{f,cr}$ demand mapping constitutes a static reference prefilter placed outside the fan speed feedback loop. The subsequent conversion to N_{f} demand involves inlet variables which can be regarded as slowly-varying relative to the characteristic times of engine dynamics and its control system. Therefore, propulsion control design may proceed by assuming that a fan speed demand is supplied as an independent reference input. The *power lever angle*, or PLA, is an alternative designation for the same pilot control.

2.2.1.3 Model Outputs

Twenty-seven outputs are included in linearization models obtained via the GUI. These outputs, in addition to many other auxiliary quantities, are written to the workspace when a simulation is carried out. The 27 model outputs of CMAPSS-1 are listed in Table 2.2, and the reader is referred to the User's Guides [2, 23] for a complete listing of additional variables. The C and D matrices obtained through

Table 2.2 CMAPSS-1 engine model outputs (adapted from [2])

Index	Output	Units
1	Fan speed, N_f	rpm
2	Core speed, N_c	rpm
3	Engine pressure ratio, EPR=$\frac{P_{50}}{P_2}$	–
4	Total pressure at fan outlet, P_{21}	psia
5	Total temperature at fan outlet, T_{21}	°R
6	Total pressure at LPC outlet, P_{24}	psia
7	Total temperature at LPC outlet, T_{24}	°R
8	Total pressure at HPC outlet, P_{30}	psia
9	Total temperature at HPC outlet, T_{30}	°R
10	Total pressure at burner outlet, P_{40}	psia
11	Total temperature at burner outlet, T_{40}	°R
12	Total pressure at HPT outlet, P_{45}	psia
13	Total temperature at HPT outlet, T_{48}	°R
14	Total pressure at LPT outlet, P_{50}	psia
15	Total temperature at LPT outlet, T_{50}	°R
16	Fan massflow, W_{21}	pps
17	Net thrust, F_n	lbf
18	Gross thrust, F_g	lbf
19	Fan stall margin	–
20	LPC stall margin	–
21	HPC stall margin	–
22	Corrected fan speed	rpm
23	Corrected core speed	rpm
24	Total bypass duct pressure, P_{15}	psia
25	Percent corrected fan speed	–
26	Static pressure at HPC outlet, P_{s30}	psia
27	Ratio, $\frac{W_f}{P_{s30}}$	pps/psi

linearization in CMAPSS-1 have dimensions 27 by 2 and 27 by 14, respectively. Naturally, most of these outputs serve only a monitoring purpose, and only a few have a corresponding sensor in the real engine.

2.2.1.4 Pre-defined Flight Condition Data

Given a set of fixed inlet conditions and health parameters, each steady value of fuel flow rate W_F corresponds to a unique equilibrium point, defined by steady fan and core speeds. Also, each one of the 27 outputs adopts a corresponding steady value. Altitude, Mach number, and sea-level temperature completely define atmospheric conditions. These three quantities can be regarded as parameters of f_1, f_2 and y_i in (2.1), (2.2), and (2.5), and must be specified as part of the linearization process. The set of inlet conditions, steady fuel flow and corresponding steady states and outputs constitutes a *flight condition*. CMAPSS-1 is distributed with a set of predefined flight condition files.

Table 2.3 Equilibrium point data for representative flight conditions from CMAPSS-1 (adapted from [2]). All health parameter inputs are zero

	FC01	FC05	FC06	FC07	FC08	FC09
Alt, ft	0.00	10000.00	20000.00	25000.00	35000.00	42000.00
Mach	0.00	0.25	0.70	0.62	0.84	0.84
TRA, °	100.00	100.00	100.00	60.00	100.00	100.00
W_F, pps	6.84	4.66	3.86	1.67	2.12	1.52
N_f, rpm	2388.00	2319.00	2324.00	1915.00	2223.00	2212.00
N_c, rpm	9051.00	8774.00	8719.00	8006.00	8346.00	8317.00
EPR	1.30	1.26	1.08	0.94	1.02	1.02
T_{48}, °R	2072.00	1947.00	1909.00	1534.00	1750.00	1744.00
Ps30, psia	522.13	371.76	206.76	163.94	183.10	130.51
LPC Rline	1.64	1.63	2.31	1.70	1.52	1.54
HPC Rline	1.95	1.96	1.98	2.03	2.00	2.03
F_n, lbf	86636.00	45830.00	25774.00	11475.00	13552.00	9647.00

Arbitrary flight conditions may be calculated in CMAPSS by specifying atmospheric conditions and fuel flow, followed by simulation under any stabilizing controller. In CMAPSS-1, the results depend on the particular scheduling functions used for VSV and VBV. CMAPSS-40k treats these actuators as control inputs, thus two constant values must be selected along with a steady fuel flow value to define the point of linearization completely. CMAPSS includes a steady-state solver, which accepts steady input values and inlet conditions to find the steady values of the two states and all outputs.

A set of predefined flight conditions is included in the CMAPSS-1 distributions. These conditions may be invoked from the GUI as the first step in creating linearized models and designing elementary controllers. Six representative flight conditions extracted from CMAPSS-1 have been summarized in Table 2.3. Appendix B contains tables for the linearized model matrices at each one of the six representative flight conditions. This information is provided as an alternative for readers not using CMAPSS. No predefined flight condition data is distributed with CMAPSS-40k, however, several typical operating regimes are presented here for future reference. Six regimes representing several stages of flight are summarized in Table 2.4. The component maps introduced in Sect. 1.3 are customarily presented in the industry in terms of corrected mass flow and speeds rather than absolute quantities. CMAPSS also uses these quantities for internal calculations purposes, as well as for displaying performance evaluation plots. The controls engineer should regard corrected quantities as aids in understanding the effects of parametric changes to the model introduced by flight condition variations.

2.2.2 Example

A sample session with CMAPSS-1 is described as a quick start-up guide for interested readers. Matlab and Simulink must be available on a Windows platform,

Table 2.4 Equilibrium point data for representative flight conditions from CMAPSS-40k. All health parameter inputs are zero

	A	B	C	D	E	F
	Ground idle	Flight idle	Approach	Max cruise	Max climb	Max T.O.
Alt, ft	1000.00	6000.00	10000.00	38000.00	6000.00	1000.00
Mach	0.10	0.50	0.50	0.75	0.40	0.15
PLA, °	40.00	44.00	52.00	68.00	73.00	78.00
W_F, pps	0.33	0.59	1.03	1.00	2.69	3.02
VSV, °	−51.40	−39.90	−23.97	−1.43	−5.96	−4.88
VBV, %	1.00	0.90	0.50	0.01	0.04	0.00
N_f, rpm	1375.77	2240.21	2987.78	3764.46	3814.22	3803.43
N_c, rpm	8624.00	9515.11	10279.50	10932.01	11440.57	11525.14
EPR	1.03	0.94	1.06	1.53	1.44	1.49
T_{48}, °R	1091.13	1211.32	1417.84	1682.52	1839.91	1877.65
SmHPC, %	37.48	38.30	33.75	19.90	22.94	22.01
F_n, lbf	1879.92	2596.65	6620.98	6658.35	20521.32	28829.92

since dynamically linked libraries (DLL files) used in Matlab S-functions were compiled for this operating system. Note, however, that the source C code for these libraries is distributed with CMAPSS for the advanced user to recompile to match his/her own architecture.

CMAPSS is distributed as a zipped file, which must be expanded in a directory where the user has write permissions. Upon doing this, the file `setup_everything.m` must be run to load basic data and set up the necessary paths.

A main dialog is brought up from which the user can choose to create a linearized model, design controllers or simulate the closed-loop engine, as shown in the diagram of Fig. 2.2.

In this example, we first obtain a linearized model at FC01 by following the GUI menus. The Simulink diagram used to inject perturbations and obtain the partial derivative information described in Sect. 2.1.1 is brought up and ran automatically. At the end of the run, the variable `SSeng_14x27_unsc` appears in the workspace, containing a state-space system object with 14 inputs and 27 outputs, without scaling. The first input corresponds to the incremental fuel flow ΔW_F with respect to the steady value defined by the flight condition, while the remaining 13 are health parameter inputs (the reader may think of them as parametric disturbances). The first two outputs are simply the two incremental states: ΔN_f and ΔN_c. To extract the state-space matrices from `SSeng_14x27_unsc`, we use the command `[A,B,C,D]=ssdata(SSeng_14x27_unsc)`. The 2x2 system matrix A is found as

```
>> A
A =
   -3.8557     1.4467
   0.46897    -4.7081.
```

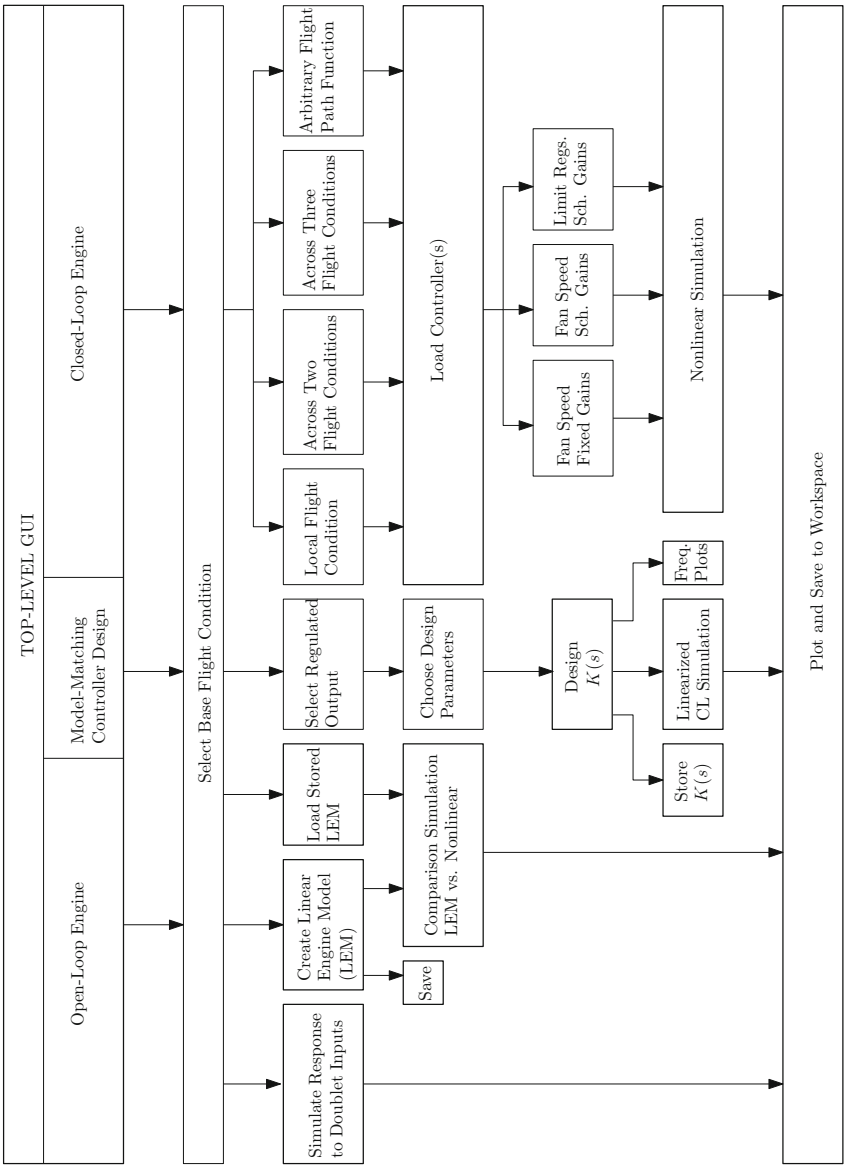

Fig. 2.2 CMAPSS functions diagram

Suppose an SISO model is desired having ΔW_f as input and ΔT_{48} as output. The following sequence of Matlab commands will store the desired model as `LEM001`:

```
>> B=B(:,1)
B =
    230.67
    653.55

>> C=C(13,:)
C =
    -0.057313  -0.32243

>> D=D(13,1)
D =
    146.37

>> LEM001=ss(A,B,C,D);
```

Thus, the transfer functions from ΔW_F to *any* of the 27 outputs at FC01 share the eigenvalues of A as poles: -3.3544 and -5.2093. In particular, the transfer function from ΔW_f to ΔT_{48} is

```
>> zpk(LEM001)
Zero/pole/gain: 146.37 (s+4.733) (s+2.301)
---------------------------
     (s+3.3544) (s+5.209)
```

Next, suppose a basic controller is to be tried for a fan speed set point change near FC01. The GUI design tool is limited to one form of compensation, consisting of lead-lag action with integration at plant input. The compensator pole is selected arbitrarily in the GUI, while the gain and zero are found from a least-squares algorithm referred to as "model-matching method". The optimization objective is to match a second-order closed-loop transfer function specified by the user by natural frequency and damping ratio. More details about the model-matching method and its limitations are given in Sect. 3.1.4. Letting the compensator pole to be -20 and choosing a closed-loop frequency of 4 rad/s and a damping ratio of 0.7, the following compensator is returned in variable `SSreg_unsc`:

```
>> zpk(SSreg_unsc)

Zero/pole/gain: 0.1122 (s+4.814)
---------------------------
     (s+20)
```

An incremental, linearized simulation may be launched from GUI for preliminary response evaluation. A Simulink diagram is brought up corresponding to this case. The default setpoint for ΔN_f may be readily changed by the user. Although the Simulink model uses scaled variables, the GUI plotting functions present the data in

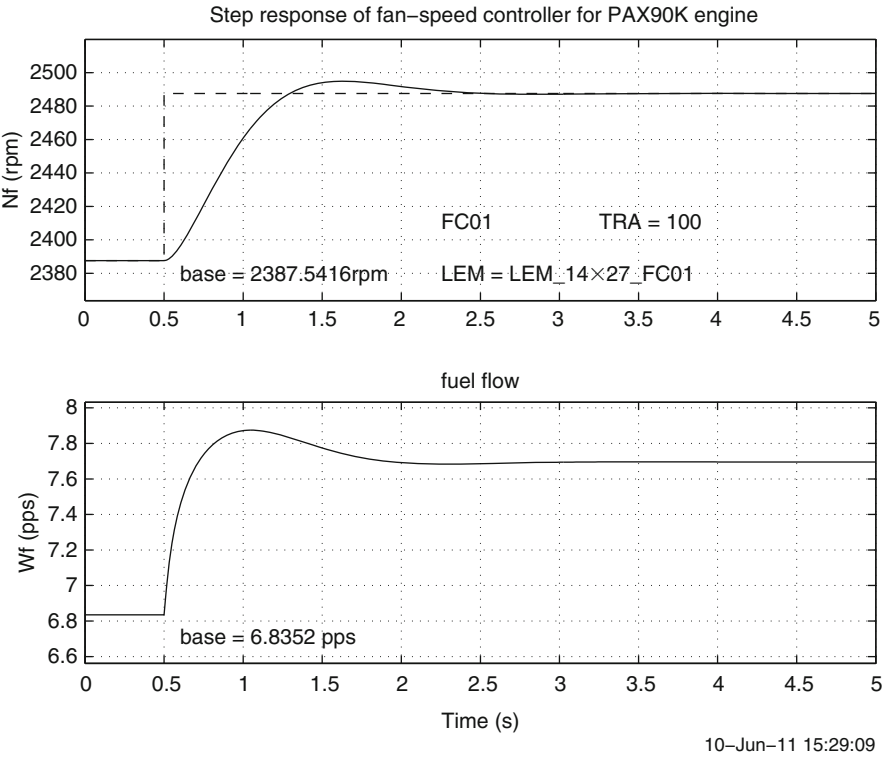

Fig. 2.3 Linearized simulation results produced by CMAPSS-1

absolute terms, as shown in Fig. 2.3. Now suppose the designed compensator is to be tested against the full nonlinear engine model, operating with a single fan speed control loop. Assuming `SSreg_unsc` still exists in the workspace, GUI dialogs may be followed to reselect FC01 as flight condition and build the closed-loop system. A Simulink diagram is brought up that contains the complete engine model and controller sections. Note that the setpoint reference is not directly given in terms of target fan speed or target incremental fan speed, but rather in terms of TRA, as described in Sect. 2.2.1.

The actual fan speed reference input can be accessed after running the simulation through variable `Nf_dmd`. In this example, a ramp input to reduce fan speed from 2387.5 to 2224.2 rpm with a slope of −300 rpm/s is used as a default. This rate and a positive rate limit of 500 rpm/s arise from a block in the Simulink diagram. The user may set these limits to `-inf` and `inf` to generate a step response. Upon doing this, various outputs of interest may be plotted using the GUI as shown in Fig. 2.4. Many other variables not accessible from the GUI are also written to the workspace.

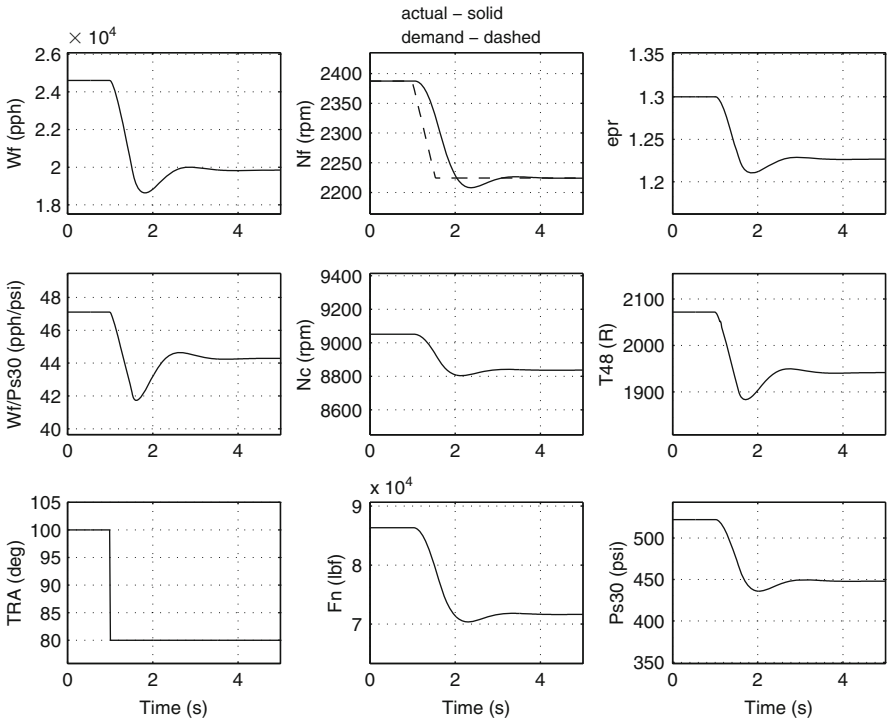

Fig. 2.4 Nonlinear engine simulation results panel produced by CMAPSS-1

2.2.2.1 Actuator Models (CMAPSS-40k)

CMAPSS-40k includes models for the three main actuators and their local control loops. Feedback is used for the fuel metering valve (FMV) and the VSV actuator, while VBV is operated in open-loop. The electrically-actuated FMV is placed around a proportional control loop to produce the W_F demanded by the engine controller. Since the valve opening dynamics from actuator current to position include a free integrator, P control is sufficient to achieve zero steady-state error (see Sect. 3.1.1) if the valve operates in a linear regime. Backlash, saturation, rate limits, and delay are present, however, as shown in Fig. 2.5. Note also that valve position – rather than actual flow – is fed back, implying that accurate calibration data must be available. If the nonlinearities and the delay are ignored, the closed-loop transfer function from desired to actual flow rate is given by

$$T_{\mathrm{FMV}}(s) = \frac{350}{s^2 + 40s + 350}$$

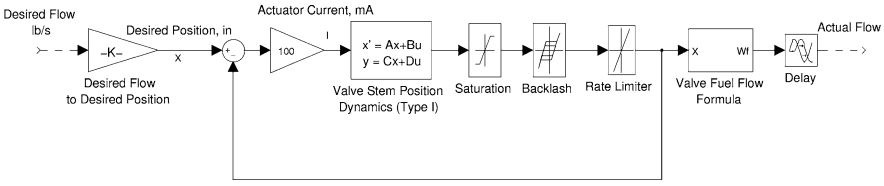

Fig. 2.5 Local control loop for the fuel metering valve (CMAPSS-40k)

with poles at -27.07 and -12.93. The VSV actuator is placed around a proportional-integral loop, which leads to the linear closed-loop transfer function:

$$T_{\mathrm{VSV}}(s) = \frac{3.75}{s + 3.75},$$

where an approximate pole-zero cancelation has been carried out. Indeed, the PI controller introduces a zero designed to nearly cancel a closed-loop pole near 0. The VBV actuator is operated in open-loop, with transfer function

$$T_{\mathrm{VBV}}(s) = \frac{23}{s + 23}.$$

An examination of Appendix C reveals that the A matrix associated with linearized engine dynamics has eigenvalues with real parts ranging from -0.75 to -4, according to flight condition. These values indicate that open-loop engine dynamics are significantly slower than those associated with the FMV and VBV actuators, but about as fast as the dynamics of the VSV actuator. The designer must decide whether to include actuator dynamics or to leave them as unmodeled dynamics and rely on the robustness properties of his/her designs. Actuator dynamics appear are cascaded at each plant input, making model extensions straightforward.

Chapter 3
Engine Control by Classical Methods

Abstract This chapter reviews and applies classical SISO design techniques (root locus and frequency domain loopshaping) to the problem of fan speed control using fuel flow rate as control input. A model-matching method is also described that is used in CMAPSS as a design tool. The shortcomings associated with the use of fixed linear compensation are illustrated with simulation examples.

As expected, classical linear compensation is adequate only to govern the engine close to a fixed operating point, as defined by the current inlet conditions and desired thrust setpoint. Engine accelerations across wide fan speed ranges, as well as thrust regulation across changing inlet conditions are handled poorly when a fixed linear controller is used. Aside from nonlinearity and parametric changes in the plant, critical variables must be maintained within safety ranges.

Linear compensation, however, is the basic building block of standard GTE control systems. Parametric changes and nonlinearity are addressed conventionally addressed with gain-scheduled linear compensators, while *limit protection logic* schemes are used to override the active linear regulator when a critical variable approaches its safety limit. In this chapter, three basic design approaches to thrust regulation by means of classical linear compensation are examined. Fuel flow rate is considered to be the only control actuator. At the end of the chapter, a CMAPSS simulation is presented that exposes the limitations of fixed-regulator schemes.

3.1 Setpoint Control via EPR or Fan Speed

As this book is being printed, no direct sensing technology yet exists that is capable of producing reliable thrust measurements suitable for feedback. Thrust estimation from other sensed quantities is very challenging due to its strong dependence on the engine's health condition, which is not precisely known [24, 25]. For this reason, alternative variables that can be reliably sensed and which are a proxy for thrust are used. Among these, EPR and N_f are commonly-used. A table-lookup routine can determine the value of EPR or N_f that results in the desired F_n setpoint, given

H. Richter, *Advanced Control of Turbofan Engines*, DOI 10.1007/978-1-4614-1171-0_3,

current inlet conditions. For all practical purposes, compensator design can proceed by assuming that a setpoint or reference profile has been given in terms of either ΔN_f or ΔEPR.

The transfer functions from ΔW_F to ΔEPR and ΔN_f are directly obtained from the linearized engine model at the appropriate flight condition, and have the form

$$\frac{\Delta \text{EPR}}{\Delta W_F} = \frac{a_0 s^2 + a_1 s + a_2}{s^2 + c_1 s + c_2}, \tag{3.1}$$

$$\frac{\Delta N_f}{\Delta W_F} = \frac{b_0 s^2 + b_1 s + b_2}{s^2 + c_1 s + c_2}. \tag{3.2}$$

Coefficients c_1 and c_2 are intrinsic to the flight condition and fixed for all engine outputs. The only assumption made about these coefficients is that they define a pair of transfer function poles having negative real parts. That is, linearized engine models are inherently stable. Nothing can be assumed about numerator coefficients, leaving open the possibility that they define transfer function zeroes with positive real parts, (*non-minimum phase zeroes*). In certain cases, however, these zeroes have real parts which are very large in comparison with the absolute value of the real parts of other zeroes and poles. Such high-frequency dynamics can usually be ignored without detriment to the accuracy of the linear model, provided the low-frequency gain of the transfer function is preserved [26]. Whenever a control technique discussed in this book is unable to handle nonminimum-phase systems, the necessary assumption will be made explicit.

3.1.1 Integral Control

It is an established fact of linear control theory [26] that a feedback compensator loop must display at least one *free integrator* (pole at the origin) for offset-free set-point attainment. In addition, such *Type 1* control loop offers enhanced disturbance rejection abilities. Indeed, step disturbances at plant input are completely rejected. Since transfer functions (3.1) and (3.2) cannot be assumed to contain a pole at the origin, a free integrator must be implemented as part of the controller.

Integral design proceeds by cascading the free integrator with the plant transfer function to form an *augmented plant* model, used as the basis for compensator selection. The zero-pole landscape associated with the augmented plant determines the simplest compensator structure to be attempted.

3.1.2 Compensator Design with the Root Locus

The classical root locus methodology is readily applied, since the plant models of (3.1) and (3.2) are of low order. Recalling classical control concepts, the objective

is to introduce a compensator and choose the loop gain so that all closed-loop poles have negative real parts, while a number of *dominant* closed-loop poles belong to a region of the complex plane which corresponds to desirable transient response. A set of poles is dominant relative to the remaining poles when their time constants are significantly larger than the remaining time constants. A time constant ratio of 8 between the fastest dominant pole and the slowest nondominant pole is typically adopted as a dominance criterion [26].

Root locus design frequently seeks to obtain *one* pair of dominant complex poles. This is because the relationship between pole locations and transient response is straightforward for the standard second-order transfer function:

$$G(s) = c\frac{w_{\mathrm{n}}^2}{s^2 + 2\zeta w_{\mathrm{n}} s + w_{\mathrm{n}}^2},$$

where $c = G(0)$ is the *static*, or *DC gain*, w_{n} is the natural frequency, and ζ is the damping ratio. Readers are referred to Dorf [26] if a review of second-order transient properties is needed. When $0 < \zeta < 1$, the response of $G(s)$ to a step input is said to be *underdamped*, and the roots of the denominator are complex and equal to $-\zeta w_{\mathrm{n}} \pm \sqrt{1-\zeta^2} w_{\mathrm{n}} i$.

The step response can be readily predicted in terms of percent overshoot and settling time according to the formulas

$$P.O. = 100\mathrm{e}^{-\left(\frac{\zeta}{\sqrt{1-\zeta^2}}\right)\pi}, \tag{3.3}$$

$$t_{\mathrm{s}} = \frac{4}{\zeta w_{\mathrm{n}}}. \tag{3.4}$$

According to these formulas, overshoot is a decreasing function of damping ratio, while settling time decreases with the absolute value of the real part of the complex poles. A combination of maximum allowable overshoot and settling time then results in a trapezoidal target region for the dominant poles, as illustrated in Figs. 3.1 and 3.2.

When more than two poles are dominant or if zeroes are present in the dominance region, the above formulas and target region do not apply, and iterative design must be carried out with the aid of a simulation package. In the simplest situation, the augmented plant has no zeroes (the coefficients of s^2 and s in the numerator are zero). As seen in Fig. 3.1 (top), proportional control is the simplest stabilizing compensator. However, the free integrator contributed by the controller causes two branches of the locus to cross the imaginary axis, introducing a limit in the loop gain for stability. When plant open-loop poles are close to the imaginary axis, slow response times will be obtained, and the design specifications become unfeasible under this compensator structure. A zero can be contributed by the compensator (PD control) to distort the root locus branches so that faster responses are possible, as seen in Fig. 3.1 (bottom). Since the free integrator belongs to the controller, an overall compensator with no more zeroes than poles is obtained. Tuning is done through the location of the zero and the value of the controller gain.

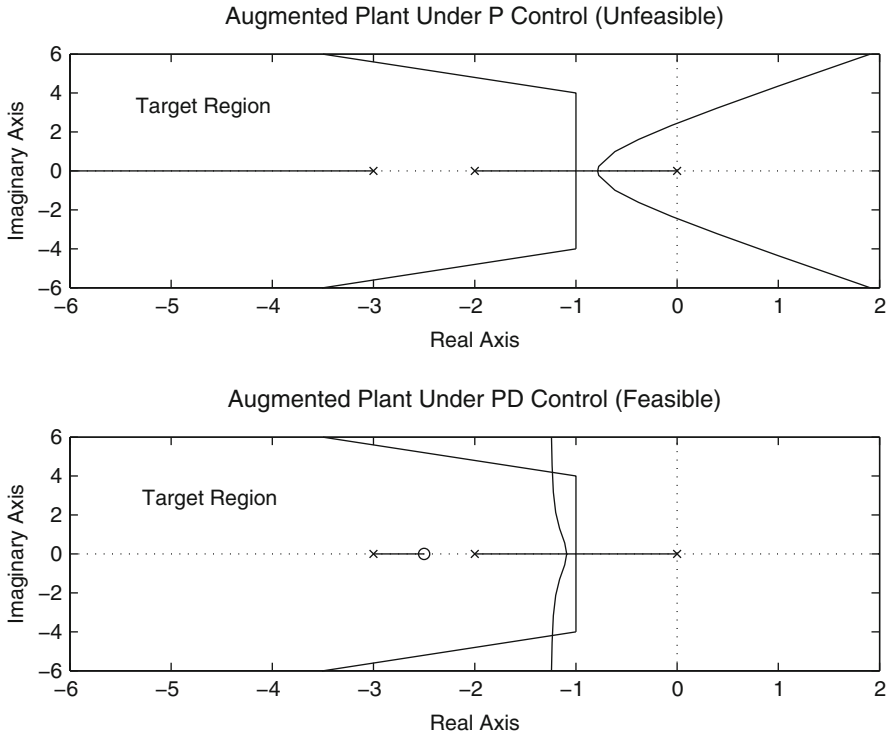

Fig. 3.1 Root locus analysis of augmented plant with no zeroes

In contrast, one or two zeroes may exist, and they may be nonminimum phase. As shown in Fig. 3.2 (top), the loop may be inherently unstable, precluding the use of a free integrator. Alternatively, it may be possible to use negative proportional gain, as shown in Fig. 3.2, followed by a compensator zero. The reader must keep in mind that the designed compensators produce ΔW_F as their output, which is allowed to be negative. This incremental control must be added to the baseline W_F corresponding to the flight condition under which the design is being carried out. Care must be exercised in keeping W_F between realistic limits by testing the linear compensator against the full nonlinear engine model.

3.1.3 Compensation in the Frequency Domain: Manual Loopshaping

Although it may be possible to meet transient response specifications using the root locus and trial-and-error procedures, the frequency domain method bridges specifications and designer inputs in a more direct fashion. Some trial-and-error is

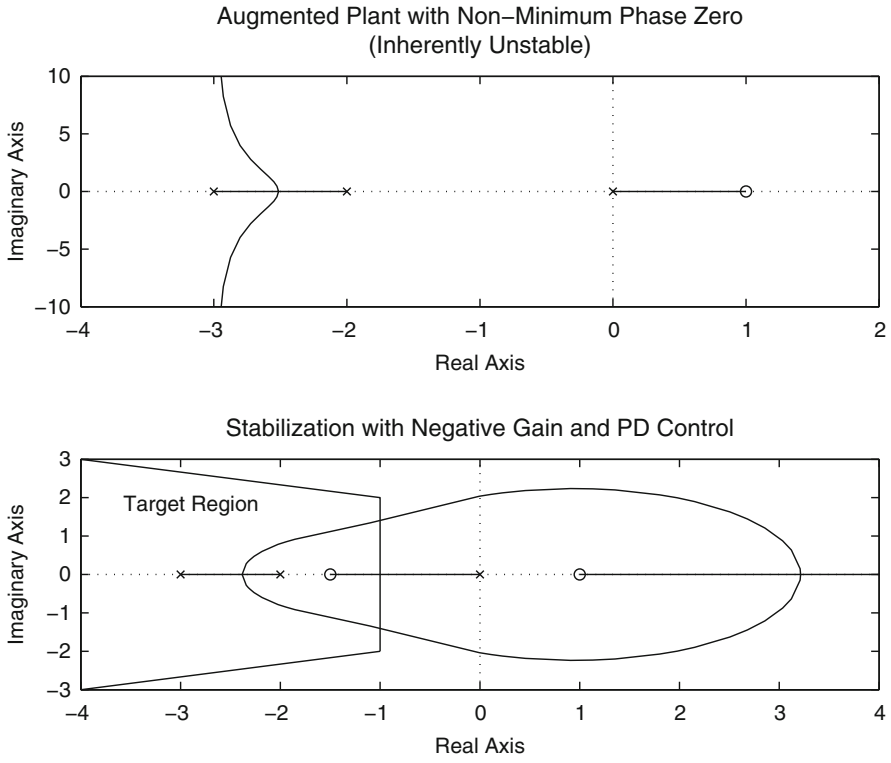

Fig. 3.2 Root locus analysis of augmented plant with non-minimum phase zero

still required to fine tune the design. In classical loopshaping, the designer attempts to reproduce the Bode plot of a target open-loop transfer function $L(s) = G(s)K(s)$ by manipulation of the zeroes, poles, and gain of $K(s)$. The target transfer function is specified by its Bode plot features. These features are obtained from an open-loop target of the form

$$L(s) = G(s)K(s) = \frac{w_{\mathrm{n}}^2}{s(s + 2\zeta w_{\mathrm{n}})}, \tag{3.5}$$

where w_{n} and ζ are chosen so that the closed-loop transfer function

$$T(s) = \frac{L(s)}{1 + L(s)} = \frac{w_{\mathrm{n}}^2}{s^2 + 2\zeta w_{\mathrm{n}} s + w_{\mathrm{n}}^2}$$

has a transient response matching the design specifications. Standard formulas [26] link w_{n} and ζ to step response characteristics such as percent overshoot and settling time:

$$PM \approx 100\zeta, \tag{3.6}$$

$$w_b \approx (-1.196\zeta + 1.85)w_n, \tag{3.7}$$

$$w_c \approx w_b/1.6, \tag{3.8}$$

where w_b is the *closed-loop bandwidth*, the frequency at which the magnitude of $T(s)$ reaches -3 dB, and w_c is the *crossover frequency*, at which the magnitude of $L(s)$ is 0 dB. These formulas are good approximations when ζ is between 0.3 and 0.8. The design process is best accomplished with the aid of software packages such as SISOtool, a graphical user interface for interactive controller design in Matlab. Assuming that zero steady-state error to step inputs is desired along with a settling time t_s and overshoot $P.0.$, the process is as follows:

1. Use (3.3) to calculate the required damping ratio ζ using the $P.O.$ specification.
2. Use (3.4) to calculate the required natural frequency w_n using ζ from above and the t_s specification.
3. Use formulas (3.6)–(3.8) to calculate the closed-loop bandwidth w_b, the target crossover frequency w_c and the target phase margin PM.
4. According to the features of the uncompensated loop $G(s)$, zeroes and poles are added to $K(s)$ and its gain is adjusted in SISOtool until $L(s)$ attains the target phase margin and crossover frequency.

If the design process leads to an $L(s)$ that attains the target phase margin and crossover frequency while having a pole-zero structure different than that of (3.5), it should be emphasized that the original time-domain specifications should still be approximately met, provided that the designed $L(s)$ contains an integrator and a dominant real pole is achieved. That is, if the dominant factor of the designed $L(s)$ has the form

$$\frac{k}{s(s+p)},$$

then it corresponds to the target of (3.5) with $w_n = \sqrt{k}$ and $\zeta = \frac{p}{2\sqrt{k}}$.

3.1.4 Edmund's Model-Matching Method

A model-matching approach due to Edmunds [27] is implemented as a controller design tool in CMAPSS-1. It is essentially an automated frequency domain loop-shaping approach, where the target is the closed loop magnitude response, defined by the tunable bandwidth and damping ratio parameters. It also includes a tunable real pole beyond the closed-loop bandwidth. A least-squares optimization process is used to arrive at a controller that produces a closed-loop frequency response having the specified real pole and the intended bandwidth and damping ratio. Note from (3.6) that the damping ratio is an indirect phase margin specification. The advantage of the method is that it requires only three parameters as user input, eliminating iterative design and thus being suitable for automated design. Unfortunately, the least-squares process may produce spurious nonminimum phase

zeroes in the controller. These zeroes may compromise the performance and stability of the loop, invalidating the design if they cannot be removed on the basis of root dominance.

3.1.5 Comparative Example

A compensator is to be designed to produce an increment of 100 RPM in fan speed under the following requirements:

1. Zero steady-state error.
2. Overshoot less than 5%.
3. Settling time near 1 s.

The designs are conducted using the CMAPSS-1 90k-class engine model at FC01. Comparisons among root locus, loopshaping, and Edmund's model matching method will be carried out. Finally, the controllers are applied to the engine model at a drastically different flight condition (FC08) to illustrate performance deterioration and motivate the need for advanced controllers. From the tables in Appendix B, the transfer function from ΔW_{F} in pounds per second (pps) to ΔN_F in RPM at FC01 can be obtained from the state-space matrices as

$$\frac{\Delta N_{\mathrm{f}}}{\Delta W_{\mathrm{F}}} = \frac{230.7s + 2032}{s^2 + 8.564s + 17.47}. \tag{3.9}$$

The plant features two stable real poles and one minimum-phase zero. The zero-steady state error requirement is addressed by including an integrator at plant input, while formulas (3.3) and (3.4) imply that the damping ratio must be greater than 0.7 and that the dominant poles must have real part less than -4. The zero of the compensator and the gain are readily tuned with the aid of design packages such as SISOtool, part of Matlab. The overall compensator becomes

$$K(s) = 0.016\frac{s + 3.70}{s}.$$

Figure 3.3 shows the root locus of the compensated loop. The closed-loop poles are $-4.19 \pm 3.67i$ and -3.85. Note that there is a real closed-loop pole in addition to the complex pair used to define the target trapezoidal region. This pole nearly cancels the zero of the plant, resulting in little deviation from the projected step response.

Manual loopshaping design is achieved by translating time domain performance specifications into a set of parameters for the target loop. Using the formulas in (3.6)–(3.8), the target crossover frequency is calculated as $w_{\mathrm{c}} = 3.62$ rad/s, and the target phase margin is $PM = 70°$. The target loop shape is achieved by including a real zero in the controller in addition to the integrator and by tuning the gain.

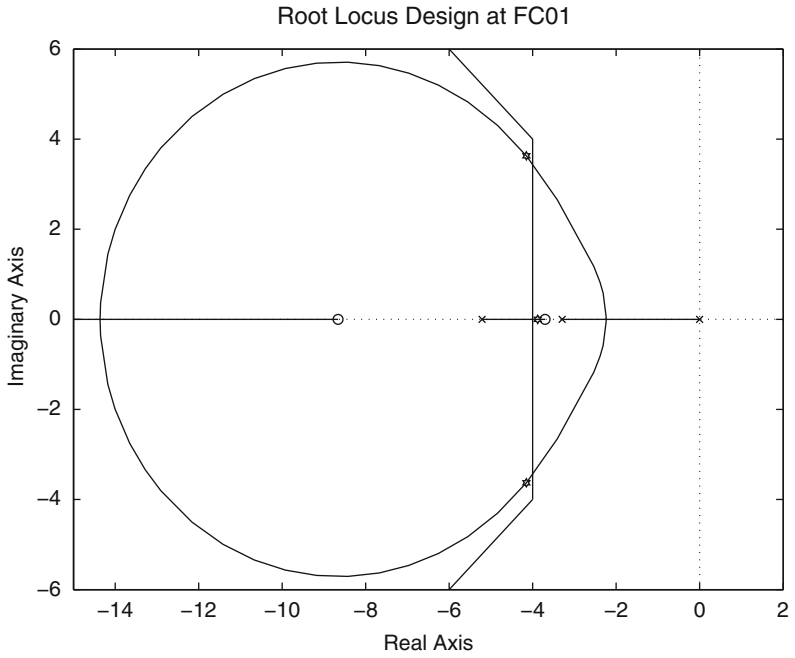

Fig. 3.3 Root locus design at FC01

The location of the zero and the value of the gain are chosen with minimal iterations in SISOtool. The resulting compensator becomes

$$K(s) = 0.012\frac{s + 3.45}{s}.$$

This controller achieves a phase margin of 76.1° and a crossover frequency of 3.99 rad/s, as shown in Fig. 3.4.

Edmund's method is applied next. Formulas (3.6)–(3.8) indicate, again, that a closed-loop bandwidth of $w_b = 5.79$ rad/s and a damping ratio of $\zeta = 0.7$ match the specifications of this example. No guidance for the selection of the real pole is offered by the model-matching method, so it is arbitrarily set to −20. The built-in model-matching solver in CMAPSS gives the controller

$$K(s) = 0.21\frac{s + 3.715}{s(s + 20)}. \tag{3.10}$$

Note that all three methods yield controllers attempting to cancel the zero of the plant. The transient response is about the same for the three designs, as shown in Fig. 3.5.

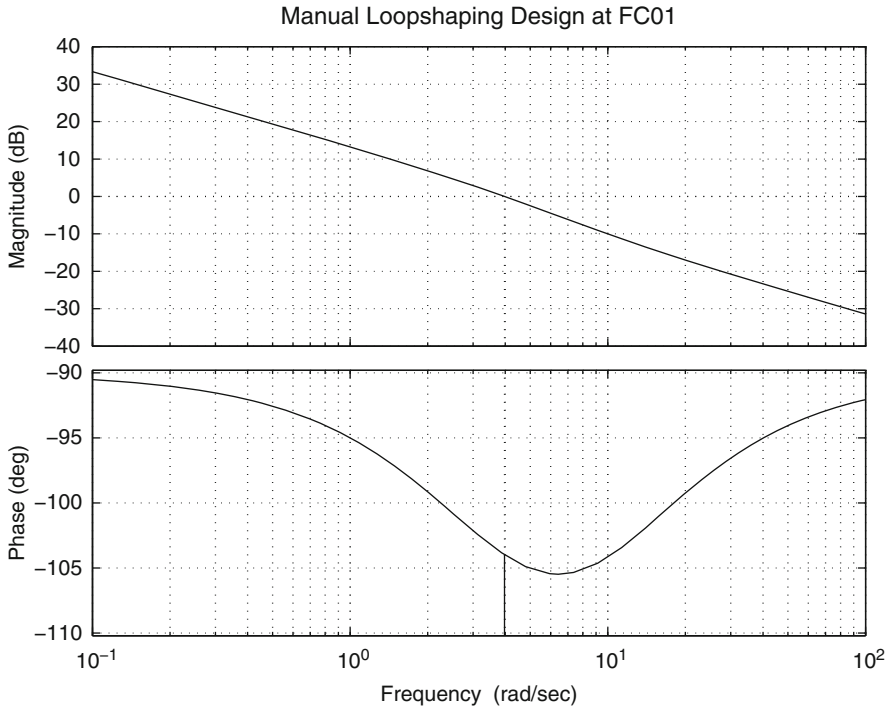

Fig. 3.4 Manual loopshaping design at FC01

3.2 Shortcomings of Fixed Linear Compensator Designs

Although only three design methods have been discussed, it should be clear that other classical compensation techniques still produce a fixed control transfer function. Advanced linear compensation methods such as $\mathcal{H}_\infty$ control and μ synthesis [28–30] ultimately deliver a fixed compensator. If properly conducted, these compensators should match the specified nominal transient response. These optimized compensators, however, offer enhanced robustness properties. In the GTE control problem, a properly designed robust compensator will maintain prescribed degrees of closed-loop stability and performance as the parameters of the plant change. As seen earlier, linearized models change due to varying inlet conditions, inherent engine nonlinearity and engine aging.

As it will become evident in the following sections and chapters, linear compensators designed on the basis of linearized models are sufficient to maintain stability in the face of plant parameter variations and inherent engine nonlinearity. *The major challenge for control design arises from the need to maintain critical operating variables within allowable limits, without undue penalties to transient response qualities*. The tendency of variables such as stall margin and turbine temperature

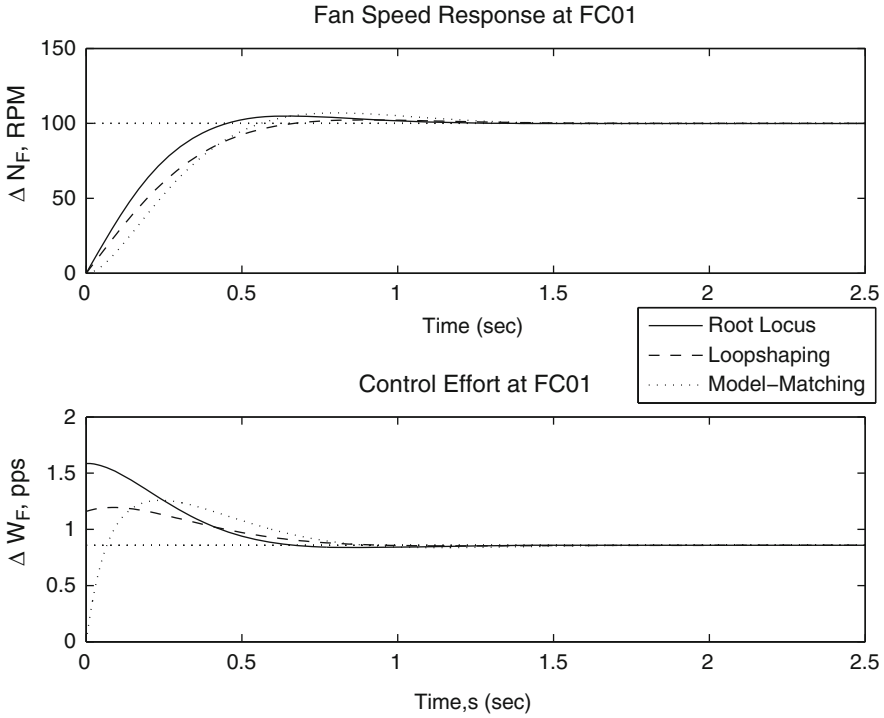

Fig. 3.5 Comparison of step responses obtained with root locus, manual loopshaping, and model-matching designs at FC01

to cross their allowable limits is exacerbated by engine aging, which increases the limit protection challenge by introducing robustness requirements.

3.2.1 Parameter Variations Across the Flight Envelope

To illustrate the extent of performance loss due to parametric changes in the plant, the loopshaped controller designed at FC01 is applied to the linearized model at FC08 (see Table 2.3), given by

$$\frac{\Delta N_{\mathrm{f}}}{\Delta W_{\mathrm{F}}} = \frac{252.2s + 1011}{s^2 + 3.919s + 3.528}.$$

A quick calculation reveals that the phase margin is reduced to 42.2°, which predicts a higher overshoot (30%), as shown in Fig. 3.6. Although both settling time and overshoot have fallen outside specifications, performance loss can be qualified as mild, and stability is not compromised even at this drastically different flight

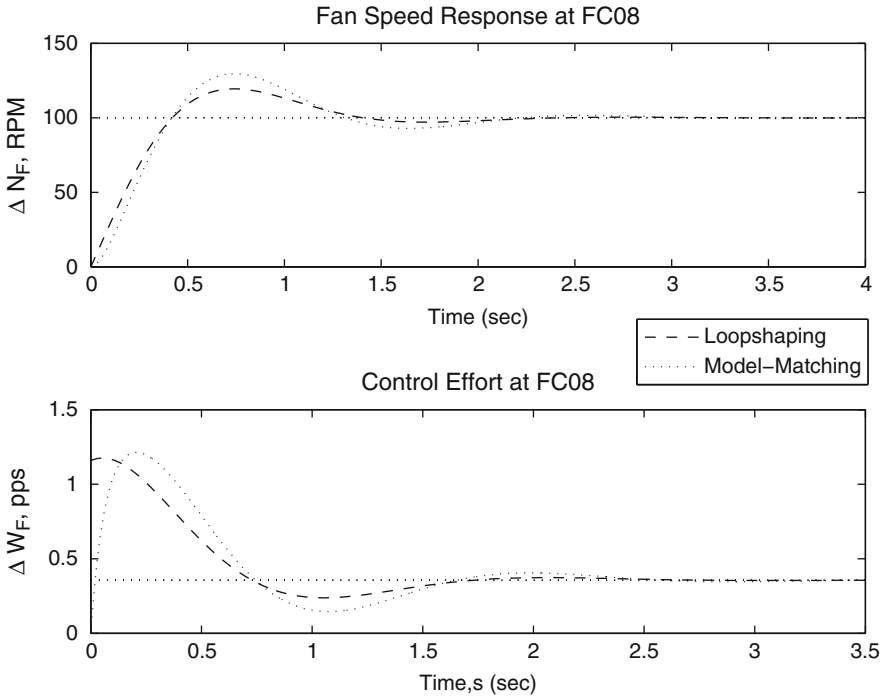

Fig. 3.6 Response of controllers designed at FC01 when applied to FC08

condition. Robust linear controllers designed with suitable methods are able to maintain specifications across a larger range of operating conditions, as shown in Chap. 4.

Gain scheduling, or the tailoring of controller gains to operating points or regions, has been used by the GTE industry for decades as the standard way to account for large parameter variations. To illustrate the extent of these variations, consider the linearized plant transfer functions at FC01 and FC08. A quick examination reveals that their four parameters (two pole locations, one zero location and a gain) have undergone significant changes. In a typical gain scheduling design, linear functions linking plant parameters to a set of *scheduling variables* are sought. Typical choices for scheduling variables in GTE control are inlet static pressure and fan speed itself. The first scheduling variable accounts for parametric changes arising from varying altitude, while the second captures intrinsic plant nonlinearity. Fixed linear compensators are then designed for various combinations of scheduling variables. Gain interpolation is used during real-time operation. Alternatively, it may be possible to parameterize controller gains using the scheduling variables as parameters. Gains are computed in real-time using formulas rather than table look-ups. Gain scheduling and its parent technique, linear-parameter-varying (LPV) control, will be examined in Chap. 5.

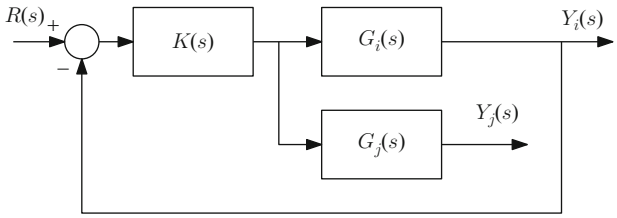

Fig. 3.7 Block diagram for output y_j under closed-loop control on y_i

3.2.2 Engine Limits

As discussed in Sect. 1.3, several critical engine variables must be kept under allowable limits at all times. Engine models such as those included in the CMAPSS family include critical variables among the outputs. Moreover, the linearization functions in all versions of CMAPSS produce output matrices for these outputs. Given a flight condition and its associated A and B matrices, a pair of C and D matrices defines a transfer function from fuel flow increment to an incremental output of choice. These output transfer functions are essential for the design of limit protection strategies. For instance, the linearized transfer function for the high-pressure turbine outlet temperature at FC01 is

$$\frac{\Delta T_{48}}{\Delta W_{\mathrm{F}}} = 146.24 \frac{(s+4.73)(s+2.30)}{s^2+8.564s+17.47},$$

where ΔT_{48} is in $^\circ R$. As expected, the denominator of the above transfer function is the same as that of (3.9). Indeed, the set of poles of the transfer function from ΔW_{F} to any system output is the constant for a fixed flight condition. The transfer function poles match the eigenvalues of state-space matrix A. Thus, the differences in the transient behavior of critical outputs are characterized by the zeroes of their respective transfer functions. Suppose $G_i(s)$, $G_j(s)$ are two output transfer functions, and $K(s)$ is a compensator placed in a feedback loop involving $G_i(s)$, as shown in Fig. 3.7. The closed-loop transfer function relative to $G_j(s)$ is given by

$$\frac{Y_j(s)}{R(s)} = \frac{G_j(s)K(s)}{1+G_i(s)K(s)}. \tag{3.11}$$

Using this formula, the closed-loop transfer function between fan speed increment demand, N_{f}, dmd and ΔT_{48} under the loopshaped fan speed controller becomes

$$\frac{\Delta T_{48}}{\Delta N_{\mathrm{f}},\,\mathrm{dmd}} = 1.756 \frac{(s+4.733)(s+3.45)(s+2.301)}{(s+3.515)(s^2+7.816s+23.92)}.$$

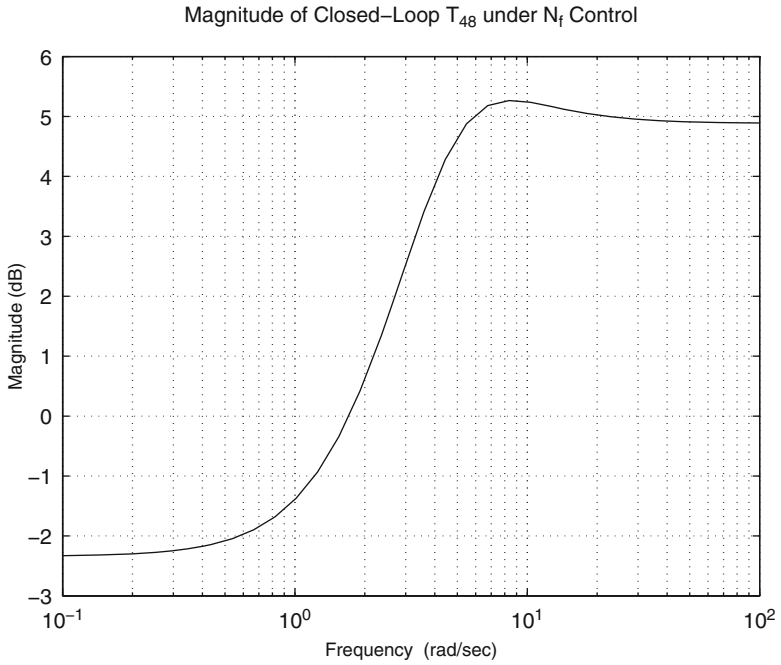

Fig. 3.8 Closed-loop magnitude response of turbine temperature under fan speed control

A frequency response plot for $\frac{\Delta T_{48}}{\Delta N_{\mathrm{f\_dmd}}}$ is shown in Fig. 3.8, revealing large magnitudes near the design crossover region. This predicts significant transient peaking of T_{48}, a highly undesirable feature. A simulation of the loopshaped controller applied to the nonlinear engine model further illustrates the limitations of a fixed-compensator approach. Gain scheduling and built-in limit protection features in CMAPSS-1 were bypassed, replacing them by the single control transfer function. The designed control transfer function $K(s)$ has ΔW_{F} as its output. To deploy the controller to the nonlinear engine, the absolute fuel flow command W_{F} must be calculated. To do this, the integrator in $K(s)$ is factored out as follows:

$$K(s) = K'(s)\frac{1}{s}.$$

If the loopshaped design is used for $K(s)$, the new compensator $K'(s)$ is of the PD type. The output of $K'(s)$ is the derivative of fuel flow rate, $\dot{W}_{\mathrm{F}}$. Thus, the absolute fuel flow command is obtained by integration of $\dot{W}_{\mathrm{F}}$, using the linearization value of W_{F} as initial condition, as shown in Fig. 3.9. Since the control transfer function is driven from the fan speed tracking error, no adjustments are required at the input (the linearization value of N_{f} at the starting flight condition cancels out at the summing node). The simulation corresponds to a TRA demand changing from 0 to 100 degrees as a step, with inlet conditions fixed at FC01 values. Figures 3.10 and 3.11 show that, although the fan speed demand is met with no offset, severe

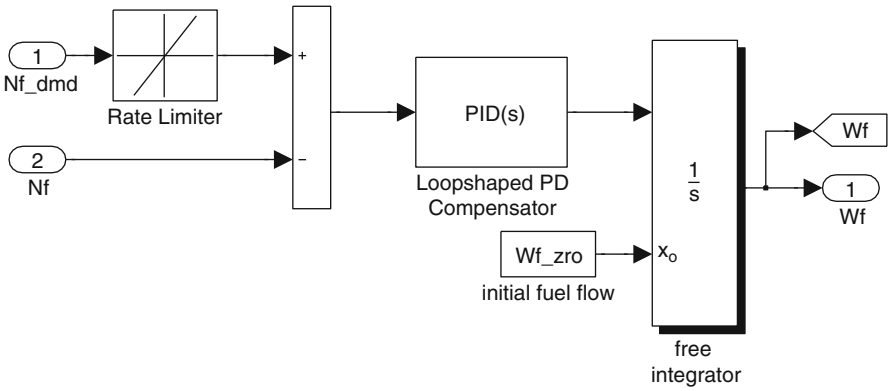

Fig. 3.9 CMAPSS implementation of an incremental linear compensator

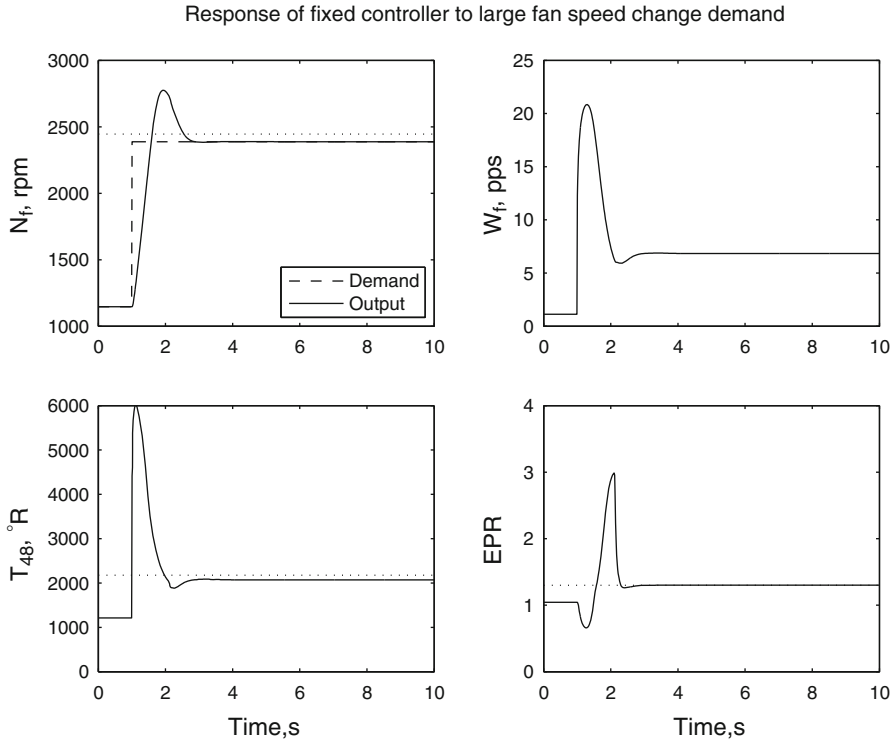

Fig. 3.10 Engine response to large fan speed demand with fixed linear compensator

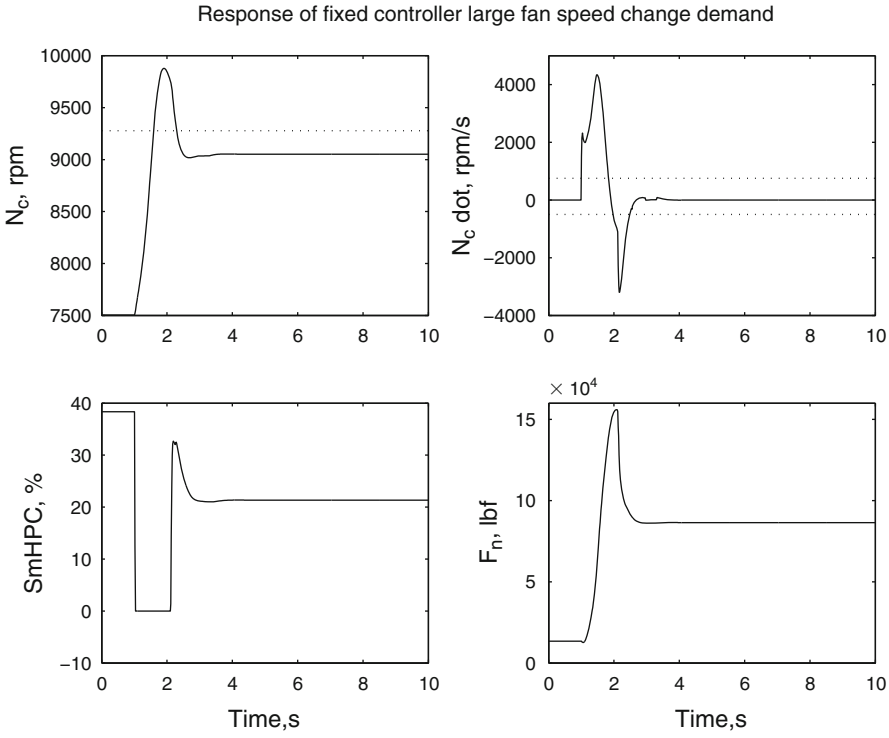

Fig. 3.11 Engine response to large fan speed demand with fixed linear compensator

transient peaking occurs in every output. Referential limits for N_f, EPR, T_{48}, $\dot{N}_c$, and SmHPC have been represented with dashed lines. All limits are exceeded by large amounts, strongly suggesting that, regardless of synthesis method, no fixed linear regulator can be found that is able to achieve limit protection and adequate fan speed response.

Finally, the CMAPSS simulation data has been represented in a compressor map in Fig. 3.12. The horizontal coordinate is the corrected flow through the HPC, calculated through (1.20), (1.21), and (1.22), with P_t and T_t taken at HPC inlet. The value from (1.22) is then divided by a scaling factor. The vertical coordinate is the HPC pressure ratio from the simulation, also scaled. The scaling factors are applied so that both coordinates are compatible with the scaling used in CMAPSS to store map data (efficiency and speed contours and R-lines). The initial and final steady conditions have been represented on the engine operating line. The trajectory starting from FC14 proceeds toward the stall limit, reaching a zero stall margin condition. The CMAPSS solver will not produce negative stall margin values hence,

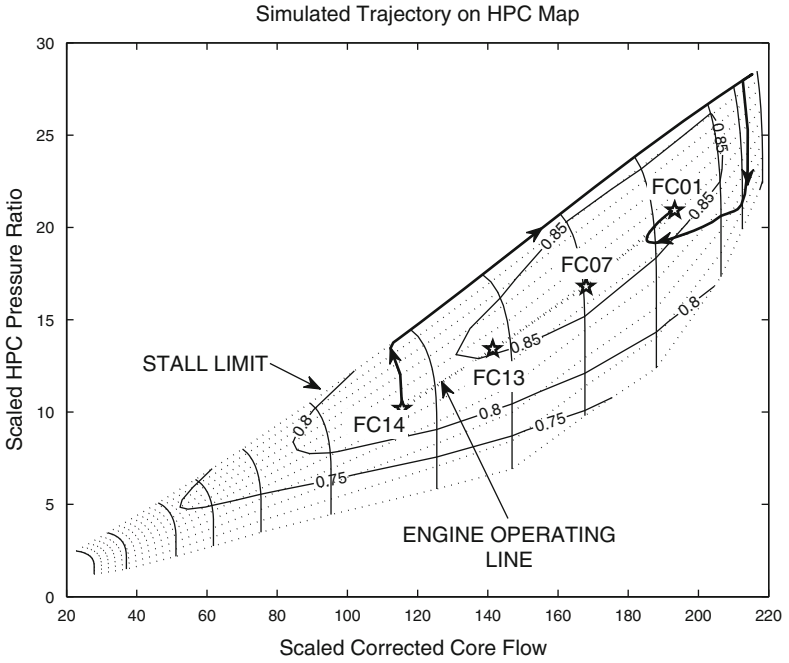

Fig. 3.12 Simulated trajectory in compressor map coordinates

the trajectory remains on the stall line for some time. It later abandons the stall line and proceeds to the regulation point at FC01. Although the simulated trajectory is inadmissible from a practical standpoint, it exemplifies the challenges associated with control design in the presence of operating limits.

Chapter 4
Engine Control by Robust State Feedback

Abstract This chapter reviews linear multivariable theory and introduces polytopic system descriptions of plant variability. The chapter also presents various methods for MIMO state-feedback synthesis, such as: LQR, $\mathcal{H}_2$, $\mathcal{H}_\infty$ and mixed-objective optimization with regional pole placement constraints. A simplified $\mathcal{H}_\infty$ compensator synthesis method is presented for SISO systems. Matlab code and simulations using the CMAPSS nonlinear engine model are included.

The purpose of this chapter is to provide an overview of robust, multivariable techniques that are directly applicable to the GTE control problem. As pointed out in Chap. 3, fixed controllers cannot be expected to preserve engine limits or to operate satisfactorily across the whole flight envelope, less so if the engine health parameters are subject to changes and unmodeled dynamics exist. However, various approaches to gain scheduling are based on a set of fixed controllers. It is therefore essential to discuss the salient features of robust multivariable control and provide practical design guidelines. The chapter assumes familiarity with the state-space pole-placement concept, at a minimum.

Actual engines incorporate real-time sensing of fan and core speeds, which are the states of the dynamic model. This fact opens the doors to many techniques based on state measurement feedback, such as the linear quadratic regulator (LQR) and sliding mode control. Although most state feedback techniques admit observer-based extensions, the performance and robustness properties attained with measurement feedback are partially lost when a state estimator is introduced. A classical example of this effect is given by the performance and robustness losses associated with observer-based implementations of linear quadratic regulation. Loop transfer recovery techniques [29] aim to restore the lost performance and robustness properties.

The chapter begins by introducing essential concepts such as singular values and signal and system norms and their computation using Matlab. Robustness and performance are then addressed using classical LQR theory. A characterization of the uncertain design plant, along with a corresponding formulation of the control objectives is then developed using the tools of $\mathcal{H}_2$ and $\mathcal{H}_\infty$ robust state feedback synthesis. A classical LQR controller and robust $\mathcal{H}$-norm based controllers are

H. Richter, *Advanced Control of Turbofan Engines*, DOI 10.1007/978-1-4614-1171-0_4,

designed for the 40k-class engine, using W_F, VSV, and VBV as actuators. The designs are then simulated in CMAPSS-40k.

4.1 Overview of Multivariable Systems Theory

Given a multi-input, multi-output (MIMO) system described by a set of state-space matrices (A, B, C, D), the single-input, single-output (SISO) transfer function $G_{ij}(s)$ from the $j-$th input to the $i-$th output is found using the formula:

$$G_{ij}(s) = C_i(sI - A)^{-1}B_j + D_{ij}, \tag{4.1}$$

where C_i is the $i-$th row of C, B_j is the $j-$th column of B, and D_{ij} is the (i, j) entry of matrix D. The arrangement of the G_{ij} in a matrix is the *transfer matrix* $G(s)$. The poles of the individual entries G_{ij} are a subset of the eigenvalues of A, where pole-zero cancelations may have occurred in individual entries. The *transmission zeroes* or *multivariable zeroes* of $G(s)$ have a special definition, and they generally differ from the zeroes of the individual $G_{ij}(s)$. The zeroes of a transfer matrix are given by the set of values of s for which $G(s)$ loses rank. Mathematically, $s = z$ is a zero of $G(s)$ with multiplicity k if there exist k linearly independent vectors v so that $G(z)v = 0$. In Matlab, the transmission zeroes are readily found with the `tzero` command.

In this book, we limit ourselves to *rational* transfer matrices; that is, matrices whose entries are ratios of polynomials. A rational transfer matrix is *proper* if its individual entries are so, that is, if they do not have more zeroes than poles. A set of state-space matrices (A, B, C, D) always results in a rational, proper transfer matrix.

4.1.1 Example

Take the 40k engine model matrices of CMAPSS-40k at Ground Idle listed in Appendix C. Define the control input vector as $u = [\Delta W_F \;\; \Delta VSV]^T$ and choose two outputs as $y = [\Delta EPR \;\; \Delta T_{48}]^T$. The following Matlab code is used to find the transmission zeroes:

```
%Assumes A,B,C,D are in the workspace
sysSS=ss(A,B,C,D); %Create system
sysTM=zpk(sysSS) %Find and display transfer matrix
tzero(sysTM) %Display transmission zeroes
```

The reader can verify that the transfer matrix returned in `sysTM` is:

$$G(s) = \frac{\begin{bmatrix} 289.0525(s+3.925)(s+1.918) & 0.1332(s-95.1)(s+3.471) \\ -10.9483(s+1.274)(s+4.64) & 0.1837(s^2+0.7075s+4.929) \end{bmatrix}}{(s+3.992)(s+2.439)}.$$

The transmission zeroes are $\{2.9121,\ -0.5659,\ -3.9919,\ -2.4389\}$, while the zeroes of the individual entries are $\{-3.925,\ -1.918,\ 95.1,\ -3.471,\ -1.274,\ -4.64, -0.3538 \pm 2.1918i\}$, a different set. A right-half plane transmission zero is found at 2.9121. As in the SISO case, such nonminimum phase zeroes introduce fundamental limitations to attainable performance and may lead to design difficulties.

To illustrate the definition of transmission zero, evaluate the transfer matrix at one of the zeroes, for instance at $z = 2.9121$. The reader can verify that $G(z)$ is very close to being singular. Numerical precision prevents the exact verification of the rank loss property. In contrast, if the zeroes of the individual transfer function entries are used for z, the rank of $G(z)$ is always 2.

4.1.2 Singular Values

Singular values are fundamental indicators of the effects of input directionality on system outputs. In one of their interpretations, they extend the notion of frequency response to multi-input systems. Recalling basic linear algebra, a vector v of constant unit length and variable direction will, in general, give rise to vectors w of different lengths upon the linear operation $w = Tv$, where T is a constant matrix. The lengths obtained for w as v is varied are bounded above and below by two quantities known as the *maximum and minimum singular values* of T:

$$\bar{\sigma}(T) = \max_{||v||=1} ||Tv||, \tag{4.2}$$

$$\underline{\sigma}(T) = \min_{||v||=1} ||Tv||. \tag{4.3}$$

Every m-by-n matrix T with complex entries can be decomposed as $T = U\Sigma V^*$, where U and V are unitary matrices and Σ is a diagonal matrix of the form

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix},$$

where

$$\Sigma_1 = \begin{bmatrix} \bar{\sigma} & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & \underline{\sigma} \end{bmatrix}.$$

The diagonal entries of Σ_1, $\bar{\sigma} = \sigma_1 > \sigma_2 > \dots > \sigma_p = \underline{\sigma}$ are the singular values, with $p = \min(m, n)$. The notation V^* is used for the complex conjugate transpose of V, and a unitary matrix is such that $V^*V = VV^* = I$. In Matlab, the singular value decomposition is found with the command `[U,S,V]=svd(T)`.

4.1.3 The Infinity Norm of a Linear System

In control systems, notions of "size" are customarily defined for transfer matrices, leading to bounds on the magnitude of the output as the input is varied under constant norm. One of such definitions is given by the *infinity norm* $||G||_\infty$, as follows: Suppose a p-by-m transfer matrix G is excited with an input vector $u(t)$ with sinusoidal components $u_j(t)$, $j = 1, 2...m$, all sharing the same frequency w but with possibly different amplitudes and phase shifts: $u_j(t) = U_j \sin(wt + \phi_j)$. Supposing output components reach a steady-state of sinusoidal oscillation of the form $y_i(t) = Y_i \sin(wt + \theta_i)$, $i = 1, 2...p$, an amplification ratio can be defined as $||Y||/||U||$, where $U = [U_1\ U_2\ ...U_m]$ and $Y = [Y_1\ Y_2\ ...Y_p]$. The infinity norm of G, denoted $||G||_\infty$, can be interpreted as the least upper bound (supremum) of the amplification ratio as frequency, input component amplitudes and phases are varied. That is,

$$||G||_\infty = \sup_{\phi_j, U_j, w} \frac{||Y||}{||U||}.$$

The frequency-domain definition of $||G||_\infty$ allows its calculation through the maximum singular value:

$$||G||_\infty = \sup_w \bar{\sigma}\left\{G(jw)\right\}. \tag{4.4}$$

For practical purposes, w is swept in a range, calculating the maximum and minimum singular values of $G(jw)$ at each point. The overall maximum of $\bar{\sigma}(G(jw))$ over all frequencies is the value of $||G||_\infty$. In Matlab (Robust Control Toolbox), an efficient numerical routine is implemented in the `hinfnorm` command. Alternatively, a routine using `svd` and `freqresp` can be easily programmed. As an example, the following sequence of Matlab commands plots the maximum and minimum singular values of the transfer function of Example 4.1.1 (assuming system matrices are available in the workspace):

```
>> sys=pck(A,B,C,D);
>> w=logspace(-1,4,500); %create a vector of 500 logarithmically
spaced frequencies between 10^(-1) and 10^4
>> Gf=frsp(sys,w); %calculate frequency response
>> [u,s,v]=vsvd(Gf); %calculate SV decomposition at each
frequency
>> vplot('liv,lm',s); %plot singular values. The norm can be
visually extracted from the plot
>> hinfnorm(sys) %accurate calculation via bisection search
```

The reader can verify that $||G||_\infty = 289$ for the transfer matrix of Example 4.1.1. Needless to say, this number is affected by the choice of units of measurement for the inputs and outputs of the system. The infinity norm may also be calculated directly with `norm(sysTM,inf)`.

4.1.4 The 2-Norm of a Linear System

The infinity norm is a measure of the *peak* amplification produced by the linear system as input frequency and direction are varied. The 2-norm measures system amplification in terms of *root mean square averages* rather than peaks:

$$||G||_2 = \sqrt{\frac{1}{2\pi}\int_{-\infty}^{\infty} \text{trace}\,\{G^*(jw)G(jw)\}\,\mathrm{d}w}. \tag{4.5}$$

The definition shows that a finite 2-norm can only be obtained if $G(\infty) = 0$, that is, if the transfer matrix is strictly proper. For rational, strictly proper transfer matrices with state-space realization $(A, B, C, 0)$, where A is stable, the 2-norm calculation reduces to

$$||G||_2^2 = \text{trace}\,(B^*QB) = \text{trace}\,(CPC^*), \tag{4.6}$$

where P and Q are the *observability and controllability Gramians*, obtained from the following Lyapunov equations:

$$AP + PA^* + BB^* = 0,$$
$$A^*Q + QA + C^*C = 0.$$

The Matlab command `h2norm` incorporates these calculations. The transfer matrix of Example 4.1.1 does not have a finite 2-norm, since it is not strictly proper. The reader may verify this fact by inspection, or numerically, using `norm(sysTM,2)`.

4.2 Robust State Feedback Synthesis

The GTE control problem, although very challenging in many respects, has a simplifying feature: *the states of the linearized plant are measurable in real-time*. That is, fan and core speeds may be used directly in the calculation of feedback laws. This permits the application of several state-based techniques for control without the need for state estimation. In this section, we are interested in assessing the stability of a feedback arrangement of plant and controller, not only when plant parameters are known, but when they are subject to bounded, uncertain variations. Figure 4.1 shows the basic state-feedback arrangement considered in this section. The linearized engine plant is described in state-space form as

$$\dot{x} = Ax + Bu + \Gamma w, \tag{4.7}$$
$$y = Cx + Du + \Lambda w + n, \tag{4.8}$$

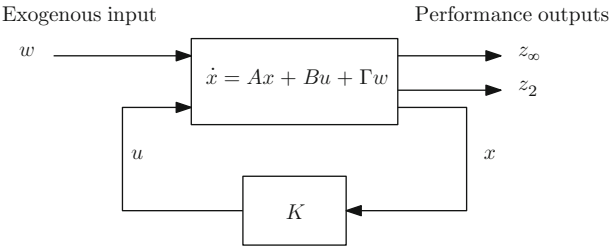

Fig. 4.1 State feedback configuration

where $x = [\Delta N_\mathrm{f} \quad \Delta N_\mathrm{c}]^\mathrm{T}$ is the state vector, $u = [\Delta W_\mathrm{F} \quad \Delta VSV \quad \Delta VBV]^\mathrm{T}$ is the vector of control inputs and w is a vector that can be used to include the effect of disturbances. In linearized models obtained with CMAPSS, w is used to represent a vector of *health parameter inputs*, which can be regarded as disturbances. As described in Sect. 2.1.3, w can be used to simulate the effects of aging and deterioration of engine components. Output vector y may be defined to reflect a combination of sensed variables, outputs that need to be regulated and quantities that need to be monitored or maintained between certain limits. A sensor noise vector n may also be considered in certain control problems. A *steady operating point* is a set of states $\bar{x}$ such that $A\bar{x} + B\bar{u} + \Gamma\bar{w} = 0$ for some fixed inputs $\bar{u}$ and $\bar{w}$. That is, given a constant health parameter vector and a constant vector of actuator inputs, the engine reaches a steady operating condition with corresponding constant values of fan and core speeds.

4.2.1 Polytopic Description of System Uncertainty

In addition to exogenous inputs, system (4.7) is subject to variations in all its describing matrices. In the GTE, these changes arise mainly from two sources:

- Intrinsic nonlinearity: the dynamic relationship between control inputs and states is nonlinear, even when operating the engine at fixed environmental conditions (altitude, aircraft speed, ambient temperature). Therefore, system matrices will vary with steady-state linearization point.
- Parametric variations: given the same set of steady states and inputs, different system matrices will be obtained through linearization if environmental conditions are changed.

It is important to note that engine health status could have been modeled as parameters of functions f_1 and f_2 in (2.1) and (2.2). Changes in these parameters would be reflected in changes to the linearized system matrices. Rather, the designers of CMAPSS have chosen to capture faults and deterioration effects by means of exogenous inputs collected in vector w.

4.2.1.1 Scheduled vs. Robust Control

The variations in A,B C, D, Γ and Λ are *deterministic* by nature. That is, they can be traced to the steady operating point in a repeatable way. In a scheduled control approach, this knowledge about the variation of system matrices with operating point is used to construct a controller whose gains are tailored to operating region. In contrast, when system matrix variations are not excessively wide, the designer may choose to ignore this knowledge and regard the variations as uncertain, but contained within certain bounds. This approach, termed "robust feedback synthesis" derives fixed controllers whose sensitivity to plant parameter changes is minimized. The complexity associated with gain-scheduled control implementations is thus avoided. As the next sections will demonstrate, robust stability with fixed linear compensators is feasible in GTE control systems. Consistent transient response qualities across the flight envelope are not obtainable with fixed compensators, however. For this reason, gain-scheduled control is used extensively in GTE control systems. Gain scheduling is covered in Chap. 5.

In this section, we focus on the robust feedback stabilization problem, that is, we attempt to find a fixed gain K capable of maintaining closed-loop stability regardless of variations in system matrices. An appropriate description of the uncertain state-space plant is given by the following *polytopic system*:

$$\dot{x} = A(\alpha)x + B(\alpha)u + \Gamma(\alpha)w, \tag{4.9}$$

where $A(\alpha)$, $B(\alpha)$ and $\Gamma(\alpha)$ are restricted to vary in a *polytope* of matrices, defined as a convex combination of k *vertices* A_i and B_i, $i = 1, 2...k$:

$$A(\alpha) = \sum_{i=1}^{k} \alpha_i A_i,$$

$$B(\alpha) = \sum_{i=1}^{k} \alpha_i B_i,$$

$$\Gamma(\alpha) = \sum_{i=1}^{k} \alpha_i \Gamma_i,$$

$$\alpha = [\alpha_1,\ \alpha_2,\ ...\alpha_k],$$

$$\sum_{i=1}^{k} \alpha_i = 1.$$

Thus, each instance of A, B and Γ can be understood as made up of "percentage" contributions of the vertices. These contributions are reflected in weight vector

α, whose entries are termed *polytopic coordinates*. Note that the influence of exogenous inputs through Γ is not relevant to stability analysis. Similar definitions apply to output matrices $C(\alpha)$, $D(\alpha)$, and $\Lambda(\alpha)$.

To generate a polytopic description of a GTE model, the designer must choose a number of representative steady operating points and perform system linearization. Simulation packages such as the CMAPSS family include a pre-defined set of such *flight conditions*, which provide good coverage of the entire flight envelope.

4.2.2 Nominal and Robust Stability

Recall that a rational transfer matrix $G(s)$ is *stable* if its entries are transfer functions whose poles have negative real parts. In a state-space realization of $G(s)$ given by the quadruple (A, B, C, D), stability is equivalent to the condition that all the eigenvalues of A have negative real parts. If no pole-zero cancellations occur, the eigenvalues of A match the set of poles of the entries of G. When the feedback control law $u = -Kx$ is applied to system (4.7), the resulting closed-loop system has the form

$$\dot{x} = (A - BK)x + \Gamma w \tag{4.10}$$

Thus, *nominal stability* is ensured if K can be chosen so that all eigenvalues of $A - BK$ have negative real parts. The pair (A, B) is called *stabilizable* when a stabilizing gain K exists. Several tests for nominal stabilizability are possible. A simple test for stabilizability is accomplished by evaluating the *modal controllability* of (A, B) as follows: the pair is first transformed to its controllability form $(\bar{A}, \bar{B})$:

$$\bar{A} = \left[\begin{array}{c|c} A_{\mathrm{nc}} & 0 \\ \hline A_{21} & A_{\mathrm{c}} \end{array}\right], \quad \bar{B} = \left[\begin{array}{c} 0 \\ \hline B_{\mathrm{c}} \end{array}\right].$$

This transformation arises from the change of state coordinates $z = Tx$, where $T = T^{\mathrm{T}}$ is an orthogonal transformation matrix. This change, referred to as a *similarity transformation*, results in system matrices $\bar{A} = TAT^{\mathrm{T}}$ and $\bar{B} = TB$, which share the same stabilizability properties as the original pair (A, B). Matrix A_{nc} describes the dynamics of the *uncontrollable subspace*, a set of state variables that are unaffected by the control input. The pair (A, B) is stabilizable if and only if A_{nc} is stable. The similarity transformation T required to put the system in controllability form can always be calculated. In Matlab, T, $\bar{A}$ and $\bar{B}$ are found directly using the `ctrbf` command.

Note that stabilizable systems need not be controllable, but controllability implies stabilizability. Controllable systems possess an empty uncontrollable subspace, and A_{nc} does not exist. When a system is controllable, a stabilizing K may be found by pole-placement methods or more sophisticated robust synthesis methods, considered later in this chapter. When the system is only stabilizable, however,

a number of poles equal to the dimension of the controllable subspace may be arbitrarily placed. The pole-placement problem is solved in z-coordinates using (A_c, B_c), resulting in a gain K_z yielding desired pole locations for $A_c - B_c K_z$. The feedback gain to be used in the original system is then calculated as $K = [* \mid K_z]T$, where $*$ denotes arbitrary values having no effect on the achieved pole locations. These values do have an impact on the magnitude of the control effort, however. For this reason, their choice is no longer immaterial in optimal feedback synthesis approaches such as LQR or $\mathcal{H}_\infty$ control.

4.2.3 *Quadratic Stability of Polytopic Systems*

Consider a polytopic system of the form

$$\dot{x} = A(\alpha)x \tag{4.11}$$

with vertices A_i, $i = 1, 2, ..k$ and $\alpha = [\alpha_1,\ \alpha_2,\ ...\alpha_k]$. The system is *quadratically stable* if all trajectories $x(t)$ starting from an arbitrary initial condition $x(0)$ converge to the origin as $t \to \infty$ and, in addition, there exists a quadratic function $V(x) = x^{\mathrm{T}} P x$, which decreases when evaluated at $x(t)$. Quadratic stability is equivalent to the existence of a k-by-k symmetric matrix P satisfying the following set of *Lyapunov inequalities* [31]:

$$\begin{aligned} A_i^{\mathrm{T}} P + P A_i < 0\ ,\ i \in \{1, 2...k\} \\ P > 0. \end{aligned} \tag{4.12}$$

The inequality signs correspond to *linear matrix inequalities*, or LMI. The notation $X > Y$ used in conjunction with matrices X and Y means that $X - Y$ is a *positive-definite* matrix. A positive-definite matrix P defines a quadratic function $f(x) = x^{\mathrm{T}} P x$ such that $f(x) > 0$ for all vectors $x \neq 0$. One test for positive-definiteness is given by the requirement that all eigenvalues of P be positive. Similarly, a negative-definite matrix defines a quadratic function that does not take positive values, and negative-definiteness is equivalent to the requirement that all eigenvalues of P be negative. For more details, the reader is referred to [32].

The quadratic stability of polytopic system (4.11) is readily evaluated in Matlab. Assuming that the vertices are available in the workspace as variables `A_1, A_2, ...A_k`, the following sequence of commands attempts to find P:

```
s1=ltisys(A_1)
s2=ltisys(A_2)
....
sk=ltisys(A_k)
polysys=psys([s1 s2 ... sk])
[tmin,P]=quadstab(polysys)
```

Quadratic stability can be conservative, as it is based on a *single* quadratic Lyapunov function defined by P. It is also possible to seek a piecewise-quadratic Lyapunov function to reduce conservativeness. The interested reader is referred, for instance, to Johansson [33] for more information.

We now focus on the polytopic description of (4.9). We wish to determine whether a state feedback gain K exists that results in quadratic stability of the closed-loop system resulting from applying the control $u = -Kx$. This is the *quadratic stabilizability* problem, which has also been formulated in terms of LMIs [31]. Quadratic stabilizability is equivalent to the existence of a symmetric, positive-definite matrix Q and a matrix Y such that

$$QA_i^{\mathrm{T}} + A_i Q - B_i Y - Y^{\mathrm{T}} B_i^{\mathrm{T}} < 0 \ , \ i \in \{1, 2...k\} \, . \tag{4.13}$$

The search for Q and Y given vertices A_i and B_i is formulated as an *LMI feasibility problem*. Feasibility can be evaluated in Matlab using the `feasp` command. Given feasible matrices Q and Y, a stabilizing feedback gain can be computed as $K = YQ^{-1}$. We defer details regarding the computation of feedback gains to Sects. 4.4 and below, where synthesis methods are considered that include robust stability and performance requirements.

4.3 Performance Measures

Consider again the feedback arrangement of Fig. 4.1. Performance outputs z_2 and z_∞ are chosen by designers to measure the effects of disturbances (vector w) on variables of interest. Denoting the transfer matrix from w to z_2 as $G_{w,z_2}(s)$, a measure of closed-loop system performance is given by the norm of $G_{w,z_2}(s)$: small values indicate that the system state and the computed control input are not very sensitive to disturbances. Minimizing the 2-norm of $G_{w,z_2}(s)$ corresponds to an optimal *linear quadratic control* problem to be discussed in the next section. Denoting the transfer matrix from w to z_∞ as $G_{w,z_\infty}(s)$, another measure of closed-loop system performance is given by the norm of $G_{w,z_\infty}(s)$. It should be noted that w can be extended to contain *reference commands* in addition to disturbances. Thus, if the control objective includes the requirement that the output y track a reference command $r(t)$, vector w can be defined as $w = [d \ \ r]^{\mathrm{T}}$, and good tracking accuracy will be obtained by minimizing the norm of $G_{w,z}(s)$, where z is defined as the tracking error: $z = r - y$. Minimizing the infinity norm of G_{w,z_∞} corresponds to an $\mathcal{H}_\infty$ *gain synthesis problem*, to be discussed in Sect. 4.6. The designer's role is to define z in a manner that best reflects the performance being sought.

4.4 LQR State Feedback Synthesis

Consider first the nominal system of (4.7), where A and B are fixed and no disturbance input is considered. Define a performance output as $z = Cx + Du$ such that $D^{\mathrm{T}}D$ is invertible. The *linear quadratic regulator problem*, or LQR, is to find the control $u(t)$ that minimizes a performance measure given by

$$J = \int_0^\infty z(t)^{\mathrm{T}} z(t) \mathrm{d}t. \tag{4.14}$$

Note that the value of J depends on the initial conditions $x(0)$. Under some conditions [28–30] which include stabilizability of (A, B), the solution takes the form of a state-feedback law $u(t) = -Kx(t)$ with a uniquely defined value of K as follows:

$$(D^{\mathrm{T}}D)^{-1}(B^{\mathrm{T}}P + D^{\mathrm{T}}C) = K, \tag{4.15}$$

$$A^{\mathrm{T}}P + PA - (PB + C^{\mathrm{T}}D)(D^{\mathrm{T}}D)^{-1}(B^{\mathrm{T}}P + D^{\mathrm{T}}C) + C^{\mathrm{T}}C = 0. \tag{4.16}$$

The optimal solution for K is the same regardless of $x(0)$. Equation (4.16) is known as the *algebraic Riccati equation*, and it yields a unique, positive-definite solution for P under the assumption that (A, B) is stabilizable. In Matlab, the Riccati equation can be solved with the `are` command. Alternatively, the `lqr` command computes P and the optimal gain K. Calling $Q = C^{\mathrm{T}}C$, $R = D^{\mathrm{T}}D$ and $N = C^{\mathrm{T}}D$, we see that $z^{\mathrm{T}}z = x^{\mathrm{T}}Qx + u^{\mathrm{T}}Ru + 2x^{\mathrm{T}}Nu$. Given an initial condition $x(0)$, a unique state trajectory $x(t)$ is obtained under the control $u = -Kx$. The term $x^{\mathrm{T}}Qx$ penalizes excessive deviations the components of $x(t)$ from zero, positive or negative. Since the term is under an integral sign and the integral takes only positive values, trajectories which converge to zero slowly will be penalized more than those rapidly approaching zero. Similar penalties are applied to the control vector u, but using a different weighting matrix. The designer can use Q and R to manage the tradeoff between a fast response without excessive peaking and the magnitude of the control effort required to produce it. The cross-term $2x^{\mathrm{T}}Nu$ is not easily linked to response performance features and may be safely omitted in most practical situations.

4.4.1 LQR with Regional Eigenvalue Constraints

A commonly-used tuning approach is to use diagonal Q and R matrices, individually penalizing state and control components by adjusting the diagonal entries of these matrices. This method is not always satisfactory, and many trial-and-error simulations could be needed. The designer may instead require that all closed-loop

eigenvalues have real parts less than a specified negative number γ. The time constants of the closed-loop eigenvalues can thus be made as small as desired, accelerating the speed of response. To do this, simply use $A + \gamma I$ in place of A when calculating the gain from (4.15) and (4.16) [34]. This method, however, does not restrict the imaginary part of the closed-loop poles, potentially leading to unacceptably low damping ratios. To address this limitation, a circular region with center at $-\gamma + 0i$ and radius ρ may be specified. The solution is found by using $\frac{1}{\rho}(A + \gamma I, B)$ as state-space description of a *discrete-time* LQR problem. For more details, the reader is referred to [34].

4.4.2 The Cheap LQR Problem and Performance Limits

Maintaining a low level of control effort is not crucial in certain problems. This may occur, for instance, when the scaling of matrix B is such that small values of u contribute significant changes to $\dot{x}$, and the allowable values of u – defined by actuator limits – are large. The performance measure used in conjunction with the cheap LQR problem is given by

$$J = \int_0^\infty x^{\mathrm{T}} Q x + \rho u^{\mathrm{T}} R u \, \mathrm{d}t, \tag{4.17}$$

where $\rho \to 0$. Recalling that $Q{=}C^{\mathrm{T}}C$, the achievable performance (i.e., the minimum attainable value of J) is influenced by the zero locations of the transfer matrix defined as $H(s) = C(sI - A)^{-1}B$. As shown in [35, 36], J will approach zero as $\rho \to 0$ if and only if $H(s)$ is minimum-phase. In this case, the minimum-phase zeroes are canceled by closed-loop poles in the limit, as $\rho \to 0$. When $H(s)$ contains non-minimum-phase zeroes, J has a nonzero minimum value and the closed-loop poles approach their mirror images relative to the imaginary axis. Any excess poles approach infinity along asymptotes that remain in the left-half of the complex plane.

The designer must therefore exercise care in choosing C to ensure that $H(s)$ is minimum-phase and performance improves uniformly as $\rho \to 0$.

4.4.3 LQR Robustness Properties

The solution for the optimal feedback gain K enjoys strong stability robustness properties. Specifically, an increasing gain margin of infinity, a decreasing gain margin of 50% and a phase margin of $\pm 60°$ are guaranteed for each control channel in conjunction with *any* choice of Q and R. That is, if we consider the SISO transfer functions from input components u_i to performance output components z_i, an unlimited gain increment or a 50% gain reduction will be tolerated without compromise to closed-loop stability. Similarly, a phase shift can be introduced (due

to unmodeled dynamics or signal delays) in a range of $\pm 60°$ without destabilizing the loop.

In the GTE control problem, such robustness properties are beneficial if actuator dynamics are to be ignored. Many actuators can be modeled as a first-order transfer function, introducing phase lags of up to 90°. Thus, if the closed-loop bandwidth is designed at appropriately small values, the phase lag due to actuator dynamics can be kept larger than $-60°$ and stability can be maintained. Similarly, the nominal gain of the actuator transfer function can be reflected in plant matrix B. Gain reductions of up to 50% due to uncertainties or changes to actuator dynamics will be tolerated.

4.4.4 Polytopic Systems

The previous sections assume that (A, B) is a fixed, known plant description. To design LQR controllers for the GTE under this assumption, a single representative steady operating point would have to be chosen, and a single gain K would be calculated for use in the entire operating envelope. If this were to be done, robust stability of the resulting polytopic system with vertices $(A_i - B_i K)$ could be readily evaluated using conditions (4.12). Even if stability is verified, performance could be unsatisfactory. A more reasonable approach is to use information about all vertices to seek a control law, which minimizes an integral quadratic cost of the form of (4.14). The optimal solution is given by a gain-scheduled state-feedback control $u = -K_i x$, where K_i are derived from a set of Riccati equations.

4.5 $\mathcal{H}_2$ State Feedback Synthesis

Consider the nominal system of (4.7) and suppose only performance output z_2 is considered in Fig. 4.1, defined as $z_2 = Cx + D_{zu}u + D_{zw}w$. The closed-loop system arising from the application of state-feedback control law $u = -Kx$ is given by

$$\dot{x} = (A - BK)x + \Gamma w,$$
$$z_2 = (C - D_{zu}K)x + D_{zw}w.$$

The transfer matrix from the exogenous input vector w to the performance output z_2 is then

$$G_{w,z_2}(s) = (C - D_{zu}K)(sI - (A - BK))^{-1}\Gamma + D_{zw}.$$

The $\mathcal{H}_2$ *feedback gain synthesis problem* is to be formulated as

Find K so that $(A - BK)$ is stable and $||G_{w,z_2}||_2 < \nu$ for some $\nu > 0$.

That is, we seek a stabilizing feedback gain that maintains the influence of the exogenous inputs on the performance outputs below a prescribed level ν. For example, choosing z_2 to be the incremental HPC stall margin Δ SmHPC would result in a closed-loop controller, which minimizes the sensitivity of stall margin to health parameter changes, in addition to rendering the closed-loop system stable.

4.5.1 Optimal $\mathcal{H}_2$ Synthesis

In an $\mathcal{H}_2$ synthesis problem, a prescribed norm value ν is targeted by solving the following LMI feasibility problem in variables X, Q, and K:

$$\begin{bmatrix} (A-BK)X + X(A-BK)^{\mathrm{T}} & \Gamma \\ \Gamma^{\mathrm{T}} & -I \end{bmatrix} < 0, \tag{4.18}$$

$$\begin{bmatrix} Q & (C-D_{zu}K)X \\ X^{\mathrm{T}}(C-D_{zu}K)^{\mathrm{T}} & X \end{bmatrix} > 0, \tag{4.19}$$

$$\mathrm{trace}(Q) < \nu^2. \tag{4.20}$$

A numerical solution to the above LMI problem is incorporated in Matlab's `msfsyn` command to be discussed in the example at the end of this chapter. An *optimal* $\mathcal{H}_2$ state feedback synthesis procedure is to attempt the LMI feasibility problem repeatedly, with decreasing values of ν, until the problem is no longer feasible.

4.5.1.1 LQR and $\mathcal{H}_2$ Equivalence

When all states are available for measurement, the $\mathcal{H}_2$ state feedback synthesis problem is equivalent to an LQR problem. Further, when only a vector of noisy measurements is available, the $\mathcal{H}_2$ problem is equivalent to a *linear quadratic Gaussian*, or LQG problem, whose solution is given by the combination of a state estimator (Kalman filter) and a state feedback law computed on the basis of the state estimates. Here, we focus in the *state measurement feedback* case, where all states are available for computation of the feedback law. Recalling that $||G_{w,z_2}||_2$ is finite only when the transfer matrix is strictly proper, we only consider the case D_{zw}=0. It can be shown [28] that the optimal $\mathcal{H}_2$ state feedback synthesis problem with plant matrices (A, B, Γ) and performance output matrices (C, D_{zu}) is equivalent to an LQR problem with Q=$C'C$, R=$D'_{zu}D_{zu}$ and N=$C'D_{zu}$. The gain K obtained as the solution of the LQR problem will be the same as the one resulting from $\mathcal{H}_2$ optimization regardless of Γ, but the optimum norm will be affected by it.

To illustrate the equivalence in detail, consider the following example system

$$A = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} -1 \\ 3 \end{bmatrix},$$
$$C = \begin{bmatrix} 1 & 1 \\ -1 & 2 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 1 \\ 2 & 2 \end{bmatrix}.$$

We leave it to the reader to verify that the solution to the LQR problem with $Q = C'C$, $R = D'_{zu}D_{zu}$, and $N = C'D_{zu}$ is given by

$$K = \begin{bmatrix} 1.13 & 0.59 \\ -1.01 & 0.54 \end{bmatrix}$$

and that the solution to the optimum $\mathcal{H}_2$ gain synthesis problem (see following section) gives the same K as above regardless of Γ. If the given Γ is used, the optimum cost is 0.4484.

4.5.2 Polytopic Systems

The $\mathcal{H}_2$ state feedback synthesis problem for polytopic systems is also tractable and can be solved through LMI feasibility. The definition of 2-norm of (4.5) is no longer appropriate for polytopic systems. Instead, the equivalent LQG definition of 2-norm is used, taking the maximum over all instances of system matrices $(A(\alpha), B(\alpha), C(\alpha))$:

$$||G||_2^2 = \max_{\alpha} \lim_{T\to\infty} E\left\{\frac{1}{T}\int_0^T y^T y \, dt\right\}, \tag{4.21}$$

where y is the output produced by polytopic system G when the input is a white noise with unit covariance. Thus, a *worst-case* approach is adopted when calculating the 2-norm of a polytopic system. A prescribed 2-norm can be targeted by solving an LMI feasibility problem whose complexity increases with the number of vertices. For further details, the reader is referred to [31]. The `msfsyn` command in Matlab admits polytopic system descriptions.

4.6 $\mathcal{H}_\infty$ State Feedback Synthesis

The $\mathcal{H}_2$ synthesis approach delineated above seeks to maintain the influence of exogenous inputs on a designer-defined performance output below a prescribed value. The 2-norm is used as a measure of the "strength" of this influence. The

$\mathcal{H}_\infty$ *state feedback synthesis* problem described here has the same objective, with the distinction that the infinity norm is used. Also, the definition of the performance output admits a nonzero D_{zw} term. Consider the nominal system of (4.7) and a performance output z_∞=$Cx+D_{zu}u+D_{zw}w$. The transfer matrix from the exogenous input vector w to the performance output z_∞ is then

$$G_{w,z_\infty}(s) = (C - D_{zu}K)(sI - (A - BK))^{-1}\Gamma + D_{zw}.$$

The $\mathcal{H}_\infty$ *feedback gain synthesis problem* is formulated as

Find K so that $(A - BK)$ is stable and $||G_{w,z_\infty}||_\infty < \gamma$ for some $\gamma > 0$.

The corresponding LMI feasibility problem is formulated as

$$\begin{bmatrix} (A - BK)X + X(A - BK)^{\mathrm{T}} & \Gamma & X(C - D_{zu})^{\mathrm{T}} \\ \Gamma^{\mathrm{T}} & -I & D_{zw}^{\mathrm{T}} \\ (C - D_{zu}K)X & D_{zw} & -\gamma^2 I \end{bmatrix} < 0, \tag{4.22}$$

$$X > 0. \tag{4.23}$$

As in the $\mathcal{H}_2$ case, an *optimal* $\mathcal{H}_\infty$ state feedback synthesis procedure is to attempt the LMI feasibility problem repeatedly, with decreasing values of γ, until the problem is no longer feasible. This process, commonly known as γ*-iteration*, is incorporated in Matlab's `hinfsyn` command. Alternatively, `msfsyn` can be used. The latter command also works with polytopic systems.

4.6.1 Polytopic Systems

The infinity norm of a polytopic system G is regarded as the maximum time-invariant norm taken over all instances of system matrices $(A(\alpha), B(\alpha), C(\alpha), D(\alpha))$ with transfer function $G_\alpha(s)$:

$$||G||_\infty = \max_\alpha \sup_w \bar{\sigma}\left\{G_\alpha(jw)\right\}. \tag{4.24}$$

As in the $\mathcal{H}_2$ case, a prescribed infinity norm can be targeted by solving an LMI feasibility problem whose complexity increases with the number of vertices. The `msfsyn` command in Matlab admits polytopic system descriptions. For further details, the reader is referred to [31] and Matlab's Robust Control Toolbox documentation.

4.7 $\mathcal{H}_2/\mathcal{H}_\infty$ Feedback Synthesis with Regional Pole Placement

The reader may question the need for the $\mathcal{H}_2$ synthesis approach, given that $\mathcal{H}_\infty$ synthesis pursues the same general objective, and does not have the restriction $D_{zw} = 0$. An intuitive justification for considering both norms can be found in their input/output interpretations. Minimizing the infinity norm amounts to reducing the peak of the closed-loop magnitude response across all frequencies. By design, the magnitude response for frequencies other than the one corresponding to the peak magnitude will also be suppressed. When high-frequency actuator dynamics are left unmodeled, good tracking and disturbance rejection calls for more selective control actions according to frequency: the controller should have an enhanced reaction to low-frequency error components, while being more insensitive to high frequency components. Subjecting all frequencies to the same peak magnitude constraint is conservative, and may lead to poor results. For this reason, frequency shaping is generally a necessary step in $\mathcal{H}_\infty$ methods. In contrast, the definition of 2-norm given in (4.21) indicates that minimizing $||G_{w,z_2}||_2$ implies reducing the *time average* (in a root-mean-square sense) of the performance output z_2. The definition given in (4.5) shows that a frequency-average is also implied. The relationship to LQR control and its integral quadratic performance measure gives the $\mathcal{H}_2$ approach a time-domain interpretation, allowing the designer to manage the performance vs. control effort constraint in a direct way. The reader wishing to examine the theoretical bases of $\mathcal{H}_\infty$ and $\mathcal{H}_2$ control in detail is referred to standard material, for instance [28–30]. These references also provide detailed coverage of frequency-shaped $\mathcal{H}_\infty$ methods.

It is possible to consider $\mathcal{H}_2$ and $\mathcal{H}_\infty$ synthesis objectives simultaneously by defining two sets of performance outputs z_2 and z_∞. The designer can balance the importance of each minimization by defining a weighted objective of the form

$$M = a||G_{w,z_\infty}||^2 + b||G_{w,z_2}||^2, \tag{4.25}$$

where a and b are non-negative weights. In the polytopic case, the norms are understood to be worst-case values, i.e., the maximum over all instances of system matrices. Again, minimization of the weighted objective is carried out under LMI feasibility constraints [37]. Unlike direct pole-placement design, norm-based minimization as presented until now does not provide a mechanism to enforce a location constraint for the closed-loop poles. Such *regional eigenvalue constraint* can be incorporated as an added LMI constraint, and the minimization problem remains tractable under some relaxing assumptions, leading to suboptimal solutions. The `msfsyn` command can be used to find a suboptimal solution to the mixed norm minimization problem, with the option of including a regional eigenvalue placement constraint. Placement regions in the complex plane can be of various shapes. The designer can obtain generally faster response times by choosing a constraint of the form $\mathrm{Re}(s) < p$, where s denotes a closed-loop pole and p is a negative number. When the constraint is fulfilled, every closed-loop pole has a time constant lesser

than $1/|p|$. A circular region also be used to prevent poles having excessively low damping. The example at the end of this chapter illustrates the use of `msfsyn` in detail, as pertaining to the GTE control problem.

4.7.1 State-Feedback Setpoint Regulation and Input Integration

The state-feedback control law $u = -Kx$ is designed to regulate the plant state from an arbitrary initial condition $x(0)$ to the origin. This law is adequate for GTE control problems where fan speed must be driven from one steady point to another. Indeed, recall that the linearized plant state is given by fan and core speed deviations from a steady operating state $[\bar{N}_\mathrm{f} \ \bar{N}_\mathrm{c}]^\mathrm{T}$: $x = [\Delta N_\mathrm{f} \ \ \Delta N_\mathrm{c}]^\mathrm{T}$, where $\Delta N_\mathrm{f} = N_\mathrm{f} - \bar{N}_\mathrm{f}$ and $\Delta N_\mathrm{c} = N_\mathrm{c} - \bar{N}_\mathrm{c}$. In nonlinear simulation or actual realtime operation, x is formed by shifting the sensed N_f and N_c by the intended reference states $\bar{N}_\mathrm{f}$ and $\bar{N}_\mathrm{c}$. Thus, $x = 0$ and $u = -Kx = 0$ at the beginning of the maneuver. Since u is the incremental actuator input, i.e.,$u^\mathrm{T} = [W_\mathrm{F} - \overline{W}_\mathrm{F} \ \ VSV - \overline{VSV} \ \ VBV - \overline{VBV}]$, constant inputs $\overline{W}_\mathrm{F}$, $\overline{VSV}$ and $\overline{VBV}$ are being applied to the engine at the initial time. A setpoint change is produced when the reference states and inputs are suddenly changed to values defining a new steady operating point. The state will no longer be zero and regulatory control action will begin according to $u = -Kx$. If the closed-loop system is stable, x and u will be driven to zero again. The actual inputs applied to the engine must be shifted by their references, that is, $[W_\mathrm{F} \ \ VSV \ \ VBV]^\mathrm{T} = -Kx + [\overline{W}_\mathrm{F} \ \ \overline{VSV} \ \ \overline{VBV}]^\mathrm{T}$. *The control system must thus include a pre-calculated list of the required references for each desired pair of fan and core speed setpoints so that the sensed variables and the calculated control can be appropriately shifted*. Figure 4.2 shows the structure of the control law as applied to the nonlinear engine. From a nominal linear model standpoint, a steady operating point is defined by the equation

$$0 = A\bar{x} + B\bar{u}.$$

When A has no poles at zero (as is the case for the open-loop engine), the equilibrium state corresponding to a constant input $\bar{u}$ can be calculated as

$$\bar{x} = -A^{-1}B\bar{u}. \tag{4.26}$$

The equation for the reverse operation is

$$B\bar{u} = -A\bar{x}. \tag{4.27}$$

If there are less inputs than states, i.e., if $m < n$, system (4.27) is underdetermined and infinitely many solutions for u exist. If $m = n$ and B is invertible, a unique solution can be found. If $m > n$ the system is overdetermined and solutions for $\bar{u}$ exist if $A\bar{x}$ is in the column span of B. This condition cannot be assumed to hold

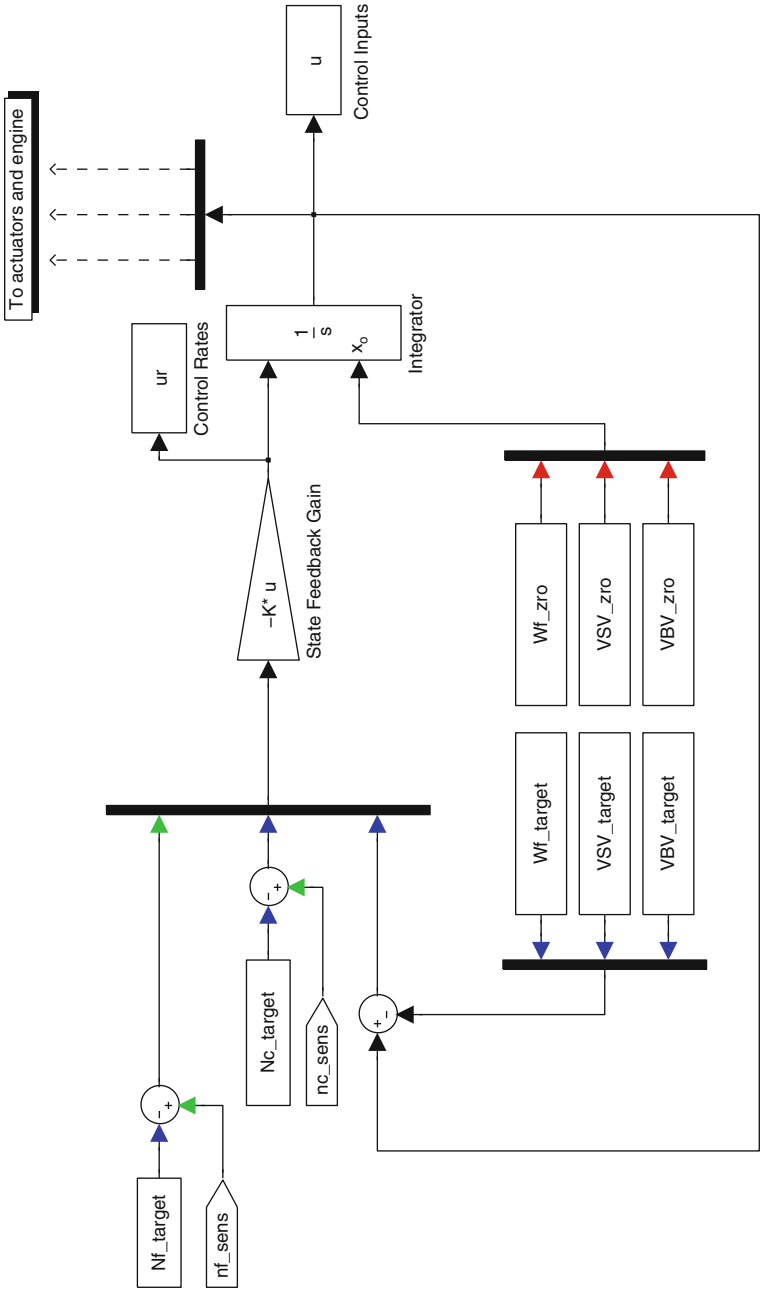

Fig. 4.2 State feedback with input integration in nonlinear engine simulation

in general. When considering three control inputs and two states for the GTE, no solutions to (4.27) can be assumed to exist for an arbitrary choice of $\bar{x}$. In addition to the structural difficulty arising from the use of three actuators with a 2-state plant, the reader must keep in mind that linearized models are only approximations to the actual nonlinear engine. The calculations for the triples $(\overline{W}_{\mathrm{F}}, \overline{VSV}, \overline{VBV})$ that yield a desired pair $(\bar{N}_{\mathrm{f}}, \bar{N}_{\mathrm{c}})$ must be performed using an accurate steady-state solver for the nonlinear engine.

Disturbances and intrinsic engine nonlinearity limit the tracking accuracy of the state feedback law $u = -Kx$. The effect of disturbances on the regulated variables can be minimized by using the synthesis approaches of this chapter, but not eliminated entirely. The ability of the control system to track varying reference commands and reject disturbances can be enhanced by including integral control action. As mentioned in Chap. 3, constant disturbances produce zero steady-state error if the control loop is of type 1, that is, if it includes a free integrator. Input integration is commonly-used in actual GTE control implementations and is found in the built-in controllers designed for CMAPSS. To achieve integral control, the following *augmented plant* description is used:

$$\dot{x} = Ax + Bu + \Gamma w, \tag{4.28}$$

$$\dot{u} = u_{\mathrm{r}}, \tag{4.29}$$

which can be compactly described as

$$\dot{x}_{\mathrm{a}} = A_{\mathrm{a}} x_{\mathrm{a}} + B_{\mathrm{a}} u_{\mathrm{r}} + \Gamma_{\mathrm{a}} w, \tag{4.30}$$

where the u_{r} is the new control input and the augmented state vector and matrices are defined as $x_{\mathrm{a}} = [x^{\mathrm{T}} \; |u^{\mathrm{T}}]^{\mathrm{T}}$ and

$$A_{\mathrm{a}} = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}, \quad B_{\mathrm{a}} = \begin{bmatrix} 0 \\ I \end{bmatrix}, \quad \Gamma_{\mathrm{a}} = \begin{bmatrix} \Gamma \\ 0 \end{bmatrix}. \tag{4.31}$$

When $(A(\alpha), B(\alpha), \Gamma(\alpha))$ are polytopic, the vertices of the corresponding polytopic augmented system can be directly computed. Moreover, the augmented polytopic system remains quadratically stabilizable when the original system is so. Hence, the various norm-based gain synthesis approaches can be applied without restrictions.

4.8 Example: CMAPSS-40k

In this section, we consider the fan speed control problem assuming that a fixed-gain feedback law of the form $u = -Kx$ is to be used across the flight envelope. The model contained in CMAPSS-40k is used for linearized plant extraction and

for realistic nonlinear simulations. We begin by choosing a six-vertex polytopic description of the linearized engine to match representative flight regimes. We then conduct a quadratic stabilizability evaluation considering input integration and then carry out several designs, observing the following:

1. Setpoint changes near different flight conditions: the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ designs with input integration are compared in their ability to produce consistently good responses near different operating regimes.
2. Effects of engine health status and faults: the disturbance rejection capabilities of the various designs are compared.

4.8.1 A Polytopic Description for the 40k Engine

Let the vector of incremental control inputs be defined as

$$u = [\Delta W_{\mathrm{F}} \;\; \Delta VSV \;\; \Delta VBV]^{\mathrm{T}}.$$

A six-vertex polytopic description $(A(\alpha), B(\alpha), \Gamma(\alpha))$ can be constructed by linearizing the engine at the six representative conditions shown in Table 2.4. The corresponding linearized system matrices are listed in Appendix C. Two outputs have been selected for the purposes of this example: the temperature at HPT outlet (T48) and the HPC stall margin (SM-HPC). The corresponding C, D, and Λ matrices used as output vertices are also listed in Appendix C.

4.8.2 Stabilizability of the Scaled Augmented Plant

The augmented vertices $A_{\mathrm{ai}}, B_{\mathrm{ai}}$ necessary to design an integral controller are obtained directly from (4.31). Direct use augmented plant matrices in the LMI feasibility problem of (4.13) may pose numerical difficulties due to disparate scaling of A_{a} and B_{a}. Indeed, B_{a} is made up of only zeroes and ones, while A_{a} contains entries ranging from zero to thousands. To avoid numerical problems, it is advisable to obtain a *balanced realization* through similarity transformation. Defining $z_{\mathrm{a}} = T x_{\mathrm{a}}$, T is selected so that the transformed pair of matrices $A_{\mathrm{bal}} = T A_{\mathrm{a}} T^{-1}$ and $B_{\mathrm{bal}} = T B_{\mathrm{a}}$ generate a matrix $[A_{\mathrm{bal}} \;\; B_{\mathrm{bal}}]$ with approximately the same row and column norms. This process is implemented in Matlab's `ssbal` command. Alternatively, the `balreal` command can be used. A transformation matrix T is obtained using an arbitrary augmented vertex and then used to transform the remaining vertices. This is a reasonable approach since all vertices share a similar scaling. Assuming that the augmented vertices are available in the workspace as

`Aa1`, `Aa2`, etc. and `Ba` (constant for all vertices), the following code can be used to evaluate quadratic stabilizability:

```
%Obtain balanced realizations
Cdummy=[1 0 0 0 0];Ddummy=zeros(1,3);
sys=ss(Aa1,Ba,Cdummy,Ddummy);
[sysbal,G,T,Ti]=balreal(sys);
[Abal1,Bbal]=ssdata(sysbal);
%Use the same transform for all augmented systems
Abal2=T*Aa2*Ti;Abal3=T*Aa3*Ti;
Abal4=T*Aa4*Ti;Abal5=T*Aa5*Ti;Abal6=T*Aa6*Ti;
%Prepare LMIs
setlmis([]);Q=lmivar(1,[5 1]);Y=lmivar(2,[3 5]);
lmiterm([-1 1 1 Q],1,1);
lmiterm([2 1 1 Q],1,Abal1',’s’);
lmiterm([2 1 1 Y],-Bbal,1,’s’);
lmiterm([3 1 1 Q],1,Abal2',’s’);
lmiterm([3 1 1 Y],-Bbal,1,’s’);
lmiterm([4 1 1 Q],1,Abal3',’s’);
lmiterm([4 1 1 Y],-Bbal,1,’s’);
lmiterm([5 1 1 Q],1,Abal4',’s’);
lmiterm([5 1 1 Y],-Bbal,1,’s’);
lmiterm([6 1 1 Q],1,Abal5',’s’);
lmiterm([6 1 1 Y],-Bbal,1,’s’);
lmiterm([7 1 1 Q],1,Abal6',’s’);
lmiterm([7 1 1 Y],-Bbal,1,’s’);
quad0=getlmis;
 %Now run feasibility problem
[tmin3,xfeas]=feasp(quad0);
%Extract feasible sols
Qfeas=dec2mat(quad0,xfeas,Q);Yfeas=dec2mat(quad0,xfeas,Y);
%Calculate a feedback gain
KT=Yfeas*inv(Qfeas);
%Restore KT to original coordinates
K=KT*T;
```

The reader can verify that a feasible Q is found and that K yields stable closed-loop vertices $(A_{\mathrm{ai}} - B_{\mathrm{ai}}K)$. Of course, the same K yields closed-loop stability for arbitrary system matrix variations within the polytope.

Note: Many books, research articles, and software assume a control input of the form $u = Kx$ rather than $u = -Kx$. Matlab's robust state feedback synthesis commands are not an exception. Using $-B$ and $-D$ in place of B and D resolves the discrepancy. This modification has been already incorporated in the code examples presented in this chapter.

4.8.3 Fixed-Gain LQR Design

An LQR gain can be found using a fixed vertex, followed by a closed-loop quadratic stability verification. As an example, the scaled vertex `Abal1,Bbal` is chosen

arbitrarily, along with unit LQR weights: $Q = I_5$ and $R = I_3$. In the following code, the resulting gain is used to calculate the closed-loop vertices and evaluate quadratic stability using `quadstab`:

```
%Set Q and R
Q=eye(5);R=eye(3);
[KT,S,E]=lqr(Abal1,Bbal,Q,R);

%Restore KT to original coordinates
K=KT*T;
%Form polytopic closed-loop system in original coordinates
sys1=ltisys(Aa1-B*K);sys2=ltisys(Aa2-B*K);
sys3=ltisys(Aa3-B*K);sys4=ltisys(Aa4-B*K);
sys5=ltisys(Aa5-B*K);sys6=ltisys(Aa6-B*K);

polysys=psys([sys1,sys2,sys3,sys4,sys5,sys6]);
[tmin,P]=quadstab(polysys)
```

The closed-loop polytopic system is verified to be quadratically stable. The fixed gain

$$K = \begin{bmatrix} -0.0108 & -0.0021 & 40.2654 & 0.3927 & -0.2583 \\ 0.0037 & -0.0056 & 0.3927 & 4.2430 & -0.0294 \\ 0.0010 & 0.0003 & -0.2583 & -0.0294 & 4.0187 \end{bmatrix} \tag{4.32}$$

produces the closed-responses shown in Fig. 4.3 when applied to the individual vertices. Although stability is guaranteed, a significant spread of response times is observed.

4.8.4 Fixed-Gain LQR in CMAPSS-40k

The gain K in (4.32) is now tested against the nonlinear engine in CMAPSS-40k. The observations made in Sect. 4.7.1 regarding controller deployment apply. The overall control law takes the form

$$u_\mathrm{r} = -Kx_\mathrm{a},$$

$$W_\mathrm{f} = \int_{W_\mathrm{f}(0)} u_\mathrm{r}(1), \tag{4.33}$$

$$VSV = \int_{VSV(0)} u_\mathrm{r}(2), \tag{4.34}$$

$$VBV = \int_{VBV(0)} u_\mathrm{r}(3), \tag{4.35}$$

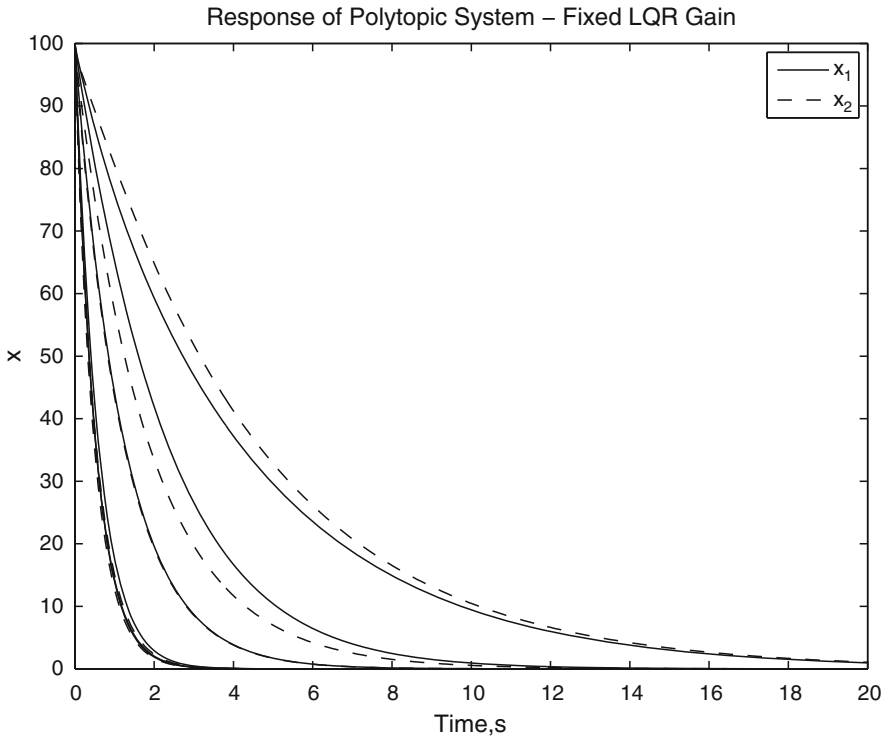

Fig. 4.3 Response of vertices with fixed LQR gain

where $x_a^T = [\bar{N}_f - N_f \;\; \bar{N}_c - N_c \;\; \overline{W}_F - W_F \;\; \overline{VSV} - VSV \;\; \overline{VBV} - VBV]$. Since the control input vector is part of the augmented state, the set of target steady values $\overline{W}_F$, $\overline{VSV}$ and $\overline{VBV}$ must be supplied to the controller. This implies that a database of such values must exist, as generated by the nonlinear engine steady-state solver. Each steady-state input triple $\bar{U} = [\overline{W}_F \; \overline{VSV} \; \overline{VBV}]$ yields a corresponding pair of steady states $\bar{X} = [\bar{N}_f \; \bar{N}_c]$ through some mapping $\bar{X} = \Phi(\bar{U})$, which is affected by inlet conditions and health parameter values. This mapping is accessible only through high-fidelity nonlinear simulation, and cannot be fully captured through functions or tables due to the uncertain nature of health parameters. To conduct the simulations of this section, the nonlinear engine was driven to a steady-state using a "native" controller supplied with CMAPSS-40k. The native controller only uses W_F as a feedback-controlled input, while VSV and VBV are injected in a feedforward fashion. The steady values of the three actuators were adopted as reference inputs for this example.

Figure 4.4 shows the response of the nonlinear engine to positive step demands in fan and core speeds. The starting operating regime is Ground Idle, labeled as "A" in Table 2.4. The settling time for the state variables is about 3 s, with no overshoot. Figure 4.5 shows the response to negative step demands in fan and core speeds. The

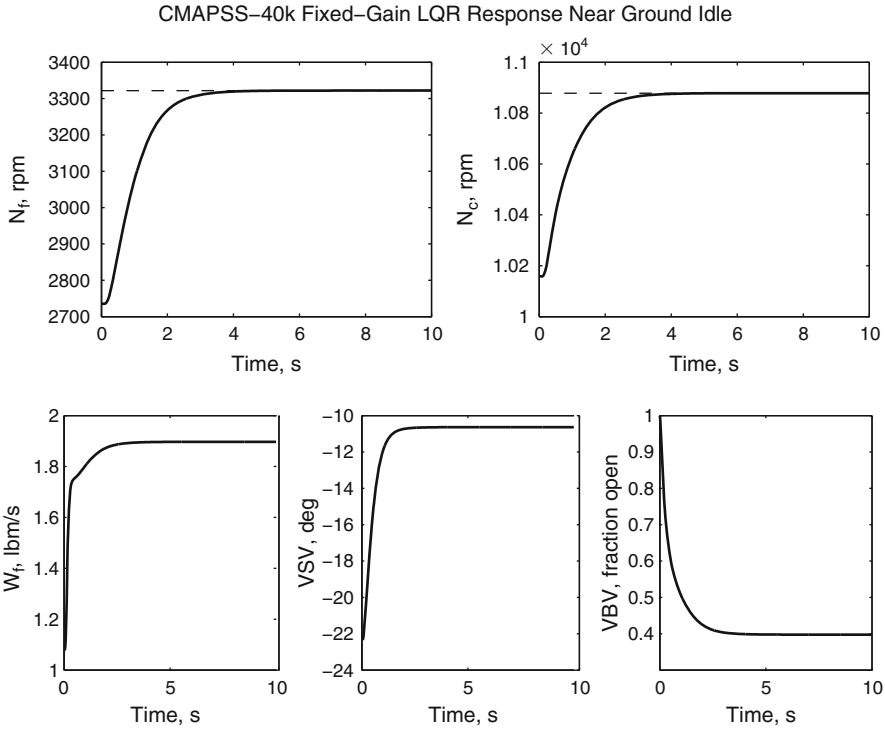

Fig. 4.4 CMAPSS-40k response with fixed LQR gain near Ground Idle

starting operating regime is Max Cruise, labeled as "D" in Table 2.4. The settling time for the state variables is now about 8 s, with no overshoot. In summary, this fixed LQR gain provides stability but is unable to maintain response speed across operating regimes. This, coupled with the need for precalculated steady references limits the applicability of integrator-based fixed-gain approaches.

4.8.5 $\mathcal{H}_2/\mathcal{H}_\infty$ Fixed Gain Synthesis: Polytopic Plant Model

We now apply the $\mathcal{H}_2$, $\mathcal{H}_\infty$ and mixed synthesis methods to the six-vertex polytopic plant and test the corresponding gains in CMAPSS-40k, comparing the results with those obtained with the "native" fan-speed controller distributed with the package. The native controller is a PI compensator around fan speed, with scheduled gains. The other two control inputs, namely VSV and VBV, are scheduled in an open-loop fashion, that is, without feedback from the state variables. Chapter 5 discusses gain scheduling in detail. We define two performance outputs: the HPT outlet temperature T_{48} and the HPC stall margin SmHPC. Allowable limits are

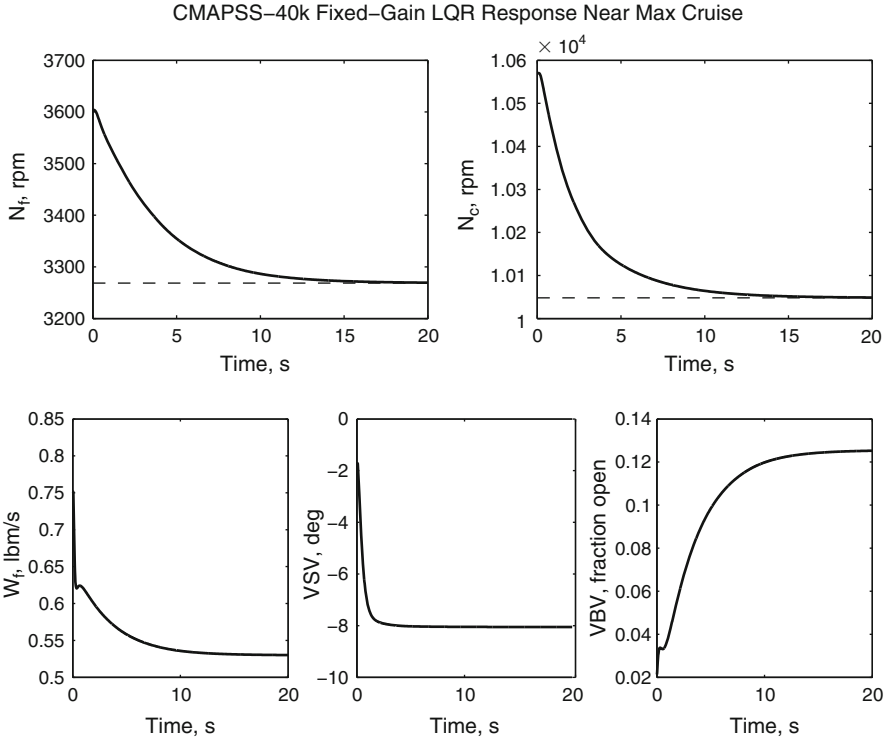

Fig. 4.5 CMAPSS-40k response with fixed LQR gain near Max Cruise

usually specified for these outputs. At the same time, these outputs tend to peak during transients. Moreover, certain health parameter inputs result in stall margin reductions. Continuing with the integral control approach, we define

$$z = [\Delta T_{48} \;\; \Delta SmHPC]^{\mathrm{T}} = C_{\mathrm{a}}(\alpha)x_{\mathrm{a}} + D_{zw}(\alpha)w + D_{zu}u_{\mathrm{r}}.$$

Augmented vertex matrices C_{ai} and D_{izw} are constructed from the data in Appendix C according to:

$$C_{\mathrm{ai}} = [C_i \mid D_i],$$
$$D_{\mathrm{izw}} = \Lambda_i .$$

Matrices C_i, D_i and Λ_i are formed for each vertex by stacking the corresponding C, D and Λ matrices for T_{48} and SmHPC. Note also that $D_{zu} = 0$, since the performance outputs are independent of the new control vector u_{r}, and that the vertices of the augmented disturbance input matrix are given by $\Gamma_{\mathrm{ai}} = [\Gamma_i^{\mathrm{T}} \mid 0]^{\mathrm{T}}$.

According to the synthesis method used in the example, we will regard both components of z as $\mathcal{H}_2$ performance outputs, both as $\mathcal{H}_\infty$ performance outputs, or a mix. Assuming that the augmented vertices already exist in the workspace as `Aa1`, `Aa2`, etc., `Ba` (constant for all vertices), `Ca1`,`Ca2`, etc., `Da` (zero for all vertices), `Gammaa1`,`Gammaa2`, etc. and `Lambda1`,`Lambda2`, etc., the following Matlab code creates the polytopic description with 2 performance outputs, using a balanced realization:

```
sys=ss(Aa1,Ba,Ca1,Da);
[sysbal,G,T,Ti]=balreal(sys);
[Abal1,Bbal,Cbal1,Dbal]=ssdata(sysbal);
Gammabal1=T*Gammaa1;
%Apply same balancing to all vertices
Abal2=T*Aa2*Ti;Cbal2=Ca2*Ti;Gammabal2=T*Gammaa2;
Abal3=T*Aa3*Ti;Cbal3=Ca3*Ti;Gammabal3=T*Gammaa3;
Abal4=T*Aa4*Ti;Cbal4=Ca4*Ti;Gammabal4=T*Gammaa4;
Abal5=T*Aa5*Ti;Cbal5=Ca5*Ti;Gammabal5=T*Gammaa5;
Abal6=T*Aa6*Ti;Cbal6=Ca6*Ti;Gammabal6=T*Gammaa6;
%Form polytopic system compatibly with msfsyn command:
%2 performance outs: [T48 SM-HPC]
sys1=ltisys(Abal1,[Gammabal1 -Bbal],Cbal1,[zeros(2,13) -Dbal]);
sys2=ltisys(Abal2,[Gammabal2 -Bbal],Cbal2,[zeros(2,13) -Dbal]);
sys3=ltisys(Abal3,[Gammabal3 -Bbal],Cbal3,[zeros(2,13) -Dbal]);
sys4=ltisys(Abal4,[Gammabal4 -Bbal],Cbal4,[zeros(2,13) -Dbal]);
sys5=ltisys(Abal5,[Gammabal5 -Bbal],Cbal5,[zeros(2,13) -Dbal]);
sys6=ltisys(Abal6,[Gammabal6 -Bbal],Cbal6,[zeros(2,13) -Dbal]);
polysys=psys([sys1,sys2,sys3,sys4,sys5,sys6]);
```

The reader should refer to Matlab documentation for details on the plant structure assumed by `msfsyn`.

4.8.5.1 $\mathcal{H}_2$ Synthesis with Regional Pole Placement

When specific penalization of control effort is not included, the $\mathcal{H}_2$-norm minimization may lead to large K and unreasonably high closed-loop bandwidths. It is possible to add a regional eigenvalue constraint to the minimization objective. In this example, both performance outputs are regarded as z_2. Note that the vertex systems have $D_{zu} = 0$ (the `zeros(2,13)` entries) so that the 2-norm is finite. The following code executes the minimization:

```
r=[2 3]; %# of z2 outputs and # of controls
obj=[0 0 0 1];
region=lmireg %choose disk with center -8, radius 6
[gopt,hopt,KT,Pcl]=msfsyn(polysys,r,obj,region);
K=KT*T;
```

Closed-loop eigenvalues are required to lie in a disk centered at −8+0i with a radius of 6. Attempts to reduce the radius and shift the center to obtain faster eigenvalues eventually leads to unfeasibility. The optimizing gain is

$$K = \begin{bmatrix} 0.0058 & 0.0055 & 10.8379 & 0.2571 & -0.0827 \\ -0.2535 & 0.0475 & -9.8546 & 9.5691 & 1.2961 \\ -0.9121 & 0.3095 & -9.7661 & 8.4027 & 12.3622 \end{bmatrix}.$$

The lowest bound for $||G_{w,z_2}(s)||_2$ returned by the program is 1.172×10^4. The order of magnitude of this quantity is linked to problem data and should not be evaluated in an absolute sense. The norm bound is useful to compare designs sharing the same performance outputs but having different eigenvalue regions. For instance, if a disk centered at −10+0i with a radius of 10 is chosen, the reader can verify that the optimum norm bound reduces to 900.27. This can be linked to a faster center for the closed-loop eigenvalues and an enlarged constraint region, as reflected in the radius specification.

Figures 4.6 and 4.7 show the responses to the same setpoint change considered in the LQR example, near Ground Idle conditions. The eigenvalue placement constraints allows to tune the controller for a faster response, matching the one obtained with the scheduled compensator. A health parameter step disturbance is injected through w at t =15 s. All components of w are equal to −0.1. Figure 4.6 shows that the ability to hold the setpoint in the presence of disturbance is roughly the same for the native controller and the $\mathcal{H}_2$ design, except for N_c, where the $\mathcal{H}_2$ design holds N_c closer to its regulation value. This is expected, since N_c is not under feedback with the native controller. Figure 4.7 shows that while the fuel flow commands are similar, the VSV and VBV inputs are not. Note that VSV saturates at the fully-open position under the $\mathcal{H}_2$ design.

4.8.5.2 Fixed-Gain $\mathcal{H}_\infty$ Design

The $\mathcal{H}_\infty$-norm minimization with a regional eigenvalue constraint is now carried out. Both performance outputs are regarded as z_∞. The `zeros(2,13)` entries in the code for the $\mathcal{H}_2$ case are changed to reflect the influence of w on the performance outputs. If Λ is used, it is difficult to find a feasible solution with the required bandwidth. Instead, the design uses 0.5Λ. Thus, the vertex definition lines must be changed to

```
sys1=ltisys(Abal1,[Gammabal1 -Bbal],Cbal1,[0.5*Lambda1 -Dbal]);
...
sys6=ltisys(Abal6,[Gammabal6 -Bbal],Cbal6,[0.5*Lambda6 -Dbal]);
polysys=psys([sys1,sys2,sys3,sys4,sys5,sys6]);
```

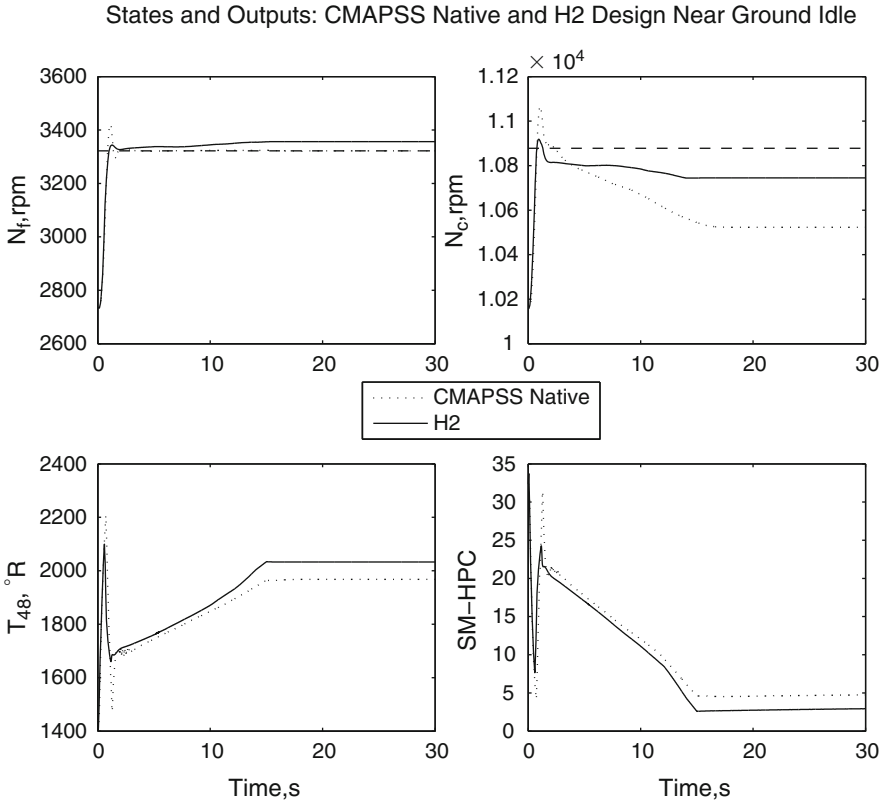

Fig. 4.6 Comparison of responses between CMAPSS-40k gain-scheduled fan speed compensator and fixed-gain $\mathcal{H}_2$ state feedback near Ground Idle: states and outputs. A step disturbance of -0.1 is applied to all health inputs at $t = 15$ s

The following code is then used to regard both performance outputs as z_∞ and carry out the minimization:

```
r=[0 3]; %# of z2 outputs and # of controls
obj=[0 0 1 0];
region=lmireg %choose disk with center -10, radius 10
[gopt,hopt,KT,Pcl]=msfsyn(polysys,r,obj,region);
K=KT*T;
```

Closed-loop eigenvalues are required to lie in a disk centered at $-10+0$i with a radius of 10. The optimizing gain is

$$K = \begin{bmatrix} -0.0013 & -0.0036 & 13.9596 & -0.0877 & 0.0374 \\ -0.0179 & 0.2482 & 231.1991 & 23.6696 & -2.4600 \\ -0.1047 & 0.2329 & 227.2949 & 8.6254 & 12.9515 \end{bmatrix}.$$

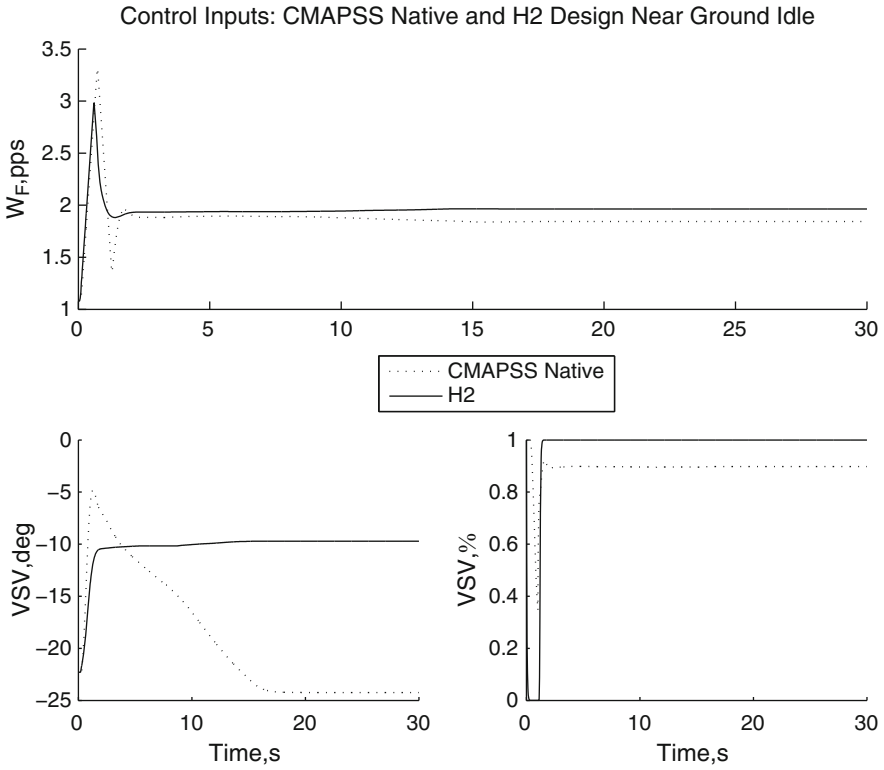

Fig. 4.7 Comparison of responses between CMAPSS-40k gain-scheduled fan speed compensator and fixed-gain $\mathcal{H}_2$ state feedback near Ground Idle: control inputs. A step disturbance of -0.1 is applied to all health inputs at $t = 15$ s

The lowest bound for γ returned by the program is 925.3. Figures 4.8 and 4.9 show the responses to the same setpoint change of the $\mathcal{H}_2$ example, again near Ground Idle conditions. The same health parameter step disturbance is injected at $t=15$ s. The closed-loop eigenvalues yield a response slightly slower than the $\mathcal{H}_2$ design and a slightly better ability to hold the setpoint in the presence of disturbance. Figure 4.9 shows that the fuel flow control input is significantly more damped than in the $\mathcal{H}_2$ design. The VSV input remains saturated for most of the transient.

4.8.5.3 Mixed $\mathcal{H}_2/\mathcal{H}_\infty$

We conclude the example with a mixed minimization objective: T_{48} is regarded as an $\mathcal{H}_2$ performance output, while SM-HPC is regarded as an $\mathcal{H}_\infty$ performance output. Since `msfsyn` assumes that the top rows of `Cbal`, `Lambda`, and `Dbal` correspond to the z_2 outputs, the matrices must be reordered:

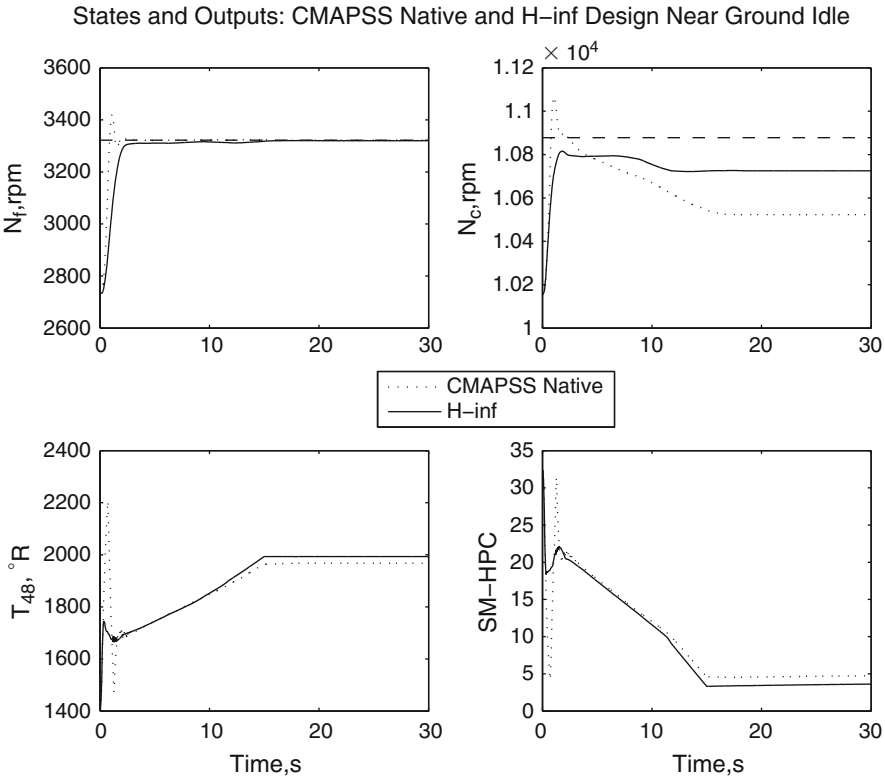

Fig. 4.8 Comparison of responses between CMAPSS-40k gain-scheduled fan speed compensator and fixed-gain $\mathcal{H}_2$ state feedback near Max Cruise: states and outputs. A step disturbance of -0.1 is applied to all health inputs at $t = 15$ s

```
sys1=ltisys(Abal1,[Gammabal1 -Bbal],
flipud(Cbal1),[0.75*Lambda1(2,:)-Dbal(1,:);zeros(1,16)]);
...
sys6=ltisys(Abal6,[Gammabal6 -Bbal],
flipud(Cbal6),[0.75*Lambda6(2,:) -Dbal(1,:);zeros(1,16)]);
polysys=psys([sys1,sys2,sys3,sys4,sys5,sys6]);
```

Again, Λ is scaled enough so that a feasible solution can be found. The following code is then used to and carry out the mixed-objective minimization with $a = 1$ and $b = 10$ as weights in (4.25):

```
r=[1 3];%# of z2 outputs and # of controls
obj=[0 0 1 10];
region=lmireg %choose disk with center -12, radius 10
[gopt,hopt,KT,Pcl]=msfsyn(polysys,r,obj,region);
K=KT*T;
```

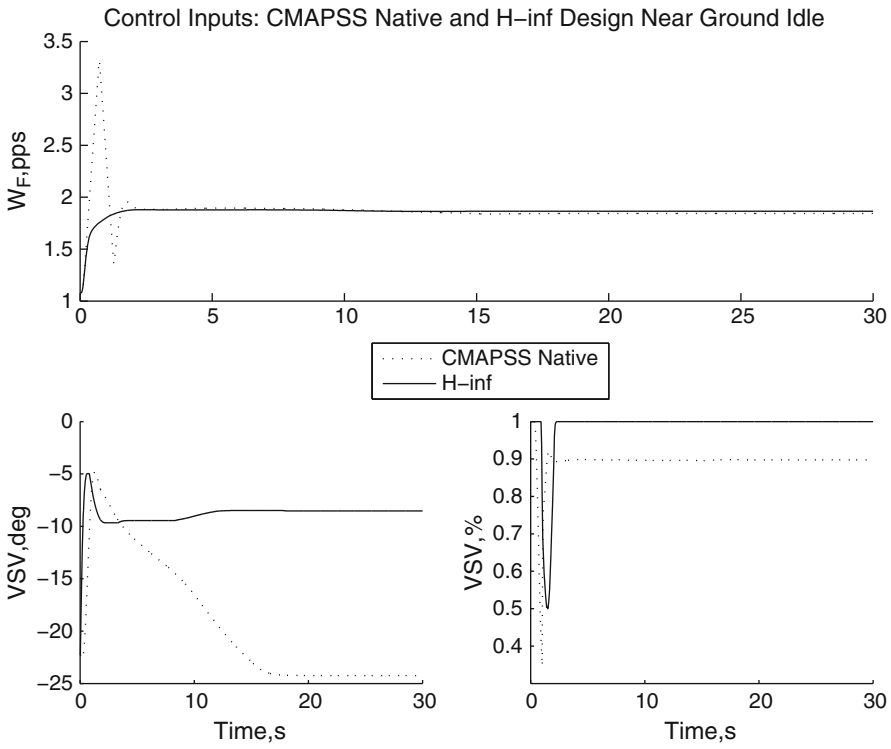

Fig. 4.9 Comparison of responses between CMAPSS-40k gain-scheduled fan speed compensator and fixed-gain $\mathcal{H}_2$ state feedback near Max Cruise: control inputs. A step disturbance of -0.1 is applied to all health inputs at $t = 15$ s

Closed-loop eigenvalues are required to lie in a disk centered at $-12+0$i with a radius of 10. The optimizing gain is

$$K = \begin{bmatrix} 0.0163 & 0.0060 & 15.3492 & 0.0148 & -0.1114 \\ -1.0027 & 0.6748 & 161.8834 & 21.7061 & 1.6351 \\ -0.8754 & 0.4571 & 126.2015 & 5.1284 & 16.6993 \end{bmatrix}.$$

The bounds for the infinity and two-norms returned by the program are 496.3 and 6,299, respectively. These figures cannot be compared with the earlier synthesis results due to changes in the definition of performance outputs. Simulation is conducted this time near the Max Cruise condition, with a shaft deceleration command and the same health disturbance inputs as before. Figures 4.10 and 4.11 show the responses. The responses are essentially equivalent to those produced by the previous norm-based designs and the native gain-scheduled controller. In summary, four gains were designed for use in an integral control approach: LQR, $\mathcal{H}_2$ for both performance outputs, $\mathcal{H}_\infty$ for both performance outputs and mixed $\mathcal{H}_2/\mathcal{H}_\infty$. The designs can be compared by measuring the closed-loop sensitivity

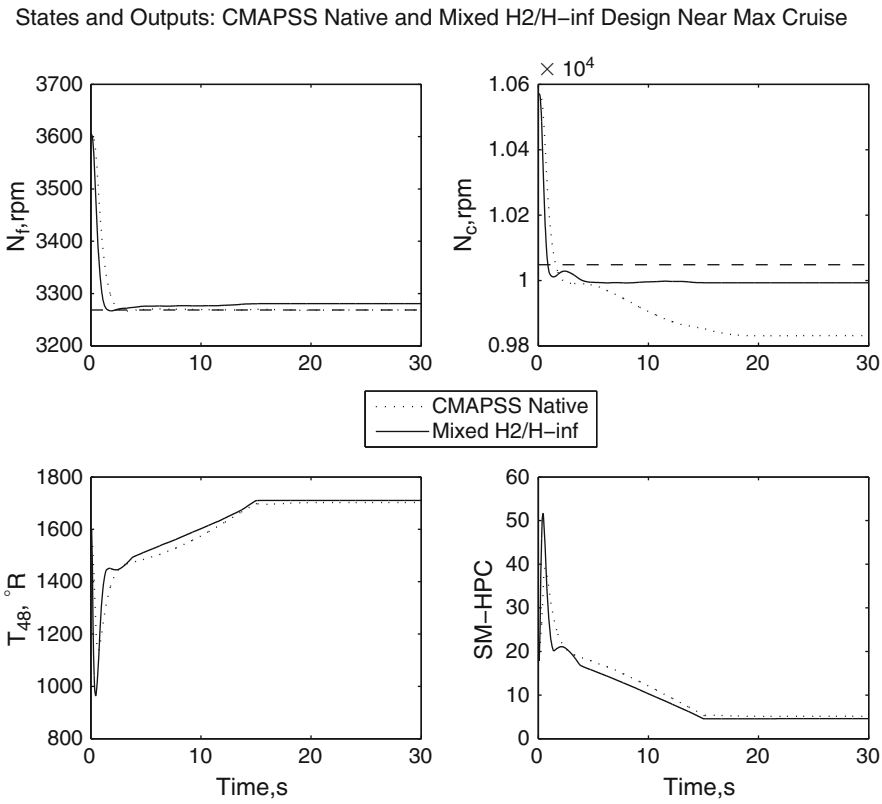

Fig. 4.10 Comparison of responses between CMAPSS-40k gain-scheduled fan speed compensator and fixed-gain mixed $\mathcal{H}_2/\mathcal{H}_\infty$ state feedback near Max Cruise: states and outputs. A step disturbance of -0.1 is applied to all health inputs at $t = 15$ s

of the performance outputs to exogenous inputs. This sensitivity can be evaluated with a plot of the maximum singular values as a function of frequency for all four designs. The worst-case peak singular value among closed-loop vertices is used. The results of this comparison are shown in Fig. 4.12.

4.9 Simplified $\mathcal{H}_\infty$ Fan Speed Control

This chapter focused on state feedback techniques that take advantage of the availability of state measurements and multiple control inputs. As discussed in Sect. 4.7.1, information about the steady map of the engine must be incorporated in state feedback laws to account for wide parametric variations and engine nonlinearity. In contrast, a classical compensator loop using fuel flow to control

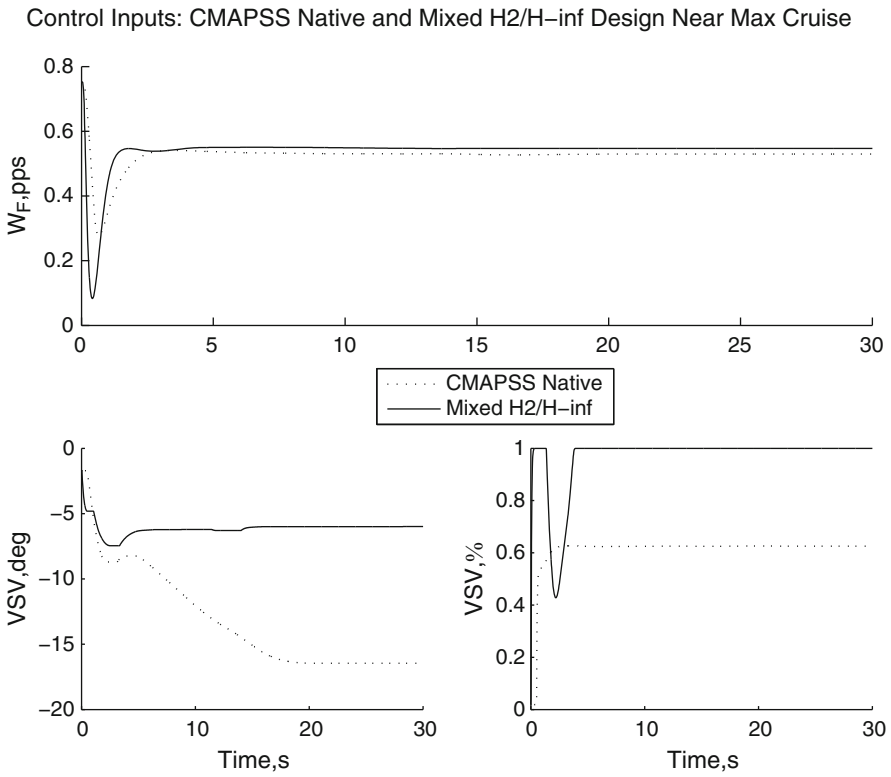

Fig. 4.11 Comparison of responses between CMAPSS-40k gain-scheduled fan speed compensator and fixed-gain $\mathcal{H}_2/\mathcal{H}_\infty$ state feedback near Max Cruise: control inputs. A step disturbance of -0.1 is applied to all health inputs at $t=1$ s

fan speed or EPR does not require reference states (other than the target fan speed setpoint). Although a single compensator is insufficient to achieve control over the whole flight envelope or maintain critical engine variables within permissible limits, control transfer functions are the basis of traditional gain-scheduled designs and limit-protection-logic arrangements. In this section, a simple, yet systematic and effective method to design a fan speed or EPR control transfer function is presented.

4.9.1 Mixed-Sensitivity $\mathcal{H}_\infty$ Design

Consider the SISO compensation loop shown in Fig. 4.13. Signals n and d represent sensor noise and output disturbance, respectively, while transfer function $G_d(s)$ can be used to "shape" the disturbance frequency spectrum. Elementary block-diagram

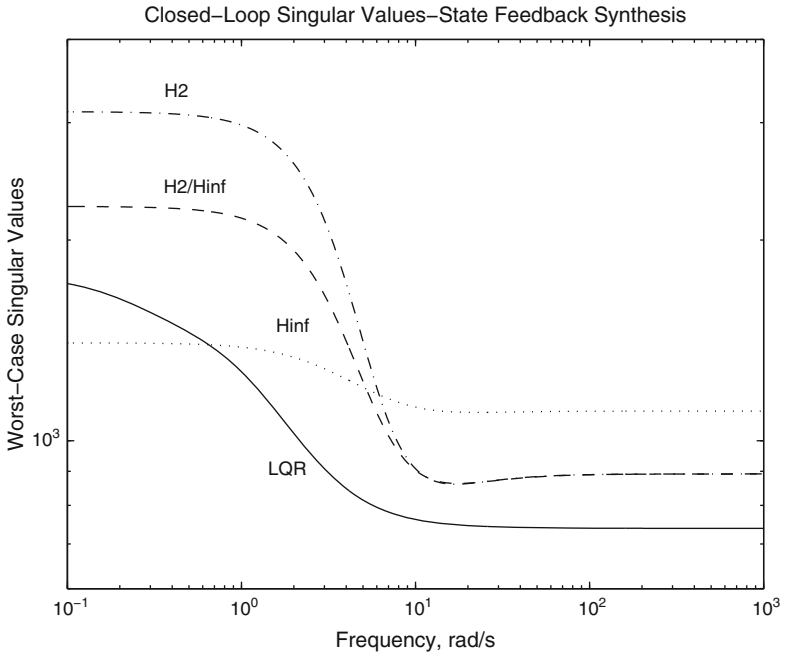

Fig. 4.12 Closed-loop comparison of worst-case maximum singular values for four synthesis methods

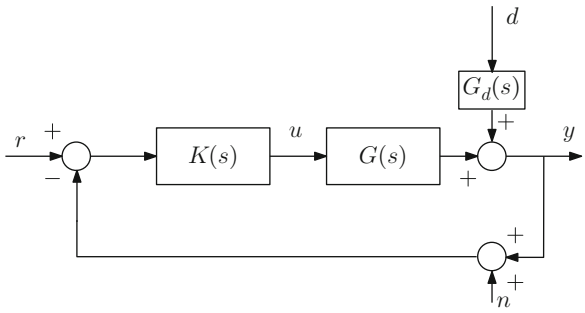

Fig. 4.13 Classical SISO feedback compensation loop

algebra shows that the output of the plant is related to the reference input r and to n and d as follows:

$$y = (I + GK)^{-1}GKr + (I + GK)^{-1}G_d d - (I + GK)^{-1}GKn, \qquad (4.36)$$

while the following expression applies to the control input:

$$u = K(I + GK)^{-1}(r - G_d d - n). \qquad (4.37)$$

Several transfer function definitions are in order. First, $T = (I + GK)^{-1}GK$ is the usual closed-loop transfer function between r and y when noise and disturbance are not considered. The transfer function $S = (I + GK)^{-1}$ is termed *sensitivity function*. This terminology is justified by the following identity:

$$\frac{\mathrm{d}T/T}{\mathrm{d}G/G} = S.$$

That is, the sensitivity function is a measure of the changes in T relative to changes in G. The closed-loop transfer function T is also referred to as *complementary sensitivity function*, since $S + T = I$ holds. This terminology is preferable, since several "closed-loop transfer functions" exist that relate u, r, y, n, and d.

With these definitions, (4.36) and (4.37) may be rewritten as

$$y = Tr + SG_d d - Tn, \tag{4.38}$$

$$u = KS(r - G_d d - n). \tag{4.39}$$

Various control objectives can be expressed in terms of desirable characteristics for T, S, and KS. First, the fundamental control objective is to force $y(t)$ to track $r(t)$ in a range of frequencies of interest. This implies that T should be as close to I (to 1 in the SISO case) as possible. In other words, the Bode magnitude of T should be flat, with a value of 0 dB, up to a design bandwidth. Note that n affects y in the same way as r. Care must be exercised in limiting the design bandwidth so that it does not encompass the sensor noise spectrum. Problems where the design bandwidth overlaps the noise spectrum require special filtering techniques. Next, SG_d must be kept small to minimize the influence of disturbances on y. Finally, KS must be small to maintain control input magnitudes within adequate ranges.

4.9.2 Frequency Weighting

The requirement that SG_d be "small" is made more precise by requiring that $|S(jw)G_d(jw)| < 1$ at all frequencies. Equivalently, $|S(jw)| < 1/|G_d(jw)|$ is required. Typically, no information about the disturbance spectrum is available, and therefore $G_d(s)$ is also unavailable. The designer may then model a generic disturbance as an input having strong low-frequency components, with little energy in the higher frequency range. These features distinguish disturbances from noise. Disturbances are then slowly-varying perturbations capable of producing large deviations in the plant output. According to this interpretation, $1/G_d(s)$ must have a high-pass characteristic. For this reason, it is commonplace to express the sensitivity minimization objective in the form

$$|S(jw)| < 1/|W_S(jw)|,$$

where $1/W_S(jw)$ is a design weight. Typical first- and second-order weights having the required characteristics are as follows:

$$W_S(s) = \frac{s/M + w_b}{s + \alpha w_b}, \tag{4.40}$$

$$W_S(s) = \frac{(s/\sqrt{M} + w_b)^2}{(s + w_b\sqrt{\alpha})^2}. \tag{4.41}$$

In both cases, w_b is the corner frequency of $1/W_S$, M is its high-frequency gain, and α is its low-frequency gain. The weights only differ in the steepness of the transition between α and M. Insensitivity to disturbances is improved by decreasing α. Note that the requirement that the magnitude of S be bounded by that of $1/W_S$ also imposes a lower-bound w_b on the bandwidth of T. Although initially conceived as a disturbance minimization objective, this form of weighting provides a mechanism to introduce a speed-of-response specification through w_b. The reader is referred to Skogestad and Postlethwaite [29] for more details.

For the purposes of controller synthesis, the sensitivity minimization objective is formulated as $||W_S S||_\infty < 1$. An upper-bound for the bandwidth of T and a roll-off rate can be specified as follows:

$$||W_T T||_\infty < 1.$$

Here, $1/W_T(s)$ is chosen to impose an upper envelope on the magnitude of $T(jw)$. A typical weight for T has the form

$$W_T(s) = \frac{s + w_b/M}{\alpha s + w_b}. \tag{4.42}$$

Design parameters w_b, α, and M may be chosen to coincide with those of the sensitivity weight, for simplicity.

Control magnitude weighting is likewise achieved with a specification of the form

$$||W_u K S||_\infty < 1.$$

A constant weight is usually sufficient and commonly chosen. All three objectives can be combined in a *mixed* or *stacked* sensitivity minimization objective of the form:

$$\min_K ||N(K)||_\infty, \tag{4.43}$$

where $N(K)$ is given by

$$N = \begin{bmatrix} W_S S \\ W_T T \\ W_u K S \end{bmatrix}. \tag{4.44}$$

The mixed-sensitivity $\mathcal{H}_\infty$ problem is readily solved using Matlab's `mixsyn`, part of the Robust Control Toolbox. Its use is shown in the CMAPSS example of the following section.

4.9.3 Example: Mixed $\mathcal{H}_\infty$ Synthesis: CMAPSS-40k

Consider the problem of controlling fan speed to a setpoint using fuel flow as a single control input. The design bandwidth is chosen as w_b=15 rad/s and the remaining parameters are $M = 2$ and $\alpha = 1 \times 10^{-4}$. The first-order weight of (4.40) is used for the sensitivity, along with the complementary sensitivity weight of (4.42). Although a solution is obtainable with a large range of control weights, the weight W_u must be chosen in the order of magnitude of 1×10^5 to obtain a compensator which can work with the nonlinear CMAPSS engine model. The following code illustrates the sequence of calculations in Matlab:

```
%Plant matrices at Ground Idle: Aa and Ba assumed
%available in the workspace
C=[1 0];D=0; %Output definition for fan speed TF
G=tf(ss(A,B,C,D));
w0=15;      %desired closed-loop bandwidth
alph=1/10000; %desired disturbance attenuation inside bandwidth
M=2 ;       %desired bound on hinfnorm(S) & hinfnorm(T)
s=tf('s'); %Laplace transform variable 's'
W1=(s/M+w0)/(s+w0*A); %Sensitivity weight
W2=1e5;                %Control weight
W3=(s+w0/M)/(A*s+w0); %Complementary sensitivity weight
[K,CL,GAM,INFO]=mixsyn(G,W1,W2,W3);
K=balred(K,3); %Perform balanced model reduction
%Plot results:
L=G*K;  %Loop transfer function
S=inv(1+L); %Sensitivity
T=1-S;      %Complementary sensitivity
sigma(GAM/W1,'k--',S,'k',GAM/W3,'k:',T,'.-k');
%Manually adjust controller for explicit integration
numK=0.032341*conv([1 3.992],[1 2.439]);
denK=[conv([1 4.876],[1 14.06]) 0];
```

The control transfer function initially returned by `mixsyn` is

$$K(s) = \frac{0.032341(s + 1.5e05)(s + 3.992)(s + 2.439)}{(s + 1.5e05)(s + 14.06)(s + 4.876)(s + 0.0015)}.$$

Two important observations regarding the structure of this compensator must be made:

1. An approximate integrator has been introduced in the controller as a result of the weighting function shapes.

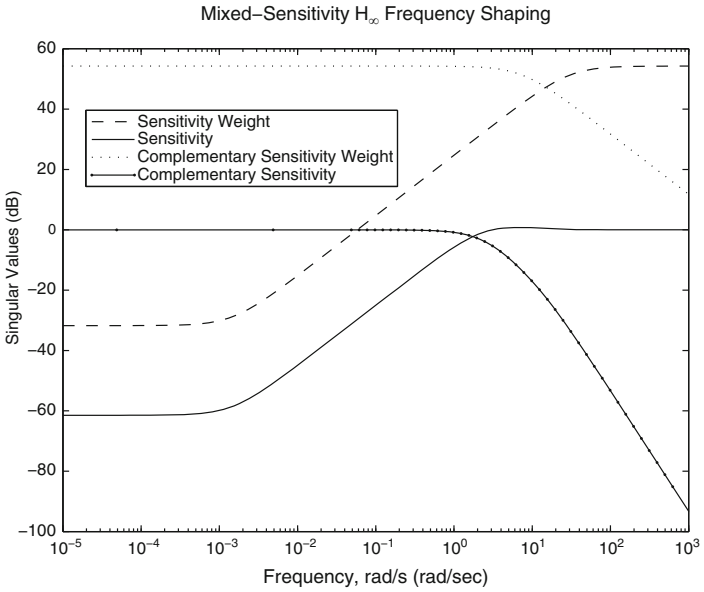

Fig. 4.14 Shaped sensitivity and complementary sensitivity functions

2. A near pole-zero cancelation is observed.
3. Some poles and zeroes have large magnitudes.

These features are not accidental, but appear rather frequently in conjunction with $\mathcal{H}_\infty$ frequency shaping techniques. Also, compensators having a large number of poles and zeroes are typical of these designs, especially when multiple weights are used, as it is the case in MIMO systems. For this reason, controller reduction is usually required before implementation. In this example, elimination of the pole and zero at -1.5×10^5 is an obvious reduction that brings the controller to 3rd order.

The approximate integrator may be modified to force an actual pole at zero. Upon doing this, the final controller becomes:

$$K(s) = \frac{0.032341(s + 3.992)(s + 2.439)}{s(s + 14.06)(s + 4.876)}.$$

Figure 4.14 shows the achieved sensitivity and complementary sensitivity functions (after reduction) in relation to their weights. This controller was implemented in CMAPSS-40k, holding the VBV and VSV actuators constant at their Ground Idle trim values during the course of simulation. Figure 4.15 shows the fan speed, core speed, and actuator responses.

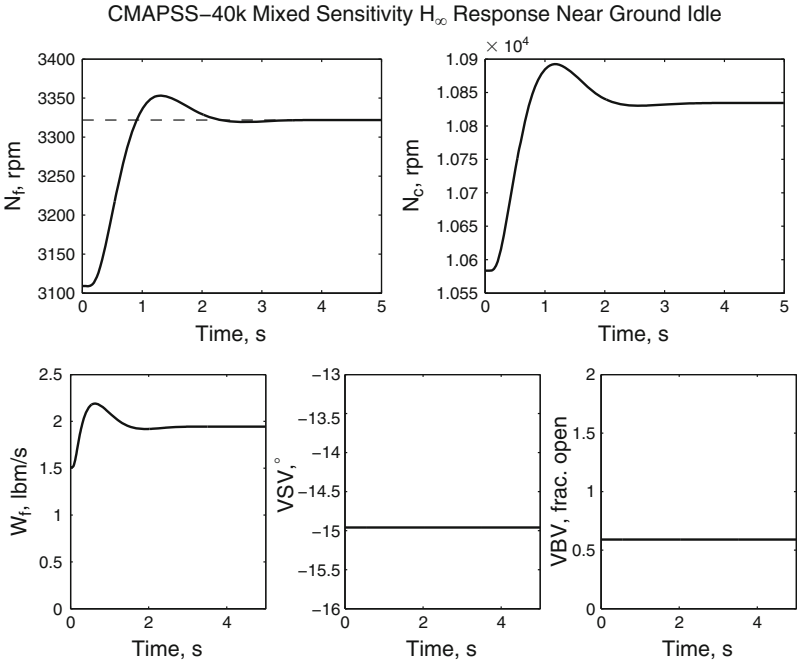

Fig. 4.15 Fan speed response with mixed sensitivity $\mathcal{H}_\infty$ design: CMAPSS-40k

4.9.4 Summary

The state-feedback design with input integration and fixed gain is conceptually simple. Systematic design procedures are available to synthesize the gain, according to various performance optimization requirements. The example shows that a fixed-gain design can be as effective as a classical gain-scheduled compensator in achieving robust stability against plant parameter variations arising from flight condition changes. Low sensitivity to health parameter inputs is also incorporated as an objective during the design process.

The responses achieved in the simulation examples are still unrealistic. The simulations of Sect. 4.8 show that the HPC stall margin, for instance, drops to 5% regardless of controller, a value regarded as unsafe in actual engine operations. Core acceleration, engine pressure ratio, and certain turbine temperatures are also subject to allowable operating limits. The limit protection features of CMAPSS-40k were disabled for these simulations. Robust nonlinear control approaches such as Model Predictive Control and Sliding Mode Control, considered in Chaps. 9 and 6, respectively, are capable of delivering superior performance. Limit management is a crucial consideration in GTE control design. Chapter 7 is entirely devoted to this problem.

Chapter 5
Gain Scheduling and Adaptation

Abstract This chapter introduces gain-scheduling and linear-parameter-varying techniques to address plant variability across the flight envelope. This chapter also introduces the concept of adaptive control and presents a basic model-reference adaptive control design. Matlab code and simulations using the CMAPSS nonlinear engine model are included.

This chapter is an overview of two control concepts that address plant variability across the flight envelope: scheduling and adaptation. These control methods are fundamentally different from robust approaches in that controller parameters are not fixed, but undergo significant changes during system operation. Gain scheduling as used in standard GTE control is described in Sect. 5.2. Next, in Sect. 5.3, a more systematic linear parameter-varying (LPV) approach to gain scheduling is presented. In Sect. 5.4, the basic concept of adaptive control is introduced, followed by a simple version of a well-established adaptive control methodology, namely Model Reference Adaptive Control (MRAC). Application of scheduling and adaptation concepts to fan speed control is illustrated throughout the chapter with CMAPSS-1 simulation examples.

5.1 Robustness, Scheduling, Adaptation

The techniques presented in Chap. 4 attempt to find a fixed controller that tolerates a range of plant parameter variations. Tolerance is understood first as the ability to maintain closed-loop stability upon uncertain plant variations. When plant matrices vary in a polytope, closed-loop stability of a given compensator can be checked using the definition of quadratic stability and associated LMI feasibility computations. Moreover, such fixed compensators can be synthesized using $\mathcal{H}_2/\mathcal{H}_\infty$ methods. Stability is necessary, but cannot be regarded as the sole criterion to define a successful GTE control implementation. Controllers must also maintain consistent transient response and tracking accuracy across the flight envelope. As demonstrated in Chap. 4, fixed controllers synthesized through $\mathcal{H}_2/\mathcal{H}_\infty$ methods offer a degree

H. Richter, *Advanced Control of Turbofan Engines*, DOI 10.1007/978-1-4614-1171-0_5,

of performance robustness, as measured by the minimum achievable norm. Still, significant tracking offsets and transient performance loss occur when attempting to use a single controller for all flight conditions.

Gain scheduling techniques address plant variations by introducing matching controller gain variations, attempting to obtain uniform transient responses across a wide range of operating conditions. The appropriate gain variations are determined during offline design and fixed for subsequent operation in the form of scheduling tables. In contrast, adaptive techniques induce control parameter variations online, as part of the control system implementation. Adaptive parameters can assume values outside the linear space generated by interpolation of scheduling table gains, thus improving the control system's *adaptability* to changes in the plant. Adaptive controllers introduce nonlinear dynamics in the closed-loop system. Therefore, their proper design requires an understanding of the theoretical bases governing their behavior. In this chapter, focus is limited to a simple form of adaptive control that is adequate for SISO plant models whose number of poles exceeds the number of zeroes by one. The transfer function from fuel flow to fan speed fits this case, and the adaptive technique is readily applied with good results.

5.1.1 Input Scheduling

Standard SISO GTE control systems use fuel flow as the only actuator used in a feedback loop ultimately aimed at controlling engine thrust. Other actuators, mainly VSV and VBV, are still actively governed during flight. Although there are no "controllers" in the usual sense that provide commands to these actuators, they are still commanded through a function of real-time measurements. The term "scheduling" is used in the GTE industry to denote this form of actuator command. In CMAPSS-40k, for instance, these actuators are commanded through functions of the corrected fan speed of (1.23), corrected core speed, and Mach number. Since the corrected speeds are functions of state variables, this form of command effectively adds decoupled SISO loops to the engine. Furthermore, it corresponds to proportional control, where the proportional gain is scheduled by Mach number and Θ, the nondimensional total temperature at LPC outlet. VSV and VBV command functions are limited to the allowable ranges of these actuators. Multivariable designs such as those presented in Chap. 4 forego of these scheduling functions and allow VSV and VBV to be commanded from a multi-output feedback controller.

5.2 Standard Gain-Scheduled GTE Control

Classical GTE gain scheduling centers on the idea of a fixed-structure (say, PI) compensator for fan speed or EPR control, whose gains are adjusted according to certain scheduling variables. These variables are chosen to reflect changes in

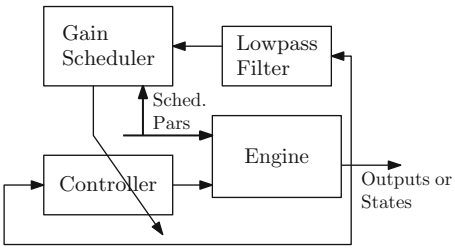

Fig. 5.1 Gain-scheduling control system

environmental conditions such as altitude, Mach number, and sea-level temperature, which act as parameters of the linearized engine models. State variables such as fan speed may also be used as scheduling variables. Note that when a state variable – or function thereof – is used as scheduling variable, a nonlinear feedback control loop is effectively introduced. This additional loop has the potential to destabilize the main control loop if not carefully designed. As shown in Fig. 5.1, it is customary to insert a low-pass filter between the state variable used for scheduling and the input of the scheduling table. This is to ensure that the changes to the control gains brought about by state variables are slow in comparison with the bandwidth of the main loop. The scheduler used in *ad-hoc* designs is nothing more than a look-up table giving control gains as a function of scheduling variables.

Although there is no universal rule for the selection of scheduling variables, it is reasonable to identify a set of physical parameters which dictate the numerical values of system matrices obtained through linearization. Recalling (2.1) and (2.2), it is clear that *all* parameters of functions f_1 and f_2, together with steady input values and a fixed set of health parameters, are needed to define an equilibrium point. Ideally, a set of scheduling variables would be comprised by all parameters. In practice, however, a few parameters can be identified that have the largest influence in the numerical values of the resulting linearized matrices. In the GTE, physical consideration and experience show that thermodynamic conditions at engine inlet, i.e., inlet static pressure and Mach number have the largest influence. Fan speed, an engine state, is frequently used as an additional scheduling variable to account for the intrinsic nonlinearity of functions f_1 and f_2. Inlet pressure may be used directly, or equivalently, altitude may be used. An additional important consideration for the selection of scheduling variables is their availability as real-time measurements.

The scheduling tables are built by selecting a controller structure and repeating a controller design for various combinations of scheduling variables. The resulting gains are then included in look-up tables. Linear interpolation is used to find controller gains from real-time measurements of the scheduling variables.

In CMAPSS-40k, fixed PI control structures are used for the fan speed, core speed, and EPR control loop options. Scheduling variables are altitude and Mach number for the P-gain, and altitude, Mach number and fan speed for the I-gain. Figures 5.2 and 5.3 are graphical representations of the P and I-gain scheduling used by default in CMAPSS-40k. As conventionally used in GTE control systems,

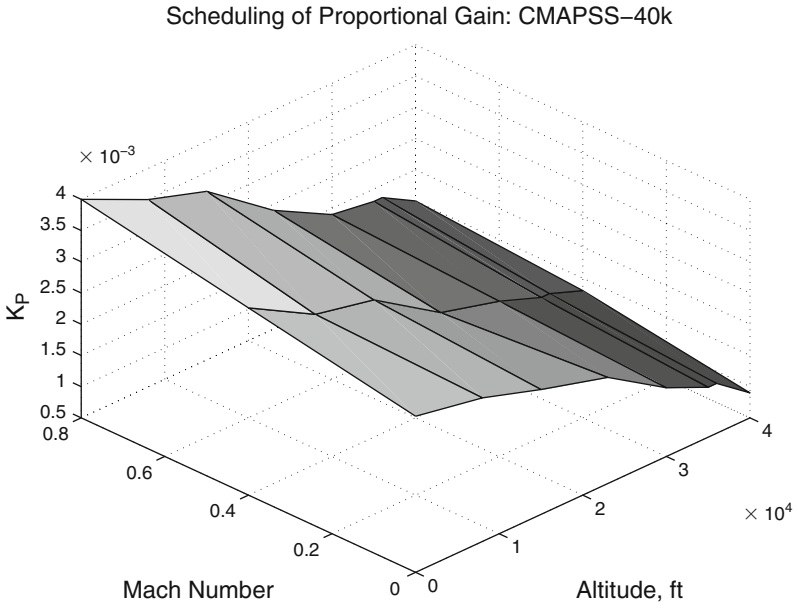

Fig. 5.2 Scheduling of proportional gain in CMAPSS-40k

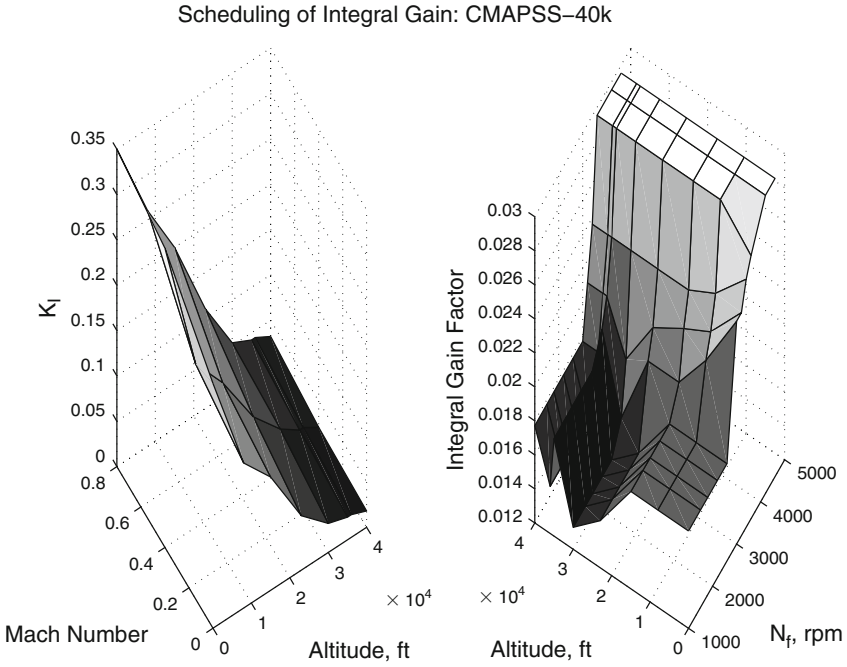

Fig. 5.3 Scheduling of integral gain in CMAPSS-40k. The value of K_i from the left plot is multiplied by the factor from the right plot

gain scheduling is a simple and effective way of achieving response specifications across the flight envelope under normal circumstances. The chief weakness of *ad-hoc* gain scheduling is its lack of robustness and its lack of adaptability to unforeseen conditions. If plant parameters change in ways that have not been accounted for in the scheduling tables, serious performance loss or instability can occur (See Shamma and Athans [38]). Engine aging and deterioration effects are in fact model parameters, whose variations are introduced as a disturbance input. These parameters are not typically used as scheduling variables. Doing so would lead to overly complex look-up tables. Besides, health parameters are not available as real-time measurements and their estimation and use in engine prognostics are challenging research problems. For more information on health parameter estimation, see references [20–22, 39].

5.3 Linear Parameter-Varying Methodologies

LPV control denotes a group of techniques based on a structured description of plant parameter variability. By imposing an LPV structure on the uncertain plant, robust designs based on $\mathcal{H}_\infty$ theory become possible. The interested reader is referred to the work of Wolodkin et al. [40], where gain scheduling for turbofan engine control is conducted under an $\mathcal{H}_\infty$ optimization objective. This approach offers superior performance in comparison with *ad-hoc* scheduling of several fixed designs.

An LPV description of the uncertain state-space plant has the form

$$\dot{x} = A(p)x + B(p)u, \tag{5.1}$$

$$y = C(p)x + D(p)u, \tag{5.2}$$

where $p = [p_1 \quad p_2 \quad p_3 \ ... p_s]$ is a vector of s parameters, and system matrices are given by

$$A(p) = A_0 + p_1 A_1 + p_2 A_2 + ... + p_s A_s, \tag{5.3}$$

$$B(p) = B_0 + p_1 B_1 + p_2 B_2 + ... + p_s B_s, \tag{5.4}$$

$$C(p) = C_0 + p_1 C_1 + p_2 C_2 + ... + p_s C_s, \tag{5.5}$$

$$D(p) = D_0 + p_1 D_1 + p_2 D_2 + ... + p_s D_s, \tag{5.6}$$

where $A_0, A_1 ..., B_0, B_1, ... C_0, C_1, ... D_0, D_1, ... D_s$ is a set of *coefficient matrices*. The parameters are chosen in the same way as scheduling parameters, that is, on the basis of knowledge or experience with the system. For fan speed control, a reasonable set of parameters is given by altitude, Mach number, and fan speed itself. For the remainder of the chapter, we assume that $p = [m \quad h \quad f]$, where m is the Mach number, h is the altitude normalized by a convenient scaling factor, and f is the fan speed, also normalized by a suitable scaling factor.

5.3.1 Obtaining an LPV Decomposition from Polytopic Vertices

Instances of $A(p)$, $B(p)$, $C(p)$, and $D(p)$ are available to the designer as an outcome of linearization at some steady-state condition determined by a fixed value of p and a set of matching equilibrium states and inputs. The coefficient matrices, however, need to be determined. When a set of instances of the system matrices corresponding to known parameter vectors are available, the coefficient matrices are determined through a generalized system of linear equations.

Suppose that a set of matrices $A(p(i)), i = 1, 2, ...r$ is available, corresponding to a set of parameter vectors $p(1)$, $p(2)$, $...p(r)$. The following linear system arises from (5.3):

$$\begin{bmatrix} I & p_1(1)I & p_2(1)I & ... & p_s(1)I \\ I & p_1(2)I & p_2(2)I & ... & p_s(2)I \\ \vdots & & & & \vdots \\ I & p_1(r) & p_2(r)I & ... & p_s(r)I \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_s \end{bmatrix} = \begin{bmatrix} A(p_1) \\ A(p_2) \\ \vdots \\ A(p_s), \end{bmatrix}. \tag{5.7}$$

This linear system has the form $RV = S$, where V is the unknown matrix. The dimensions of R are nr-by-ns, where n is the dimension of each square matrix $A(p)$. If $r = s$ and $p(i)$ are chosen so that R has full rank, the solution can be found as $V = R^{-1}S$. If $r > s$, the system is overdetermined and an exact solution may be found only for a rather restrictive set of problem data in R and S. An approximate solution for V can be found by minimizing the 2-norm (largest singular value) of the residual matrix $RV - S$. The solution to this case is $V = R^{+}S$, where R^{+} denotes the Moore–Penrose pseudoinverse [41] of R, which can be calculated in Matlab using the `pinv` command. The parameter vector chosen for GTE fan speed control has $s = 3$. Thus, three instances of system matrices would be necessary to obtain the LPV coefficient matrices using R^{-1}. Three flight conditions cannot be expected cover the operating envelope, implying that the pseudoinverse must be used.

5.3.1.1 CMAPSS-1 LPV Decomposition

The 14 conditions listed in Appendix B are readily used to obtain a set of four coefficients for $A(p)$ and $B(p)$. Altitude is normalized by 10,000 and fan speed by 3,000 to define the parameter vector. Matrix $A(p)$, for instance, is decomposed as

$$A(m, h, f) = \begin{bmatrix} 0.1974 & 0.4174 \\ 0.1285 & 1.0424 \end{bmatrix} + m \begin{bmatrix} -1.0000 & 0.0626 \\ 0.1002 & -0.2370 \end{bmatrix} + h \begin{bmatrix} 0.7476 & -0.2551 \\ -0.0653 & 0.7102 \end{bmatrix} + f \begin{bmatrix} -5.0210 & 1.3617 \\ 0.5981 & -7.2702 \end{bmatrix}.$$

To illustrate the magnitude of the error associated with the use of the pseudoinverse, consider $m = 0.7$, $h = 2$, and $f = 0.775$, corresponding to FC06. Computation of A from the LPV decomposition results in

$$\begin{bmatrix} -2.8971 & 1.0059 \\ 0.5314 & -3.3350 \end{bmatrix}.$$

This represents a 6.1% error in matrix 1-norm (largest column absolute value sum) relative to the true value of A at FC06. Similar errors exist for all other matrices and flight conditions.

5.3.2 A Simple LPV Approach to Fan Speed PI Control

Since fan speed increment is the first state in (5.1), an output y defined as fan speed increment will have constant $C = [1 \;\; 0]$ and $D = 0$. This implies that the transfer function from fuel flow increment $u = \Delta W_{\mathrm{F}}$ to $y = \Delta N_{\mathrm{f}}$ always has the form

$$G(s) = \frac{k(s+z)}{(s+c_1)(s+c_2)}, \tag{5.8}$$

where c_1 and c_2 may be complex conjugates. In CMAPSS-40k, linearization at high PLA levels tends to give complex poles, while real poles are seen at low PLA settings. In CMAPSS-1, real poles are observed for all 14 flight conditions. The technique presented in this section is restricted to plants with real poles only. The technique is based on the same premise as conventional gain scheduling: scheduling variables (in this case, parameters p) are available as real-time measurements from sensors. Then system matrices $A(p)$ and $B(p)$ can be computed in real-time using the LPV decomposition of (5.3) and (5.4). This information can be used to calculate the gains of a controller whose structure has been predetermined to meet performance and stability-related objectives.

Denote the entries of $A(p)$ and $B(p)$ as $a_{ij}(p)$ and $b_i(p)$, respectively, for $i = 1, 2$ and $j = 1, 2$. Transfer function $G(s)$ may then be parameterized by these entries by using a scalar version of (4.1):

$$G(s) = \frac{b_1 s - b_1 a_{22} + b_2 a_{12}}{s^2 - (a_{11} + a_{12})s + a_{11}a_{22} - a_{12}a_{21}}. \tag{5.9}$$

Recalling Chap. 3, the PI controller structure is a suitable choice to meet the zero steady-state error requirement and obtain adequate transient responses. The control transfer function $K(s) = K_p + K_i/s$ leads to the closed-loop characteristic equation

$$s^3 - (a_{11} + a_{22} - K_p b_1)s^2 - (a_{12}a_{21} - a_{11}a_{22} - K_i b_1 - K_p a_{12} b_2 + K_p a_{22} b_1)s + K_i a_{12} b_2 - K_i a_{22} b_1 = 0.$$

This equation possesses three roots, of which at least one must be real, since complex roots appear in conjugate pairs. An approach to selecting control gains is to enforce a pole-zero cancelation in the closed-loop transfer function. Recall that the zeroes of $G(s)K(s)$ are the same as those of the closed-loop transfer function $T(s)$, up to pole-zero cancelations. In this technique, the zero introduced by the PI controller at $s = -K_i/K_p$ is to cancel a real root of the closed-loop characteristic equation. This is done to remove part of the response variability arising from plant parameter changes. To place a constrain on K_i and K_p so that the sought pole-zero cancelation occurs, a symbolic computation process must be carried out. First, $s = -K_i/K_p$ is substituted in the closed-loop characteristic equation. When this is done, the characteristic equation can be factored as $K_i\,Q(K_i, K_p) = 0$, where Q is a quadratic polynomial in K_i. Then two nonzero solutions for K_i are found from $Q(K_i, K_p) = 0$:

$$K_i^+ = -\frac{K_p}{2}\left(a_{11} + a_{22} + \sqrt{\Delta}\right), \tag{5.10}$$

$$K_i^- = -\frac{K_p}{2}\left(a_{11} + a_{22} - \sqrt{\Delta}\right), \tag{5.11}$$

where $\Delta = a_{11}^2 - 2a_{11}a_{22} + a_{22}^2 + 4a_{12}a_{21}$. It can be readily verified that $\frac{1}{2}\left(a_{11} + a_{22} \pm \sqrt{\Delta}\right)$ are the eigenvalues of $A(p)$. Hence, the restriction of the technique to real eigenvalues is justified to prevent complex controller gains. Thus, each solution produces the cancelation of one of the two real plant poles. Furthermore, although the objective was to cancel a *closed-loop* pole, (5.10) and (5.11) show that the only way to achieve this is by direct cancelation of the plant *open-loop* pole. Hence, after cancelation, the loop transfer function has the form

$$G(s)K(s) = \frac{k'(s+z)}{s(s+c)},$$

where z is the plant zero, c is the un-canceled plant pole, and k' is a new gain.

Equations (5.10) and (5.11) place a constraint on the relationship between K_i and K_p, but do not completely determine their values. A root locus argument can be used to show that sufficiently high values of K_p will result in an approximate cancelation of the plant zero. Furthermore, increasing K_p leads to faster responses and insensitivity from plant parameters. Indeed, assuming without loss of generality that $c_1 > c_2$ in (5.8), it is clear that using K_i^+ will result in the cancelation of c_1 and using K_i^- results in the cancelation of c_2. Assume that K_i^+ is chosen. The open-loop transfer function is then

$$G(s)K(s) = K_p k \frac{(s+z)}{s(s+c_2)}.$$

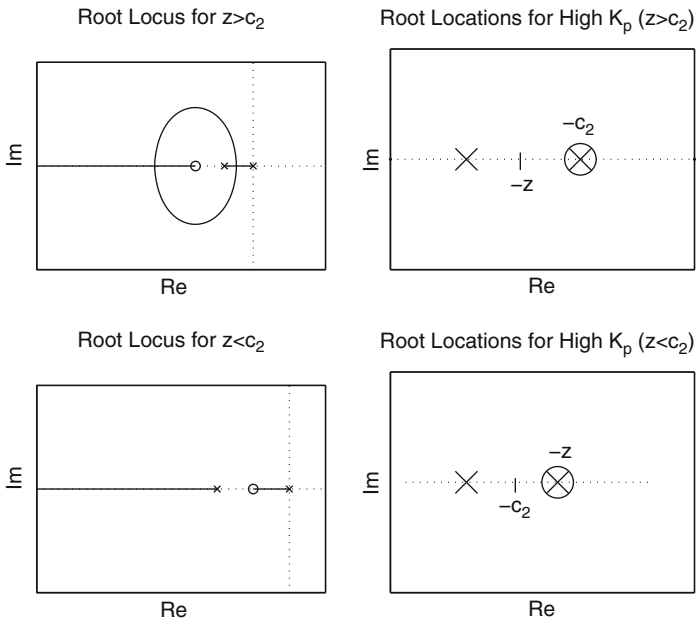

Fig. 5.4 Possible root loci and closed-loop root locations: pole-zero cancelation design with K_i^+ and high K_p

Two possible root loci are shown in Fig. 5.4, according to the relative magnitudes of z and c_2. In both cases, a sufficiently high value of K_p will place a closed-loop pole at the location of the plant zero. The closed-loop transfer function is nearly independent from plant parameters. The remaining real pole proceeds to $-\infty$ as K_p is increased. This observation indicates that K_p does not need to be scheduled, but rather set at an appropriate value, best determined from simulation trials. The same general behavior is seen if K_i^- is chosen. In a real-time implementation, the eight coefficient matrices are stored in memory, from where the entries of $A(p)$ and $B(p)$ are computed. A fixed K_p and K_i^+ or K_i^- as calculated from (5.10) and (5.11) are used as PI gains.

5.3.2.1 Linearized Study: CMAPSS-1

A simulation study is conducted with $K_p = 0.1$, using the eight coefficient matrices of the LPV decomposition. Unit step responses corresponding to the 14 plants controlled with the LPV-based PI controlled are shown in Fig. 5.5. The results show that, in this case, K_i^+ tends to produce faster responses than K_i^-+ and that a constant K_p is sufficient to produce consistent transient responses across wide plant parameter changes.

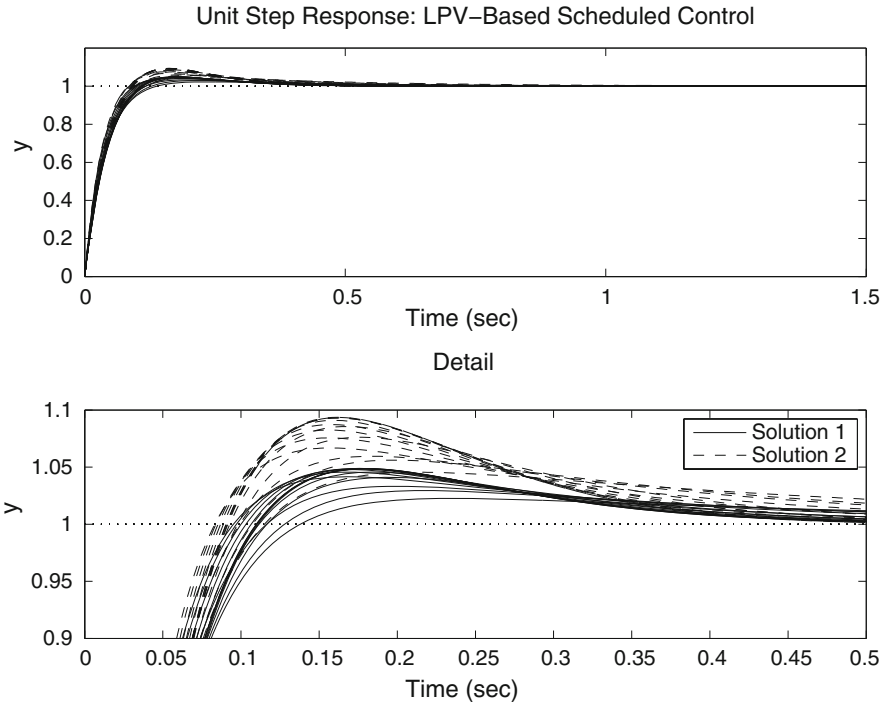

Fig. 5.5 Linearized responses of LPV-based scheduled controller across 14 flight conditions: CMAPSS-1

5.3.2.2 LPV Scheduling in Nonlinear Engine: CMPASS-1

A simulation is now conducted using the nonlinear 90k engine. Altitude, Mach number, and fan speed are varied during the simulation, mimicking the parameter changes taking place during takeoff and climb. Initially, the engine is at sea level, the Mach number is zero, and the TRA is 20. This corresponds to FC13 in Appendix B, or near-idle conditions, where $N_{\mathrm{f}} = 1{,}497$ rpm, as shown in Table 2.3. Altitude is changed to 20,000 ft., Mach number to 0.7, and TRA to 100, which correspond to FC06, where $N_{\mathrm{f}} = 2{,}324$ rpm. The transition between altitude and Mach number parameters is taken as a 1-second ramp. Although no aircraft is capable of such fast altitude and airspeed changes, these parameter changes are useful to benchmark engine control systems. The fan speed reference input passed to the control system is given by a ramp having a slope of 500 rpm/s, the maximum admissible in CMAPSS-1. The proportional gain K_p is maintained at 0.1, and the formula for K_i^+ is used. Figure 5.6 shows that the LPV-based scheduling of the P-gain produces very accurate fan speed demand tracking. The lower plot shows that the control input is more aggressive in comparison with the native scheduled controller. Figure 5.7 shows the variation of K_i.

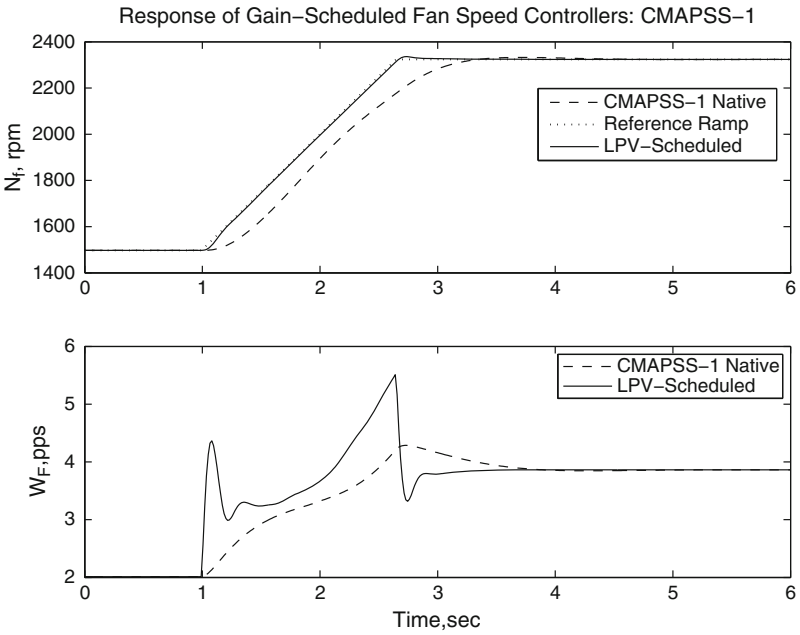

Fig. 5.6 Comparison of nonlinear engine responses. LPV-based scheduled controller and CMAPSS-1 native scheduled regulator

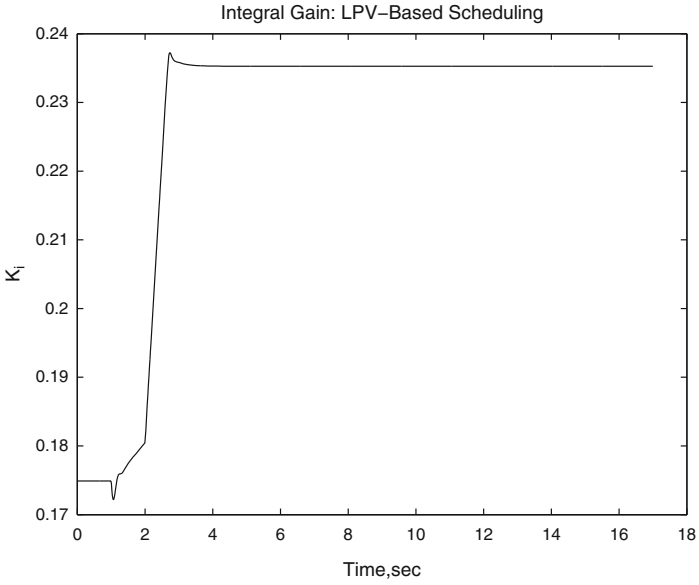

Fig. 5.7 LPV-based scheduled controller in CMAPSS-1 nonlinear engine: integral gain variation

5.3.3 Other LPV Approaches

The LPV-based pole-zero cancelation approach presented above represents an improvement over standard point design interpolations in a practical sense. Indeed, formulas replace tables, making it possible to generate gains with finer resolution in a compact format. The offline generation of coefficient matrices is very systematic and easily extensible to a larger number of vertices. The method, however, does not have any explicit robustness properties. Many multivariable robust approaches to gain scheduling using the LPV parameterization were developed in the 1990s. For a theoretical basis, see Kamen and Khargonekar [42], Apkarian and Gahinet [43] or Packard and Kantner [44]. For application of these techniques to aircraft engines, see Wolodkin et al. [40], Balas [45] or extensions of the LPV parameterization that allow polynomial dependence on coefficient matrices, but include linear dependence on controller parameters. Such *polynomial LPV synthesis* has been developed by SNECMA, a French aerospace manufacturer, see Henrion [46] and Gilbert [47].

5.4 Overview of Adaptive Control

The central idea of adaptive control schemes is to introduce a controller structure and a set of parameters Θ which, along with feedback measurements, determine the value of the control input at every instant. In ideal circumstances, when the plant is known exactly, Θ could be computed so that the closed-loop system has desirable characteristics. The premise justifying the use of adaptive control is that plant parameters are either uncertain or changing in time. Then Θ cannot be calculated beforehand. Various adaptive schemes introduce an *adaptation law*, or *parameter update law* to refine an initial guess of Θ during the course of system operation. Numerous adaptive schemes have been proposed, see, for instance, Åstrom and Wittenmark [48] or Ioannou [49]. Three architectures are frequently used: model-reference adaptive control (MRAC), indirect adaptive control, and self-tuning regulators. As Fig. 5.8 shows, the MRAC scheme introduces a reference model, which specifies the desired dynamics of the controlled plant. The MRAC is a dynamic system whose parameters Θ are adjusted so that the error between the reference model and the actual plant outputs is driven to zero. This approach can be categorized as *direct*, in that no attempt is made to estimate the unknown plant parameters as an intermediate step for the computation of the control input. The *MIT rule* was used in the early beginnings of adaptive control to update the parameters:

$$\frac{\mathrm{d}\Theta}{\mathrm{d}t} = -\gamma e \frac{\partial e}{\partial \Theta},$$

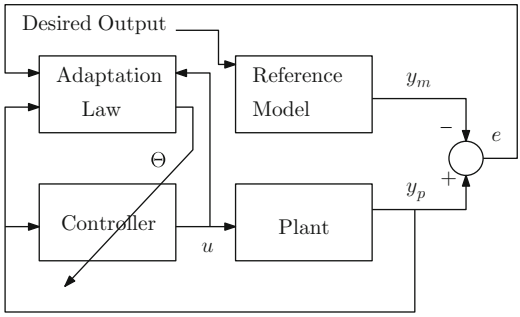

Fig. 5.8 Model-reference adaptive control system schematic

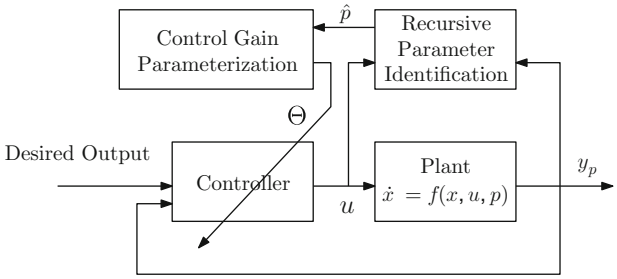

Fig. 5.9 Indirect adaptive control system schematic

where e is the error between plant and reference model outputs and γ is a positive constant governing the speed of adaptation. This rule was subsequently shown to lead to unpredictable system behavior, ranging from poor performance and slow adaptation to unstable closed-loop systems (see Anderson [50]). Present-day adaptation and control law synthesis are firmly grounded on Lyapunov stability theory.

Indirect adaptive approaches include a plant parameter estimator and a control design algorithm as an intermediate step, as shown in Fig. 5.9. The controller has been parameterized and tuned in terms of plant parameters, much like what was done in Sect. 5.3.2 using the LPV parameterization. A parameter estimator constantly updates plant parameters, which determine controller gains. Computationally, the only distinction between indirect adaptive control and LPV-based gain scheduling is the mechanism used to arrive at plant parameters: A recursive estimator is used in adaptive control. In contrast with the static, predetermined LPV parameterization, no offline information about plant variability is needed in the indirect adaptive case. A *self-tuning regulator* is a form of indirect adaptive control where plant parameter estimates are used to conduct controller design leading to the gains to be implemented in a fixed-structure controller, as shown in Fig. 5.10. This differs from other indirect schemes in that plant parameters are not directly used in the controller parameterization. Depending on the type of controller design to be performed, self-tuning regulators could represent a significant

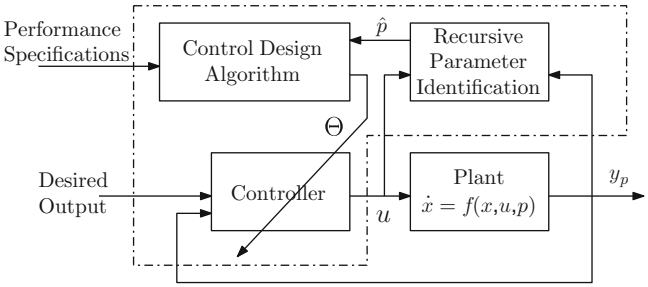

Fig. 5.10 Self-tuning regulator schematic

real-time computational burden. Some of the burden associated with the design can be transferred to offline computation, resulting in explicit implementations (see Grimble [51]).

5.4.1 Relative-Degree 1 MRAC

An MRAC scheme originally developed by Feuer and Morse [52] and further elaborated by Ioannou [49] is now summarized. This form of MRAC is applicable to linear SISO plants whose parameters are unknown, but whose *relative degree* is known to be equal to one. Recall that the relative degree of a linear transfer function is defined as the number of poles minus the number of zeroes. In addition, the plant transfer function is required to be minimum-phase and to have a high-frequency gain with known sign. Let the plant model be given in transfer function form as

$$Y_p(s) = G(s)U(s) = k\frac{N(s)}{D(s)}U(s), \tag{5.12}$$

where $N(s)$ and $D(s)$ are the numerator and denominator polynomials such that the degree of $D(s)$ is higher than the degree of $N(s)$ by one. The polynomials are assumed to be *monic*, that is, the leading coefficient (the coefficient of the highest power of s) of $N(s)$ and $D(s)$ must be one. In this case, the magnitude of the plant frequency response approaches an asymptote of the form $\frac{|k|}{w}$, where k is the high-frequency gain. These assumptions are satisfied by the transfer function from fuel flow to fan speed, upon which many standard GTE designs are based. The reference model is given by a transfer function of the form

$$Y_m(s) = W(s)R(s),$$

where r is the reference input to be tracked by y_p and $W(s)$ must be of relative degree one and also satisfy the following assumptions:

1. All poles of $W(s)$ must have negative real parts.
2. $W(s)$ must be minimum-phase.
3. The real part of $W(jw)$ must be nonnegative for all $w \geq 0$.
4. The high-frequency gain of $W(s)$ has the same sign as that of $G(s)$.

Conditions (2) and (3) above are satisfied by *strictly positive real* (SPR) transfer functions, which can be generated for any relative degree using two matrix conditions due to Kalman and Yakubovich, see [48]. In this section, we only consider reference models of the form

$$W_m(s) = \frac{1}{\tau s + 1}, \tag{5.13}$$

which satisfy all assumptions and are sufficient for an introductory exposition of MRAC methods. The speed of response of the reference model can be tuned using time constant τ.

This form of MRAC uses four controller gains assembled in a parameter vector $\Theta = [\Theta_1 \;\; \Theta_2 \;\; \Theta_3 \;\; \Theta_4]^{\mathrm{T}}$. The control input is calculated as

$$u = \Theta^{\mathrm{T}}\omega, \tag{5.14}$$

where $\omega = [\omega_1 \;\; \omega_2 \;\; y_p \;\; r]^{\mathrm{T}}$. Quantities ω_1 and ω_2 are the outputs of first-order filters of the form

$$\dot{\omega}_1 = F\omega_1 + gu_{\mathrm{p}}, \tag{5.15}$$

$$\dot{\omega}_2 = F\omega_1 + gy_p, \tag{5.16}$$

whose initial conditions are set as $\omega_1(0) = \omega_2(0) = 0$. An arbitrary nonzero value is chosen for g and F is chosen so that the above filters are stable, that is, $F < 0$. Finally, the parameter adaptation law is given by

$$\dot{\Theta} = -Ge\omega \text{ sign } (k_p k_{\mathrm{m}}), \tag{5.17}$$

where k_{m} is the model's high-frequency gain, G is a tunable positive-definite matrix and the error has been defined as

$$e = y_p - y_m.$$

Matrix G controls the rate of parameter adaptation.

5.4.2 Example: CMAPSS-1

The MRAC schemed is tested first using the 14 linearized plant models of CMAPSS-1. A step command for ΔN_{f} of 100 rpm is used as reference input. The reference model was chosen as

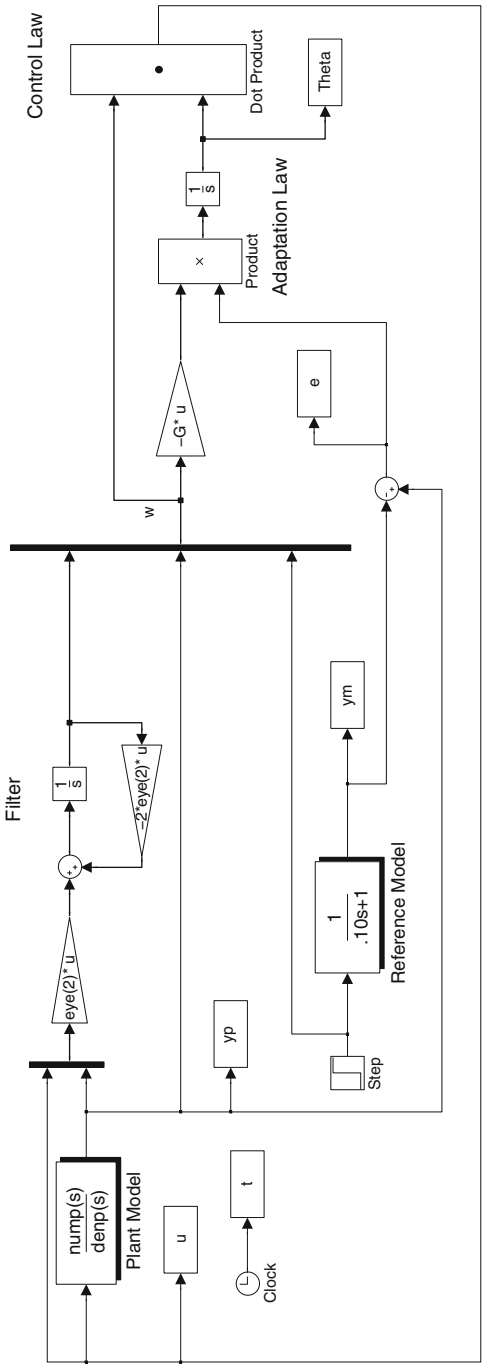

Fig. 5.11 Simulation diagram for model-reference adaptive control with linearized plant model

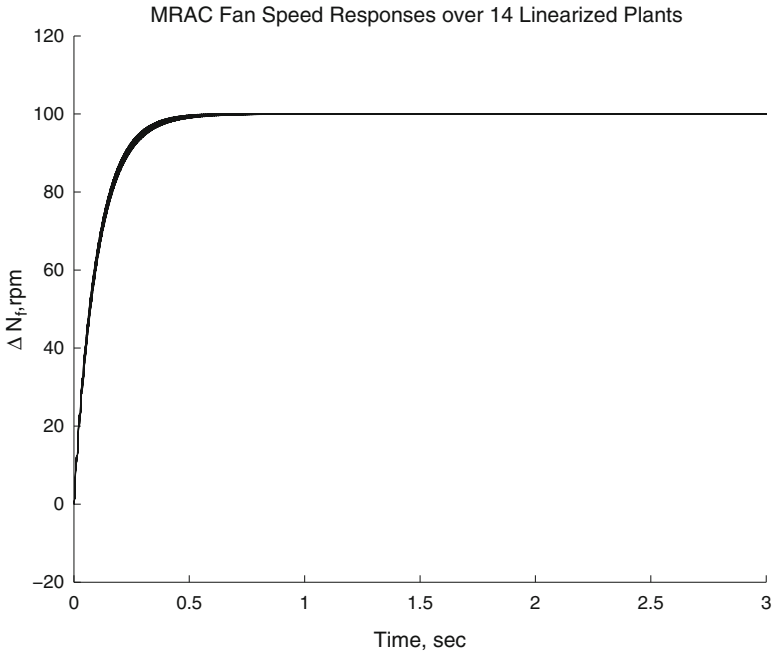

Fig. 5.12 Fan speed responses of model-reference adaptive control over 14 linearized plants

$$W_m(s) = \frac{1}{0.1s + 1}.$$

The adaptive gains were chosen with almost no trial-and-error as

$$G = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}, \quad F = -2 \quad g = 1.$$

The reader interested in reproducing these results may refer to Fig. 5.11, which shows the Simulink implementation. The initial parameter vector was chosen as $\Theta_0 = [0 \;\; 0 \;\; 0 \;\; 0]^{\mathrm{T}}$. The step response simulation was repeated using the 14 transfer functions from incremental fuel flow to incremental fan speed using the data from Appendix B, and without modifications to G, F, g, or Θ_0. Figure 5.12 shows that the fan speed responses closely match each other and the reference model: there is no overshoot and the settling time is about four time constants. Figure 5.13 shows how parameters adapt and converge to different values to achieve the specified model-matching objective.

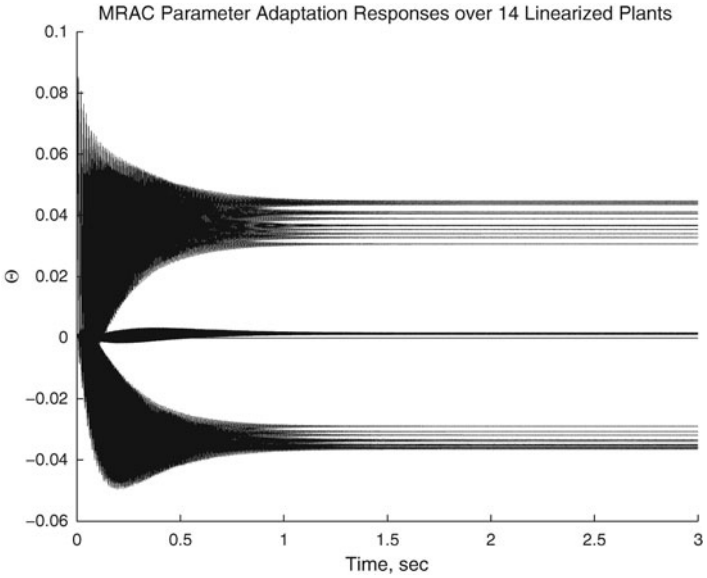

Fig. 5.13 Parameter adaptation responses of model-reference adaptive control over 14 linearized plants

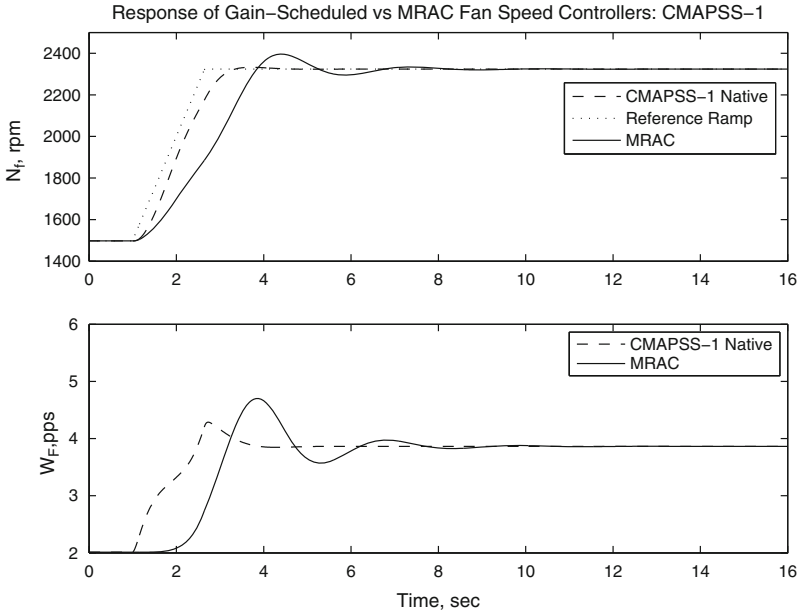

Fig. 5.14 Comparison of nonlinear engine responses: model-reference adaptive control and CMAPSS-1 native scheduled regulator

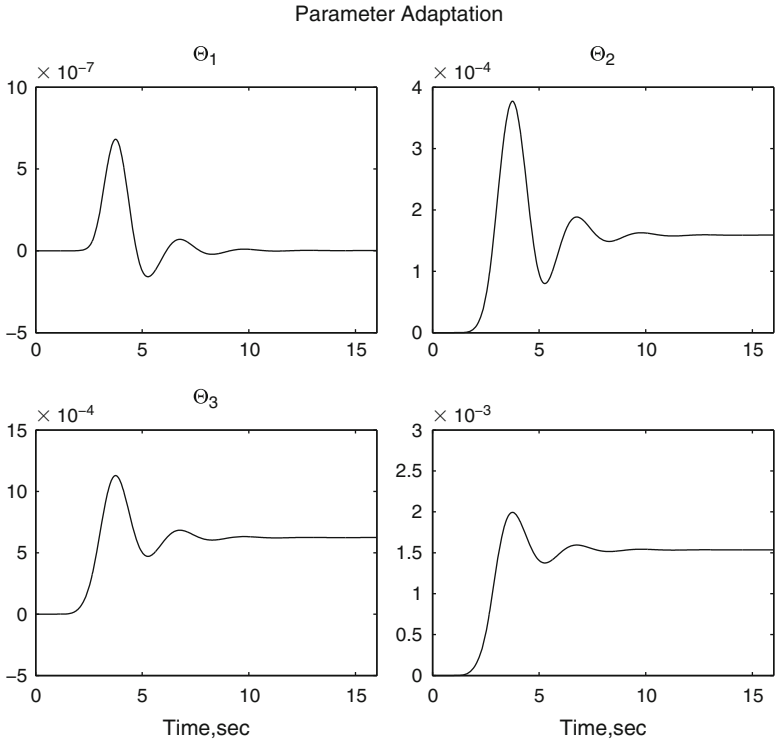

Fig. 5.15 Model-reference adaptive control in nonlinear engine: parameter adaptation histories

The relative-degree 1 MRAC design is now applied to the 90k nonlinear engine of CMAPSS-1. Altitude, Mach number, and fan speed are varied during simulation, the same way as in the example of Sect. 5.3.2.2. The MRAC system was tuned with $G = 1 \times 10^{-8}\ I_4$, $\tau = 0.5$, and initial parameter vector $\Theta_0 = [0 \quad 0 \quad 0 \quad 0]^{\mathrm{T}}$. Figure 5.14 shows the fan speed and fuel flow input responses corresponding to the MRAC in comparison with the CMAPSS-1 native gain-scheduled fan speed controller. Even with zero as initial parameter guesses, the adaptive control system is able to attain zero offset and reasonable transient response characteristics. Figure 5.15 shows how parameters are adapted and converge to steady values. A "bootstrapping" tuning procedure may be used to improve transient response. After a first simulation with $\Theta_0 = 0$, the resulting steady values of Θ may be used as Θ_0. A new simulation is run, with an improvement in transient response. The first simulation converged to $\Theta = [0 \quad 0.0002 \quad 0.0006 \quad 0.0015]^{\mathrm{T}}$. Figure 5.16 shows how performance is significantly improved by setting Θ_0 to these new values. The process may be repeated to produce further improvements, but the refinement process will tend to tailor Θ_0 to a particular simulation, and the same transient qualities will not be observed for a new set of plant parameter changes. A good choice for Θ_0 is best obtained by a simulation study comprising several plant parameter variation

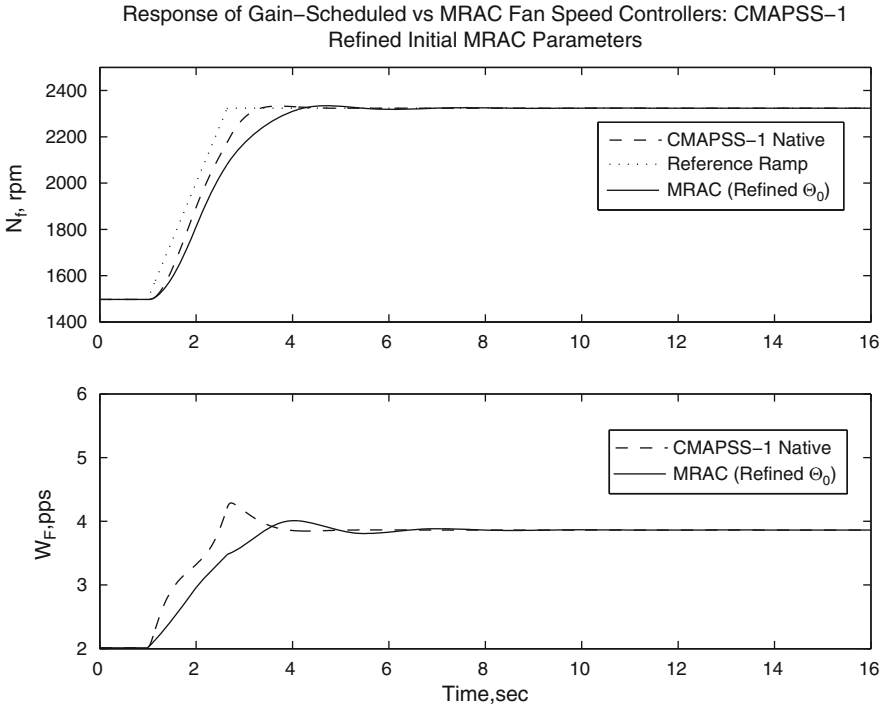

Fig. 5.16 Response of model-reference adaptive control with initial parameter refinement

scenarios. Theoretical results guarantee that the error will converge to zero and that the parameters will converge to steady values. Instability can be observed in simulation, however, due to interaction of adaptive dynamics and numerical solution algorithms. The CMAPSS simulation presented here used a 2nd-order Runge–Kutta method (Heun's algorithm) with a fixed step size of 0.015. Under these conditions, choosing the entries of G as 1×10^{-7} leads to large numerically-induced oscillations. Values near 5×10^{-7} cause unboundedness of Θ and ω.

Chapter 6
Sliding Mode Control of Turbofan Engines

Abstract This chapter introduces the concept of sliding mode control and elaborates in its robustness properties and commonly-used tuning approaches. This chapter also presents MIMO versions of the sliding mode regulator and setpoint tracker, as well as a simplified SISO design. Linear and nonlinear engine simulations using CMAPSS are included.

Sliding Mode Control (SMC) denotes a family of nonlinear techniques having particularly strong robustness and disturbance rejection properties. These properties arise from invariance conditions imposed on the system as part of the SMC formulation. Regardless of its initial value, the state of the system is required to reach a set called *sliding manifold*, or *sliding surface* in finite time, and remain in the set beyond the reaching time. The system is then said to undergo a *sliding mode*. Depending on certain conditions assumed for the disturbance and plant parameter variations, the system can be made totally insensitive to these uncertainties while in sliding mode. The ability to reach the sliding mode in finite time is guaranteed under mild conditions on the disturbance and parameter uncertainties.

When the system is in sliding mode, the states are constrained by the relationship defining the sliding surface. Suppose $x = [x_1 \;\; x_2]^{\mathrm{T}}$ is the state vector of a second-order system and the sliding surface is defined as the set $\mathcal{S} = \{(x_1, x_2) : x_1 + x_2 = 0\}$. Then the states are constrained to be the negative of each other during the sliding mode. This implies that two states are no longer necessary to describe system dynamics during the sliding mode. Only one expression is needed for the derivative of one of the states, and the other state is determined from the algebraic constraint $x_1 + x_2 = 0$. This *order reduction* is a universal characteristic of SMC systems having sliding manifolds of the form $\mathcal{S} = \{x : s(x) = 0\}$, where x is the state vector and $s(x)$ is the *sliding function*.

The reduced-order dynamics governing the system in the sliding mode must be made stable by proper choice of sliding function. As elaborated further in the chapter, this represents one of the two essential SMC design tasks. The other is to ensure that the system actually reaches the sliding mode and that its invariance

H. Richter, *Advanced Control of Turbofan Engines*, DOI 10.1007/978-1-4614-1171-0_6,

holds for a range of disturbances and parameter variations. Exact enforcement of the invariance property relies on *chattering control*, an infinitely fast, finite-amplitude switching of the control input. Although theoretical chattering cannot be reproduced by any physical actuator, or even simulated in a computer, approximations to switching control laws are available that offer a manageable tradeoff between high-frequency control activity and relaxed forms of robustness. These relaxed robustness properties still exceed what is achievable with fixed linear controllers.

The chapter begins with an example providing insight into the motivations behind the SMC concept. Next, in Sect. 6.2, a MIMO SMC regulator is presented, along with suitable methods to design various control parameters. In Sect. 6.3, an output tracking design is introduced, and a simplified SISO fan speed tracking design is developed in Sect. 6.4. Simulation examples using CMAPSS-1 are included throughout the chapter.

6.1 Motivation Example: On–Off Rocket Thruster Control

Consider the problem of controlling the velocity of a projectile so that it trails a moving target by a constant distance, as shown in Fig. 6.1. The x position of the projectile is controlled using a thruster capable of being either on or off. Suppose that the target moves with a known constant acceleration $\ddot{r}$. System dynamics are given by the following differential equation:

$$m\ddot{x} = u,$$

where m is the mass of the projectile and u is the force produced by the thruster, limited to the two values $\{U_{\min}, U_{\max}\}$. Projectile dynamics are given in state-space form by:

$$\dot{x}_1 = x_2,$$
$$\dot{x}_2 = \frac{u}{m},$$

where x_1 is the position of the projectile and x_2 its velocity. Suppose first that m is known exactly. The velocity of the projectile must match that of the moving target, if a constant distance is to be maintained. Define a sliding function as the difference between the velocities of the target and projectile:

$$s = \dot{r} - x_2.$$

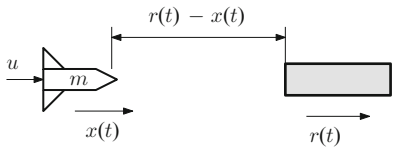

Fig. 6.1 Schematic of projectile velocity tracking system

The control objective can be expressed in terms of this function as the requirement that $s = 0$ is reached in finite time, and that the zero value is maintained thereafter. If $\dot{r} > x_2$ at any given time, the target is escaping, s is positive and must be reduced, that is, $\dot{s} < 0$ is needed. If $\dot{r} < x_2$ at any given time, the projectile is on a collision course, s is negative and must be increased, that is $\dot{s} > 0$ is required. *Clearly, s and its derivative must have opposite signs*. That is, we may enforce

$$\dot{s} = -\eta \text{ sign } (s), \tag{6.1}$$

where η is a positive quantity affecting the "strength" of the reaction of $\dot{s}$ to the sign of s. Note that s is essentially an error term, and that similar reactions to the sign of the error are commonplace in household and industrial thermostats. Taking the derivative of s from the definition, we obtain the following control law from (6.1):

$$u = m(\ddot{r} + \eta \text{ sign } (s))$$

Assuming that $\ddot{r}$ is a positive constant, we see that u can give rise to only two values. The design is completed by choosing two values of η so that u matches the available thruster forces:

$$\begin{aligned} m(\ddot{r} + \eta^+) &= U_{\max}, \\ m(\ddot{r} - \eta^-) &= U_{\min}. \end{aligned}$$

Substituting control law (6.1) into the state-space description of the plant yields:

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= \ddot{r} + \eta \text{ sign } (\dot{r} - x_2), \end{aligned}$$

where the *signum function* sign(s) is defined as

$$\text{sign}(s) = \begin{cases} 1, & s > 0 \\ -1, & s < 0 \end{cases}.$$

Although the sign of zero is undefined, software packages assign it an arbitrary value. In Matlab, for instance `sign(0)` returns zero. In the sliding mode, $s = 0$, that is, $\dot{r} = x_2$, reducing the closed-loop system description to one differential equation and one algebraic equation:

$$\begin{aligned} \dot{x}_1 &= \dot{r}, \\ x_2 &= \dot{r}. \end{aligned}$$

Only the state equation for $\dot{x}_1$ is needed. The second state is x_2 is determined from the input $\dot{r}$ through an algebraic relationship (simple identity in this case).

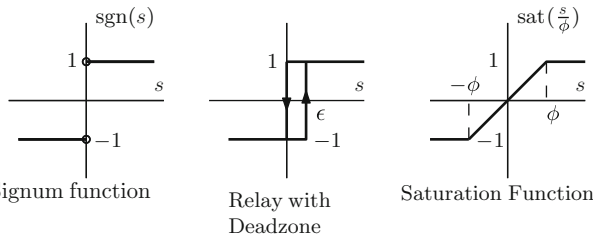

Fig. 6.2 Signum function, relay and saturation function characteristics

Real-time operation and even system simulation are not possible with the control law of (6.1). In the first case, no actuator is capable of instantaneous switching between two distinct values. In the second, the number of consecutive zero crossings of the variable s becomes unbounded, resulting in the failure of numerical integration methods. It must be made clear, however, that chattering control functions and state trajectories arising from them are valid mathematical constructs. Their proper description requires the notion of *differential inclusion*, an extension of the notion of differential equation. The interested reader is referred to Utkin [53], one of the main proponents of SMC.

From an engineering perspective, suitable approximations to ideal switching are often sufficient. A clear example is given by thermostats, and more generally, relay control systems. Relays are designed to have a *deadzone*, a set of values of s where switching is not allowed. A typical deadzone implementation of the control law of (6.1) is as follows:

$$u = \begin{cases} U_{\max}, & s \text{ increasing and } s \geq \epsilon \\ U_{\min}, & s < 0 \end{cases}.$$

The positive quantity ϵ is the dead band, in this example situated entirely on the nonnegative side of s. Another commonly-used approximation to the signum function is given by the *saturation function*, defined as

$$\text{sat}(s/\phi) = \begin{cases} 1, & s \geq \phi \\ s/\phi, & -\phi < s < \phi \\ -1, & s \leq -\phi \end{cases}, \tag{6.2}$$

where ϕ is a small positive constant known as *boundary layer thickness*, used to adjust the accuracy of the approximation. The signum function and its approximations by relay with deadzone and saturation function are illustrated in Fig. 6.2.

This system is now simulated using $m = 1$, $U_{\max} = 10$, $U_{\min} = 0$. Figure 6.3 shows that the projectile quickly matches the velocity of the target, thus trailing it at a constant distance. The relay used $\epsilon = 0.01$ and a switch-off point of

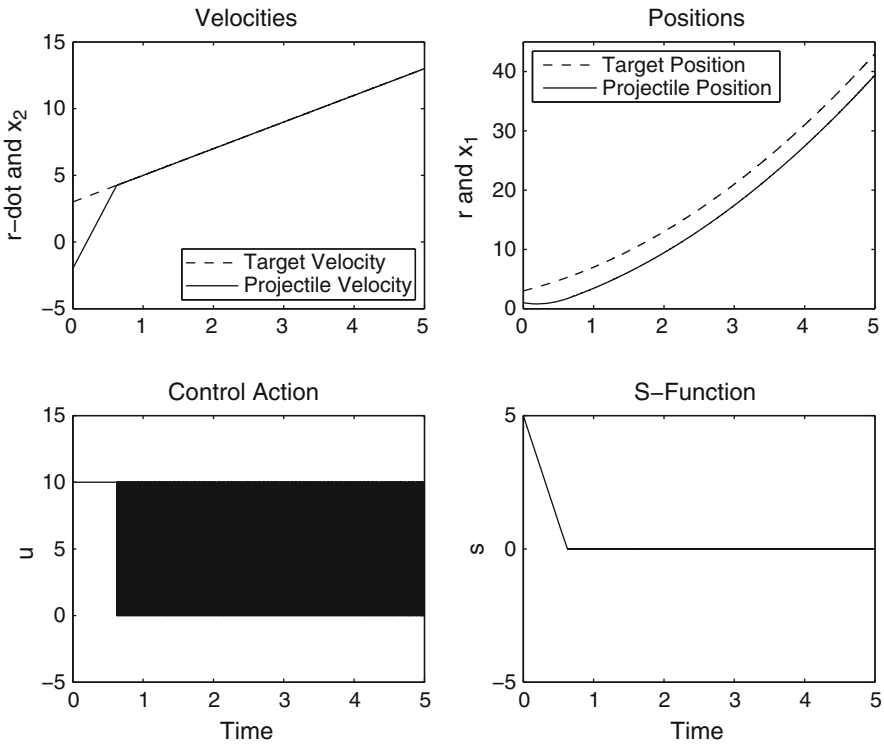

Fig. 6.3 Response of the projectile velocity tracking system using a relay with deadzone

$s = 0$. This results in high-frequency switching of the thruster between the off and on states. The s-function reaches zero in less than 1 s. Figure 6.4 shows system trajectories when the saturation function is used in place of the relay, with $\phi = 0.01$. The velocity of the target is matched within a small error, and control chattering is greatly reduced. Figure 6.5 compares the tracking accuracies of relay with deadzone and saturation function.

6.1.1 Adding Uncertainty and Disturbance

Although the above velocity tracking problem could have been formulated using a simple classical linear controller, it is the outstanding robustness properties what makes SMC a far better solution. Consider that the mass is not exactly known, but that a nominal value and maximum uncertainty are available. That is, take

$$m = m_0 + \Delta m,$$

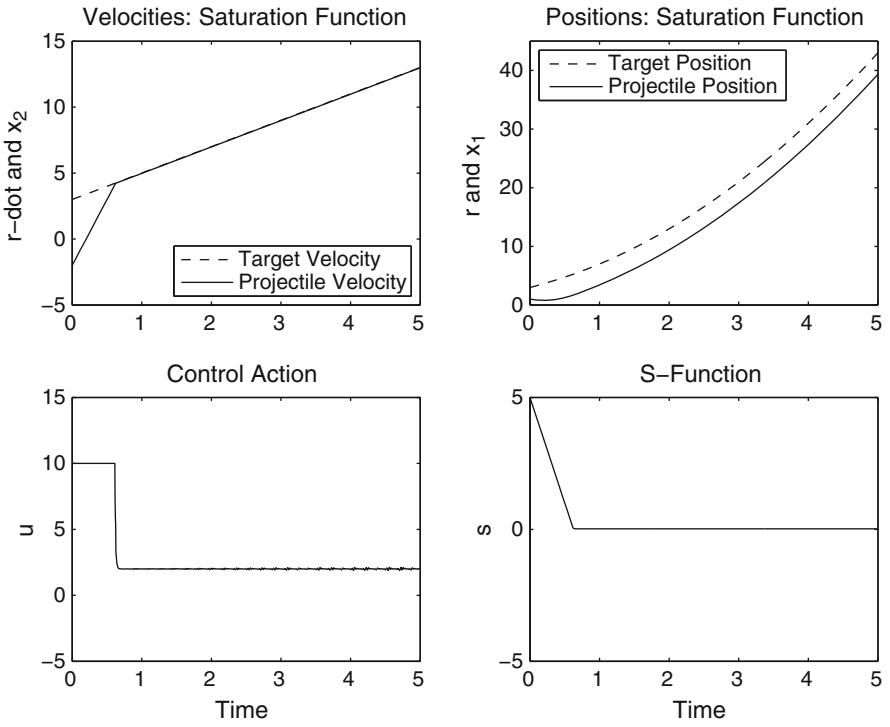

Fig. 6.4 Response of the projectile velocity tracking system using a saturation function

where m_0 is the nominal value and $|\Delta m| \leq b_m$, for some positive bound. We consider that $b_m < m_0$, so that the uncertain mass remains positive. Also, consider that an unknown disturbance force $f(t)$ (due to aerodynamic drag, for example) acts opposing the direction of motion. The disturbance force is assumed to be bounded in absolute value by a constant, that is

$$|f(t)| \leq b_f.$$

The differential equation describing projectile motion is now

$$(m_0 + \Delta m)\ddot{x} = u + f(t)$$

and the corresponding state-space description is

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= \frac{u}{m_0 + \Delta m} + \frac{f(t)}{m_0 + \Delta m}. \end{aligned}$$

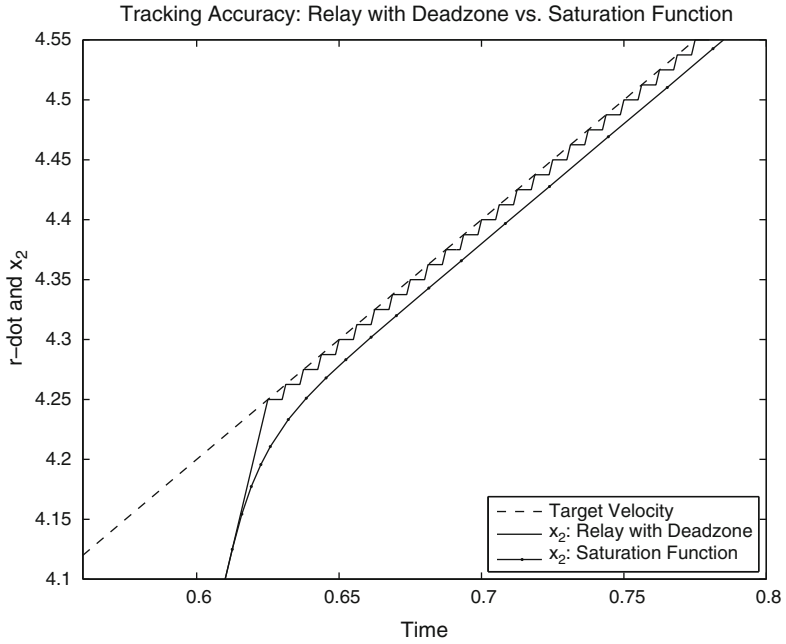

Fig. 6.5 Comparison of velocity tracking accuracies: relay with deadzone vs. saturation function

Define the sliding function in the same way as before, that is: $s = \dot{r} - x_2$. If Δm and $f(t)$ were exactly known, they could be incorporated in the control law to produce a cancelation of terms and achieve $\dot{s} = -\eta \text{ sign}(s)$ as done before. That is, one could choose

$$u = (m_0 + \Delta m)\left(\ddot{r} + \eta \text{ sign } (s) - \frac{f(t)}{m_0 + \Delta m}\right).$$

Since Δm and $f(t)$ are unknown, we show that the control law

$$u = m_0(\ddot{r} + \eta \text{ sign } (s))$$

may still be used, provided η is chosen appropriately. Instead of requiring $\dot{s} = -\eta$ sign(s), we define a function as

$$V = \frac{1}{2}s^2.$$

This function is a measure of the distance of the system state to the set defined by $s = 0$, and it is nonnegative. Therefore, we impose the requirement that the time derivative of V be negative whenever $s \neq 0$. This requirement guarantees that s will reach zero in finite time. Indeed, note that $\dot{V} = s\dot{s}$, thus s and $\dot{s}$ have opposite signs. Then s will decrease if it is positive, and it will increase if it is negative. Once $s = 0$, $V = 0$ and $\dot{V} = 0$. Now, V cannot increase due to the imposed requirement

on $\dot{V}$, nor can it decrease, since V has already reached its minimum value of zero. The only possibility is that s remain at zero indefinitely.

The requirement

$$s\dot{s} < 0 \tag{6.3}$$

is known as in SMC theory as a *sliding mode attractiveness* condition. Continuing with the example, we derive the requirements on η so that the set defined by $s = 0$ is attractive. By the assumptions on disturbance and mass uncertainty bounds, the following inequalities hold:

$$|\Delta m| \leq b_m < m_0, \tag{6.4}$$

$$0 < m_0 + b_m \leq |m_0 + b_m|, \tag{6.5}$$

$$\left| \frac{\ddot{r} m_0}{m_0 + \Delta m} \right| \leq \frac{\ddot{r} b_m}{m_0 - b_m}, \tag{6.6}$$

$$|f(t)| \leq b_f, \tag{6.7}$$

$$\left| \frac{f(t)}{m_0 + \Delta m} \right| \leq \frac{b_f}{m_0 - b_m}. \tag{6.8}$$

Now, the derivative of V is

$$\dot{V} = s\dot{s} = s \left[\ddot{r} \frac{\Delta m}{m_0 + \Delta m} - \frac{m_0}{m_0 + \Delta m} \eta \text{ sign } (s) - \frac{f(t)}{m_0 + \Delta m} \right].$$

Thus, if η is selected to satisfy

$$\eta > \left(\frac{m_0 + \Delta m}{m_0} \right) \left(\frac{\ddot{r} b_m + b_f}{m_0 - b_m} \right)$$

inequalities (6.4)–(6.8) show that η is large enough to "overcome" the uncertainties and guarantee that $s\dot{s} < 0$. Thus, the sliding surface will be reached in finite time and will remain invariant regardless of the value of Δm and the actual disturbance function $f(t)$. Velocity tracking during the sliding mode is *totally insensitive* to such parameter variations and disturbance.

In this example, we consider that only two control values are available, which must be matched to corresponding values of η. That is,

$$\eta^+ = \frac{U_{\max}}{m_0} - \ddot{r} > \left(\frac{m_0 + \Delta m}{m_0} \right) \left(\frac{\ddot{r} b_m + b_f}{m_0 - b_m} \right)$$

$$\eta^- = \ddot{r} - \frac{U_{\min}}{m_0} > \left(\frac{m_0 + \Delta m}{m_0} \right) \left(\frac{\ddot{r} b_m + b_f}{m_0 - b_m} \right).$$

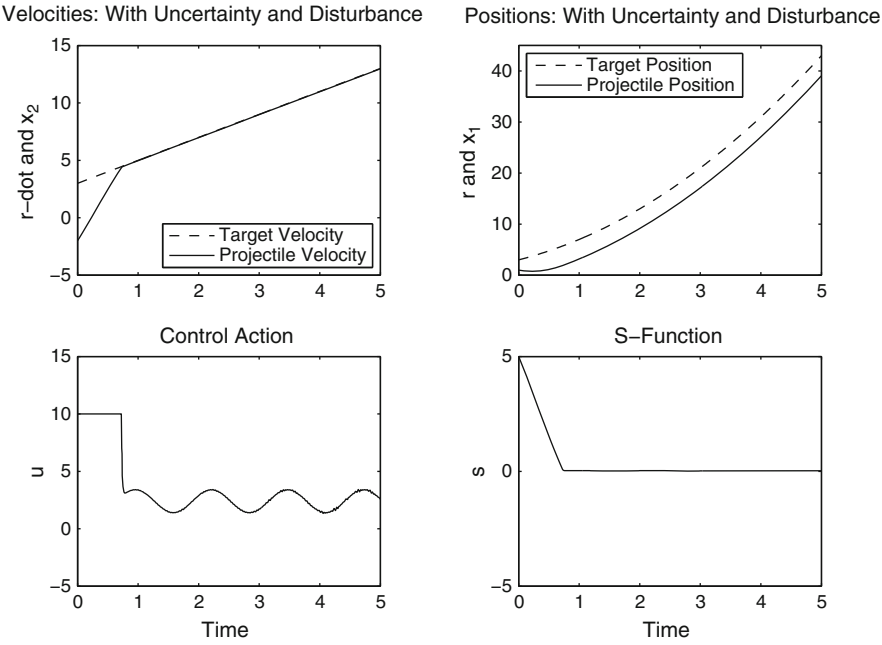

Fig. 6.6 Response of the projectile velocity tracking system with parametric uncertainty and disturbance

Therefore, for the problem to be feasible, a minimum level of thruster force is required for each direction:

$$U_{\max} > m_0 \left[\ddot{r} + \left(\frac{m_0 + b_m}{m_0} \right) \left(\frac{\ddot{r} b_m + b_f}{m_0 - b_m} \right) \right],$$

$$U_{\min} < m_0 \left[\ddot{r} - \left(\frac{m_0 + b_m}{m_0} \right) \left(\frac{\ddot{r} b_m + b_f}{m_0 - b_m} \right) \right].$$

As an example, consider a 20% uncertainty in the nominal mass, that is, $m_0 = 1$ and $|\Delta m| \leq 0.2 = b_m$. Consider that the disturbance force is bounded as $|f(t)| \leq 0.1 = b_f$. For simulation, a sinusoidal disturbance with an amplitude equal to b_f and a frequency of 5 rad/s was used along with a perturbed mass of $m_0 + \Delta m = 1.2$. The value of $U_{\min} = 0$ used earlier is not adequate for this "uncertainty budget". In fact, the above inequalities for $U_{\max}$ and $U_{\min}$ indicate that negative thruster action will be required, at a level $U_{\min} = -0.1$. Simulation is conducted with $U_{\min} = -0.5$ and $U_{\max} = 10$. The saturation function is used with $\phi = 0.01$. Figure 6.6 shows that the velocity tracking objective is met perfectly (up to saturation function approximation). Note that the control input "absorbs" the sinusoidal disturbance, applying force corrections to the projectile so that the disturbance is not propagated to the velocity.

The rather powerful robustness and total insensitivity properties seen in this example hold for a special case of disturbances and parameter variations known as *matched uncertainties*. As elaborated in the following sections, general uncertainty can be split into a matched component, which is entirely canceled by the control input, and an unmatched component, whose effects can be minimized by proper system tuning.

6.2 Multivariable SMC Regulator

The state-space regulation problem for uncertain plants is now considered. Let the plant description be given by

$$\dot{x} = (A + \Delta A)x + (B + \Delta B)u + \Gamma w, \tag{6.9}$$

where ΔA and ΔB can be used to reflect uncertainties or time variations in A and B. The state equation may be rewritten as

$$\dot{x} = Ax + Bu + d(x, u, t), \tag{6.10}$$

where $d(x, u, t) = \Delta Ax + \Delta Bu + \Gamma w(t)$ represents a generalized uncertainty term. The dimensions of Γ are assumed to be n-by-h, corresponding to an h-component disturbance w. Also, the state vector is of dimension n and there are m control inputs. For reasons that will justified below, it is assumed that $m < n$, B is full-rank and (A, B) is a controllable pair.

6.2.1 Matched Uncertainties

If the generalized uncertainty can be expressed as

$$d(x, u, t) = B\zeta(t, x, u) \tag{6.11}$$

for some new uncertainty vector $\zeta(t, x, u)$, then the uncertainties are said to be *matched*. Matched uncertainties thus enter the plant through the same channels as the control inputs. It is useful to analyze the conditions that must be met for uncertainties to be of the matched type. Dropping the arguments of d and ζ from the notation, note that d has dimensions n-by-1, so ζ must be m-by-1. Regarding (6.11) as a system of n linear equations where ζ is a vector of m unknowns, and continuing with the assumption that $m < n$, we see that the system of equations is overdetermined. Recalling basic linear algebra, a solution for ζ will exist only if d is a linear combination of the columns of B, (assumed to be full-rank). Mathematically, this is expressed as

$$d \in \text{col } B, \tag{6.12}$$

where col denotes the linear space spanned by the columns of a matrix.

In our GTE problem, there are two states, thus only one actuator may be used in SMC designs based on (A, B) to meet the assumption that $m < n$. The generalized uncertainty ζ becomes a scalar, and d must be parallel to B for the matching condition to be satisfied.

As the reader may appreciate, it is difficult to satisfy the matching condition exactly. However, as illustrated by Khalil [54], a general uncertainty may be decomposed into matched and unmatched components. SMC will be totally insensitive to the matched portion, without additional design effort, while the effects of the unmatched component can be minimized by proper design. Thus, every stable SMC tuning reduces the sensitivity of the closed-loop system to general uncertainties.

6.2.2 Control Law Development

The regulation objective is to drive the state from an arbitrary initial point to zero. Introduce a vector of m sliding variables as

$$s = Gx,$$

where G is an m-by-n full-rank matrix of *sliding coefficients*. The sliding coefficients will be selected by the designer to prescribe desired dynamics to the nominal system in the sliding mode defined by $s = \dot{s} = 0$. It is assumed that this choice satisfies the *tranversality condition* that GB is nonsingular.

The first stage of the control action is to bring the system state to the sliding mode. This stage of the control law derivation is known as "sliding mode attractiveness" or "reachability problem". As done in the introductory example of Sect. 6.1.1, the condition $s\dot{s} < 0$ guarantees that $s = 0$ will be reached in finite time. One way of enforcing this condition for nonscalar sliding functions is to make

$$\dot{s} = -\Xi \overrightarrow{\text{sign}}(s),$$

where Ξ is a diagonal matrix of positive gains and $\overrightarrow{\text{sign}}(s)$ is a vector whose components are $\text{sign}(s_i)$, $i = 1, 2, ..m$. If the generalized disturbance were known, the hypothetical control input u_{eq} required to achieve this condition could be found using $\dot{s} = G\dot{x} = G(Ax + Bu + d)$:

$$u_{\text{eq}} = -(GB)^{-1}(GAx + \Xi \overrightarrow{\text{sign}}(s) + Gd).$$

This control input is known as *equivalent control*. Under the action of u_{eq}, the sliding mode is eventually reached and $s = 0$ subsequently, regardless of the variation of d. Substituting u_{eq} into the plant description of (6.10), the sliding mode dynamics become

$$\dot{x} = (A - B(GB)^{-1}GA)x - B(GB)^{-1}Gd + d. \tag{6.13}$$

Now, if the uncertainty is matched, $d = B\zeta$, so the terms containing d in the above equation cancel out, reducing the sliding dynamics to

$$\dot{x} = (A - B(GB)^{-1}GA)x = A_{\text{eq}}x. \tag{6.14}$$

This *total insensitivity* to matched uncertainty in the sliding mode is one of the trademark features of SMC. The other one is its ability to reach $s = 0$ robustly, for which the matching condition is not required, as seen next. Since d is unknown, it cannot be part of the control law. A valid SMC control law can be specified simply by omitting the uncertainty term:

$$u = -(GB)^{-1}(GAx + \Xi \overrightarrow{\text{sign}}(s)). \tag{6.15}$$

It should be clear that the stability of the sliding mode dynamics is determined by (6.14). Since d has been omitted, it remains to verify that the sliding mode can still be established. For this, the sign of $s_i\dot{s_i}$ under control law (6.15) is examined. We have

$$s_i\dot{s_i} = s(-\eta_i \text{ sign } (s_i) + G_i d),$$

where η_i is the i-th diagonal entry in Ξ and G_i is the i-th row of G. Following a reasoning similar to the one used in Sect. 6.1.1, it is clear that the reaching condition is ensured if $d(t, x, u)$ is bounded and η_i is chosen large enough. Explicitly, it is necessary that $\eta_i > ||G_i d||$. Using the triangle inequality, the following sufficient condition is obtained:

$$\eta_i > ||G_i||\Delta,$$

where the generalized uncertainty satisfies $||d(t, x, u)|| \leq \Delta$. Obtaining such a bound for the most general form of uncertainty is impractical, since it depends on the peak magnitudes of x and u, which are in turn influenced by η_i. When no uncertainties are considered for A and B, the bound is readily obtained from the assumed bound on disturbance vector w and the norm of Γ.

6.2.3 Reduced-Order Dynamics and Sliding Coefficient Selection

When the system is in sliding mode, the states satisfy the algebraic constraint $Gx = 0$. This implies that a set of states is linked to the remaining states through an algebraic relationship. Since G is m-by-n and full-rank and x is n-by-1, the case $m = n$ would reduce the sliding set to the singleton set defined by $x = 0$, and insensitivity to uncertainties would not hold until the system state has reached zero. Control chattering would be immediately propagated to the system state, in absence of sliding dynamics acting as a filter. In turn, when $m > n$, G contains redundant rows, cannot be full-rank and GB becomes singular. Therefore, the assumption that $m < n$ is justified and maintained.

The algebraic constraint $Gx = 0$ represents an underdetermined and homogeneous system of linear equations. Since G is full-rank, m state variables can be expressed as linear functions of the remaining $n-m$ states. Equations (6.13) and (6.14) must therefore contain redundant state variables. Indeed, it is possible to show that A_{eq} has at least m zero eigenvalues, regardless of how G is chosen. For stable sliding mode dynamics, A_{eq} must have $n - m$ eigenvalues with negative real parts. If G is chosen appropriately, there will be exactly m zero eigenvalues. Moreover, $I - B(GB)^{-1}G$ has $n - m$ unity eigenvalues and m zero eigenvalues. This matrix appears as the disturbance input distribution matrix in (6.13).

As an example, consider the following pair of system matrices

$$A = \begin{bmatrix} 1 & 2 & -1 & 2 \\ 2 & 2 & 0 & 1 \\ -1 & 4 & 2 & 0 \\ 2 & 4 & 4 & -1 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 1 \\ 1 & -1 \\ 2 & 4 \\ 1 & 1 \end{bmatrix}$$

and the following sliding coefficient matrix:

$$G = \begin{bmatrix} 0.9674 & 0.7188 & 0.6433 & 0.4440 \\ -1.2572 & -0.9562 & 1.1158 & -1.2753 \end{bmatrix}.$$

The reader can verify that the eigenvalues of A_{eq} are 0, 0, -1 and -2. Here, two states can be written as a function of the other two when the system is in sliding mode. This may be done by direct algebraic manipulation of the equation $Gx = 0$. The redundant state equations can likewise be eliminated from the sliding mode dynamics of (6.13) and (6.14).

This elimination can be made systematic by introducing a convenient similarity transformation, which puts the system in *regular form*. Let T be an invertible n-by-n matrix satisfying

$$TB = \begin{bmatrix} 0 \\ B_2 \end{bmatrix}, \tag{6.16}$$

where B_2 is an m by m matrix. When B is full-rank, such matrix can always be found by the following procedure: the first $n - m$ rows of T are chosen as vectors which are orthogonal to each column of B, so that the zero matrix is obtained. This is always possible, since $m < n$, thus the space spanned by the columns of B has dimension m. This space can be extended into a basis of an n-dimensional space using $n - m$ additional vectors, which are orthogonal to col (B). The remaining m rows of T are chosen so that T is full-rank, but are otherwise arbitrary, resulting in some B_2 matrix. Edwards and Spurgeon [55] present a convenient method and list a Matlab program to find an orthogonal matrix T using the QR decomposition of B.

Introduce a change of coordinates $z = Tx$. In terms of these coordinates, the plant has the state-space description:

$$\begin{aligned} \dot{z}_1 &= A_{11}z_1 + A_{12}z_2 + \gamma_1, \\ \dot{z}_2 &= A_{21}z_2 + A_{22}z_2 + B_2u + \gamma_2, \end{aligned} \tag{6.17}$$

where

$$TAT^{-1} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{6.18}$$

and γ_1 and γ_2 are, respectively, the first $n-m$ and last m rows of Td. Note that A_{11} is square with dimension $n-m$ and A_{12} is $n-m$ by m. To understand the order reduction introduced by the sliding mode, express s in z coordinates as

$$s = Gx = GT^{-1}z = G_{z1}z_1 + G_{z2}z_2,$$

where G_{z1} contains the first m columns of GT^{-1} and G_{z2} the remaining m. G_{z2} acts as a scaling factor and does not influence system dynamics in the sliding mode. Since the design of G will be conducted by specifying GT^{-1} first, we may assume that G_{z2} has been chosen to be invertible. Usually, $G_{z2} = I$ is chosen by designers for simplicity. Adopting this choice, we rederive control law (6.15) in z coordinates:

$$u = -B_2^{-1}[(G_{z1}z_1A_11 + A_{21})z_1 + (G_{z2}A_{12} + A_{22})z_2 + \Xi\overrightarrow{\text{sign}}(s)]. \tag{6.19}$$

Substituting this law into (6.17) yields the following description of the closed-loop dynamics:

$$\dot{z}_1 = A_{11}z_1 + A_{12}z_2 + \gamma_1, \tag{6.20}$$

$$\dot{z}_2 = -G_{z1}(A_{11}z_1 + A_{12}z_2) - \Xi\overrightarrow{\text{sign}}(s) + \gamma_2. \tag{6.21}$$

Under the SMC law, the sliding variables obey the relationship:

$$\dot{s} = -\Xi\overrightarrow{\text{sign}}(s) + G_{z1}\gamma_1 + \gamma_2.$$

Thus, (6.21) can be rewritten as:

$$\dot{z}_2 = -G_{z1}(\dot{z}_1 - \gamma_1) + \dot{s} - G_{z1}\gamma_1 = -G_{z1}\dot{z}_1 + \dot{s}.$$

This implies that z_2 may be integrated as follows

$$z_2(t) = -G_{z1}z_1(t) + s(t) + C,$$

where C is a constant reflecting initial conditions. Noting that the above equation is just a rearrangement of the definition of s, we see that $C = 0$.

A better representation of closed-loop system dynamics is obtained by regarding s as a state variable, along with z_1. The following description accounts for system behavior during the reaching phase as well as the sliding mode:

$$\dot{z}_1 = (A_{11} - A_{12}G_{z1})z_1 + A_{12}s(t) + \gamma_1(t), \tag{6.22}$$

$$\dot{s} = -\Xi \overrightarrow{\text{sign}}(s) + G_{z1}\gamma_1(t) + \gamma_2(t). \tag{6.23}$$

Once more we observe that if the entries of Ξ are chosen large enough in relation to the disturbance bounds, s will reach zero in finite time.

Equation (6.23) shows that the evolution of s is unaffected by z_1. Then, $s(t)$ may be regarded as an exogenous input to (6.22) alongside γ_1. For the overall nonlinear system to be stable, it is then sufficient that $(A_{11} - A_{12}G_{z1})$ have all its eigenvalues with negative real parts.

Turning attention now to the sliding regime, system dynamics reduce to

$$\dot{z}_1 = (A_{11} - A_{12}G_{z1})z_1 + \gamma_1(t).$$

If the disturbance is matched $Td = TB\zeta$. In this case $\gamma_1 = 0$, as seen from (6.16). Since the closed-loop dynamic equations in z coordinates were obtained from x coordinates by similarity transformation, the eigenvalues of the corresponding system matrices must be preserved. Therefore, from (6.22) and (6.23), we see that the eigenvalues of $(A_{11} - A_{12}G_{z1})$ must match the nonzero eigenvalues of A_{eq} of (6.14). This result provides a commonly-used approach to prescribe desired dynamics for the sliding mode: direct pole placement based on the pair (A_{11}, A_{12}), with G_{z1} acting as the state-feedback gain.

It can be shown that (A_{11}, A_{12}) is a controllable pair whenever (A, B) is controllable, justifying the assumption at the beginning of this section. Under this assumption, the designer may freely assign $n - m$ eigenvalues to A_{eq}. The resulting G_{z1} is then combined with $G_{z2} = I$, and the desired sliding coefficients are found as $G = [G_{z1}\ I]T$.

6.2.4 *Utkin and Young's LQ Method*

Direct pole placement is simple and convenient for small values of $n - m$. For problems of higher dimensionality, pole placement for SMC presents the same difficulties found in state feedback design by manual pole placement: multiple pole locations are not easily linked to transient response characteristics.

Many methods have been developed over the years to substitute manual pole placement in conjunction with SMC. The *eigenstructure assignment* approach seeks to minimize the effects of unmatched uncertainties on the sliding dynamics. As seen in (6.22), G_{z1} not only determines the eigenvalues of $(A_{11} - A_{12}G_{z1})$, but also

partially determines the effects of γ_1 on z_1. Eigenstructure methods take advantage of the unused degrees of freedom associated with pole placement to minimize unmatched disturbance effects. The interested reader is referred to Edwards and Spurgeon [55] for a detailed exposition.

A simple and intuitive approach to select G is given by the quadratic minimization method developed by Utkin and Young [56]. The objective is to minimize a cost function of the form

$$J = \int_{t_r}^{\infty} x(t)^{\mathrm{T}} Q x(t) \mathrm{d}t \tag{6.24}$$

subject to system dynamics in the sliding mode, where t_r denotes the time at which the sliding mode begins and Q is a symmetric, positive-definite weight matrix. As explained in detail in Utkin and Young [56] and Edwards and Spurgeon [55], the quadratic minimization problem can be expressed in z-coordinates as

$$\min \quad J = \int_{t_r}^{\infty} z_1^{\mathrm{T}} \hat{Q} z_1 + v^{\mathrm{T}} Q_{22} v \mathrm{d}t \text{ subject to,} \tag{6.25}$$

$$\dot{z}_1 = \hat{A} z_1 + A_{12} v, \tag{6.26}$$

where

$$\hat{Q} = Q_{11} - Q_{12} Q_{22}^{-1} Q_{21}, \tag{6.27}$$

$$\hat{A} = A_{11} - A_{12} Q_{22}^{-1} Q_{21}, \tag{6.28}$$

$$v = z_2 + Q_{22}^{-1} Q_{21} z_1 \tag{6.29}$$

and Q_{ij} are block matrices arising from partitioning TQT^{-1} in the same way as TAT^{-1}. The minimization problem of (6.25) and (6.26) has the standard LQR form, with v regarded as the control input. The solution is then a static state feedback law of the form

$$v = -K z_1,$$

where K may be obtained using Matlab's `lqr` command. Substituting the optimal solution for v into (6.29) yields

$$z_2 = -(K + Q_{22}^{-1} Q_{21}) z_1.$$

Since the minimization is conducted under the constraint that the system is in the sliding mode, we have $s = 0$, thus $z_2 = -G_{z1} z_1$. By direct comparison, the optimizing value of G_{z1} is given by

$$G_{z1} = K + Q_{22}^{-1} Q_{21}.$$

6.2.5 SMC Regulator Example: CMAPSS-40k

In this example, an integral multivariable SMC regulator is designed to meet setpoint command changes. The starting and ending flight conditions are the same as in the example of Sect. 4.8. The SMC controller was designed using the linearized matrices corresponding to the Ground Idle condition listed in Table 2.4. To facilitate design, the plant is first scaled as follows:

$$x_s = T_x x = \begin{bmatrix} \frac{1}{500} & 0 \\ 0 & \frac{1}{1000} \end{bmatrix} x,$$

$$v = T_u u = \begin{bmatrix} \frac{1}{5} & 0 & 0 \\ 0 & \frac{1}{100} & 0 \\ 0 & 0 & 1 \end{bmatrix} u.$$

These scalings give rise to a set of transformed plant matrices of the form $(T_x A T_x^{-1}, T_x B T_u^{-1})$. The augmented plant is then assembled using the scaled matrices. Since the augmented plant has five states and there are three inputs, the sliding dynamics are described by second-order dynamics. Consequently, the pole-assignment problem conducted using the pair (A_{11}, A_{12}) requires the specification of two closed-loop eigenvalues. Choosing -2.5 and -3 as desired pole locations, the following sliding coefficient vector is obtained in scaled coordinates:

$$G_s = \begin{bmatrix} -0.0658 & 0.2296 & 1.0000 & 0 & 0 \\ 0.0563 & -0.1349 & 0 & 1.0000 & 0 \\ 0.0003 & -0.0010 & 0 & 0 & 1.0000 \end{bmatrix}.$$

The controller is implemented in CMAPSS in scaled form, by appropriately scaling the state measurements as part of the control algorithm. The computed control rate vector $\dot{v} = v_r$ must be de-scaled and integrated prior to its application to the engine. The switching gains and boundary layer parameters were manually tuned as $\eta_1 = \eta_2 = 0.5$, $\eta = 1$, $\phi_1 = 0.005$, $\phi_2 = 0.01$, and $\phi = 0.05$.

The SMC regulator requires the specification of target steady values for all augmented state components. Moreover, the two steady speeds must be the equilibrium values corresponding to the target steady inputs. These requirements are the same as those applicable to the integral state-feedback regulators considered in Sect. 4.7.1. The designer must have access to the steady map of the nonlinear engine to determine a valid target state. Figures 6.7–6.9 show the results corresponding to a fan speed increment demand starting at Ground Idle conditions. It can be observed that fan and core speeds and control inputs reach their targets with high accuracy and without noticeable chattering. The settling time for fan speed is roughly 2.13 s.

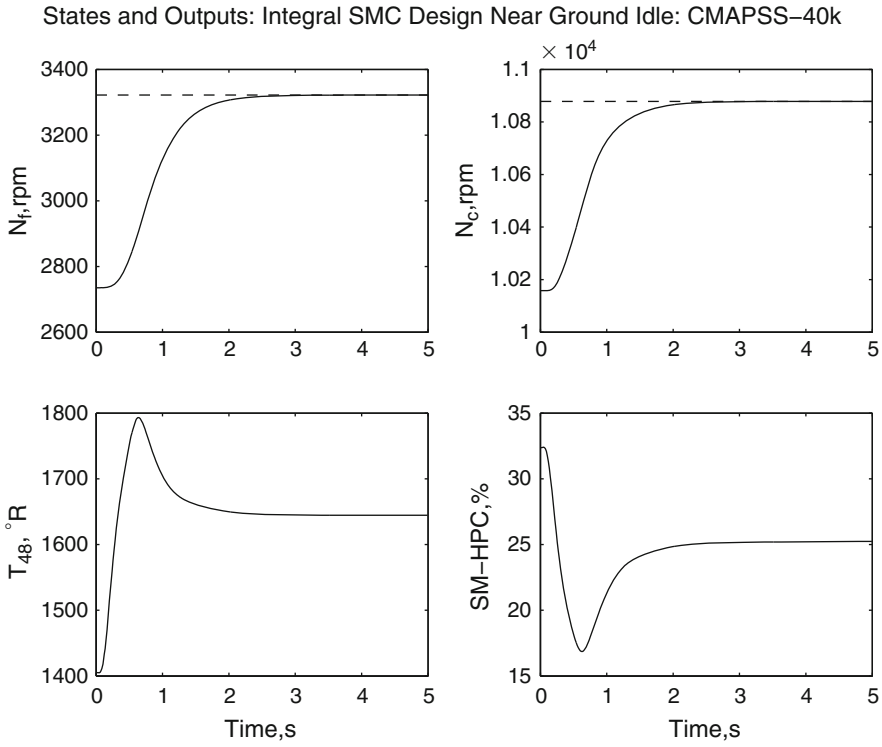

Fig. 6.7 Response of integral multivariable SMC design near Ground Idle: states and outputs

The same design is now used near Max Cruise condition, this time with a fan speed decrease command. Figures 6.10–6.12 show the results. Although the augmented state still reaches its setpoint with high accuracy, the transient response is significantly slower. The robustness of the SMC design ensures convergence to the boundary layer despite the unmatched uncertainty introduced by changes in A and B. However, these changes have an influence in the overall transient behavior, in particular during the sliding mode. Adaptive SMC techniques exist that can be used to achieve more consistent transient response characteristics. The interested reader is referred to the work of Kuo and co-workers [57] for a simple PID-like definition of sliding surface with adaptive gains.

6.3 SMC Output Setpoint Following

Recalling the state feedback designs of Sect. 4.7.1, a desired output setpoint can be achieved in nonlinear engine simulation by appropriately shifting the measured plant state. This requires knowledge of the steady map of the nonlinear plant, so that

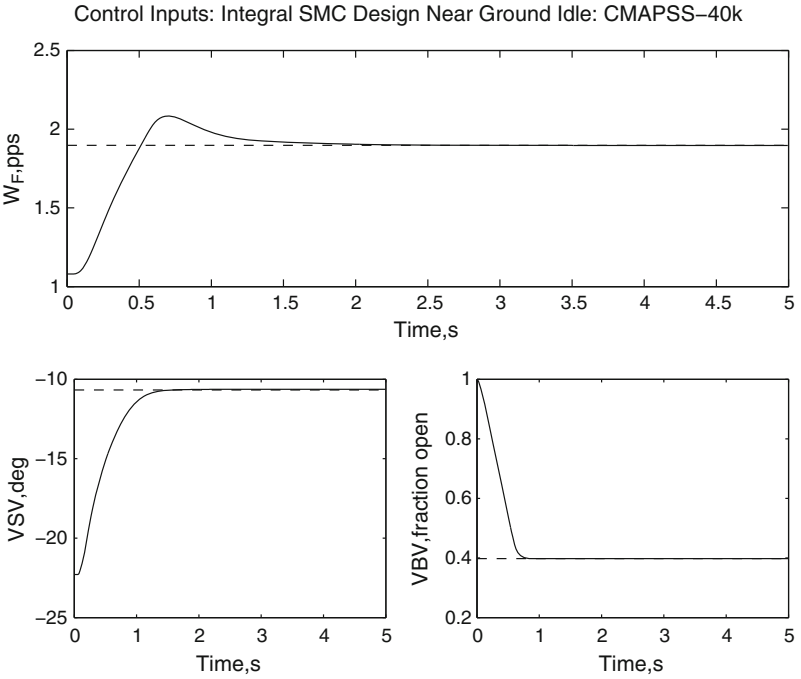

Fig. 6.8 Response of integral multivariable SMC design near Ground Idle: control inputs

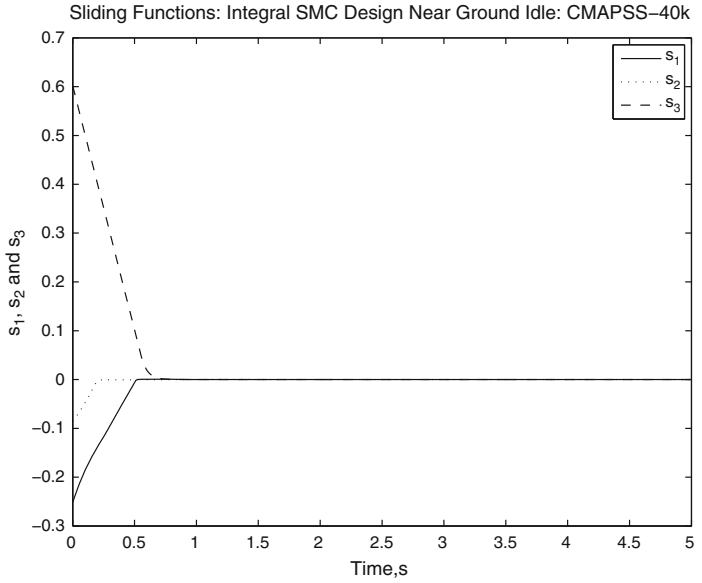

Fig. 6.9 Response of integral multivariable SMC design near Ground Idle: sliding variables

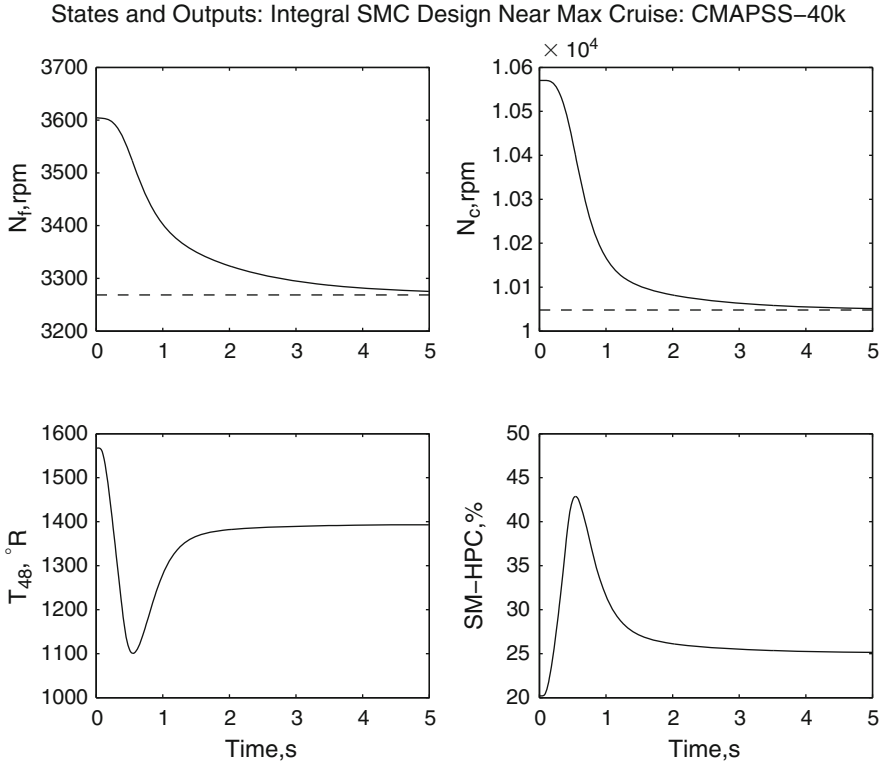

Fig. 6.10 Response of integral multivariable SMC design near Max Cruise: states and outputs

the correct reference states can be determined. Similar state shifting can be used in conjunction with SMC to drive the states to desired values.

Alternatively, the generation of reference states can be brought online, as part of the overall control law. When the plant model is accurately known, reference states can be generated by a simulation of the plant model under some stabilizing control. The simulated control is designed so that model outputs follow a set of reference inputs. This simulation runs in parallel with the control system, feeding the required reference states to the SMC controller. By doing this, only output references are specified, and previously calculated state references become unnecessary. When the plant model is linear, the "virtual control" used to stabilize the model may be as simple as a state-feedback law with reference prefilter.

Assume that the nominal plant model is given by

$$\dot{x} = Ax + Bu, \tag{6.30}$$

$$y = Cx + Du, \tag{6.31}$$

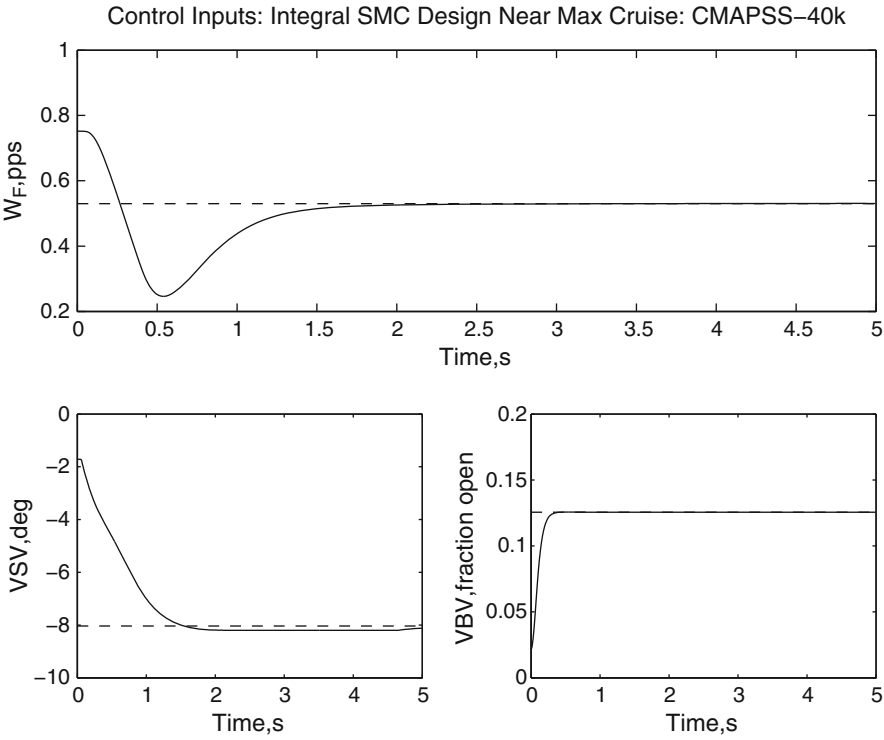

Fig. 6.11 Response of integral multivariable SMC design near Max Cruise: control inputs

where u and y have dimensions m-by-1. The objective is to specify an SMC control law so that the closed-loop system is stable and $y \to r$ as $t \to \infty$, where r is a vector of time-varying references to be tracked. Note that uncertain terms have been omitted, but the same observations made for the regulator design apply: matched uncertainties will not affect sliding mode dynamics, and sufficiently high values of the switching gain guarantee the existence of a sliding mode when general uncertainties are present. Assume, as before, that B is full-rank and (A, B) is controllable.

The regulator design presented here introduces a *reference model*, much like the one used in conjunction with the model reference adaptive control of Sect. 5.4.1. Here, the reference model is introduced to feed the SMC tracker a set of suitable *reference states*. The reference model has the same structure as the nominal plant under the action of a state feedback control. The state feedback control applied to the reference model is designed so that its outputs track the references, at least in the steady-state. The SMC is then formulated so that the plant state converges to the model state in a robust fashion. The nominal plant outputs will then track the references, matching their steady values.

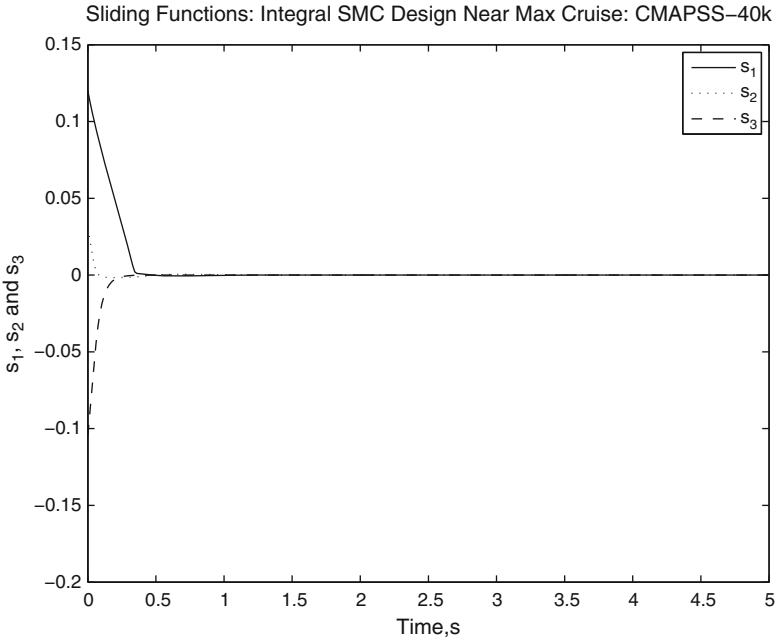

Fig. 6.12 Response of integral multivariable SMC design near Max Cruise: sliding variables

Let the reference model be given by

$$\dot{x}_m = Ax_m + Bu_m, \tag{6.32}$$

$$u_m = -Kx_m + Pr, \tag{6.33}$$

where K is a feedback gain chosen to stabilize $(A - BK)$ and P is a static reference prefilter designed so that $y_m = Cx_m + Du_m = r$ at steady state. At steady state, one has $\dot{x}_m = 0$, so $x_m = -(A - BK)^{-1}BPr$. The steady model output is then $y_m = [D - (C - DK)(A - BK)^{-1}B]Pr$. Then P must be chosen as

$$P = \left[D - (C - DK)(A - BK)^{-1}B\right]^{-1}. \tag{6.34}$$

Define the sliding function as $s = Ge$, where $e = x - x_m$. As done for the regulator design, the control input is obtained from the requirement that $\dot{s} = -\Xi \overrightarrow{\text{sign}}(s)$. The control law is then:

$$u = -(GB)^{-1}\left[GAe - (GB)u_m + \Xi \overrightarrow{\text{sign}}(s)\right]. \tag{6.35}$$

The reader may readily verify that the closed-loop dynamics under this control law reduce to

$$\dot{e} = A_{\text{eq}} e - B(GB)^{-1}\, \Xi\, \overrightarrow{\text{sign}}\,(s), \tag{6.36}$$

$$\dot{s} = -\Xi\, \overrightarrow{\text{sign}}\,(s). \tag{6.37}$$

Clearly, s reaches zero in finite time and e converges to zero asymptotically if A_{eq} is designed as done for the regulator design.

6.3.1 Example: Linearized CMAPSS-40k Model

Consider the CMAPSS-40k linearized model matrices near Ground Idle and define the outputs to be the incremental fan speed, core speed, and HPC stall margin. We conduct an integral design using an augmented plant model as in the regulator example. Suppose K is chosen via an LQR design on the augmented model matrices, as done in the example of Sect. 4.8.3. The scaling used in this example is the same as the one used in the SMC regulator example. The value of K from (4.32) is converted to scaled coordinates according to $K_s = T_u K T_x^{-1}$ to give

$$K_s = \begin{bmatrix} -1.0800 & -0.4200 & 40.2654 & 7.8540 & -0.0517 \\ 0.0185 & -0.0560 & 0.0196 & 4.2430 & -0.0003 \\ 0.5000 & 0.3000 & -1.2915 & -2.9400 & 4.0187 \end{bmatrix}.$$

The static prefilter gain is then computed from (6.34) as

$$P_s = \begin{bmatrix} 0.0146 & 0.0006 & -0.1632 \\ -0.0015 & 0.0012 & 0.0008 \\ -0.1280 & 0.0240 & -5.6871 \end{bmatrix}.$$

The reference model and control law computations are implemented in scaled coordinates, as shown in the Simulink diagram of Fig. 6.13. Next, we illustrate Utkin and Young's LQ method for the selection of sliding coefficients. Taking Q to be a diagonal matrix with entries $1, 1, 10, 10, 10$, the following matrix of sliding coefficients is obtained:

$$G_s = \begin{bmatrix} 0.1148 & 0.1263 & 1.0000 & 0 & 0 \\ -0.0484 & 0.1356 & 0 & 1.0000 & 0 \\ -0.0005 & -0.0005 & 0 & 0 & 1.0000 \end{bmatrix},$$

which places the poles of A_{eq} at -4.46 and -5.55. The design is completed by selecting the switching gains η_i (diagonal entries of Ξ) and values for the boundary layer parameters ϕ_i. Unlike the regulator example, s is computed on the basis of the

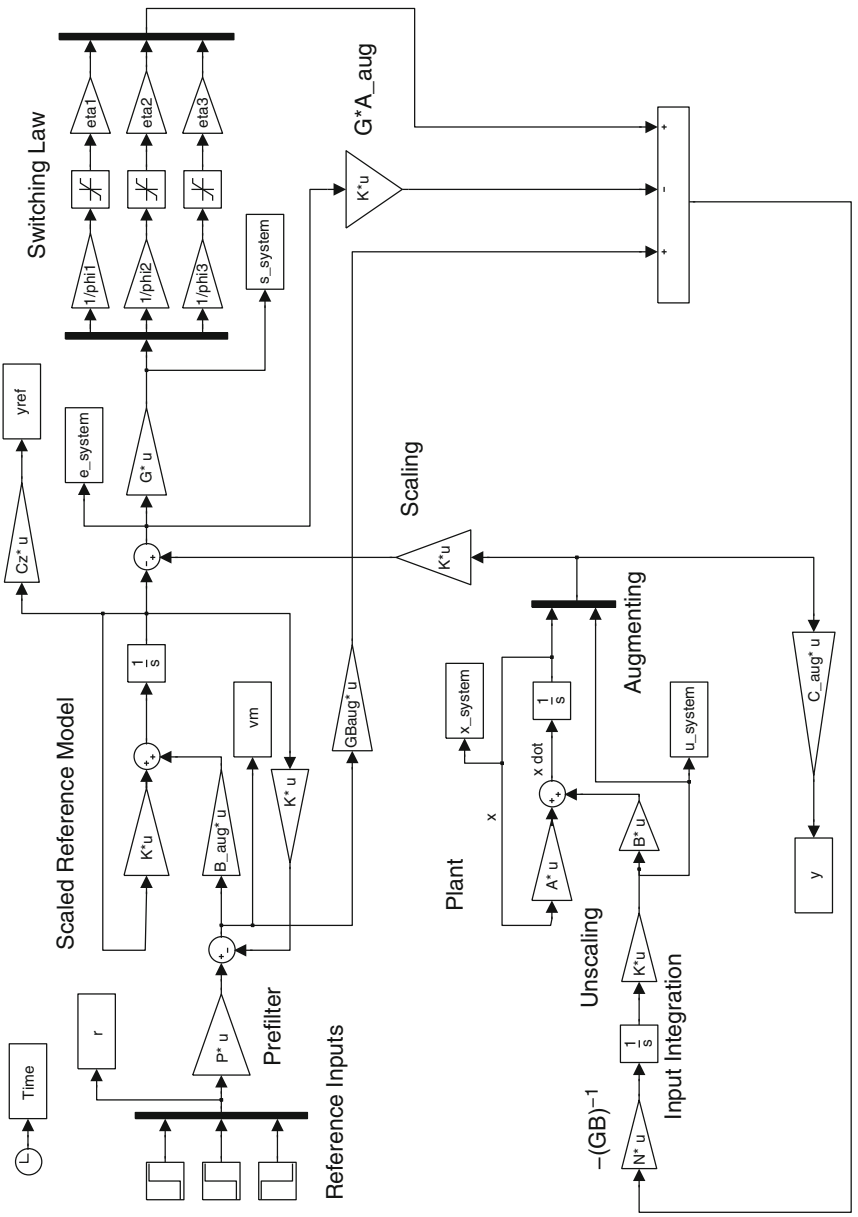

Fig. 6.13 Simulink diagram of SMC output setpoint tracking design

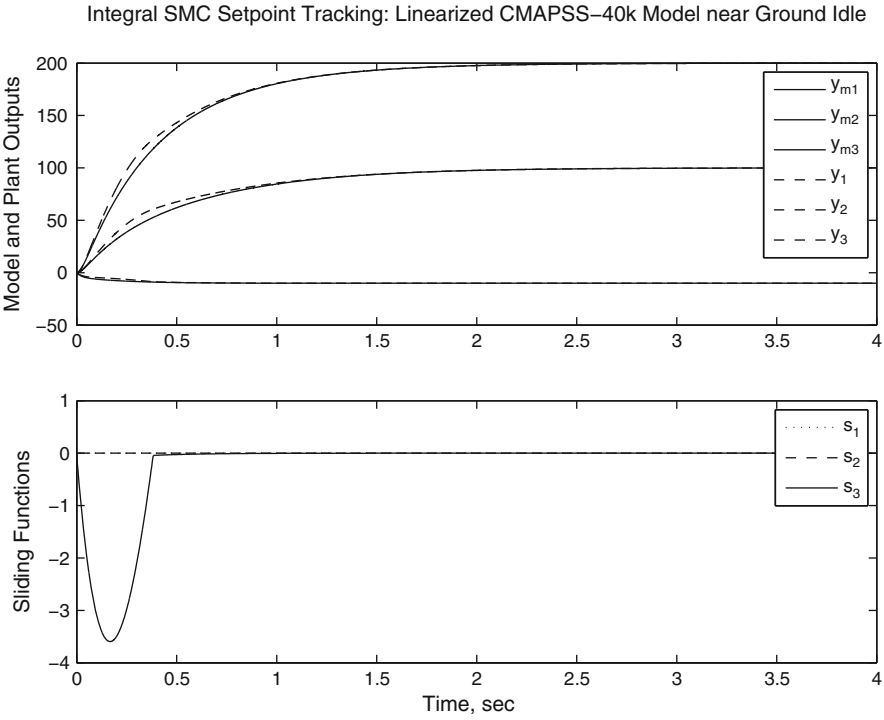

Fig. 6.14 Response of integral multivariable SMC output tracking design near Max Cruise. The simulation uses the linearized CMAPSS model

error between the state and its reference value. This requires that η_i be re-tuned to obtain an adequate reaching time. In this example, $\eta_1 = \eta_2 = 500$, $\eta_3 = 50$ are chosen, along with $\phi_i = 0.1$ for $i = 1, 2, 3$. Figure 6.14 shows plant outputs closely following their setpoint references. Convergence of the sliding variables to zero is also shown.

As presented, the reference model approach will give good results for small excursions of the states from their points of linearization. For larger transfers, however, significant offsets will be observed. Sliding mode robustness is rendered useless, mainly due to the attempt to capture the engine's steady map in formula (6.34). Although the SMC system will track the reference state in a robust fashion, tolerating changes in A and B, references states are still generated according to the linearized model. It is certainly possible to incorporate the LPV techniques of Sect. 5.3 to correct the reference model in real-time. The accuracy of reference state generation can be significantly improved.

Another limitation of this approach is that some combinations of steady reference inputs may require corresponding steady control inputs falling outside actuator saturation limits. Finding a restricted range for the reference inputs requires knowledge of the nonlinear engine's steady map, defeating the purpose of introducing a reference model.

6.4 Simplified SISO Integral SMC Design

We conclude the chapter by presenting a simplified integral control version of SMC that can be applied to linear plants with one input and one output. This is directly applicable to fan speed control using W_F as control input. This SMC formulation is also the basis for the advanced limit protection design of Chap. 8.

Assume that the plant is described in state-space form by:

$$\dot{x} = Ax + Bu, \tag{6.38}$$

$$\dot{u} = u_r, \tag{6.39}$$

$$y_{\text{reg}} = Cx, \tag{6.40}$$

where y_{reg} is a system output which is required to achieve a setpoint value r. It is assumed that A is n-by-n and nonsingular and that (A, B) is controllable. Define an auxiliary output for this system as

$$y = Gx + u, \tag{6.41}$$

where G is a vector of sliding coefficients to be designed. Let the sliding function be defined as $s = y - \bar{y}$, where $\bar{y}$ is a constant chosen to achieve $y_{\text{reg}} = r$ at steady state (see (6.45)).

Taking the derivative of s and imposing the reaching condition $\dot{s} = -\eta \operatorname{sign}(s)$ results in

$$\dot{s} = G[A \mid B]x_{\text{a}} + u_r = -\eta \operatorname{sign}(s),$$

where x_{a} is the augmented state defined as $x_{\text{a}}^{\text{T}} = [x \mid u]$. The expression for the control rate is then

$$u_r = -G[A \mid B]x_{\text{a}} - \eta \operatorname{sign}(s). \tag{6.42}$$

The reaching condition $\dot{s} = -\eta \operatorname{sign}(s)$ can be integrated to give a useful formula for the time required for s to reach the zero value. Denoting the reaching time by t_{r}, we have

$$s(t_{\text{r}}) = 0 = s(0) - \eta \int_0^{t_{\text{r}}} \operatorname{sign}(s)\mathrm{d}t.$$

Ideally, the sign of s must be constant until the sliding surface is reached. Thus, $\operatorname{sign}(s) = \operatorname{sign}(s(0))$. Integrating and solving for t_{r} gives:

$$t_{\text{r}} = \frac{|s(0)|}{\eta}. \tag{6.43}$$

Consider now the dynamics of the system in the sliding mode. From the definition of s, $y = \bar{y}$ must hold. Thus, the augmented control state must be related to the plant state by $u = \bar{y} - Gx$. Substituting this into (6.38) yields the following expression:

$$\dot{x} = (A - BG)x + B\bar{y}. \tag{6.44}$$

The above n-th order relationship already represents dynamics in the sliding mode, which are of reduced order relative to the augmented plant. Note that no transformation to the regular form is required, since direct pole placement on $A - BG$ can be carried out. If $A - BG$ is stable, a steady state will be reached where $\dot{x} = 0$ and $x = \bar{x}$ is the steady state calculated as

$$\bar{x} = -(A - BG)^{-1} B \bar{y}.$$

The corresponding steady value for the regulated output is then $C\bar{x}$. Equating this to r gives the required value for $\bar{y}$:

$$\bar{y} = -\frac{r}{C(A - BG)^{-1} B}. \tag{6.45}$$

Formula (6.45) was derived on the assumption that the plant is truly linear. The same shortcomings observed with formula (6.34) apply: the nonlinear engine's steady map cannot be captured using a single relationship derived from a linearized model. Our approach to obtain accurate setpoint attainment has been to use reference steady values for the state, derived from the nonlinear steady map as part of various control formulations.

The SMC approach of this section can be conveniently adjusted to use such predetermined state references. Express the sliding function as

$$s = Gx + u - \bar{y} = Gx + u - G\bar{x} - \bar{u} = G(x - \bar{x}) + (u - \bar{u}). \tag{6.46}$$

Control law (6.42) can now be implemented with $\bar{x}$ and $\bar{u}$ directly, which can be found from the true nonlinear steady map so that the regulated output reaches the intended setpoint.

6.4.1 Example: CMAPSS-1

The simplified design is now applied to the fan speed regulation problem. The design plant is the transfer function from ΔW_{F} to ΔN_{f} at FC07. Matrices A and B corresponding to this condition are listed in Appendix B. The design comprises three elements:

1. Choosing G to assign eigenvalues to the sliding mode dynamics.
2. Selecting values for the switching gain η and boundary layer thickness ϕ.
3. Finding the values of the reference states $\bar{x}$ and $\bar{u}$ from the actual steady map to match the desired fan speed increment.

For the first task, direct pole placement is straightforward, since $A - BG$ is of second order. The poles can be chosen to reflect desirable time constant and damping values. Taking a damping ratio $\zeta = 0.7$ and a natural frequency $w_{\mathrm{n}} = 11.43$

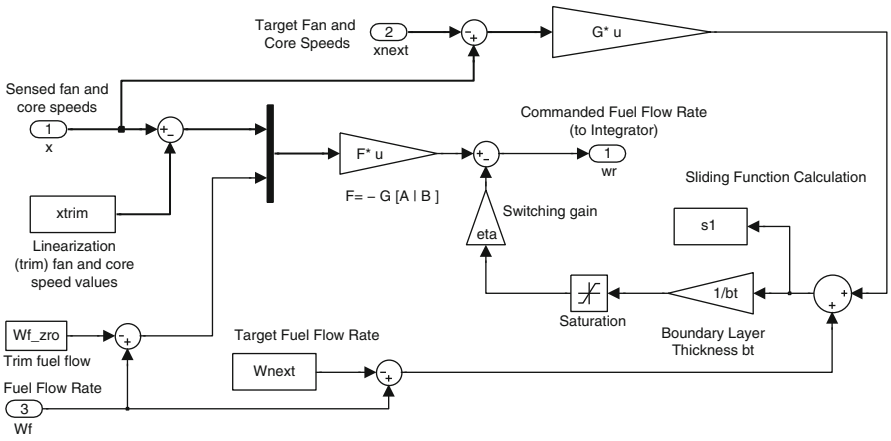

Fig. 6.15 CMAPSS Implementation of integral SMC fan speed controller

rad/s corresponds to pole locations at $-8 \pm 8.1616i$, which have an associated settling time of 0.5 s. The pole placement calculation is done using Matlab's `place` command, yielding the following value for the sliding coefficients:

$$G = [0.1405 \quad -0.0318].$$

Various criteria must be considered when choosing η. The time to reach $s = 0$ is determined by η, and also by G, since $s(0)$ depends on this parameter. Robustness against uncertainties typically require a minimum value for η. Finally, limit protection considerations also apply. In this example, η is tuned on the basis of reaching time alone. The calculation of $s(0)$ requires knowledge of the reference states and a closer look at the SMC implementation in CMAPSS.

Recalling that the linearized states are fan and core speed increments, and that $\bar{x}$ and $\bar{u}$ are likewise incremental, (6.46) can be rewritten in terms of sensed and target variables as follows:

$$s = G\left(\begin{bmatrix} N_\mathrm{f} - N_{\mathrm{f}0} \\ N_\mathrm{c} - N_{\mathrm{c}0} \end{bmatrix} - \begin{bmatrix} \bar{N}_\mathrm{f} - N_{\mathrm{f}0} \\ \bar{N}_\mathrm{c} - N_{\mathrm{c}0} \end{bmatrix}\right) + W_\mathrm{F} - W_{\mathrm{F}0} - (\overline{W}_\mathrm{F} - W_{\mathrm{F}0}).$$

The linearization, or "trim" values $N_{\mathrm{f}0}$, $N_{\mathrm{c}0}$ and $W_{\mathrm{F}0}$ cancel out, reducing the expression for s to

$$s = G\left(\begin{bmatrix} N_\mathrm{f} - \bar{N}_\mathrm{f} \\ N_\mathrm{c} - \bar{N}_\mathrm{c} \end{bmatrix}\right) + W_\mathrm{F} - \overline{W}_\mathrm{F}.$$

Thus, s can be calculated online from the sensed speeds, actual fuel flow, and predetermined values for $\bar{N}_\mathrm{f}$, $\bar{N}_\mathrm{c}$, and $\overline{W}_\mathrm{F}$. The SMC section of the CMAPSS implementation is shown in Fig. 6.15. Returning now to the calculation of η,

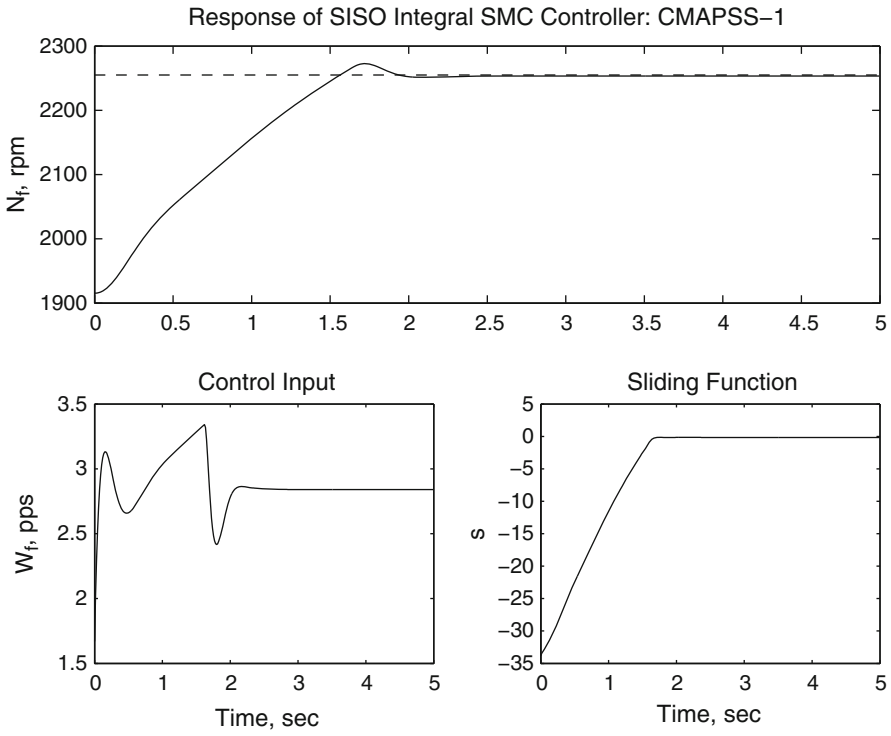

Fig. 6.16 Response of integral SMC fan speed controller: CMAPSS-1

consider a TRA change of 40° from the trim value of 60° defining FC07. The corresponding trim values are $N_{f0} = 1915.3$ rpm, $N_{c0} = 8006.3$ rpm, and $W_{F0} = 1.67$ pps. The steady values after the 40° TRA increment are obtained from CMAPSS as $\bar{N}_f = 2254.8$ rpm, $\bar{N}_c = 8481.2$ rpm, and $\overline{W}_F = 2.85$ pps. These figures can be used to calculate the initial value of the sliding function: $s(0) = -33.78$. Equation (6.43) can be used to tune η to obtain a desired reaching time. For a reaching time less than 1.5 s, we choose $\eta = 25$. The boundary layer thickness ϕ is chosen to obtain a suitable tradeoff between chattering and regulation accuracy. Trial-and-error is the best way to tune this parameter when the nonlinear engine is being simulated. In this example, $\phi = 0.05$ was initially chosen, but resulted in significant control chattering. A value of $\phi = 0.5$ was finally adopted. As Fig. 6.16 shows, this value produces a very small offset between the actual and target steady fan speeds, and no control chattering is observed.

Chapter 7
Engine Limit Management with Linear Regulators

Abstract This chapter describes the min–max logic arrangement used in standard engine control systems to maintain critical variables within the permissible bounds. A thorough analysis of this arrangement is conducted using the concept of positive invariance. The shortcomings of the min–max approach are made evident in simulations. A brief description of an acceleration-limiting approach is also included.

The control techniques presented so far do not address the need to maintain critical engine variables within permissible limits. As described in Sect. 1.3, engine outputs such as stall margins and turbine temperatures must be kept between safe limits at all times. The values chosen for safety limits depend on particular engine characteristics, and typically reflect a tradeoff between high engine performance and engine durability and operational safety. Indeed, large shaft accelerations are desirable for improved maneuverability, since they produce fast thrust responses to pilot commands. Large accelerations lead to transient reductions in stall margin, however, increasing the danger of compressor surge. Large accelerations also correlate with large turbine temperature transients. Since blade wear rate increases with temperature, transient peak temperature suppression should be included as an objective when designing fan speed controllers.

Constraints can be placed on system inputs and outputs. Temperatures, stall margins, and pressure ratios are examples of output variables that may be subject to constraints as part of a particular design process. Fuel flow rates and VSV and VBV openings are inputs whose ranges are constrained. Input constraints can arise from the same considerations as output constraints, that is, to address engine durability or safety, or from the presence of physical limits in actuator systems. An example of a safety-related input constraint is a minimum fuel flow rate requirement, imposed to maintain the combustion chamber away from lean blowout conditions. An example of a physical actuator constraint is given by the fact that a valve may not be more than 100% open or less than fully closed. When such *actuator saturation* effects are ignored during controller design, the implemented closed-loop controller may oscillate or even become unstable. Designs which do not explicitly address

H. Richter, *Advanced Control of Turbofan Engines*, DOI 10.1007/978-1-4614-1171-0_7,

saturation must be validated via simulation studies to evaluate the possibility and effects of saturation modes during real-time operation.

In this chapter, a widely used control architecture intended to achieve limit protection is examined. The *min–max* approach, presented in Sect. 7.1, is used in many GTE control systems. CMAPSS-1 and CMAPSS-40k are distributed with min–max limit management implementations, representative of the actual control systems used in commercial engines such as General Electric's GE90 and Pratt and Whitney's PW2000.

The concept of set invariance is introduced in Sect. 7.2. This concept is instrumental to the analysis of constrained control systems in general. Some basic invariant set constructions are presented to illustrate the application of the theory to GTE control. In Sect. 7.3, min–max architecture based on state feedback controllers is analyzed for its invariance properties. Detailed examples using the CMAPSS-1 model are given throughout the chapter.

7.1 The Min–Max Limit Management Logic

The *min–max* selector arrangement is representative of actual aircraft engine control systems. Similar arrangements have also been used in industrial processes where a number of system outputs must be kept between prescribed limits as a main output is controlled between setpoints. *The premise is that a single control input is available for the tasks of controlling the main output and maintaining a set of outputs between desired limits*. Figure 7.1 shows the min–max arrangement with

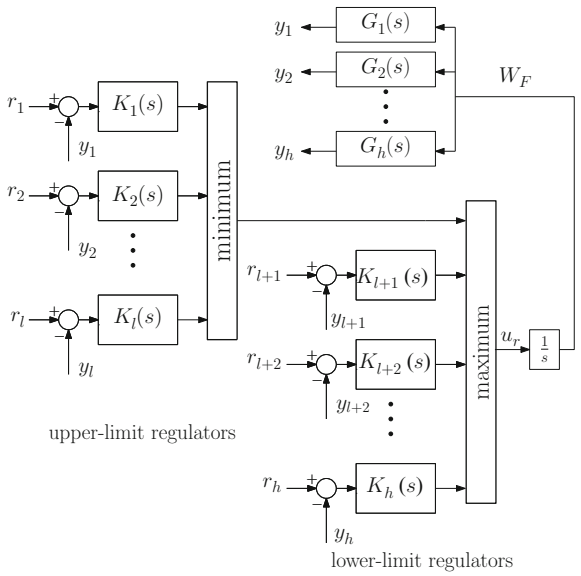

Fig. 7.1 Min–max limit management logic with linear compensators and integral control

linear compensators. A version of the min–max arrangement with state feedback controllers is used for analysis in Sect. 7.3, and a version replacing the linear compensators with sliding mode controllers is presented in Chap. 8.

As shown in Fig. 7.1, the technique uses input integration. The fuel flow control input W_F is applied to the engine, resulting in a set of h outputs characterized by their linearized transfer functions from W_F. Outputs are divided into two groups: a set of *upper-limited outputs*, numbered as y_i, for $i = 1, 2...l$ and a set of *lower-limited outputs*, numbered with $i = l + 1, l + 2, ...h$. The reader should note that transfer functions $G_i(s)$ for $i = 1, 2...h$ share the same poles, since they arise from the same linearized state-space matrix A. The static gains $G_i(0)$ and zero locations will differ, due to differences in matrices C and D defining the outputs. It is also useful to remember that GTE dynamics are such that neither the transfer functions $G_i(s)$ nor A have poles at the origin.

The min–max system is conceived to prevent selected variables from crossing their limits, by activating their regulators as needed. If a variable approaches its limit, its regulator should take over and attempt to drive the output to the prescribed limit without overshooting it. The diagram of Fig. 7.1 shows that upper-limited variables $y_i(s)$ each have a controller $K_i(s)$ with a classical error feedback structure. The setpoint corresponds to the desired output limit, r_i. Each upper-limited controller produces a "candidate" control rate u_{ri}. The minimum rate among upper-limit regulator outputs is preliminarily chosen. A separate set of lower-limit controllers produce their own candidate control rates. The maximum control rate among lower-limit regulator rates and the "winner" from the min stage is selected as the rate to be integrated, producing the fuel flow input command applied to the plant.

The above description cannot be expected to satisfy the average controls engineer. Unfortunately, few works can be found in the open literature that conduct a detailed analysis of this system. Is the system always stable? Could it produce endless switching among regulators? Why are upper-limited variables associated with the min selector and lower-limited variables to the max selector? Under which conditions is limit preservation guaranteed? It turns out that some of these questions have simple and thorough answers, while others are either unresolved or require analysis beyond the scope of this book.

The question of stability of min–max implementations using linear regulators, for instance, has been treated with nonlinear techniques applicable to sector-bounded nonlinearities, such as Popov's criterion and the famous Small Gain Theorem (see Glattfelder and Schaufelberger [58], for instance). These tools provide only sufficient conditions, resulting in very conservative stability assessments. Johansson [33] has analyzed a conceptually similar system using the tools of piecewise-linear systems and multiple Lyapunov functions.

The author of this book proposed that the linear regulators be replaced by sliding mode controllers. Although analysis becomes more complex due to the nonlinearity inherent to sliding mode control, a full global asymptotic stability proof was achieved and simple design guidelines generated, see Richter [59]. This is presented in Chap. 8.

The question of limit protection effectiveness is difficult to establish for the transient regime when dynamic compensators (control transfer functions) are used. When static feedback laws of the form $u = -Kx + Pr$ are used instead, some simplifications occur, facilitating analysis. In the following sections, the dynamic compensation case is studied for its limit protection properties at steady-state only, while the static feedback case is analyzed for both transient and steady-state limit preservation properties.

7.1.1 Default Index Assumptions: Min and Max Operators

Although the generic mathematical operations of taking the minimum or maximum require no explanations, assumptions must be made regarding their behavior in the event of non-unique minimum or maximum values among operands. For the remainder of the book, we adopt the assignment rules made by the min and max operations implemented in many computer languages. Let $\{N(i)\}$ represent an indexed collection of real numbers, for $i=1, 2, ...n$ and let $\bar{N}$ and $\underline{N}$ denote, respectively, the maximum and minimum values found in the collection. If there exists a single index $\bar{i}$ such that $N(\bar{i})=\bar{N}$, $\bar{i}$ is chosen as the outcome of the max selection process. In general, if a set of maximizing indices exists, that is set $\{I(j)\}$ is such that $N(I(j))=\bar{N}$, then the lowest of such indices is chosen: $\bar{i}=\min(I(j))$. For example, take the set $N=\{3,\ 5,\ -2,\ 5,\ 5,\ -2\}$. Here, $\bar{N}=5$ and $I=\{2,\ 4,\ 5\}$, thus $\bar{i}=2$. A similar convention is adopted for the min selector. In the same example, $\underline{i}=3$. This convention will be referred to as *default index assumption* in what follows.

7.1.2 Static Properties of the Min–Max Arrangement with Dynamic Compensators

Three fundamental questions pertaining to the operation of the min–max arrangement are decided next: determining the regulator that becomes active at the initial time ($t = 0$), determining the regulator that remains active during a steady-state regime ($t \to \infty$), and establishing conditions for limits to be preserved at steady-state. As shown later in the chapter, a min-only or a max-only arrangement is sufficient in some cases. Because of this and to facilitate understanding, analysis is separated into min-only, max-only, and min–max cases.

7.1.2.1 Initial Regulator: Min-Only Case

Suppose only the min selector is used to select a control rate u_r among candidate rates u_{ri}, $i \in L$. The min selection law is expressed as

$$u_r = \min_{i \in L} \{u_{ri}\}. \tag{7.1}$$

Suppose $i = i_0$ is the regulator at the initial time. Then the following inequality must hold for all j:

$$u_{ri_0}(0) \leq u_{rj}(0).$$

If there exists $j \neq i_0$ such that $u_{rj} = u_{ri_0}$, then it is necessary that $i_0 < j$ due to the default index assumptions. The control rates are expressed in terms of transfer functions as

$$U_{ri_0}(s) = \frac{K_{i_0}(s)}{1 + G'_{i_0}(s)K_{i_0}(s)} R_{i_0}(s), \tag{7.2}$$

$$U_{rj}(s) = K_j(s)(R_j(s) - G_j(s)U_{ri}(s)), \tag{7.3}$$

where $G'_{i_0}(s) = G_{i_0}(s)/s$. Recalling the *initial value theorem* [26] and noting that $R_{i_0}(s) = r_{i_0}/s$ and $R_j(s) = r_j/s$, we have

$$u_{ri_0}(0) = \lim_{s \to \infty} sU_{ri_0}(s) = \lim_{s \to \infty} \frac{r_i K_{i_0}(s)}{1 + G'_{i_0}(s)K_{i_0}(s)}. \tag{7.4}$$

Note that G'_{i_0} has a pole at $s = 0$ and that it is strictly proper, since G_{i_0} is proper and does not have zeroes at $s = 0$. Therefore, if $K_{i_0}(s)$ is proper, we have $u_{ri_0}(0) = K_{i_0}(\infty)r_{i_0}$. When $K_{i_0}(s)$ is an improper PD transfer function, the derivative term is impulsive when r_{i_0} is constant. The initial regulator observed in simulations or real-time deployments will depend on the approximation involved in the implementation of the derivative term. Therefore, we restrict our formulas for the initial regulator to the case when all controllers are proper.

Now, using the initial value theorem on u_{rj}, we have

$$u_{rj}(0) = \lim_{s \to \infty} sU_{rj}(s) = \lim_{s \to \infty} \left\{r_j K_j(s) - sG_j(s)U_{ri_0}(s)\right\}. \tag{7.5}$$

The reader can verify that if $K_j(s)$ is proper $u_{rj}(0)$ has a well-defined constant value:

$$u_{rj}(0) = K_j(\infty)r_j.$$

Therefore, the initial regulator is calculated as the smallest index i_0 satisfying

$$K_{i_0}(\infty)r_{i_0} \leq K_j(\infty)r_j \tag{7.6}$$

for all $j \in L$.

7.1.2.2 Initial Regulator: Max-Only Case

Now consider that only the max selector is used to select a control rate u_r among candidate rates u_{ri}, $i \in H$. The max selection law is expressed as

$$u_r = \max_{i \in H} \{u_{ri}\}. \tag{7.7}$$

Suppose $i = i_0$ is the regulator at the initial time. The initial values of the rates u_{ri} are the same regardless of the type of selector, max or min. The initial selection is therefore given by the smallest index i_0 satisfying

$$K_{i_0}(\infty) r_{i_0} \geq K_j(\infty) r_j \tag{7.8}$$

for all $j \in H$.

7.1.2.3 Initial Regulator: Min–Max Case

The min–max case has a subtlety that requires careful attention. Referring back to Fig. 7.1, the min–max selection law is given by

$$u_r = \max_{k \in H} \left\{ \min_{j \in L} \{u_{rj}\}, \; u_{rk} \right\}, \tag{7.9}$$

where u_{rj} are the min-selected regulator outputs and u_{rk} are the max-selected regulator outputs. Again, the values of the initial rates are the same regardless of the selection mechanism. Suppose i_0 is the index of the regulator selected at $t = 0$. If $i_0 \in L$, the following inequalities must be true:

$$u_{ri_0} \leq u_{rj} \text{ for all } j \in L, \tag{7.10}$$

$$u_{ri_0} \geq u_{rk} \text{ for all } k \in H. \tag{7.11}$$

If $i_0 \in H$, however, the following inequalities apply:

$$u_{ri_0} > u_{rk} \text{ for all } k \in L, \tag{7.12}$$

$$u_{ri_0} > \min_{j \in L} \{u_{rj}\}. \tag{7.13}$$

Note that strict inequality must be used, since the winner of the min stage is applied to the first port of the max selector. Thus, equality between the winner of the max stage and the winner of the min stage would result in $i_0 \in L$ due to the default index assumptions. Also note that u_{ri_0} is required to be greater than the *minimum* of the rates produced by regulators in L, but not necessarily greater than each. This implies that a two-step iterative process must be followed to determine i_0. Assume first that

$i_0 \in L$ and find the smallest index satisfying inequalities (7.10) and (7.11). If no such index can be found, it must be that $i_0 \in H$. Inequalities (7.12) and (7.13) are then tested using the known value of the winner of the min stage.

The reader should observe that the calculations for the initial regulator in the min, max and min–max cases are not influenced by whether a variable is upper- or lower-limited. Only the type of selector must be taken into account.

7.1.2.4 Steady Regulator: Min-Only Case

Here, we assume that the closed-loop system is asymptotically stable, with a fixed steady-state regulator selection denoted as i^*. Applying the *final value theorem* [26] to the Laplace expression for the control rate of (7.2), we obtain

$$u_{ri*}(\infty) = \lim_{s\to 0} sU_{ri*}(s) = \lim_{s\to 0} \frac{r_{i*}K_{i*}(s)}{1 + G'_{i*}(s)K_{i*}(s)}. \tag{7.14}$$

The reader can verify that $u_{ri*} = 0$ whenever K_{i*} and G_{i*} are proper and do not have zeroes at the origin. Furthermore, the same is true if K_{i*} is a PD transfer function and G_{i*} is proper with no zeroes at the origin. This is expected, since the active loop is of type I, and the steady-state error to constant inputs is zero. Since K_{i*} is driven by the error, its output will be zero at steady-state.

Applying the final value theorem to $U_{rj}(s)$ from (7.3), we obtain

$$u_{rj}(\infty) = \lim_{s\to 0} sK_j(s)\left[\frac{r_j}{s} - G_j(s)U_{ri*}(s)\right].$$

If $G_j(s)$ and $K_j(s)$ are proper with no poles at the origin, this reduces to

$$u_{rj}(\infty) = K_j(0)\left[r_j - r_i\frac{G_j(0)}{G_{i*}(0)}\right]. \tag{7.15}$$

Therefore, i^* is the smallest index satisfying

$$K_j(0)\frac{r_{i*}G_j(0)}{G_{i*}(0)} \le K_j(0)r_j, \tag{7.16}$$

for all $j \in L$. Note that

$$\frac{r_{i*}G_j(0)}{G_{i*}(0)} = \bar{y}_{j/i*},$$

where $\bar{y}_{j/i*}$ is the steady value attained by y_j when i^* is the steady regulator. If $K_j(0) > 0$, inequality (7.17) reduces to $\bar{y}_{j/i*} \le r_j$. This is interpreted verbally as follows: "i^* is the regulator for which outputs do not exceed their setpoints at steady-state". If $K_j(0) < 0$, the inequality becomes $\bar{y}_{j/i*} \ge r_j$. This corresponds to: "i^* is the regulator for which outputs do not become smaller than their setpoints at steady-state".

7.1.2.5 Steady Regulator: Max-Only Case

The final regulator index for this case is the smallest index i^* satisfying

$$K_j(0)\frac{r_{i^*}G_j(0)}{G_{i^*}(0)} \geq K_j(0)r_j \tag{7.17}$$

for all $j \in H$. If $K_j(0) > 0$, inequality (7.17) reduces to $\bar{y}_{j/i^*} \geq r_j$. This is interpreted verbally as follows: "i^* is the regulator for which outputs do not become smaller than their setpoints at steady-state". If $K_j(0) < 0$, the inequality becomes $\bar{y}_{j/i^*} \leq r_j$. This corresponds to: "i^* is the regulator for which outputs do not exceed their setpoints at steady-state".

The steady properties of the isolated min and max selectors provide a basic guideline for design. If all outputs are upper-limited and $K_j(0) > 0$ for all j, a max regulator provides steady-state limit protection. If all outputs are lower-limited and $K_j(0) > 0$ for all j, a max selector provides steady-state limit protection. This seems to be the assumption justifying the assignments found in the standard min–max arrangements used in the aerospace industry.

As the reader may appreciate, the more general case where outputs have different signs of $K_j(0)$, some being lower-limited and some being upper-limited are not covered by the min-only or max-only arrangements. The min–max arrangement was introduced as an attempt to cover these cases.

7.1.2.6 Steady Regulator: Min–Max Case

The same logic used for determining the initial regulator is followed, using the appropriate formulas for the steady rates. If $i^* \in L$, the following inequalities hold:

$$0 \leq u_{rj} \text{ for all } j \in L, \tag{7.18}$$

$$0 \geq u_{rk} \text{ for all } k \in H. \tag{7.19}$$

If $i^* \in H$, however, the following inequalities apply:

$$0 > u_{rk} \text{ for all } k \in L, \tag{7.20}$$

$$0 > \min_{j \in L}\left\{u_{rj}\right\}. \tag{7.21}$$

7.1.3 Example: CMAPSS-1

Consider the transfer functions from incremental fuel flow to incremental fan speed (rpm), incremental HPT temperature (°R) and HPC stall margin (%) near FC01:

$$G_1(s) = \frac{\Delta N_f(s)}{\Delta W_F(s)} = \frac{230.9s + 2000}{s^2 + 8.504s + 17.16},$$

$$G_2(s) = \frac{\Delta T_{48}(s)}{\Delta W_F(s)} = \frac{146.2s^2 + 1027s + 1528}{s^2 + 8.504s + 17.16},$$

$$G_3(s) = \frac{\Delta SmHPC(s)}{\Delta W_F(s)} = \frac{-4.405s^2 - 28.81s - 20.49}{s^2 + 8.504s + 17.16}.$$

Suppose that a set of controllers is selected that independently stabilize the above transfer functions under an integral control loop (disregarding transient response qualities), as follows:

$$K_1(s) = 0.21\frac{s + 3.715}{s + 20},$$

$$K_2(s) = 0.1,$$

$$K_3(s) = \frac{-s - 1}{s + 2}.$$

Suppose $r_1 = 10$, $r_2 = 20$, and $r_3 = -10$. In this example, $G_1(\infty) = 0$, $G_2(\infty) = 146.2$ and $G_3(\infty) = -4.405$. Also, $K_1(\infty) = 0.21$, $K_2(\infty) = 0.1$, and $K_3(\infty) = -1$. Similarly, $G_1(0) = 116.55$, $G_2(0) = 89.044$, $G_3(0) = -1.194$ and $K_1(0) = 0.039$, $K_2(0) = 0.1$ and $K_3(0) = -0.5$.

If only a min selector is used, (7.6) can be used to determine that the initial regulator is $i_0 = 2$ and the final regulator is $i^* = 1$. If only a max selector is used, $i_0 = i^* = 3$. Suppose now that u_{r1} and u_{r2} are associated with a min selector, while the min-preselection and u_{r3} are associated with a max selector. It can be readily verified that $i_0 = i^* = 3$ as well.

The concept of *positive invariance* is introduced next as an essential tool to study limit regulator behavior.

7.2 Basic Set Invariance Concepts

Set invariance theory is a unifying umbrella under which many techniques for constraint handling have been developed. Although a detailed account of set invariance and derived techniques is out of the scope of this book, some definitions and results will be included. For an excellent introduction to the topic, readers are referred to the survey by Blanchini [60].

Given a dynamical system with state vector x, a *positively invariant* set is a subset $\mathcal{S}_o$ of the state-space formed by all initial conditions resulting in trajectories, which remain in $\mathcal{S}$ for all subsequent times. That is, $\mathcal{S}_o$ is positively invariant if $x(0) \in \mathcal{S}_o$ implies $x(t) \in \mathcal{S}_o$ for all $t > 0$, justifying the *positive* invariance qualification.

A result by Nagumo [61] establishes a necessary and sufficient condition for a set to be positively invariant with respect to a given dynamical system. For the purposes

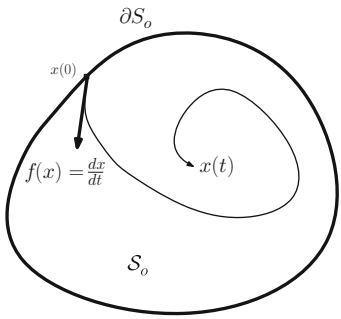

Fig. 7.2 Illustration of Nagumo's invariance condition. $\mathcal{S}_o$ is the invariant set and $\partial\mathcal{S}_o$ its boundary

of this book, invariant sets of simple descriptions are considered, such as ellipsoids, half-spaces, and real intervals. Roughly, Nagumo's condition is equivalent to the requirement that the vector $\dot{x}$ evaluated along the boundary of $\mathcal{S}_o$ be directed toward its interior, as illustrated in Fig. 7.2. Nagumo's condition is rather intuitive, and becomes more so when x is a scalar, so that the invariant set is an interval, as shown next.

7.2.1 Positive Invariance of an Interval

An interval $(-\infty, b]$ is invariant for a generic real variable $z(t)$ if $\dot{z}(t) \leq 0$ at $z = b$. Similarly, an interval $[a, \infty)$ is invariant if $\dot{z}(t) \geq 0$ at $z = a$. When an interval is invariant and $z(t_1)$ belongs to the interval for some $t_1 > 0$, then $z(t)$ will remain in the interval for $t \geq t_1$. The concept is readily applied to limit protection: the interval $(-\infty, 0]$ must be invariant for the error $e = r - y$ when y is a lower-limited output. Conversely, $[0, \infty)$ must be invariant for the error of upper-limited variables.

7.2.2 Ellipsoidal Invariant Sets for Linear Systems

An n-dimensional ellipsoid is described by the inequality

$$x^{\mathrm{T}} P x \leq 1,$$

where P is a symmetric, positive-definite matrix. Consider first an autonomous linear state-space system described $\dot{x} = Ax$ and an ellipsoidal set $\mathcal{S}_o$ described as

$$\mathcal{S}_o = \mathcal{E} = \{x \ : \ x^{\mathrm{T}} P x \leq 1\}.$$

For $\mathcal{E}$ to be PI, Nagumo's theorem requires that $\dot{x} = f(x) = Ax$ be directed toward the interior of $\mathcal{E}$, when x is taken at its boundary, defined by $h(x) = x^{\mathrm{T}} P x = 1$.

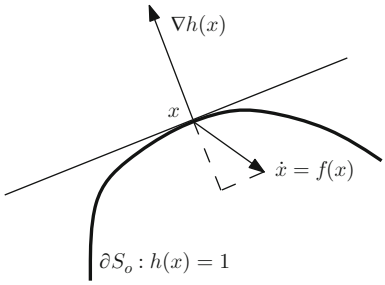

Fig. 7.3 Invariance enforcement by negative gradient projection

This can be ensured by the condition that the projection of $f(x)$ onto a vector normal to the boundary be negative, as shown in Fig. 7.3. The invariance condition is expressed as

$$\nabla h(x).f(x) < 0.$$

For an ellipsoidal boundary we have $\nabla h(x) = 2x^{\mathrm{T}}P$, thus the condition becomes $2x^{\mathrm{T}}PAx < 0$. The reader may recall that any square matrix can be decomposed into a symmetric component and an antisymmetric component. Decompose PA as follows:

$$PA = \frac{1}{2}(PA + A^{\mathrm{T}}P) + \frac{1}{2}(PA - A^{\mathrm{T}}P).$$

The first term is symmetric, while the second is antisymmetric. Thus,

$$2x^{\mathrm{T}}PAx = x^{\mathrm{T}}(PA + A^{\mathrm{T}}P)x + x^{\mathrm{T}}(PA - A^{\mathrm{T}}P)x.$$

Recalling that the cross-terms of a quadratic form $xY^{\mathrm{T}}x$ cancel out when Y is antisymmetric, the invariance condition reduces to the following *Lyapunov inequality*

$$PA + A^{\mathrm{T}}P < 0. \tag{7.22}$$

The above condition can be applied to evaluate the invariance of an ellipsoid relative to the closed loop system $\dot{x} = (A - BK)x$ resulting from using the state feedback control law $u = -Kx$:

$$P(A - BK) + (A - BK)^{\mathrm{T}}P < 0. \tag{7.23}$$

Note that inequality (7.23) is satisfied *for some* P when K stabilizes $A - BK$, but that a predetermined P need not satisfy the inequality for all stabilizing K. A family of invariant ellipsoids associated with a given stabilizing K can be found by turning inequality (7.23) into equality, using an arbitrary positive-definite, symmetric matrix Q:

$$P(A - BK) + (A - BK)^{\mathrm{T}}P = -Q. \tag{7.24}$$

All ellipsoids of the form $x^{\mathrm{T}}Px = a$, with $a \geq 0$ are then PI relative to the closed-loop dynamics.

7.2.3 Invariance of a Half-Space

State and output constraints can be formulated as a set of linear inequalities in the state variables: $\mathcal{G} = \cap \mathcal{G}_i$ for $i = 1, 2..m$, where $\mathcal{G}_i = \{x : G_i x \leq 1\}$. Application of Nagumo's result to a system $\dot{x} = f(x)$ and an individual linear constraint $G_i x \leq 1$ results in the following condition for invariance:

$$G_i f(x) \leq 0 \text{ along } G_i x = 1. \tag{7.25}$$

In the case of a linear state-space system (A, B) under state feedback $u = -Kx$, this reduces to

$$G_i(A - BK)x \leq 0 \text{ along } G_i x = 1. \tag{7.26}$$

In general, condition (7.26) cannot be satisfied for all points belonging to the boundary defined by $G_i x = 1$. Geometrically, if G_i is not parallel to $G_i(A - BK)$, the boundary will be divided into three subsets: a subset where $G_i(A - BK)x = 0$, a subset where $G_i(A - BK) > 0$ and a subset where the condition is satisfied. When several constraints exist, methodologies have been developed to ensure that points belonging to the set, where $G_i(A - BK)x > 0$ do not satisfy the other constraints. *Polyhedral invariant set theory* provides means to construct invariant sets using linear segments. Given a constraint set defined by design requirements, polyhedral invariant sets can be constructed with little conservativeness. An invariant set construction is conservative if it excludes points of the constraint set which actually lead to permissible trajectories. Polyhedral sets require a large number of vertices to achieve low conservativeness. The interested reader is referred to Blanchini's survey for an introduction [60].

7.2.4 Ellipsoidal Operating Sets

Invariant sets are used to determine the range of allowable initial conditions so that the ensuing trajectories do not violate the constraints. When error dynamics are used as a basis, an invariant set description can be used to determine the allowable range of the setpoints so that no limit violations will occur in the transient or steady regimes. Such *operating sets* are thus invariant subsets of the constraint set. A simple approach to finding an operating set is to find the largest invariant ellipsoid contained in the constraint set. For this, ellipsoids which are tangent to individual constraints are found. The smallest of such ellipsoids will be tangent to one of the constraints and tangent or interior to the remaining ones, in addition to being invariant.

Given a linear constraint $G_i x \leq 1$, and an invariant ellipsoid family defined by matrix P, the ellipsoid which is tangent to the boundary $G_i x = 1$ can be found by solving the following constrained optimization problem:

$$\max V = x^{\mathrm{T}} P x \text{ subject to}$$
$$G_i x = 1.$$

This is readily solved using Lagrange multipliers, yielding the following formula for the maximum value of $x^{\mathrm{T}} P x$:

$$V_i = \frac{1}{G_i P^{-1} G_i^{\mathrm{T}}}.$$

Thus, the largest invariant ellipsoid tangent to the i-th constraint is given by

$$\mathcal{E}_i = \{x \ : \ x^{\mathrm{T}} P x \leq V_i\},$$

where P satisfies inequality (7.24). The overall invariant ellipsoid can be found by taking the smallest V_i and matrix P. As an introductory example, consider the double-integrator plant $G(s) = 1/s^2$ with state-space matrices A and B as follows:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Suppose the constraints are given by $|x_1| \leq 1$ and $|x_2| \leq 1.5$. The plant is stabilized with a state-feedback law of the form $u = -Kx$, where K is chosen so that $A_{\mathrm{c}} = A - BK$ has eigenvalues with negative real parts. Suppose a K is designed using an LQR approach, with $Q = I$ and $R = 1$. This yields $K = [1 \ \sqrt{3}]$, which places the poles of A_{c} at $-\sqrt{3}/2 \pm i/2$. The following P satisfies Lyapunov inequality (7.23):

$$P = \begin{bmatrix} 1.5847 & 0.5489 \\ 0.5489 & 0.6339 \end{bmatrix}.$$

The constraints are expressed as $G_1 x = [1 \ 0]x \leq 1$, $G_2 x = [-1 \ 0]x \leq 1$, $G_3 x = [0 \ 1/1.5]x \leq 1$ and $G_4 = [0 \ -1/1.5]x \leq 1$. The values of V_i are calculated as $V_1 = V_2 = 1.109$ and $V_3 = V_4 = 1$. Thus, the ellipsoid described by $x^{\mathrm{T}} P x \leq 1$ is tangent the third and fourth constraints and interior to the first and second constraints, in addition to being positively invariant. Figure 7.4 shows the ellipsoidal boundary in relation to the constraints. A few trajectories have also been plotted.

The dotted line represents a trajectory which satisfies the constraints but whose initial condition is not captured by the ellipsoid. In contrast, the dashed line is a trajectory whose initial point belongs to the constraint set but which produces constraint violation, and is correctly excluded by the ellipsoidal set. An example of the ellipsoidal construction as applied to the GTE problem is given in Sect. 7.5.2.

A less conservative invariant set construction which retains the simplicity of ellipsoids is given by the semiellipsoidal approach of O'Dell [62]. In this approach, an invariant ellipsoid which may exceed the constraints is sought under the condition that its intersection with the constraints retains the positive invariance property. The methodology regards K and P as free parameters, optimized to yield an ellipsoid of maximum volume. The resulting semiellipsoidal set is an approximation to the maximal recoverable set under state feedback, i.e., the largest set of initial conditions yielding trajectories which proceed to the origin without constraint violation. The maximizing argument K does not offer any performance guarantees, however.

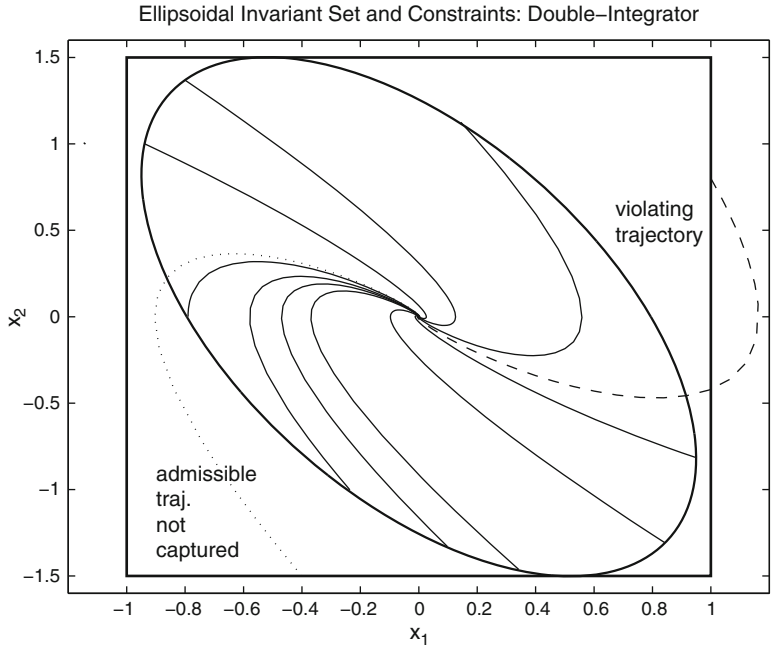

Fig. 7.4 Ellipsoidal invariant set and constraints: double-integrator system

The operating set approach amounts only to "passive" constraint validation. If invariance cannot be established without unreasonable conservativeness, a different control law must be found or the constraints relaxed. In contrast, approaches such as the min–max arrangement constitute *active* means to enforce the constraints as the system operates. The use of a *reference governor* allows to exploit set invariance properties in an active way. The key idea is to filter reference inputs to force the state to remain in an invariant set, see, for instance, Bemporad [63].

7.3 Min–Max Limit Management with Integral State Feedback Controllers

Analysis of the min–max arrangement is somewhat simplified when state feedback controllers are used instead of dynamic compensators. Consider a linear plant with input integration, described by the following augmented state-space model:

$$\dot{x}_{\mathrm{a}} = A_{\mathrm{a}} x_{\mathrm{a}} + B_{\mathrm{a}} u_r, \tag{7.27}$$

$$y_i = C_{ai} x_{\mathrm{a}}, \tag{7.28}$$

where $x_{\mathrm{a}}^{\mathrm{T}} = [x^{\mathrm{T}}\ u]$ is the augmented state vector and A_{a} and B_{a} are defined as in (4.31). Outputs $y_i = C_i x + Du$ are defined by their augmented matrices $C_{ai} = [C_i\ \ D_i]$. Consider the static state feedback law

$$u_{ri} = -K_i x_{\mathrm{a}} + P_i r_i, \tag{7.29}$$

where K_i is such that $A_{\mathrm{a}} - B_{\mathrm{a}} K_i$ is stable. Although K_i could be chosen to be constant across regulators to guarantee stability and achieve desirable transient properties for the main controlled output, the min–max arrangement would become useless. To see this, note that $u_{ri} - u_{rj}$ is constant for a given pair i, j when a constant K is used. Thus, the regulator selected by min and max at $t = 0$ becomes permanent and no switching occurs. Assume, then, that the min, max, and min–max arrangements involve distinct feedback gains.

The value of the *prefilter gain* P_i is chosen to ensure that $y_i = r_i$ in the steady-state, and must therefore vary with i (see (7.35)).

7.3.1 Closed-Loop Behavior Under a Fixed Regulator

Suppose the i-th regulator is active at all times. The closed-loop system becomes

$$\dot{x}_{\mathrm{a}} = A_{ci} x_{\mathrm{a}} + B_{\mathrm{a}} P_i r_i, \tag{7.30}$$

where $A_{ci} = A_{\mathrm{a}} - B_{\mathrm{a}} K_i$. Since K_i is designed so that A_{ci} has eigenvalues with negative real parts, the augmented state reaches a steady-state value of $\bar{x}_{ai}$ satisfying

$$0 = A_{\mathrm{a}} \bar{x}_{ai} + B_{\mathrm{a}} \bar{u}_{ri}, \tag{7.31}$$

$$0 = \bar{u}_{ri} = -K_i \bar{x}_{ai} + P_i r_i, \tag{7.32}$$

where (7.32) is obtained from the requirement that the last component of the augmented state derivative be zero at steady-state, that is, $\dot{u} = \bar{u}_{ri} = 0$. Equations (7.31) and (7.32) imply that $A_{\mathrm{a}} \bar{x}_{ai} = 0$ and that P_i must be related to $\bar{x}_{ai}$ as follows:

$$P_i = K_i \frac{\bar{x}_{ai}}{r_i}. \tag{7.33}$$

The value of $\bar{x}_{ai}$ is determined by the requirement that $y_i = r_i$ at steady-state. Separating the first n components of (7.31), we have:

$$0 = A\bar{x} + B\bar{u},$$

where $\bar{x}_{ai}^{\mathrm{T}} = [\bar{x}^{\mathrm{T}}\ \ \bar{u}]$. Since A has no poles at the origin, $\bar{x} = -A^{-1} B \bar{u}$. The steady output is then

$$\bar{y}_i = C_i \bar{x} + D_i \bar{u} = (-C_i A^{-1} B + D_i)\bar{u} = G_i(0) r_i.$$

For $\bar{y}_i = r_i$, we must have $\bar{u} = r_i / G_i(0)$ and $\bar{x} = -A^{-1} B r_i / G_i(0)$, thus

$$\bar{x}_{ai} = \frac{r_i}{G_i(0)} \begin{bmatrix} -A^{-1} & B \\ & 1 \end{bmatrix}, \tag{7.34}$$

$$P_i = \frac{K_i}{G_i(0)} \begin{bmatrix} -A^{-1} & B \\ & 1 \end{bmatrix}. \tag{7.35}$$

7.3.2 Closed-Loop Behavior Relative to a Fixed Index

It is convenient to shift the augmented state variable by the steady-state value corresponding to an arbitrary index i. That is, define

$$\tilde{x}_{ai} = x_{ai} - \bar{x}_{ai}. \tag{7.36}$$

Relevant quantities are now expressed in terms of $\tilde{x}_{ai}$. The i-th and j-th outputs become

$$y_i = C_{ai} x_{\mathrm{a}} = C_{ai}(\tilde{x}_{ai} + \bar{x}_{ai}) = r_i + C_{ai} \tilde{x}_{ai}, \tag{7.37}$$

$$y_j = C_{aj} x_{\mathrm{a}} = C_{aj}(\tilde{x}_{ai} + \bar{x}_{ai}) = \bar{y}_{j/i} + C_{aj} \tilde{x}_{ai}, \tag{7.38}$$

where $\bar{y}_{j/i}$ is the steady value of y_j when i is active at steady-state. The tracking errors are expressed in terms of the new state variable as

$$e_i = r_i - y_i = -C_{ai} \tilde{x}_{ai}, \tag{7.39}$$

$$e_j = r_j - y_j = \bar{e}_{j/i} - C_{aj} \tilde{x}_{ai}, \tag{7.40}$$

where $\bar{e}_{j/i}$ denotes the steady-state error in y_j when i is active. The control rates are expressed as

$$u_{ri} = -K_i(\tilde{x}_{ai} + \bar{x}_{ai}) + P_i r_i = -K_i \tilde{x}_{ai}, \tag{7.41}$$

$$u_{rj} = -K_j(\tilde{x}_{ai} + \bar{x}_{ai}) + P_j r_j, \tag{7.42}$$

where (7.32) has been used.

The reader should observe that (7.37)–(7.42) use an arbitrary index i as a reference, but are valid regardless of the active regulator. The next two expressions for the derivatives of tracking errors e_i and e_j, however, assume that i is the active regulator, and this is reflected in the notations $\dot{e}_{i/i}$ and $\dot{e}_{j/i}$.

$$\dot{e}_{i/i} = -C_{ai} A_{ci} \tilde{x}_{ai}, \tag{7.43}$$

$$\dot{e}_{j/i} = -C_{aj} A_{ci} \tilde{x}_{ai}. \tag{7.44}$$

Equations (7.30) and (7.31), and the definition of $\tilde{x}_{ai}$ from (7.36) have been used in the derivation of (7.43) and (7.44). Finally, the following expression for the difference between control rates will be useful in subsequent developments:

$$u_{ri} - u_{rj} = (K_j - K_i)\tilde{x}_{ai} + K_j\bar{x}_{ai} - P_j r_j. \tag{7.45}$$

7.3.3 *Static Properties of the Min–Max Arrangement with State Feedback*

The static analysis conducted for the min–max arrangement with dynamic compensators is repeated for the state feedback case. The same fundamental issues are discussed: determining the regulator that becomes active at the initial time ($t = 0$), determining the regulator that remains active during a steady-state regime ($t \to \infty$), and establishing conditions for limits to be preserved at steady-state. Unlike min–max systems with control transfer functions, initial conditions have an effect in the initial regulator selection. Since linear systems are considered, initial conditions do not determine steady-state properties, however.

7.3.3.1 Initial Regulator: Min-Only Case

Using (7.45) directly, the initial regulator i_0 must satisfy

$$(K_j - K_{i_0})\tilde{x}_{ai_0}(0) \leq -K_j\bar{x}_{a_{i_0}} + P_j r_j$$

for all $j \in L$. Using identity (7.34), i_0 is determined as the smallest index satisfying

$$(K_j - K_{i_0})\tilde{x}_{ai_0}(0) \leq -M_j \frac{r_{i_0}}{G_{i_0}(0)} + P_j r_j \tag{7.46}$$

for all $j \in L$ where the scalar M_j is defined as

$$M_j = K_j \begin{bmatrix} -A^{-1} & B \\ & 1 \end{bmatrix}. \tag{7.47}$$

7.3.3.2 Initial Regulator: Max-Only Case

Simply reversing inequality (7.46), we see that initial regulator i_0 is the smallest index satisfying

$$(K_j - K_{i_0})\tilde{x}_{ai_0}(0) \geq -M_j \frac{r_{i_0}}{G_{i_0}(0)} + P_j r_j \tag{7.48}$$

for all $j \in H$.

7.3.3.3 Initial Regulator: Min–Max Case

The logic is the same as for the dynamic compensator case: if $i_0 \in L$, the following inequalities must hold:

$$(K_j - K_{i_0})\tilde{x}_{ai_0}(0) \leq -K_j \bar{x}_{a_{i_0}} + P_j r_j \text{ for all } j \in L, \tag{7.49}$$

$$(K_j - K_{i_0})\tilde{x}_{ai_0}(0) \geq -M_k \frac{r_{i_0}}{G_{i_0}(0)} + P_k r_k \text{ for all } k \in H. \tag{7.50}$$

If $i_0 \in H$, however, the following inequalities apply:

$$(K_j - K_{i_0})\tilde{x}_{ai_0}(0) \geq -M_k \frac{r_{i_0}}{G_{i_0}(0)} + P_k r_k \text{ for all } k \in H, \tag{7.51}$$

$$(K_j - K_{i_0})\tilde{x}_{ai_0}(0) > \min_{j \in L} \left\{ u_{rj}(0) \right\}. \tag{7.52}$$

A guess must be made regarding whether $i_0 \in L$ or $i_0 \in H$ and the corresponding inequalities verified.

7.3.3.4 Steady Regulator: Min-Only Case

Suppose i^* is the index of the steady regulator. Then $\dot{u} = u_{ri^*} = 0$ and $\tilde{x}_{ai^*} = 0$. Equation (7.45) is used directly to determine that i^* is the smallest index satisfying

$$M_j \frac{r_{i^*}}{G_{i^*}(0)} \leq P_j r_j$$

for all $j \in L$. If P_j is designed so that $y_j = r_j$ in steady-state (i.e., according to (7.35)), then the reader can verify that i^* is the smallest index satisfying

$$\frac{1}{G_j(0)} M_j \left(\bar{y}_{j/i^*} - r_j \right) \leq 0, \tag{7.53}$$

where $\bar{y}_{j/i^*} = G_j(0) r_{i^*} / G_{i^*}(0)$ is the steady value attained by y_j when i^* is active at steady state. This result is interpreted in a similar way as done for the dynamic compensator case, where $M_j / G_j(0)$ plays the role of the compensator's low-frequency gain $K(0)$: when $M_j / G_j(0) > 0$, i^* is the regulator for which outputs do not exceed their setpoints at steady-state. If $M_j / G_j(0) < 0$, i^* is the regulator for which outputs do not become smaller than their setpoints at steady-state. Note from (7.35) that $M_j / G_j(0) \neq 0$, otherwise the tracking task would not be possible.

7.3.3.5 Steady Regulator: Max-Only Case

In this case, i^* is the smallest index satisfying

$$\frac{1}{G_j(0)} M_j \left(\bar{y}_{j/i^*} - r_j \right) \geq 0 \tag{7.54}$$

for all $j \in H$. When $M_j / G_j(0) > 0$, i^* is the regulator for which outputs do not become smaller than their setpoints at steady-state. When $M_j / G_j(0) < 0$, i^* is the regulator for which outputs do not exceed their setpoints at steady-state.

In summary, when $M_j / G_j(0) > 0$, the min selector protects upper-limited outputs at steady-state and the max selector protects lower-limited outputs at steady-state. When $M_j / G_j(0) < 0$, the reverse steady-state protections are afforded.

7.3.3.6 Steady Regulator: Min–Max Case

If $i^* \in L$, the following inequalities must hold:

$$M_j \frac{r_{i^*}}{G_{i^*}(0)} \leq P_j r_j, \tag{7.55}$$

$$M_k \frac{r_{i^*}}{G_{i^*}(0)} \geq P_k r_k \tag{7.56}$$

for all $j \in L$ and all $k \in H$. If there exist values of j or k for which the above inequalities fail, then it must be that $i^* \in H$ and the following inequalities must hold:

$$M_k \frac{r_{i^*}}{G_{i^*}(0)} \geq P_k r_k, \tag{7.57}$$

$$\min \left\{ -M_j \frac{r_{i^*}}{G_{i^*}(0)} + P_j r_j \right\} < 0 \tag{7.58}$$

for all $k \in H$, with the minimum in inequality (7.58) is taken over $j \in L$. These conditions can be interpreted following a similar process as done for the isolated min and max cases. Suppose first that $i^* \in L$. Inequalities (7.55) and (7.56) can be written as

$$\frac{M_j}{G_j(0)} (\bar{y}_{j/i^*} - r_j) \leq 0, \tag{7.59}$$

$$\frac{M_k}{G_k(0)} (\bar{y}_{k/i^*} - r_k) \geq 0 \tag{7.60}$$

Table 7.1 Guidelines for the association of outputs to selectors based on steady-state characteristics. When using the min–max arrangement, $i^* \in L$ is required

Case	Sign of $M_j/G_j(0)$	Association
I: All upper-limited,	+	Min only
$\text{sign}\left(\frac{M_j}{G_j(0)}\right)$ constant	–	Max only
II: All lower-limited,	+	Max only
$\text{sign}\left(\frac{M_j}{G_j(0)}\right)$ constant	–	Min only
III: All upper-limited,	+	Min in min–max
mixed $\text{sign}\left(\frac{M_j}{G_j(0)}\right)$	–	Max in min–max
IV: All lower-limited,	+	Max in min–max
mixed $\text{sign}\left(\frac{M_j}{G_j(0)}\right)$	–	Min in min–max
Mixed limits,		Use min–max
V: mixed $\text{sign}\left(\frac{M_j}{G_j(0)}\right)$		As in III and IV

for all $j \in L$ and all $k \in H$. Thus, min-linked outputs y_j such that $M_j/G_j(0) > 0$ and max-linked y_k outputs such that $M_k/G_k(0) < 0$ will remain below their setpoints at steady-state. Similarly, min-linked outputs y_j such that $M_j/G_j(0) < 0$ and max-linked y_k outputs such that $M_k/G_k(0) > 0$ will remain above their setpoints at steady-state. Now suppose design parameters are chosen so that $i^* \in H$. Inequality (7.60) holds for all $k \in H$, but min-linked outputs must satisfy the following:

$$0 > \min\left\{\frac{M_j}{G_j(0)}(r_j - \bar{y}_{j/i^*})\right\}.$$

When $M_j/G_j(0) < 0$ we have

$$\min\left\{\frac{M_j}{G_j(0)}(r_j - \bar{y}_{j/i^*})\right\} = \frac{M_j}{G_j(0)} \max\left\{r_j - \bar{y}_{j/i^*}\right\}.$$

It follows that $r_j < \bar{y}\, j/i^*$ for all $j \in L$, indicating that min-linked variables are lower-limited by r_j, matching the min-only result. However, if $M_j/G_j(0) > 0$, we can only say:

$$0 > \min\left\{r_j - \bar{y}_{j/i^*}\right\}$$

and no bounds can be guaranteed for $\bar{y}_{j/i^*}$. Thus, if the design requires $i^* \in H$ for some reason, only lower-limited outputs having $M_j/G_j(0) < 0$ should be linked to the min-selector.

The preceding analysis can be regarded as a design guideline for the association of outputs to selectors. This is done solely on the basis of steady-state characteristics. The guidelines have been summarized in Table 7.1.

7.3.4 Example: CMAPSS-1 Linearized Model

Consider the same transfer functions of Example 7.1.3. Matrices A and B of the state-space realization are as follows:

$$A = \begin{bmatrix} -3.8420 & 1.4125 \\ 0.5310 & -4.6623 \end{bmatrix}, \quad B = \begin{bmatrix} 230.9226 \\ 653.7255 \end{bmatrix}.$$

For $y_1 = \Delta N_f$, $C_1 = [1 \;\; 0]$ and $D_1 = 0$. The values of matrices C and D for ΔT_{48} are $C = [-0.1022 \;\; -0.2952]$ and $D = 146.24$. Finally, a state-space realization of the transfer function to ΔSmHPC using the same A and B has $C = [0.0189 \;\; 0.0066]$ and $D = -4.4052$. Note that this flight condition is close, but not equal to FC01 listed in Appendix B

Consider first that y_1 must be driven to a setpoint $r_1 = 100$ rpm while preventing $y_2 = \Delta T_{48}$ from exceeding $r_2 = 200°$R. That is, y_2 is an upper-limited output. The individual regulators are designed using an LQR approach for the augmented plant, with $Q = C_{a1}^{\mathrm{T}} C_{a1}$ and $Q = C_{a2}^{\mathrm{T}} C_{a2}$, where $C_{a1} = [C_1 \;\; D_1]$ and $C_{a2} = [C_2 \;\; D_2]$ are the augmented output matrices. Using $R = 1$ for both designs, the resulting state-feedback gains are

$$K_1 = [0.7098 \;\; 0.0840 \;\; 20.9204],$$
$$K_2 = [-0.1006 \;\; -0.2870 \;\; 144.7929].$$

The required values of P_1 and P_2 for perfect tracking when the regulators are used independently are $P_1 = 1$ and $P_2 = 1$, with $\bar{x}_{a1}^{\mathrm{T}} = [100 \;\; 131.7106 \;\; 0.8581]$, as determined from formulas (7.35) and (7.34). Suppose that a min selector is used and, for illustrative purposes, that the input integrator has initial condition $u(0) = 1$. Then $\tilde{x}_{a1}(0) = [-100 \;\; -131.7106 \;\; 0.1419]$. The initial regulator is then $i_0 = 2$ by application of condition (7.46). Similarly, condition (7.53) predicts that $i^* = 1$. This is confirmed by the simulation results of Fig. 7.5. If the min selector is not used and a feedback loop is established to control y_1 to its setpoint, y_1 has a settling time of about 0.4 s and zero steady-state error by design. Output y_2, however, exceeds its intended limit by far during the transient regime and settles at $\bar{y}_{2/1} = 76.39 < r_2$. If the min selector is used, u_{r2} is active from $t = 0$ until $t = 0.22$, which causes y_2 to be regulated toward its limit. A regulator switching occurs near $t = 0.22$ and $i = 1$ becomes active for all subsequent times. Note that y_1 overshoots its limit by about 2%. Thus, this example shows that the min selector preserved limits during the transient regime for y_2 but not for y_1, motivating further analysis of limit protection properties. This is done in Sect. 7.3.5. Now suppose ΔT_{48} is not a concern, but that the HPC stall margin must be kept above certain limit to minimize the risk of stall or surge. Define $y_2 = \Delta$ SmHPC and suppose that the stall margin must not decrease more than 10% from its steady value at FC01, that is y_2 must be lower-limited by $r_2 = -10$. References to the standard min–max arrangement used in the GTE

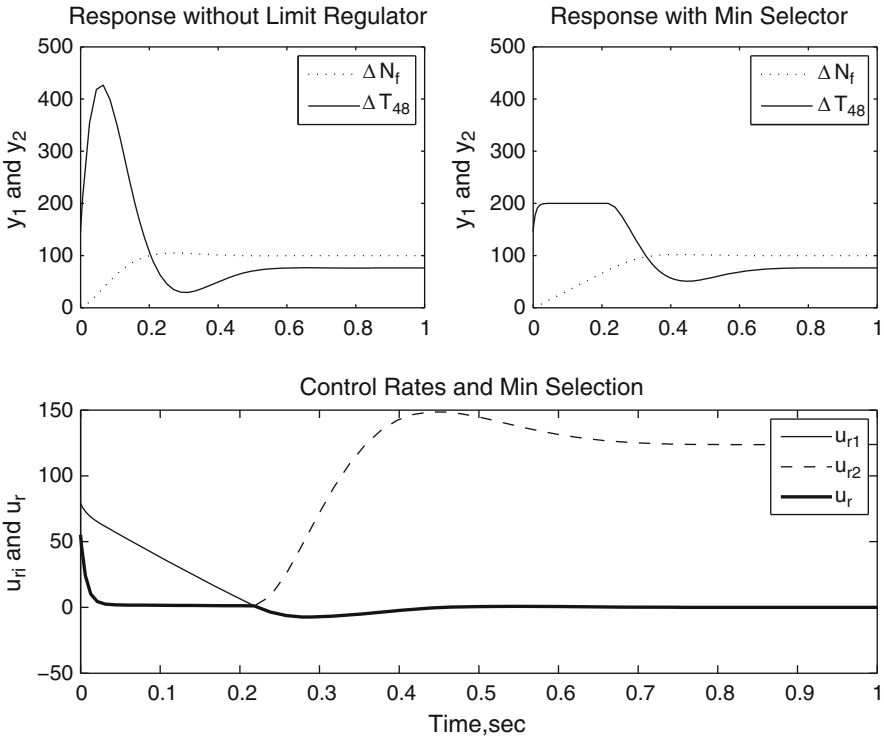

Fig. 7.5 Response of state-feedback system with min selector: CMAPSS-1 linearized model with upper-limited output $y_2 = \Delta T_{48}$

industry assume a priori that the regulators of lower-limited outputs are applied to the ports of a max selector. As this example demonstrates, the choice of selector must take additional considerations into account. Suppose K_2 is designed using an LQR approach with $Q = C_{a2}^{\mathrm{T}} C_{a2}$ and $R = 1$, and K_1 and r_1 are maintained. Now,

$$K_2 = [-0.0097 \quad -0.0045 \quad 3.0135]$$

and $P_2 = -1$. If, according to conventional wisdom, the max selector is used, it can be verified that $i_0 = 1$ and $i^* = 2$. As shown in Fig. 7.6, the results are disastrous. Since $i^* = 2$, the lower limit imposed on y_2 is preserved at steady-state, but $\bar{y}_{1/2} = 975.98$, far above r_1. Also, y_2 becomes smaller than its limit during the transient. If a min selector is used instead, the results are acceptable if fan speed is allowed to overshoot. In this case, $i_0 = 2$ and $i^* = 1$. Since $\bar{y}_{2/1} = -1.02$, the lower limit imposed in y_2 is still preserved at steady-state, and moreover, transient limit protection is observed. To illustrate the initial and steady properties of the min–max arrangement, consider the following output definitions: $y_1 = \Delta N_{\mathrm{f}}$, $y_2 = \Delta T_{48}$, $y_3 = \Delta$ Ps30 and $y_4 = \Delta(W_{\mathrm{F}}/\mathrm{Ps30})$. The last two outputs are the incremental

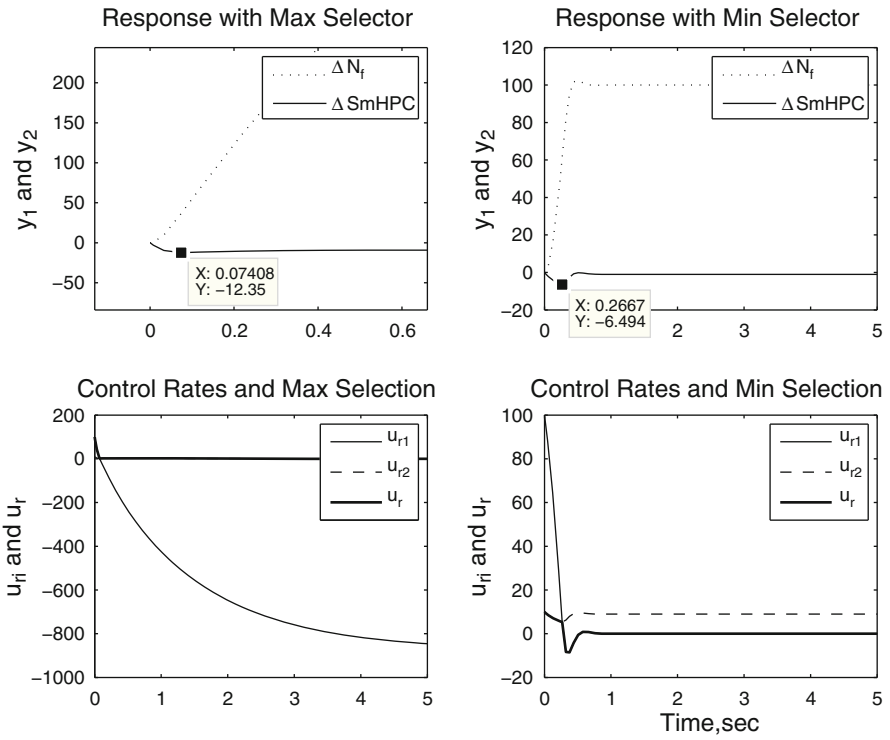

Fig. 7.6 Response of state-feedback system with max and min selectors: CMAPSS-1 linearized model with lower-limited output $y_2 = \Delta$ SmHPC

static pressure at HPC outlet, and the incremental ratio between fuel flow and Ps30. To prevent lean blowout conditions in the combustor, y_3 is lower-limited. A low limit is usually imposed on y_4 to prevent the LPC from stalling.

The linearized C and D parameters for y_3 and y_4 can be found in Appendix B. The state feedback gains are designed using an LQR approach with $Q = C_{ai}^{\mathrm{T}} C_{ai}$ and $R = 1$ as before. The resulting gains are

$$K_3 = [0.0639 \;\; 0.1534 \;\; 25.2464],$$
$$K_4 = [-0.0039 \;\; -0.0085 \;\; 3.5781]$$

with corresponding values of $P_3 = 1$ and $P_4 = 1$. The two upper-limited outputs are linked to the min selector, while the lower-limited outputs are linked to the max selector. Consider setpoints as follows: $r_1 = 100$, $r_2 = 200$, $r_3 = -50$, and $r_4 = -20$. Formulas (7.49)–(7.52) can be used to determine that $i_0 = 1$. Likewise, formulas (7.55)–(7.58) indicate that $i^* = 1$. The response to this case is shown in Fig. 7.7. Since y_3 and y_4 have DC gains $G_3(0) = G_4(0)$ with the same sign as $G_1(0)$, their DC value increases when $r_1 > 0$. Therefore, they move away from their

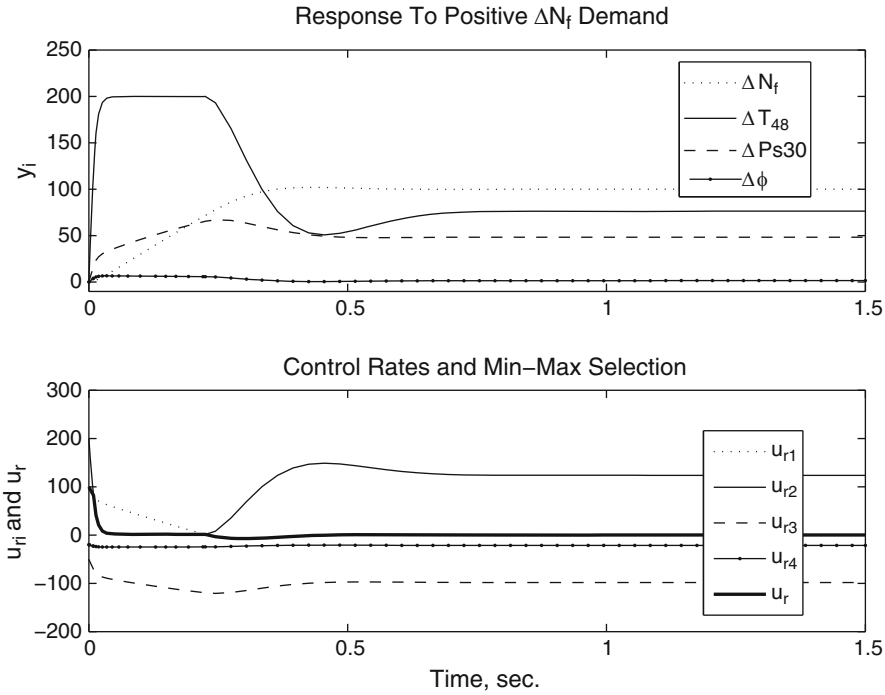

Fig. 7.7 Response of state-feedback system with min–max selectors: CMAPSS-1 linearized model with upper-limited output $y_2 = \Delta T_{48}$ and lower-limited outputs $y_3 = \Delta$ Ps30 and $y_4 = \Delta W_F$/Ps30 (fan speed increase demand)

negative limits and their regulators are never active. As in the previous example, the T_{48} limit regulator becomes active in the transient regime. Now suppose $r_1 = -100$ rpm. In this case, y_3 and y_4 decrease from their initial values. It can be verified that $i_0 = 4$ and $i^* = 1$. As shown in Fig. 7.8, the 4th regulator remains active for some time after $t = 0$ and y_3 and y_4 are maintained above their lower limits at steady state, as well as during the transient.

7.3.5 Transient Limit Protection Analysis

The invariance condition for intervals presented in Sect. 7.2.1 is now applied to study the limit protection properties of the min–max approaches with static state feedback. Analysis is divided into three cases corresponding to min-only, max-only, and min–max selectors. For upper-limited outputs, the error $e = r - y$ must not become negative. That is, $[0, \infty)$ must be positively invariant. For lower-limited outputs, e must not become positive, that is, $(-\infty, 0]$ must be positively invariant.

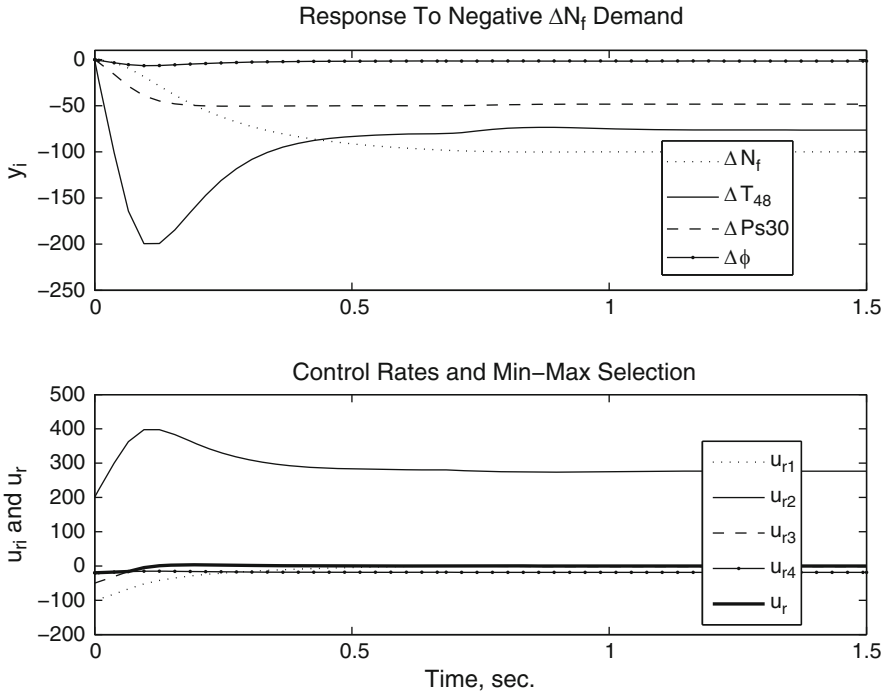

Fig. 7.8 Response of state-feedback system with min–max selectors: CMAPSS-1 linearized model with upper-limited output $y_2 = \Delta T_{48}$ and lower-limited outputs $y_3 = \Delta$ Ps30 and $y_4 = \Delta W_F$/Ps30 (fan speed decrease demand)

The sign of $\dot{e}$ must be examined along the boundary $e = 0$ to determine invariance. Given a fixed i and $j \neq i$, five sets are relevant:

1. $\mathcal{U}_{ij}$: the set where $u_{ri} = u_{rj}$
2. $\mathcal{E}_i$: the set where $e_i = 0$
3. $\mathcal{E}_j$: the set where $e_j = 0$
4. $\dot{\mathcal{E}}_i$: the set where $\dot{e}_{i/i} = 0$
5. $\dot{\mathcal{E}}_{j/i}$: the set where $\dot{e}_{j/i} = 0$

Set $\mathcal{U}_{ij}$ represents the boundary between the two half-spaces which are the regions of activity of each regulator, and the roles of the other sets are clear from their definitions. For the min-only or max-only cases, the sets are hyperplanes, as represented in Figs. 7.9 and 7.10. Sets $\dot{\mathcal{E}}_i$ and $\dot{\mathcal{E}}_{j/i}$ divide the space into two halves, one where the error increases and the other where the error decreases, and this has been indicated with plus or minus signs.

For upper-limited variables, the invariance condition requires that the derivative of the error be nonnegative whenever the error is zero. Now, as seen in Figs. 7.9 and 7.10, $\mathcal{E}_i$ and $\mathcal{E}_{j/i}$ are divided into two regions, corresponding to positive and

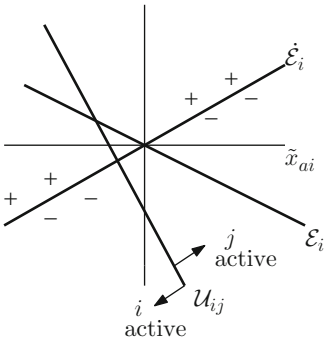

Fig. 7.9 Schematic of the regions corresponding to sets $\mathcal{U}_{ij}$, $\mathcal{E}_i$ and $\dot{\mathcal{E}}_i$

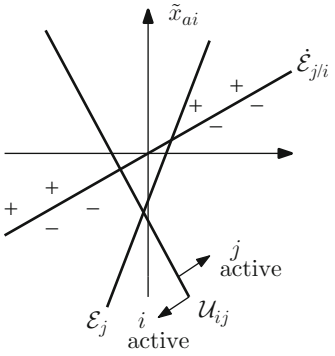

Fig. 7.10 Schematic of the regions corresponding to sets $\mathcal{U}_{ij}$, $\mathcal{E}_j$ and $\dot{\mathcal{E}}_{j/i}$

negative error derivatives. Assuming that i is the active regulator, *limit protection is ensured by requiring that the regions of $\mathcal{E}_i$ and $\mathcal{E}_{j/i}$ where, the error increases lie outside the region where i is active*. That is, the intersection of $\dot{\mathcal{E}}_i$ and $\mathcal{E}_i$ must be outside the half space where i is active, and the intersection of $\dot{\mathcal{E}}_{j/i}$ and $\mathcal{E}_j$ must likewise be outside the half space where i is active.

Equations (7.39) and (7.43) are combined to determine the set $\dot{\mathcal{E}}_i \cap \mathcal{E}_i$, resulting in the following system of linear equations on $\tilde{x}_{ai}$:

$$\begin{bmatrix} -C_{ai} \\ -C_{ai}A_{ci} \end{bmatrix} \tilde{x}_{ai} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{7.61}$$

The matrix characterizing the above system of equations is full-rank, unless C_{ai} and $C_{ai}A_{ci}$ are parallel vectors. For this to happen, there would have to exist a scalar λ such that $C_{ai}A_{ci} = \lambda C_{ai}$. Taking the transpose of this equation, we get

$$A_{ci}^{\mathrm{T}} C_{ai}^{\mathrm{T}} = \lambda C_{ai}^{\mathrm{T}}$$

that is, C_{ai}^{T} would have to be an eigenvector of A_{ci}^{T}. Assuming that K_i has been verified not to produce this exception, the only solution to system (7.61) is $\tilde{x}_{ai} = 0$.

Likewise, (7.40) and (7.44) are combined to determine the set $\dot{\mathcal{E}}_{j/i} \cap \mathcal{E}_j$, resulting in the following system of linear equations on $\tilde{x}_{ai}$:

$$\begin{bmatrix} -C_{aj} \\ -C_{aj} A_{ci} \end{bmatrix} \tilde{x}_{ai} = \begin{bmatrix} \bar{e}_{j/i} \\ 0 \end{bmatrix}. \tag{7.62}$$

System (7.62) does not have a unique solution in general. In this case, it becomes necessary to find particular ones that assist in arriving at invariance conditions. This is done separately for the min-only, max-only, and min–max cases.

7.3.5.1 Transient Limit Protection: Min-Only Case

Suppose all variables are upper-limited. To guarantee that y_i will not exceed its upper limit, we require that $u_{ri} - u_{rj} > 0$ along solutions to (7.61). If C_{ai}^{T} is not an eigenvector of A_{ci}^{T}, $\tilde{x}_{ai} = 0$ is the only solution. Using (7.45), we see that the required inequality reduces to $K_j \bar{x}_{ai} > P_j r_j$ for all $j \in L$. Using (7.34) and the definition of M_j from (7.47), this becomes:

$$M_j \frac{r_i}{G_i(0)} > P_j r_j \tag{7.63}$$

for all $j \in L$. Noting that inequality (7.63) is the opposite of inequality (7.53), we see that transient limit protection cannot be guaranteed for $i = i^*$, even for intervals of time where i^* is active! This can be confirmed in the min-only case of Example 7.3.4, where $i^* = 1$ and y_1 exhibits overshoot while its own regulator is active. In the max-only case where $i^* = 2$, y_2 violates its limit in the transient regime, even when its regulator is active. These shortcomings are removed by using sliding mode controllers instead of linear regulators (See Chap. 8).

Now consider the condition for y_j not to exceed its upper limit while i is active. For this, we require that $u_{ri} - u_{rj} > 0$ along solutions to (7.62). Since there is no explicit solution in this case, the solution that minimizes $u_{ri} - u_{rj}$ is sought, followed by a requirement that the minimum solution be positive. Mathematically, the invariance condition can be expressed as

$$(K_j - K_i)\tilde{x}_{ai}^* > -K_j \bar{x}_{ai} + P_j r_j,$$

where $\tilde{x}_{ai}^*$ is the solution to the following constrained minimization problem:

$$\tilde{x}_{ai}^* = \arg\min (K_j - K_i)\tilde{x}_{ai}$$

subject to

$$\begin{bmatrix} -C_{aj} \\ -C_{aj} A_{ci} \end{bmatrix} \tilde{x}_{ai} = \begin{bmatrix} \bar{e}_{j/i} \\ 0 \end{bmatrix}. \tag{7.64}$$

A negative test for invariance can be conducted by finding *any* solution to system (7.62) and checking whether $u_{ri} \leq u_{rj}$ is satisfied for all $j \in L$ for that particular value of $\tilde{x}_{ai}$. If so, the set $\dot{\mathcal{E}}_{j/i} \cap \mathcal{E}_j$ contains at least one point for which the active regulator is i and invariance for y_j cannot be concluded.

7.3.5.2 Transient Limit Protection: Max-Only Case

The corresponding invariance conditions for y_i and y_j are obtained by reversing the inequalities in (7.63) and (7.64), and are the same whether upper- or lower-limited variables are involved. That is, invariance for y_i is guaranteed if:

$$K_j \bar{x}_{ai} < P_j r_j \tag{7.65}$$

for all $j \in H$. Similarly, invariance for y_j is guaranteed if:

$$(K_j - K_i)\tilde{x}^*_{ai} < -K_j \bar{x}_{ai} + P_j r_j,$$

where $\tilde{x}^*_{ai}$ is the solution to the following constrained minimization problem:

$$\begin{aligned} \tilde{x}^*_{ai} &= \arg\min (K_j - K_i)\tilde{x}_{ai} \\ &\quad \text{subject to} \\ \begin{bmatrix} -C_{aj} \\ -C_{aj} A_{ci} \end{bmatrix} \tilde{x}_{ai} &= \begin{bmatrix} \bar{e}_{j/i} \\ 0 \end{bmatrix}. \end{aligned} \tag{7.66}$$

Although the invariance conditions for the min-only and max-only cases were developed used two regulators, their validity extends to more than two regulators by considering all possible pairs of regulators.

7.3.5.3 Transient Limit Protection: Min–Max Case

In this case, analysis must be made in groups of three regulators, an arbitrary fixed regulator i assumed active, a regulator j in the min group, and a regulator k in the max group. Two cases must be considered: $i \in L$ and $i \in H$. Assuming $i \in L$, the region where i is active is given by $\mathcal{U}_L \cap \mathcal{U}_H$, where:

$$\mathcal{U}_L = \left\{ \tilde{x}_{ai} \ : \ u_{ri} - u_{rj} \leq 0 \text{ for all } j \in L \right\},$$

$$\mathcal{U}_H = \left\{ \tilde{x}_{ai} \ : \ u_{ri} - u_{rk} \geq 0 \text{ for all } k \in H \right\}.$$

Invariance will be guaranteed if $\dot{\mathcal{E}}_i \cap \mathcal{E}_i$ lies outside $\mathcal{U}_L \cap \mathcal{U}_H$. As before, if $\tilde{x}_{ai} = 0$ is the only element of $\dot{\mathcal{E}}_i \cap \mathcal{E}_i$, the invariance conditions reduce to

$$K_j \bar{x}_{ai} > P_j r_j, \tag{7.67}$$

$$K_k \bar{x}_{ai} < P_k r_k \tag{7.68}$$

for all $j \in L$ and for all k in H. For a regulator $i \in H$, the region where it is active is given by $\mathcal{U}_H \cap \mathcal{U}_l$, where $\mathcal{U}_l$ is defined as

$$\mathcal{U}_l = \left\{ \tilde{x}_{ai} \ : \ u_{ri} > \min \left\{ u_{rj} \right\}, j \in L \right\}.$$

The reader will observe that this set has a complex description. Establishing the invariance of y_j while i is active is also rather difficult. First, j must be assumed to begin to either L or H. For each case, subcases corresponding to $i \in L$ and $i \in H$ must be contemplated. This process is best handled numerically, with the aid of a computer program.

7.4 Example: CMAPSS-1 Linearized Model

Continuing with Example 7.3.4, consider first the problem of establishing transient limit protection when only the min selector is used. It can be verified that C_{a1} is not parallel to $C_{a1}A_{ci}$ and therefore $\tilde{x}_{a1} = 0$ is the only solution to system (7.61). Thus, no transient limit protection can be guaranteed for i^*. Although (7.63) is only a sufficient condition, simulation showed that y_1 overshoots its limit in the transient regime, even for intervals when its own regulator is active. To see whether invariance can be established for y_2 while $i^* = 1$ is active, note that $\bar{y}_{2/1} = 76.39$, so $\bar{e}_{2/1} = 123.61$. An exact solution to system (7.62) can be readily found using the pseudoinverse as $\tilde{x}_{a1}^{\mathrm{T}} = [25.4488 \quad 4.2484 \quad -0.8189]$. Evaluating $u_{r1} - u_{r2}$ at this value using (7.45) gives -247.25, indicating, again, that there is a point where $e_2 = \dot{e}_{2/1} = 0$ with $i = 1$ being active. Therefore, invariance for y_2 cannot be concluded during the transient regime for periods where $i = 1$ is active. Simulation shows, however, that y_2 remains below its limit. The conservativeness of the sufficient condition can be traced to the fact that the solution found for $\tilde{x}_{a1}$ does not belong to the actual state trajectory followed by the system. A refined version of the invariance condition would require taking into account such trajectories, which depend on the previous switching history.

Now consider the max-only case of Example 7.3.4. Again, no transient limit protection can be guaranteed for $i^* = 2$ for periods where this regulator is active. Condition (7.63), is satisfied for $i = 1$, however. Therefore, y_1 will not exceed its limit when its own regulator is active. In this example $i_0 = 1$, and $i = 1$ remains active for a short time, where $y_1 < r_1$, verifying transient invariance for this case.

A regulator switching occurs and the active regulator becomes $i^* = 2$ for all future times. Invariance cannot be concluded for y_2 while $i = 2$ is active, and simulation confirms that y_2 becomes smaller than its limit.

The preceding steady and transient limit protection analysis, as well as the examples, suggest that the standard min–max arrangement may be ill-conceived, at least when used in conjunction with linear regulators. Although steady limit preservation is achieved in a simple fashion, there is no clear way to guarantee that outputs will remain within their prescribed bounds in the transient regime. When sliding mode controllers are used, stability and limit protection are guaranteed under most cases, as described in Chap. 8.

7.5 Alternative Minimum-Interaction Design: $\mathcal{H}_\infty$ Approach

A reasonable alternative to using limit regulators is to design the state feedback gain so that one of the outputs (say, fan speed) has a suitably fast transient response while the other outputs are suppressed during transients, minimizing the possibility of limit violations. This problem can be formulated using the multiobjective $\mathcal{H}_\infty$ state feedback synthesis discussed in Chap. 4. Adequate fan speed response is obtained by the built-in robust stability property, together with a regional eigenvalue placement constraint. Limited outputs are regarded as performance outputs z_2 or z_∞. The latter choice is convenient when output disturbances are included, since the $\mathcal{H}_\infty$ norm remains finite even for nonstrictly proper transfer functions.

7.5.1 Example: CMAPSS-1

The constant feedback gain $\mathcal{H}_\infty$ design is now applied to the fan speed control problem near FC01. System parameters are given in Appendix B. Recalling the developments of Chap. 4, the robust state feedback synthesis approach handles the system description given below:

$$\dot{x}_\mathrm{a} = A_\mathrm{a} x_\mathrm{a} + B_\mathrm{a} u_r + \Gamma_a w, \tag{7.69}$$

$$z_\infty = C x_\mathrm{a}, \tag{7.70}$$

where w is a vector of disturbances. Matrix Γ has not been listed in Appendix B, but may be obtained through linearization using the CMAPSS-1 interface. The objective is to find a feedback gain K stabilizing the augmented closed-loop matrix $A_\mathrm{c} = A_\mathrm{a} - B_\mathrm{a} K$ while minimizing the infinity norm of the transfer function from w to z_∞. The designer controls fan speed response by specifying a target region for the closed-loop eigenvalues of A_c. The $\mathcal{H}_\infty$ minimization objective translates into transient suppression of the limited outputs. The presence of a disturbance

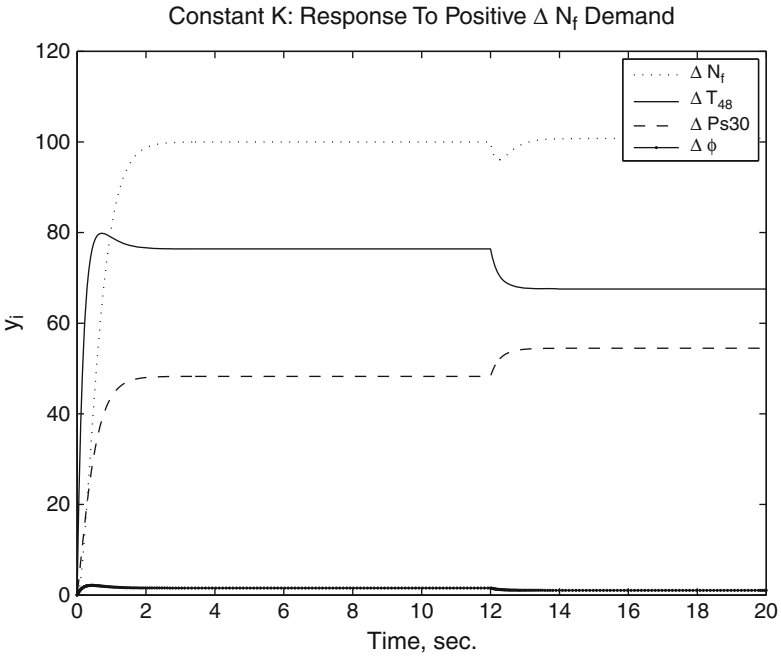

Fig. 7.11 Response of limited outputs for minimum interaction design: positive ΔN_f demand

component introduces a design tradeoff. The larger the influence of disturbances (as measured through matrix Γ), the smaller the closed-loop allowable bandwidth. This can be observed when carrying the design using, for instance, the `msfsyn` command: Γ may need to be scaled to maintain feasibility.

In this example, Γ is scaled to 10% of their linearization value. A disk centered at -10 with radius 7 is used as the target region, resulting in an $\mathcal{H}_\infty$ norm of 169 and a feedback gain as follows:

$$K = [0.0001 \quad -0.0008 \quad 3.8233],$$

which places the closed-loop eigenvalues of A_c at -5.444, -3.8419, and -3.0417. Using a fan speed demand of $\Delta N_\mathrm{f} = 100$ rpm, it can be verified that the corresponding steady values of the limited outputs are $\Delta T_{48} = 75.835°$ R, Δ Ps30=48.26 psi and $\Delta\Phi = 1.56$ pps/psi. As Fig. 7.11 shows, the transient peak of ΔT_{48} remains under the limit of $r_2 = 200$ that was used in the previous examples. A step health parameter disturbance was applied at $t = 12$ s. Figure 7.12 corresponds to $\Delta N_\mathrm{f} = -100$ rpm. In this case, all limited outputs remain within the bounds used in the previous examples.

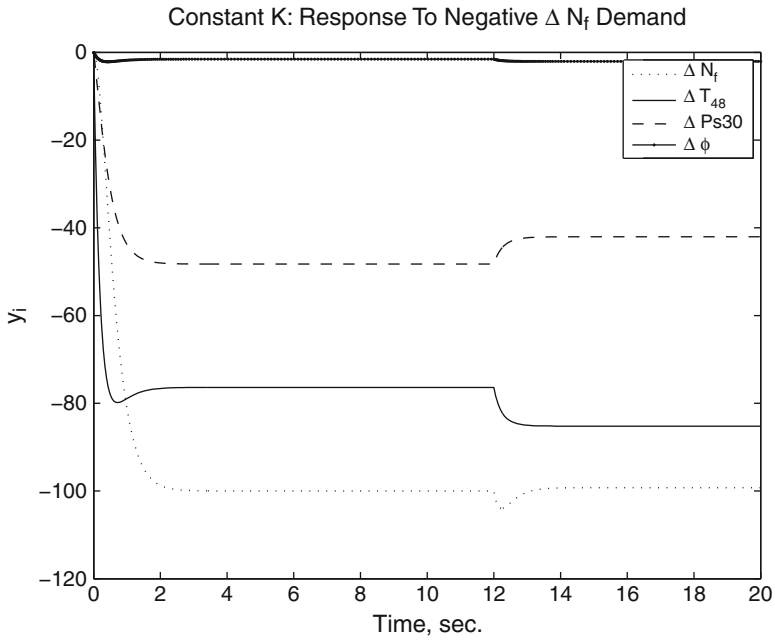

Fig. 7.12 Response of limited outputs for minimum interaction design: negative ΔN_{f} demand

7.5.2 Example: Ellipsoidal Invariant Set: CMAPSS-1 Linearized Model

An ellipsoidal invariant set is now derived for the closed-loop system of the previous example, first using the feedback gain obtained through $\mathcal{H}_\infty$ synthesis. Define a shifted state vector as done in Sect. 7.3.2:

$$\tilde{x}_{\mathrm{a}} = x_{\mathrm{a}} - \bar{x}_{\mathrm{a}},$$

where $\bar{x}_{\mathrm{a}}$ represents a target state so that $y_1 = r_1$ when the system reaches steady-state. The state-feedback law for fan speed setpoint tracking is $u_r = -Kx_{\mathrm{a}} + Pr_1$, which can be written in terms of $\tilde{x}_{\mathrm{a}}$ as follows:

$$u_r = -Kx_{\mathrm{a}} + Pr_1 = -K(\tilde{x}_{\mathrm{a}} - \bar{x}_{\mathrm{a}}) + Pr_1 = -K\tilde{x}_{\mathrm{a}},$$

where (7.33) has been used. Recalling that $A_{\mathrm{a}}\bar{x}_{\mathrm{a}} = 0$, substitution of the above control law into system (7.69) yields the following closed-loop dynamics

$$\dot{\tilde{x}}_{\mathrm{a}} = (A_{\mathrm{a}} - B_{\mathrm{a}}K)\tilde{x}_{\mathrm{a}}.$$

Now, the constraints are formulated in terms of x_a from the limited output specifications:

$$y_2 = C_{a2}\tilde{x}_a \le r_2,$$
$$y_3 = C_{a3}\tilde{x}_a \ge r_3,$$
$$y_4 = C_{a3}\tilde{x}_a \ge r_4.$$

Note that the first output (incremental fan speed) has not been regarded as a limited output, but if necessary, it can be directly added to the list of constraints. The constraint vectors are now normalized so that they correspond to the assumed description $G_i x \le 1$. For this, inequalities reflecting positive upper bounds are simply divided by the value of the bound. For negative lower bounds, the left-hand and right-hand sides of the inequality are switched and the inequality divided by the negative of the bound. According to this, the three constraints adopt the desired form, with $G_2 = C_{a2}/r_2$, $G_3 = C_{a3}/r_3$, and $G_4 = C_{a4}/r_4$.

An arbitrary symmetric, positive-definite matrix Q may be used to obtain a family of invariant ellipsoids from the solution of Lyapunov equation (7.24). For each constraint, the largest admissible ellipsoid from this family is found by calculating its V_i value. The smallest V_i, say V, is then selected, and the operating set for $\tilde{x}_a$ is defined as $\tilde{x}_a^T P \tilde{x}_a \le V$. Although this guarantees invariance, the orientation of the ellipsoid may not be the best for the given constraints. An improvement is to find P to maximize the volume of the ellipsoid, subject to the invariance condition (7.23) and $G_i \tilde{x}_a \le 1$, see O'Dell [62].

As an example, take $Q = I$. The Lyapunov equation is solved using `lyap((Aa-Ba*K)',Q)` to yield

$$P = \begin{bmatrix} 0.1342 & 0.0303 & 7.3913 \\ 0.0303 & 0.1184 & 11.4089 \\ 7.3913 & 11.4089 & 2397.3 \end{bmatrix}.$$

The values of V_i are calculated as $V_2 = 1438.7$, $V_3 = 8780.9$, and $V_4 = 9526.1$. Therefore, the operating set for $\tilde{x}_a$ is described by the inequality $\tilde{x}_a^T P \tilde{x}_a \le 1438.7$.

This result can be given a practical interpretation by considering the meaning of $\tilde{x}_a$. The operating set restricts the distance between $x_a(0)$ and $\bar{x}_a$, the target state. Recalling that $x_a^T = [\Delta N_f \;\; \Delta N_c \;\; \Delta W_F]$ is the incremental state relative to a steady linearization point, we see that $x_a(0) = 0$ if the setpoint change maneuver starts at the linearization point. For such cases, we can restrict the target state to guarantee limit protection by enforcing $\bar{x}_a^T P \bar{x}_a \le 11.3$. Using (7.34), this inequality becomes

$$\left(\frac{r_1}{G_1(0)}\right)^2 \begin{bmatrix} -A^{-1}B \\ 1 \end{bmatrix}^T P \begin{bmatrix} -A^{-1}B \\ 1 \end{bmatrix} \le 1,$$

which in this example reduces to $|r_1| \leq 38.3$, a conservative figure. Now consider the problem of finding a stabilizing K that maximizes the volume of an invariant ellipsoid contained in the constrained set. This problem can be solved using O'Dell's techniques [62], yielding the following combination of P and K:

$$P = \begin{bmatrix} 0.00000219 & 0.00000537 & 0.00064 \\ 0.00000537 & 0.00001354 & 0.00121 \\ 0.00064366 & 0.00121329 & 1.12223 \end{bmatrix},$$

$$K = [0.000578 \;\; 0.001120 \;\; 0.877210].$$

This feedback gain places the closed-loop eigenvalues at -1.251, -2.965, and -5.165, producing a somewhat slower response in comparison with the value obtained through $\mathcal{H}_\infty$ synthesis. The minimum value of V is now 1 and the bound for r_1 becomes 78.8 rpm, a less conservative value. This value is still far from the maximum value of r_1 for which a limited output reaches its limit. Indeed, since all closed-loop poles are real, no output will exhibit overshoot. Thus, an upper-limited variable y_j can only reach r_j in the steady state. For this, r_{i*} must be chosen so that $\bar{y}_{j/i*} = r_j$. In this example, for y_2 to reach $r_2 = 200$, r_1 must be chosen as $r_2 G_1(0)/G_2(0) = 261.8$ rpm.

7.6 Acceleration and Deceleration Limiting

In addition to limits placed on the magnitudes of critical variables such as turbine temperature, shaft speeds, combustor pressure and engine pressure ratio, the core shaft acceleration must also be maintained between prescribed bounds. An upper bound is introduced to protect the engine against surge and stall, while the lower limit is introduced to provide safety against engine flame-out. Recalling (2.2), we see that core acceleration $\dot{N}_c$ depends on fuel flow, W_F. A traditional way to maintain core acceleration below its prescribed upper bound is simply to override the value of u_r calculated by the min-selected regulators, replacing it with a constant rate of zero pps/sec whenever the acceleration reaches its upper limit. Minimum acceleration is likewise maintained by replacing the rate produced by the max stage with $u_r = 0$ whenever the acceleration reaches its lower limit.

In CMAPSS-1, these operations are implemented by override switches triggered by $\dot{N}_c$, as shown in Figs. 7.13 and 7.14. The threshold value for $|\dot{N}_c|$ is 500 rpm/s. Note that a similar acceleration limiting scheme could, in principle, be applied to $\dot{N}_f$. When thrust is being controlled indirectly by a feedback loop on N_f, however, such scheme is not necessary, since reference ramps are usually commanded for this variable. Hence, fan acceleration should follow response specifications. When an EPR loop is established as a means to achieve thrust control, N_f and N_c may be regarded as upper-limited variables, and corresponding acceleration limiters included.

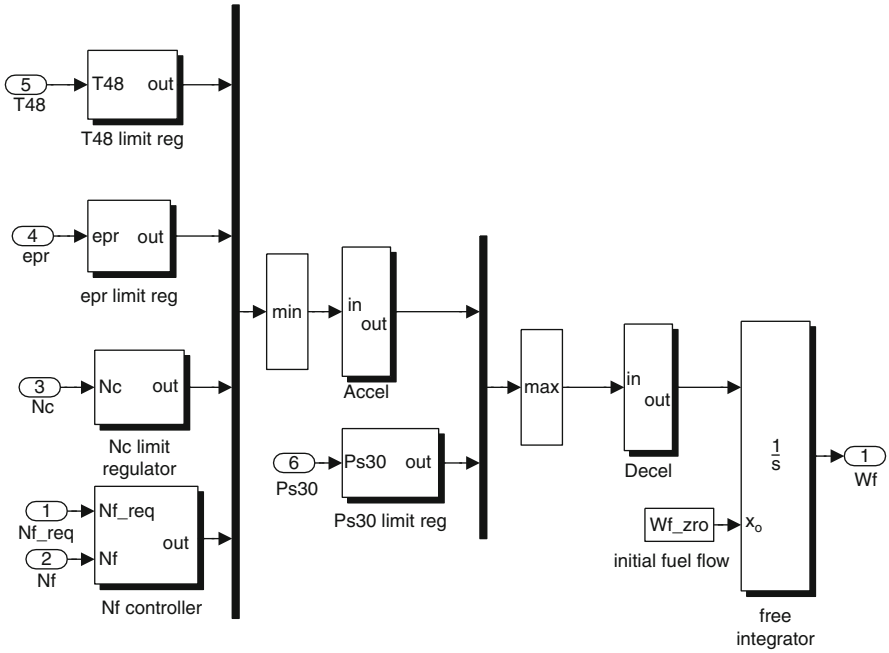

Fig. 7.13 Schematic of CMAPSS-1 min–max arrangement with acceleration and deceleration limiters

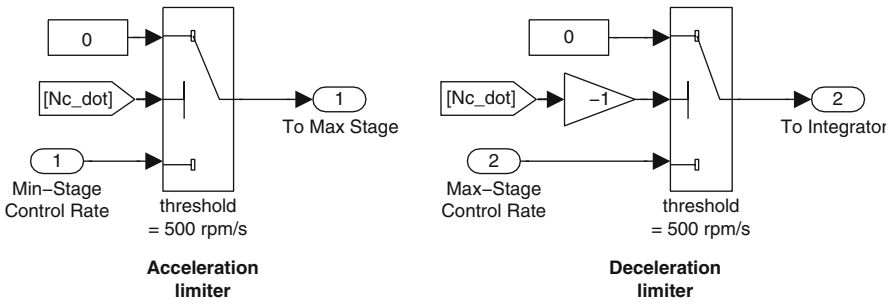

Fig. 7.14 Schematic of CMAPSS-1 acceleration and deceleration override switches

7.6.1 "N-Dot" Control and Acceleration Scheduling

The so-called "N-dot" control concept exploits the algebraic relationship between shaft accelerations and fuel flow, as observed in (2.1) and (2.2). In principle, if the inverse of functions f_1 and f_2 were available, one could compute values of fuel flow resulting in a desired instantaneous value for $\dot{N}_f$ or $\dot{N}_c$. In practice, no such inverses are available for real-time computations, and uncertain and unmeasurable time-varying parameters participate in the definitions of f_1 and f_2.

Since fuel flow is related to acceleration by an uncertain algebraic relationship, the latter may be controlled by establishing a PI loop using the former as control input. Assuming that good tracking properties are attainable, thrust control is achieved by providing adequate acceleration references, or *acceleration schedules*. These references are shaped so that their integral corresponds to a desirable shaft speed response. In addition, acceleration schedules may be used to introduce limit protections in critical engine variables. For more details on Ndot control, readers are referred to Link and Jaw [14] and Spang and Brown's survey [64].

Chapter 8
Engine Limit Management with Sliding Modes

Abstract This chapter develops a method to maintain critical engine variables within allowable limits, without the disadvantages associated with the standard min–max approach. Guidelines for the association of sliding mode regulators to logic max or min selectors are given, along with an $\mathcal{H}_2/\mathcal{H}_\infty$ sliding coefficient synthesis method. Simulations using the CMAPSS nonlinear engine model are included.

Sliding modes constitute a powerful tool to achieve the simultaneous objectives of robust output regulation and limit protection. Research conducted by the author in collaboration with NASA [73] indicate that many shortcomings of the standard min–max approach can be removed by replacing linear regulators with SMC. A single-input version of the *max-min/SMC* approach was available at the time this book was printed, and is presented in this chapter. Thus, the developments of this chapter assume that only one input is available to manage both regulation and limit protection objectives.

The central idea of the max–min/SMC approach is to *define sliding functions as the difference between a limited variable and its permissible limit*. One of such functions is defined for each limited output, in addition to a sliding function defined for the main regulated output (fan speed in the GTE problem). Recalling from Chap. 6 that since convergence of the s variable to zero is one-sided, if follows that outputs will not cross their limits when their corresponding SMC regulator is active.

Remarkably, the technique also assures that outputs will not cross their limits even when some other regulator is active. This represents a significant improvement over the min–max arrangement with linear regulators of Chap. 7, where transient limit protection cannot be guaranteed.

Establishing stability is of the utmost importance for the development of new control laws. In the max–min/SMC approach, asymptotic stability is guaranteed, ensuring that suitably-defined error states converge to zero. The approach constitutes a *hybrid dynamical system*, in that discrete variables exist that interact with the continuous system state. The relevant discrete variable in the max–min/SMC

H. Richter, *Advanced Control of Turbofan Engines*, DOI 10.1007/978-1-4614-1171-0_8,

approach is q, the index of the currently active SMC regulator. This variable takes on integer values, reflecting the number of regulators being implemented. A notion of stability must also be considered for q. The max-min/SMC approach has the property that q undergoes a finite number of transitions before it settles at a steady value.

The limit-preserving and stability properties of the max–min/SMC approach are ensured by following simple guidelines for associating regulators with the max or min selectors, and then tuning each SMC regulator independently. That is, a *separation property* applies, akin to the well-known property of linear observers used in combination with linear state feedback control [65]. Note that the sliding coefficients corresponding to the SMC limit regulators are no longer design freedoms: they are defined by the C matrix of each limited output. Because of this, the technique is applicable to *minimum-phase outputs* only, since the eigenvalues of the matrix defining sliding mode dynamics coincide with the zeroes of the transfer function from input to limited output. The reader may wish to re-visit Chap. 3, where the effects of right-half plane zeroes are discussed. Thus, limit regulator design entails the selection of switching gains η and boundary-layer parameters ϕ. The sliding coefficients corresponding to the main SMC regulator are design freedoms, as elaborated below. Basic design is conducted by choosing these coefficients on the basis of the main output regulation task alone. An advanced design technique is also possible, where interaction between main and limit regulators is addressed in a mixed $\mathcal{H}_2/\mathcal{H}_\infty$ synthesis framework similar to that of Chap. 4.

The detailed stability argument for max–min/SMC is involved and out of the scope of this book. Interested readers are referred to Richter [59] for a complete mathematical proof. Here, the control law is developed and the salient stability and limit-preservation properties are described.

8.1 System Description, Assumptions and Control Objectives

The architecture of max–min/SMC is the same as the max–min arrangement of Chap. 7, with the linear regulators replaced by sliding mode regulators, as illustrated in Fig. 8.1.

Due to the requirement that the main output be precisely regulated, integration is used at plant input. The regulators, thus, provide control input rates. Letting $L = \{1, 2, ...l\}$ and $H = \{l + 1, l + 2, ...h\}$, the max–min selection law is expressed as

$$u_r = \max_{k \in H} \left\{ \min_{j \in L} \left\{ u_{\mathrm{rj}} \right\}, \quad u_{\mathrm{rk}} \right\}, \tag{8.1}$$

where u_{rj} are the min-linked regulator outputs and u_{rk} are the max-linked regulator outputs. Some studies characterizing the behavior of related schemes have appeared

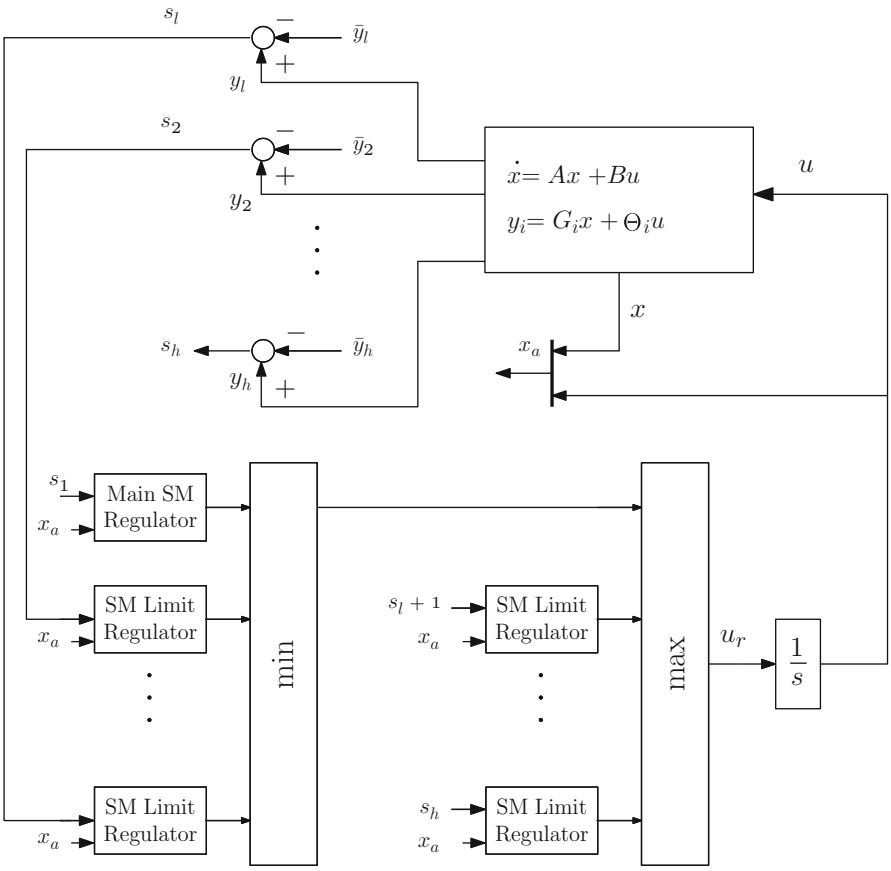

Fig. 8.1 Max-Min/SMC arrangement

in the research literature [66–68]. Recently, the (nonasymptotic) stability of this particular scheme under linear regulators has been analyzed by Johansson [33] using piecewise-quadratic Lyapunov functions. Even for linear regulators, a complete characterization of closed-loop behavior that includes essential issues such as determining which regulator will be active at steady-state or how to design the regulators to address performance requirements does not exist in the open literature. Limit protection is an indispensable consideration in the GTE problem; however, few works addressing the max–min arrangement have appeared [64, 69]. Of particular importance is the observation that limit regulators may become active even when the auxiliary outputs are far from their limits, causing a degradation in the response of the main output due to an overriding control objective [69]. Also, as established in Chap. 7, the max–min architecture with linear regulators does not ensure transient limit protection, a less-than-desirable feature.

We consider linearized models for the engine, with a single control input u (typically fuel flow). Considering that the controller includes integral action, the state-space description of the design plant is:

$$\dot{x} = Ax + Bu + \Gamma w \tag{8.2}$$

$$\dot{u} = u_r, \tag{8.3}$$

where x is n-by-1 and u and u_r are scalars. The above model captures the effect of uncertainties and exogenous inputs through vector w and its input matrix Γ, of compatible dimensions. Assume that a set of outputs is defined as

$$y_i = G_i x + \Theta_i u \tag{8.4}$$

for $i = 1, 2, ...h$, with G_i a 1-by-n vector and Θ_i a scalar.

We make three key assumptions: A is nonsingular, $\Theta_i \neq 0$ and matrices $A_{\text{eq}.i}$ defined in (8.5) have eigenvalues with negative real parts for $i = 1, 2, ...h$.

$$A_{\text{eq},i} = A - \frac{BG_i}{\Theta_i}. \tag{8.5}$$

The assumption on $A_{\text{eq},i}$ is equivalent to the requirement that the outputs defined by (8.4) are minimum-phase relative to the state-space system of (8.2). Note that when A contains a zero eigenvalue, the corresponding integrator can be factored out from the transfer function between u and y_i, resulting in a nonsingular A. The input integrator is not implemented explicitly as part of the control law. When $\Theta_i = 0$ for some i, further modifications are required [59].

8.1.1 Control Objectives

Without loss of generality, let y_1 be the output whose setpoint is to be transferred with zero steady-state error. This must be achieved under constraints of the form $y_k \leq \bar{y}_k$ and $y_l \geq \bar{y}_l$, where k are the indices of the upper-limited outputs and l are the indices of the lower-limited outputs. In addition, usual transient response specifications apply for the design of the main output regulator.

8.1.2 Sliding Mode Control Laws

Define sliding variables as

$$s_i = y_i - \bar{y}_i \tag{8.6}$$

for $i \in L \cup H$, where $\bar{y}_i = G_i \bar{x}_i + \Theta_i \bar{u}_i$. The reference variables $\bar{x}_i$ and $\bar{u}_i$ are selected to be equilibrium pairs, that is, so that $A\bar{x}_i + B\bar{u}_i = 0$. The standard SMC control law is obtained by requiring that $s_i = 0$ in finite time (reaching

phase). Beyond the reaching phase, $s_i = 0$ must become invariant (sliding phase). The system then evolves with reduced-order dynamics matching the zero dynamics associated with output s_i. Thus, a minimum-phase assumption is required. For a single SMC regulator (fixed i), the control law given below in (8.7), where η_i is a positive constant, forces the function $\frac{1}{2}s_i^2$ to have derivative $s_i\dot{s}_i = -\eta_i \text{ sign } (s_i)$. implying that the set $s_i = 0$ is reached in finite-time, with subsequent invariance.

$$u_{\text{ri}} = -\frac{1}{\Theta_i}\left(G_i(Ax + Bu) + \eta_i \text{ sign } (s_i)\right) \tag{8.7}$$

In view of the definition of s_i, a limit regulator, if operated alone, causes its corresponding limited output to attain the limit value in finite time without overshoot. Under the max–min selection logic, the closed-loop system is given by (8.2), (8.3), (8.4), (8.6), (8.7), and (8.1). The controller implements (8.6), (8.7), (8.1), and (8.3).

8.2 Behavior Under a Fixed Regulator

Let i and j be two fixed regulator indices and define the augmented state as $x_a \triangleq [x^T|u]^T$, let $\bar{x}_{\text{ai}} = [\bar{x}_i^T|\bar{u}_i]^T$ and define the augmented state relative to i as $\tilde{x}_a \triangleq x_a - x_{\text{ai}}$. Using this definition, it is straightforward to derive the following identities pertaining to system behavior under the control law of (8.7):

$$\dot{\tilde{x}}_a = A_i\tilde{x}_a - \frac{1}{\Theta_i}B_i\eta_i \text{ sign } (s_i) \tag{8.8}$$

$$s_j = J_j\tilde{x}_a + \Delta_{j,i} \tag{8.9}$$

$$\dot{s}_{j|i} = \Theta_j\left(\Gamma_{j,i}\tilde{x}_a - \frac{\eta_i}{\Theta_i}\text{ sign } (s_i)\right). \tag{8.10}$$

where:

$$A_i = \begin{bmatrix} A & B \\ -\frac{G_i}{\Theta_i}A & -\frac{G_i}{\Theta_i}B \end{bmatrix}, \quad B_i^T = [0_{1\times n}|1] \tag{8.11}$$

$$J_j = [G_j|\Theta_j], \quad \Delta_{j,i} = J_j(\bar{x}_{\text{ai}} - \bar{x}_{\text{aj}}) \tag{8.12}$$

$$\Gamma_{j,i} = \left[\frac{G_j}{\Theta_j} - \frac{G_i}{\Theta_i}\right]\left[A\middle|B\right] \tag{8.13}$$

The notation $\dot{s}_{j|i}$ is interpreted as "the derivative of s_j when i is the active regulator". When $i = j$, we simply write $\dot{s}_i$. Note that $\Delta_{i,i} = 0$ and $\Gamma_{i,i} = 0$ for $i \in L \cup H$. It follows from standard sliding mode theory that for each i, the

spectrum of A_i is formed by the eigenvalues of $A_{\text{eq},i}$ from (8.5) and zero. The closed-loop system resulting from applying the input of (8.7) to system (8.2), (8.3) is more conveniently described in terms of the derivatives of the s variables, as before, and the *rate* of x. In fact, define $X_r \triangleq \dot{x}$. The closed-loop system dynamics are expressed as

$$\dot{X}_r = A_{\text{eq},i} X_r - B \frac{\eta_i}{\Theta_i} \text{ sign } (s_i) \tag{8.14}$$

$$\dot{s}_{j|i} = \Theta_j \left[\left(\frac{G_j}{\Theta_j} - \frac{G_i}{\Theta_i} \right) X_r - \frac{\eta_i}{\Theta_i} \text{ sign } (s_i) \right]. \tag{8.15}$$

The rate system is a convenient description, since $A_{\text{eq},i}$ characterizes the dynamics of the sliding mode, facilitating the description of asymptotic properties.

8.2.1 Determination of the Steady Regulator Index

Define a switching function $q(x, u)$ with values in $L \cup H$. The minimum (min), maximum (max) switching functions are expressed by (8.16) and (8.17), respectively.

$$q_{\min} = \underset{i \in L}{\arg\min} \{u_{\text{ri}}\} \tag{8.16}$$

$$q_{\max} = \underset{j \in H}{\arg\max} \{u_{\text{rj}}\}. \tag{8.17}$$

When the above equations yield nonunique values, an assignment is made according to a predefined arbitrary rule. For the remainder of this chapter, $q_{\min} = \min(i, j)$ and $q_{\max} = \min(i, j)$ are assumed whenever $u_{\text{ri}} = u_{\text{rj}}$. When a max–min arrangement is used, it is assumed that the min preselection is applied to the first port of the max selector, so that the min input is used in case of equality with the max preselection. These assumptions will be referred to as *default index assumptions*.

As done for the max–min arrangement for linear regulators of Chap. 7, a procedure to determine the steady regulator index is developed next.

Under the min switching law, system (8.2), (8.3) has a unique equilibrium point at $(\bar{x}_{i^*}, \bar{u}_{i^*})$, where $i^* \in L$ is the index such that $\frac{\text{sign } (\Delta_{j,i^*})}{\Theta_j} \leq 0 \; \forall j \in L$.

Given system parameters, it is straightforward to compute the terminal regulator index. All $\Delta_{j,i}$ combinations are computed. For the min law, an index i^* is sought that satisfies $0 \leq -\frac{\eta_j}{\Theta_j} \text{ sign } (\Delta_{j,i^*})$ for all $j \in L, \; j \neq i^*$.

Under the max switching law, system (8.2), (8.3) has a unique equilibrium point at $(\bar{x}_{i^*}, \bar{u}_{i^*})$, where $i^* \in H$ is the index such that $\text{sign}\,(\Delta_{j,i^*}) \geq 0\ \forall\, j \in H$. This index is termed *terminal regulator index*.

The determination of the terminal index for the max and max–min switching laws is presented next.

Under the max–min switching law of (8.1), system (8.2), (8.3) has a unique equilibrium point at $(\bar{x}_{i^*}, \bar{u}_{i^*})$, where $i^* \in L \cup H$ is the index satisfying condition (8.18):

$$0 \geq -\frac{\text{sign}\,(\Delta_{k,i^*})}{\Theta_k} \quad \forall k \in H \tag{8.18}$$

and either condition (8.19) or condition (8.20):

$$0 \leq -\frac{\text{sign}\,(\Delta_{j,i^*})}{\Theta_j} \quad \forall\, j \in L \tag{8.19}$$

$$0 > \min_{j \in L} \left\{ -\frac{\text{sign}\,(\Delta_{j,i^*})}{\Theta_j} \right\}. \tag{8.20}$$

When condition (8.19) is satisfied, the terminal regulator index $i^* \in L$. Otherwise, condition (8.20) is satisfied and $i^* \in H$.

A simple algorithm to identify the ending regulator i^* in the max–min case follows:

1. Assume that $i^* \in L$ and take $i^* = 1$.
2. Check condition (8.18). If true, check condition (8.19). If true, i^* is the ending regulator. If not, take the next $i^* \in L$ and re-check.
3. If the final regulator is not found in L, repeat the above steps, checking condition (8.20) instead of (8.19).

8.3 Summary of Stability Properties

In [59], a proof of global asymptotic convergence to the equilibrium point $\bar{x}_{\text{ai}}$ is developed that relies only on the assumptions stated at the outset. The proof is based on attractiveness properties of each individual sliding set, together with considerations about the geometry of the regions of $\mathbb{R}^{n+1}$ in which each regulator is active under any of the min, max or max-min switching logic. Here, the relevant stability properties are summarized, omitting the lengthier proofs. The reader is referred to [59] for detailed proofs.

8.3.1 Stability: Min or Max Switching

All trajectories of System (8.2), (8.3) under control input (8.7) and the min switching law converge asymptotically to the unique equilibrium point x_{ai^*}.
The property $\max\{u_{ri}\} = -\min\{-u_{ri}\}$ can be used to infer stability for the max case using the proof of the min case.

8.3.2 Stability : Max–Min Switching

The max–min case requires additional analysis, as index selection cannot be expressed in terms of min only. However, the property $\max\{a_k - b_j\} = \max\{a_k\} - \min\{b_j\}$ for any two collections of numbers $\{a_k\}$ and $\{b_j\}$ proves useful in reducing the proof to the already-studied min and max cases. An important property of the max–min arrangement is that there exists a finite time after which switching is restricted to happen either among the min or the max selectors, whichever group contains the terminal index. In what follows, and without loss of generality, it is assumed that the terminal regulator index belongs to the min set, that is, $i^* \in L$.

All trajectories of System (8.2), (8.3) under control input (8.7) and the max–min switching law converge asymptotically to the unique equilibrium point x_{ai^*}. Moreover, the total number of switchings from the L set to the H set is at most equal to the number of regulators in the H set.

8.4 Invariance Properties: Limit Protection

The results of this section show that the min, max, and max–min designs actually maintain outputs within limits. In summary, it will be shown that *when the min switching law is used alone, outputs whose* Θ *is positive will be protected against upper-limit violations and outputs whose* Θ *is negative will be protected against lower-limit violations. Conversely, the max switching law alone protects outputs whose* Θ *is positive against lower-limit violations and outputs whose* Θ *is negative against upper-limit violations.* A max–min scheme is used to cover additional combinations of signs of Θ and upper or lower limits.

Recalling the definitions of Sect. 7.2.1, an interval $(-\infty, b]$ is invariant for a generic real variable $z(t)$ if $\dot{z}(t) \leq 0$ at $z = b$. Similarly, an interval $[a, \infty)$ is invariant if $\dot{z}(t) \geq 0$ at $z = a$. When an interval is invariant and $z(t_1)$ belongs to the interval for some $t_1 > 0$, then $z(t)$ will remain in the interval for $t \geq t_1$. For the proposed technique to be effective, the interval $(-\infty, 0]$ must be invariant for the s_j of upper-limited variables, in view of the definition of s_j for limited output y_j. Conversely, $[0, \infty)$ must be invariant for the s_j of lower-limited variables.

8.4.1 Invariance Under Min Switching

Let y_j be a limited variable. The derivative of s_j when i is active is given by (8.15). When i is active, we must have $u_i \leq u_j$, so:

$$\frac{\dot{s}_{j|i}}{\Theta_j} = \Gamma_{j,i}\tilde{x}_a - \frac{\eta_i}{\Theta_i}\,\text{sign}\,(s_i) \leq -\frac{\eta_j}{\Theta_j}\,\text{sign}\,(s_j).$$

Noting that the inequality changes to equality for $j = i$, it is clear that $\frac{\dot{s}_j}{\Theta_j} \leq 0$ at $s_j = 0$ under *any* regulator. If $\Theta_j > 0$, upper-limit protection is guaranteed. If $\Theta_j < 0$, lower-limit protection is guaranteed.

8.4.2 Invariance Under Max Switching

Following the same reasoning used for the min case, it is clear that $\frac{\dot{s}_j}{\Theta_j} \geq 0$ at $s_j = 0$ under any regulator. If $\Theta_j > 0$, lower-limit protection is guaranteed. If $\Theta_j < 0$, upper-limit protection is guaranteed.

8.4.3 Invariance Under Max–Min Switching

One would expect that the max–min arrangement guarantee invariance of any real interval $[a, b]$ containing zero, regardless of the sign of Θ, but this is not the case. An exception occurs for s_j when $j \in L$ and the active regulator belongs to H. This lack of symmetry arises from the fact that for $q \in H$ to be active it is necessary that u_{rq} be greater than the *minimum* of all u_{rl}, $l \in L$, but not for every u_{rl} in L. In contrast, for $q \in L$ to be active, u_{rl} must be greater than *every* u_{rh}, $h \in H$. Indeed, suppose $q = i \in L$ is active and consider a variable s_j and its derivative along the boundary $s_j = 0$:

$$\dot{s}_{j|i}/\Theta_j = u_{\text{ri}} - u_{\text{rj}} - \eta_j\,\text{sign}\,(s_j)/\Theta_j = u_{\text{ri}} - u_{\text{rj}}. \tag{8.21}$$

If $j \in L$, it is necessary that $u_{\text{ri}} - u_{\text{rj}} \leq 0$, while one must have $u_{\text{ri}} - u_{\text{rj}} \geq 0$ if $j \in H$. Thus, while $q \in L$, s_j is upper-bounded by zero if $\Theta_j > 0$, and it is lower-bounded by zero if $\Theta_j < 0$. Now consider $q = i \in H$ to active. Equation (8.21) still applies. If $j \in H$, it is necessary that $u_{\text{ri}} - u_{\text{rj}} \geq 0$. Thus, while $q \in H$, all variables s_j *associated to the max selector* will be upper-bounded by zero when $\Theta_j < 0$ and will be lower-bounded by zero if $\Theta_j > 0$. The difficulty arises when considering $j \in L$ while the active regulator is in H. The difference $u_{\text{ri}} - u_{\text{rj}}$ may be positive,

negative or zero, and invariance does not apply. Fortunately, separate arguments can be made which maintain the validity of the approach under commonly found circumstances. These arguments are elaborated in the next section.

8.5 Additional Considerations

For the remainder of the article, it is assumed that regulators are assigned to selectors so as to exploit the invariance properties described above. These assignment rules have been summarized in Table 8.1.

8.5.1 Limited Output Consistency

The results of this chapter are directly applicable to setpoint changes, implying that initial and final plant states $[x^T|u]^T$ are equilibrium points. Then it is always possible to redefine variables so that the initial input u, state x, and outputs y_j are zero. Frequently, it occurs that the sign of the DC gain of the transfer functions from u to y for the limited outputs coincides with the sign of Θ. The steady plant input–output relationships have the form

$$\bar{y}_j = \Theta_j(1 - G_j A^{-1} B/\Theta_j)\bar{u}$$

for $j = 1...h$. If $1 - G_j A^{-1} B/\Theta_j > 0$, then the sign of steady input $\bar{u}$ will match that of the limit $\bar{y}_j$ when $\Theta_j > 0$ and will be of the opposite sign when $\Theta_j < 0$. This has useful implications for the behavior of min-variables when $q \in H$, where invariance was not found. The following heuristic reasoning applies: if $q \in H$ because an upper-limited variable from the max group is reaching its (positive) limit, then u will be negative, since Θ_j must be negative according to the assignment rules. Any y_i among the min-selected variables which is upper-limited will be driven away from its limit by the negative $\bar{u}$, since $\Theta_i > 0$ by the assignment rules. The same reasoning can be followed for other combinations. This behavior is confirmed in simulation.

Table 8.1 Guidelines for the association of sliding mode regulators to selectors

Limit	Sign of Θ	Selector
Upper	+	Min
Upper	−	Max
Lower	+	Max
Lower	−	Min
Regulated output	n.a.	min

8.6 Design Process

This paper deals with a single control input and a single controlled output, whose setpoint is to be changed. Of all outputs, exactly one is the controlled variable, while the rest are limited variables. The G_i and Θ_i for the limited outputs are given by the system definition and are thus not design freedoms. The fundamental assumption that they define minimum-phase outputs must hold for this technique to work, however. The designer may freely choose all switching gains η_i and the reference states $\bar{x}_i$ and $\bar{u}_i$ so that they constitute equilibrium pairs and so that they correspond to the desired setpoint for the controlled variable and to the limits $\bar{y}_i$. Next, upper-limited variables such that $\Theta_i > 0$ and lower-limited variables having $\Theta_i < 0$ are placed under the min selector, while upper-limited variables with $\Theta_i < 0$ and lower-limited variables such that $\Theta_i > 0$ are placed under the max selector, as summarized in Table 8.1.

For the purposes of showing stability, no distinction was made between regulated and limited outputs, and the sliding function for the regulated variable was defined as the difference between the output and a limit. Suppose $y_0 = C_0 x + D_0 u$ is the true system output to be regulated (i.e., fan speed). Since output setpoint regulation is equivalent to the selection of a reference pair $(\bar{x}, \bar{u})$, one may introduce a "limited" variable and associated regulator for the purpose of reaching the reference state. Without loss of generality, suppose that $y_1 = G_1 x + \Theta_1 u$ is such variable. For y_0 to reach its setpoint, $(\bar{x}_1, \bar{u}_1)$ must be chosen under the restriction that $G_1 \bar{x} + \Theta_1 \bar{u}$ equals the desired setpoint, and G_1 and Θ_1 must be chosen to satisfy the minimum-phase assumption and for good performance during the sliding mode. The regulator for y_1 is then placed under the min selector. Under nominal conditions, the designer ensures that $i^* = 1$, so that y_1 attains the commanded setpoint. This is easily accomplished by manipulation of Θ_1 and η_1, since the choices do not affect the ability to place the eigenvalues of $A_{\text{eq},1}$, nor compromise system stability.

8.6.1 MultiObjective Control: Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Feedback Gain Synthesis

Although the above alone will guarantee stability and limit protection in both transient and steady states, eigenvalue placement for $A_{\text{eq},1}$ using G_1 may be carried out under additional, performance-oriented constraints. The interaction among the controlled and limited variables may be minimized in the sense of a mixed-sensitivity $\mathcal{H}_\infty$ approach. Consider the following generic control system in state-space form:

$$\dot{x} = Ax + B_1 w + B_2 u \tag{8.22}$$

$$z_{i,\infty} = C_i x + D_{i,1} w + D_{i,2} u \tag{8.23}$$

$$z_{j,2} = C_j x + D_j u, \tag{8.24}$$

where $i = 1, 2, ..I$ and $j = 1, 2..J$ are the indices of performance outputs $z_{i,\infty}$ and $z_{j,2}$. The objective is to find a feedback gain K that:

1. Stabilizes (A, B_2) under the control input $u = -Kx$ and
2. Minimizes a weighted objective function of the form $\alpha||T_\infty||_\infty^2 + \beta||T_2||_2^2$,

where T_∞ is the closed-loop transfer matrix from the exogenous inputs w (i.e., disturbances) to the performance outputs $z_{i,\infty}$ and T_2 is the closed-loop transfer matrix from w to the performance outputs $z_{i,2}$. The weighting coefficients α and β reflect design priorities and may be set to zero. The subindices represent the infinity and 2-norms, respectively, which are commonly-used in the standard $\mathcal{H}_\infty$ and LQG problems [28, 29]:

$$||T_\infty||_\infty = \max_w \bar{\sigma}\left(T_\infty(jw)\right) \tag{8.25}$$

$$||T_2||_2^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty} \text{trace}\left(T_2(jw)T_2^*(jw)\right)\text{d}w, \tag{8.26}$$

where $\bar{\sigma}$ denotes maximum singular value and * denotes complex-conjugate transpose. These norms were used in Chap. 4 as part of a robust state feedback gain synthesis approach.

Note that the performance outputs are stacked together, allowing the designer to include multiple objectives in the norm minimization. This generic problem may be solved with additional constraints, for instance the requirement that the eigenvalues of $(A - B_2K)$ lie in certain region of the complex plane. Software tools such as the `msfsyn` function within Matlab's Robust Control Toolbox are available to solve these problems.

In the context of aircraft engine controls, w may represent actual disturbances or may be used to capture the effects of engine aging and deterioration. In fact, matrix Γ in (8.2), representing the influence of health parameter perturbations, may be directly obtained by linearization. In the context of sliding mode control, w represents an *unmatched disturbance* if Γ does not belong to the column space of B [53, 55, 70]. This means that w cannot be regarded as an additive component to the control input u. Consequently, w may not be exactly canceled out by u even if it were known or accurately estimated. Since it cannot be assumed that health parameter perturbations will be of the matched type, a reasonable design objective is to minimize the influence of w on the regulated variable.

The generic multiobjective synthesis approach described above may be directly applied to guide the selection of G_1, the sliding coefficients for the main regulator. To do this, system dynamics under an ideal sliding regime with i^* as the active regulator are considered. Under these conditions, the control input has the state-feedback form $u = -Kx$, with $K = \frac{1}{\Theta_{i^*}}G_{i^*}$. The $\mathcal{H}_2$ objective is included by considering $z_{j,2} = y_{j|i^*}$. That is, the (transient) excursions of y_j under control input $u = -\frac{1}{\Theta_{i^*}}G_{i^*}x$ are to be minimized in an $\mathcal{H}_2$-sense. For this, the transfer functions of interest are of the form

$$T_{j,2}(s) = G_j(sI - A)^{-1}B + \Theta_j. \tag{8.27}$$

Note that T_2 is formed by stacking the $T_{j,2}$ together. Also note that T_2 in the generic problem regards w as the input. Here, $B_1 = B_2 = B$ has been used to reflect the intuitive requirement that u must not unduly excite y_j.

The $\mathcal{H}_\infty$ objective is included by considering a single performance output $z_{i^*,\infty} = y_{i^*}$. The corresponding transfer function is

$$T_{i^*,2}(s) = C_0(sI - A)^{-1}B_1 + D_0, \tag{8.28}$$

where C_0 and D_0 define the regulated output. Note that $G_{i^*} \neq C_0$ and $\Theta_i^* \neq D_0$ in general.

8.7 Design Examples

8.7.1 Linearized Simulation Study

We consider the problem of changing fan speed between two setpoints with limits in T_{48}, EPR, high-pressure compressor stall margin (SmHPC), and HPC exit static pressure (Ps30). Note that no real-time sensing of the stall margin is possible, however the SM controller only requires fan and core speeds as feedback measurements. The CMAPSS model linearized at flight condition FC07 listed in Appendix B is used.

This system is open-loop stable, so the assumption that A is nonsingular is satisfied. We consider the excursions of EPR and T_{48} from the values in Table 2.3 to be upper limited at 0.35 and 400°R, respectively, while those of Ps30 and SmHPC to be lower-limited at −85 psia and −15%. The values of the limits are representative of actual engine operations. These limited outputs are defined as in (8.4) with

$$\begin{aligned} G_2 &= [0.0071 \;\; 0.0177], & \Theta_2 &= -18.4743 \\ G_3 &= [0.0244 \;\; -0.2665], & \Theta_3 &= 410.4741 \\ G_4 &= [-0.0037 \;\; 0.1599] \times 10^{-3}, & \Theta_4 &= 0.0461 \\ G_5 &= [0.0017 \;\; 0.0855], & \Theta_5 &= 25.5719, \end{aligned}$$

where indices 2, 3, 4, and 5 correspond to SmHPC, T_{48}, EPR, and Ps30, respectively. It can be verified that these outputs are minimum-phase, satisfying the minimum-phase assumption. Given a fan speed increment setpoint and the set of limits, the corresponding reference states $(\bar{x}_i, \bar{u}_i)$ are readily computed by enforcing equilibrium conditions and using the output definitions, that is

$$\begin{aligned} A\bar{x}_i + B\bar{u}_i &= 0 \\ G_i\bar{x}_i + \Theta_i\bar{u}_i &= L_i, \end{aligned}$$

where L_i are the values of the limits. Note that the reference states for the fan speed regulator are found using $C_1 = [1 \; 0]$, $D_1 = 0$, and L_1 equal to the desired setpoint

for ΔN_f. However, the fan speed regulator does not use C_1 and D_1 as sliding function coefficients, since limit preservation is not required for this variable. To find G_1 and Θ_1, we solve the $\mathcal{H}_2$ minimization with regional eigenvalue placement described in the design section. The optimization is carried out with the `msfsyn`, which is part of Matlab's Robust Control Toolbox. The eigenvalue placement region was specified as the half-space $\mathrm{Re}(s) \leq -4$. Setting $G_1 = 1$ arbitrarily, the solution returns

$$G_1 = [0.0118 \;\; 0.0026],$$

which places the eigenvalues of $A_{\mathrm{eq},1}$ at -5.26 and -4.40. The regulators are now associated with the min or max selectors according to Table 8.1. The fan speed regulator is applied to the min regulator. Since $\Theta_2 < 0$ and y_2 is lower-limited, the output limit regulator #2 is applied to the min selector. Similarly, since Θ_3 and Θ_4 are positive and y_3 and y_4 are upper-limited, the corresponding regulators are applied to the min selector. Finally, $\Theta_5 > 0$ and y_5 is lower-limited, so the regulator is associated with the max selector. Under nominal conditions, the designer wishes that #1 be the terminal regulator. Since the only controller parameters left to be specified are the switching gains $\eta_i > 0$, and because their choice does not compromise stability, tuning is straightforward and requires little or no iteration. The η gains are adjusted until $i^* = 1$ is predicted and a satisfactory response is observed in simulation. For this example, it can be verified that setting $\eta_i |\Theta_i| = 15$ for $i = 1, 2, 3, 4, 5$ results in $i^* = 1$.

Figures 8.2 and 8.3 show fan speed and auxiliary output responses to a setpoint of $\Delta N_f = 340$ rpm with all limit regulators disabled. As expected, it can be observed that y_1 attains its setpoint; however, some auxiliary variables exceed their limits. Specifically, T_{48} and SmHPC incur transient violations. EPR does not reach its limit, and Ps30 has a DC gain of the opposite sign as N_f, causing it to move away from its lower-limit. Note that the settling time for N_f is about 1 s.

Figures 8.4 and 8.5 show the responses obtained when the limit regulators are enabled. It can be seen that T_{48} and SmHPC now "ride" their limits during the transient regime. Naturally, the fan speed response will show some performance degradation in terms of transient behavior, but its ability to reach the setpoint will not be hindered, since the design ensures that $i^* = 1$ under nominal conditions. The settling time is now about 1.4 s. Note that the limit $\Delta T_{48} = 400°\mathrm{R}$ has been made very "tight" for illustrative purposes. Considering that the absolute limit on T_{48} used in realistic engine controls is close to 2,200°R, Table 2.3 indicates that ΔT_{48} could be chosen as high as 657°R. Under these conditions, the same design would result in a faster settling time.

To illustrate activation of the lower limit, suppose now that fan speed is to be reduced, that is, the setpoint is $\Delta N_f = -340$ rpm. It can be verified that $i^* = 1$ and $i_0 = 1$ still hold under the same design parameters. Figures 8.6 and 8.7 show that Ps30 reaches its lower limit and holds it for some time.

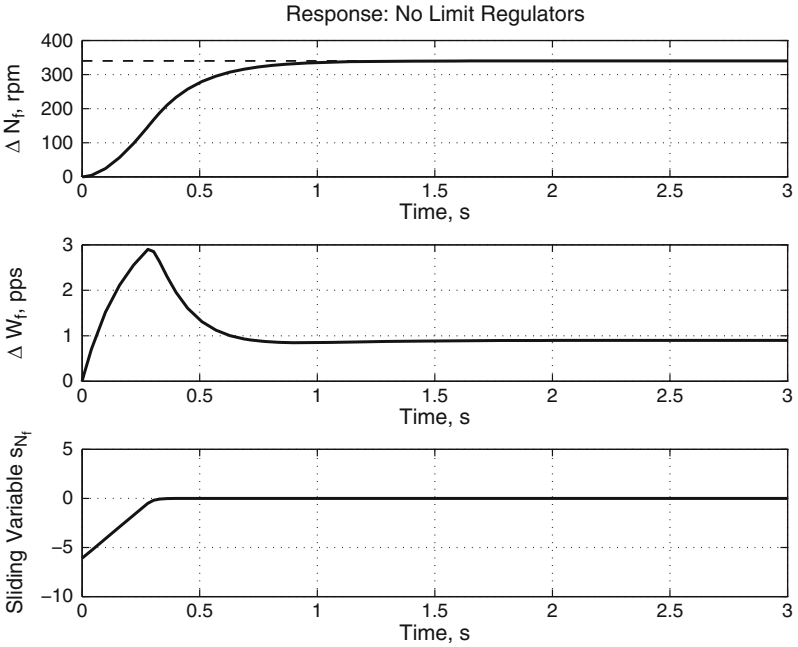

Fig. 8.2 Fan speed response with limit regulators removed: positive setpoint change

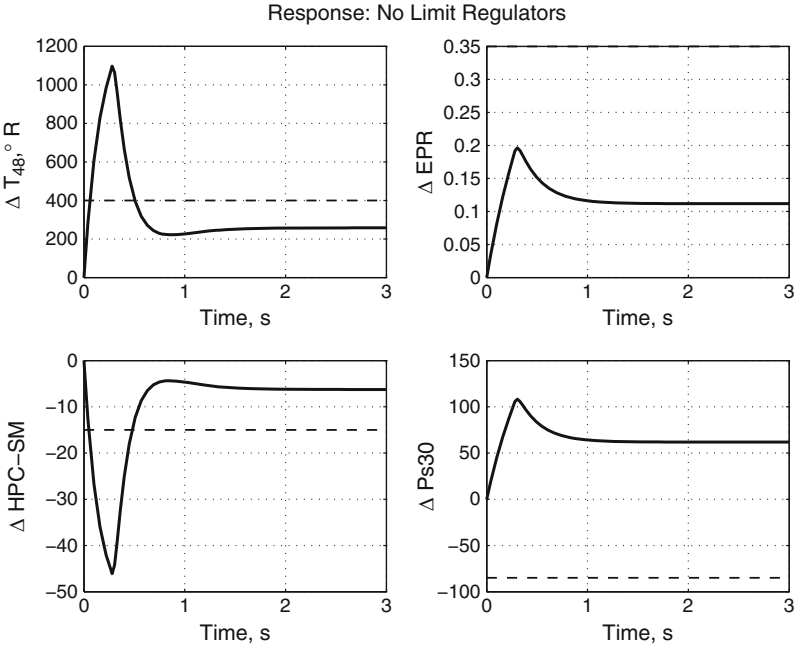

Fig. 8.3 Auxiliary output response with limit regulators removed: positive setpoint change

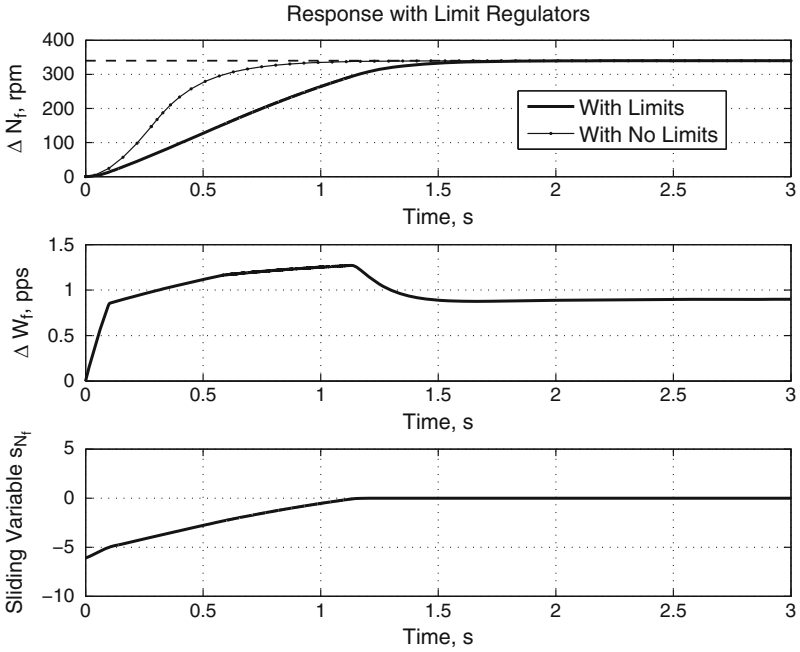

Fig. 8.4 Fan speed response with limit regulators enabled: positive setpoint change

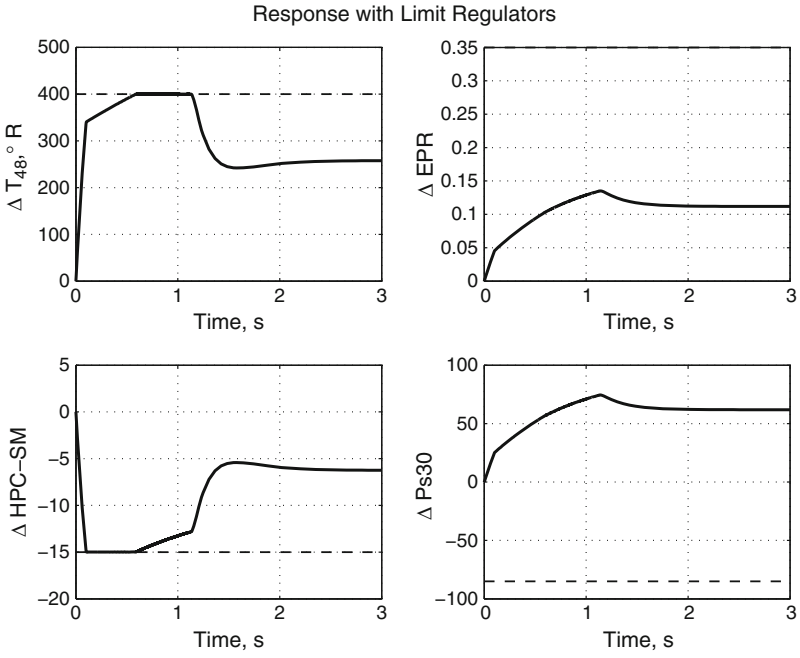

Fig. 8.5 Auxiliary output response with limit regulators enabled: positive setpoint change

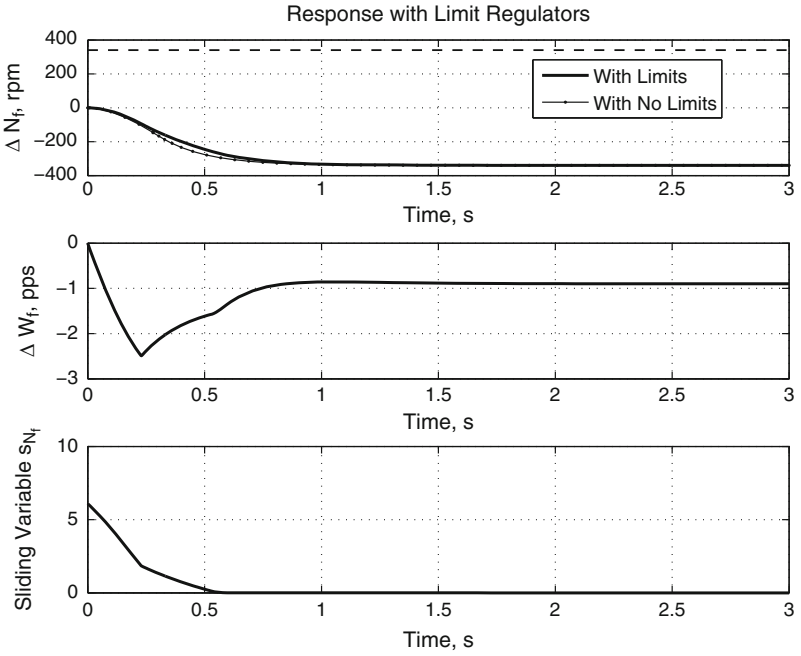

Fig. 8.6 Fan speed response with limit regulators enabled: negative setpoint change

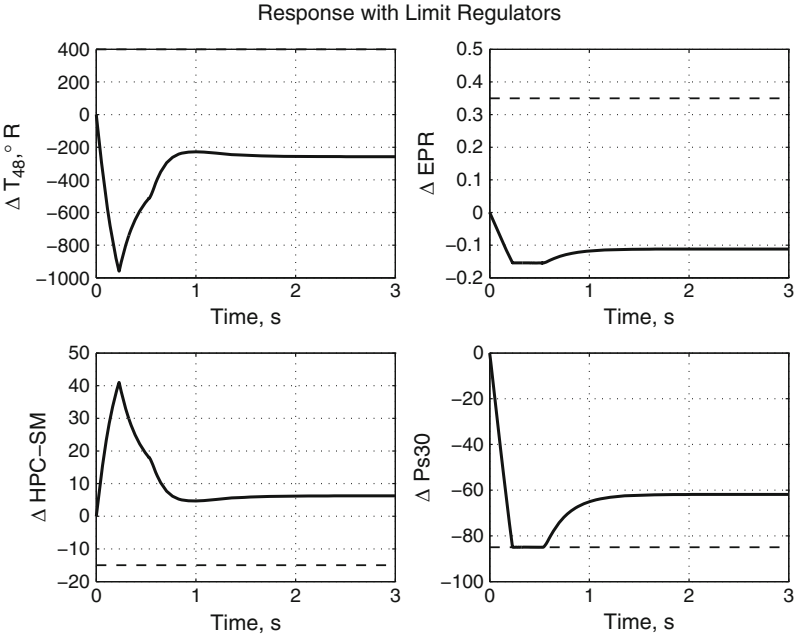

Fig. 8.7 Auxiliary output response with limit regulators enabled: negative setpoint change

8.7.2 CMAPSS Implementation: Upper Limit on T_{48}

The above example was meant to illustrate the details of the design process. The simulations used linearized models, however. The simulation studies of this section are conducted in CMAPSS-40k, and strongly suggest that the combination of sliding mode regulators and max–min selectors is robust enough to produce satisfactory responses and limit protection behavior when applied to a high-fidelity nonlinear engine simulation. In turn, the nonlinear simulation provides evidence for the feasibility of this technique to be deployed in real-time to an actual engine. Note that the control law of (8.7) carries an insignificant computational burden, especially when compared to other control strategies aiming to handle constraints, like model predictive control. In this example, minimizing the effects of health parameter changes is also included among the design objectives. A fan speed regulator and a T_{48} regulator are considered, both associated with a single min selector. The increment request for N_f is the same as in the previous example, but the T_{48} limit is reduced to 1,900°R ($\Delta T_{48} = 357$°R to study the behavior of each design under tight limits. Among the health parameter perturbations discussed in Sect. 8.1, the HPT flow modifier and the HPT efficiency modifier have the largest influence on T_{48}. They correspond to a specific Γ matrix (see (8.2)), obtained during linearization. At the flight condition considered in this example matrix Γ is given by:

$$\Gamma = \begin{bmatrix} -505.4 & 152.6 \\ 4325.2 & -1030.5 \end{bmatrix}.$$

Step changes from 0% to 3% in each component of w will be considered in the CMAPSS example, simulating the effect of a sudden fault. A comparison is to be made between the proposed design and the max–min strategy with linear regulators, which is the default in CMAPSS. Each linear regulator (including that for the regulated output) is designed using the so-called KQ technique due to Edmunds [27]. In summary, each regulator is restricted to be of the lead-lag type. The pole of the regulator is arbitrarily set, and the zero and the gain are found by model-matching optimization, whereby the closed-loop system is required to meet a target closed-loop bandwidth w_b and damping ratio ζ for a pair of dominant poles. Although this classical technique may be satisfactory for independent loops, it does not incorporate any information about the interaction of the regulators through the max–min selector, nor does it use information about the values of the setpoints or limits. As a result, poor performance may be observed, even when a limited variable stays far from its limit. For illustrative purposes, a KQ design was conducted using $\zeta = 0.7$ and $w_b = 4$ rad/s, which according to classical compensation design techniques [26] should produce a settling time of 1.45 s assuming that the regulator is active at all times. These parameters and a real pole at $s = -20$ were used for both N_f and T_{48} regulators. The resulting compensators are

$$K_{N_f}(s) = \frac{0.1196s + 0.1868}{s + 20}$$

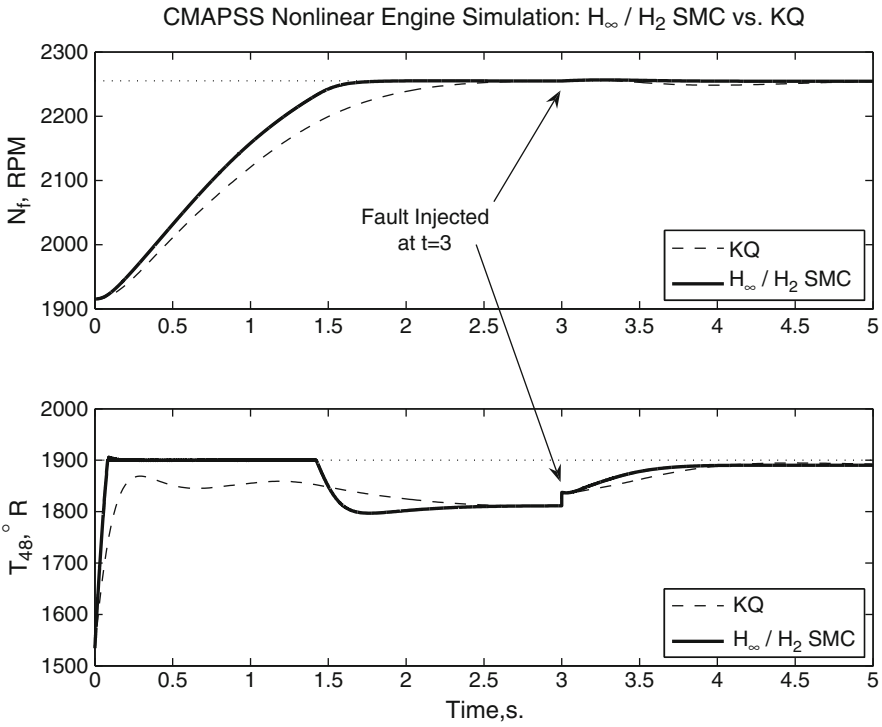

Fig. 8.8 CMAPSS simulations: Fan speed response with T_{48} limit regulator and simulated fault

$$K_{T_{48}}(s) = \frac{-0.01017s + 0.1073}{s + 20}.$$

Note that the KQ design introduces a right-half plane zero in the T_{48} loop, raising concerns about the general validity of Edmunds' approach. Separately, a sliding mode max–min design with mixed $\mathcal{H}_2$ norm minimization was carried out, following the same steps as in the previous example, including performance output z_∞ to capture the effects of w on N_f. The regional pole placement constraint was maintained, this time using $\mathrm{Re}(s) \leq -5$. Finally, $\alpha = 1$ and $\beta = 1$ were used. The resulting sliding coefficient vector for the N_1 regulator was $G_1 = [0.0168 \;\; 0.0014]$, with Θ_1 arbitrarily set to 1. The eigenvalues of $A_{\mathrm{eq},1}$ are $-5.0112 \pm 1.5396i$. The switching gains were set at $\eta_1|\Theta_1| = \eta_2|\Theta_2| = 15$. The resulting controller was implemented in CMAPSS, and a sample comparison simulation was run. The results are summarized in Figs. 8.8 and 8.9. It is clear that the new design outperforms KQ by a large amount in terms of fan speed response, maximization of the available limit, and low sensitivity to a simulated fault. Note that the settling time with the proposed design is 40% shorter than the one obtained with the KQ design. The example also indicates that a design based on a linearized plant is adequate for deployment to the nonlinear model, although scheduling or gain adaptation is likely

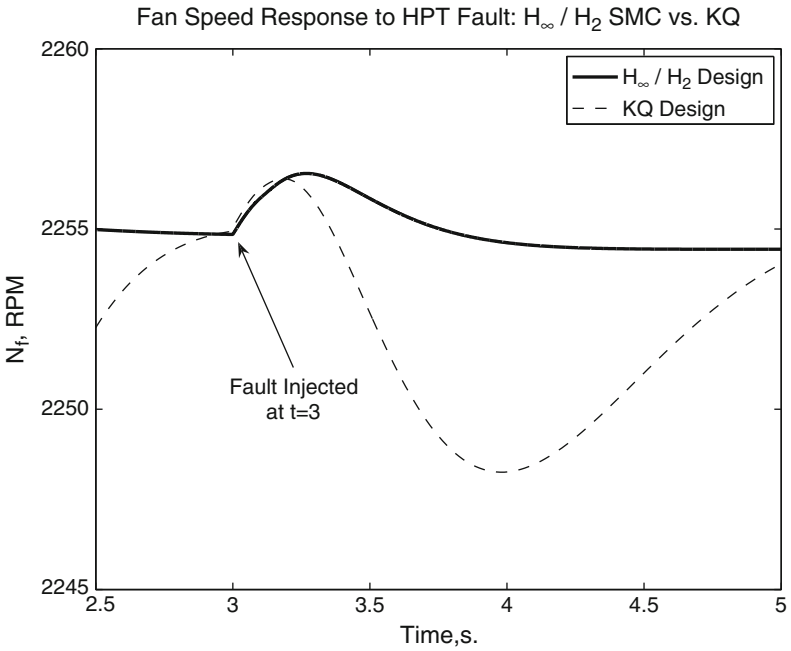

Fig. 8.9 Detail of Fig. 8.8: Response to fault input

to be necessary to cover the whole flight envelope or to accommodate larger setpoint changes.

8.7.3 CMAPSS Implementation with Multiple Limit Regulators

In this example, a more realistic implementation of the max/min SMC approach is described. As in the simulation example, T_{48} and EPR are regarded as upper-limited and Ps30 as lower-limited. A limit regulator cannot be directly implemented for the stall margin, however, since this output is not sensed in the actual engine. A related variable usually referred to as Φ can be used instead to achieve a minimum stall margin requirement. This variable is defined as the ratio of fuel flow rate to static HPC outlet pressure Ps30:

$$\Phi = \frac{W_F}{\text{Ps30}}.$$

An allowable range for the value of Φ is typically used to calculate corresponding allowable values of W_F on the basis of the current value of Ps30 [71]. That is, the value of W_F calculated by the control system is passed through a saturation function with variable limits which are determined from the minimum and maximum values allowed for Φ and the current value of Ps30. In this example, however, we consider

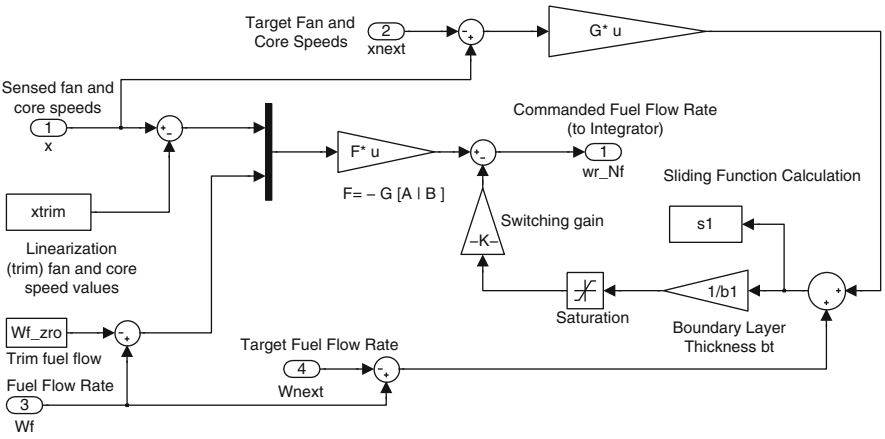

Fig. 8.10 CMAPSS simulation diagram for max–min/SMC: Main output regulator

only an absolute lower limit on Φ, enforced by an SMC limit regulator. Since the D coefficient of the linearized output $\Delta\Phi$ is positive, the limit regulator is associated to the max selector. Note that if an upper limit is specified for Φ in addition to the lower limit, an extra regulator must be associated to the min selector.

The example corresponds with a burst and chop maneuver starting at FC07. A fan speed demand is created that corresponds to a step increase in TRA from 60 to 100°. The opposite TRA change is used for the chop portion. The sliding coefficients for the main output regulator are determined using the $\mathcal{H}_2/\mathcal{H}_\infty$ approach. Fan speed is regarded as the z_∞ performance output and z_2 includes all limited outputs. Equal weights for the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ objectives are specified, and the target eigenvalue region is taken as a disk centered at -8 with radius 2.

The sliding coefficients returned by `msfsyn` are

$$G = [0.0452 \quad -0.0010],$$

which places the poles of A_{eq} at -9.99 and -6.00. The design is completed by specifying a set of switching gains that provide an adequately fast response while ensuring that the fan speed regulator is active at steady state. The switching gain for the main regulator was chosen as $\eta_1 = 1$, while $\eta_j = 15|\Theta_j|$ was chosen for all limit regulators. Boundary layer thickness parameters must be chosen according to orders of magnitude projected for the s variables. $\phi = 0.5$ was chosen for the main SMC regulator, together with $\phi_2 = 0.01$, $\phi_3 = \phi_4 = \phi_5 = 0.1$.

Figures 8.10 and 8.11 show the Simulink implementations of the main and limit regulators, where the T_{48} regulator has been used as an example of the latter. Figure 8.12 shows the behavior of the system state and main sliding function. The fan speed response has a fast settling time of about 0.8 s.

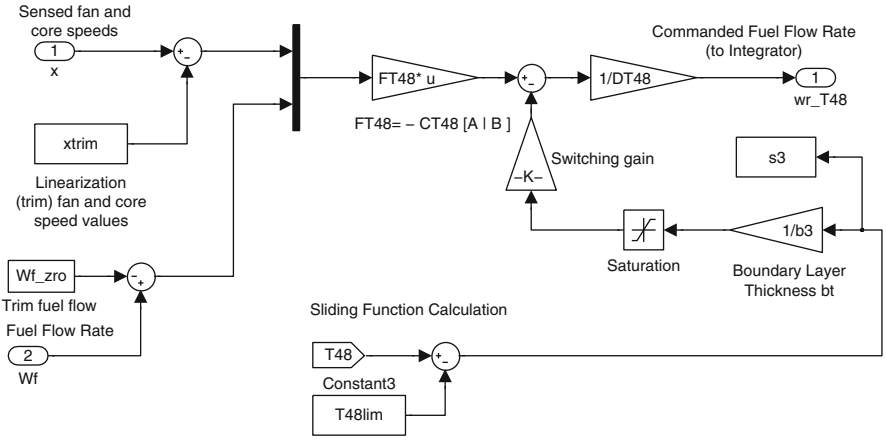

Fig. 8.11 CMAPSS simulation diagram for max–min/SMC: Limited output regulator

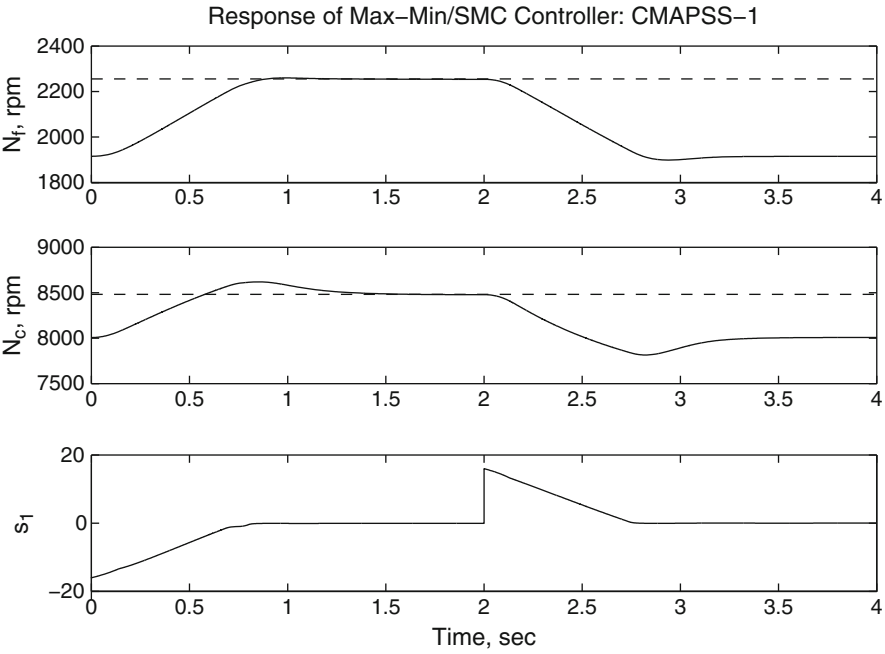

Fig. 8.12 CMAPSS simulation of max–min/SMC with $\mathcal{H}_2/\mathcal{H}_\infty$ tuning: State variable and sliding function responses

As shown in Fig. 8.13, T_{48} tends to peak during the burst transient, but SMC limit regulator effectively maintains the variable at the exact value of the limit (2,175°) as long as necessary. Similarly, Φ tends to undershoot pronouncedly during the chop

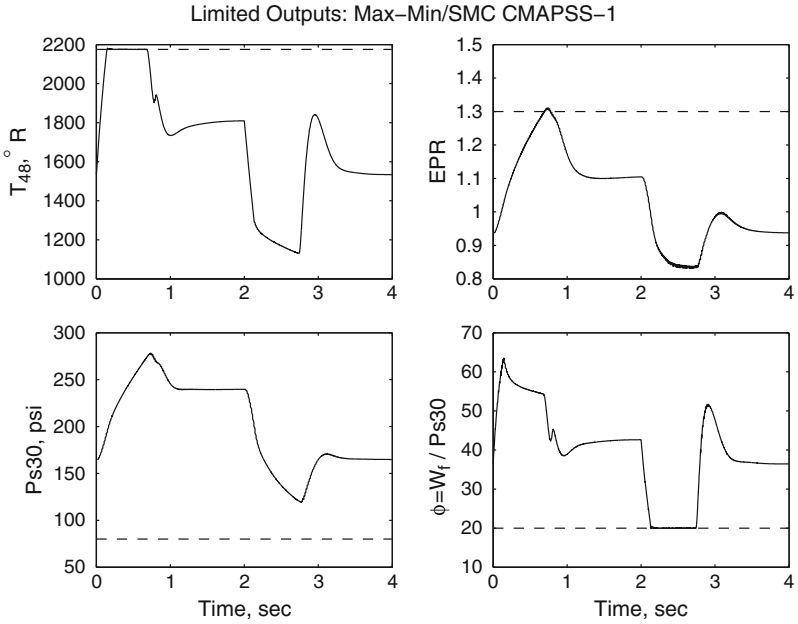

Fig. 8.13 CMAPSS simulation of max–min/SMC with $\mathcal{H}_2/\mathcal{H}_\infty$ tuning: Limited output responses

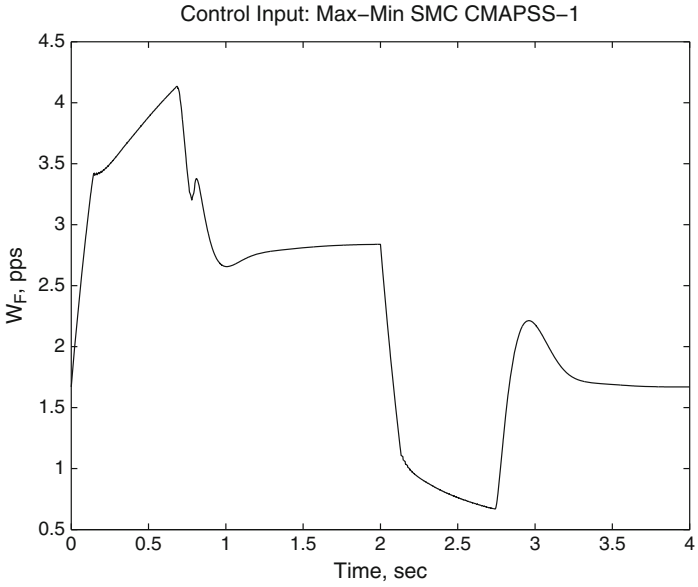

Fig. 8.14 CMAPSS simulation of max–min/SMC with $\mathcal{H}_2/\mathcal{H}_\infty$ tuning: Fuel flow input

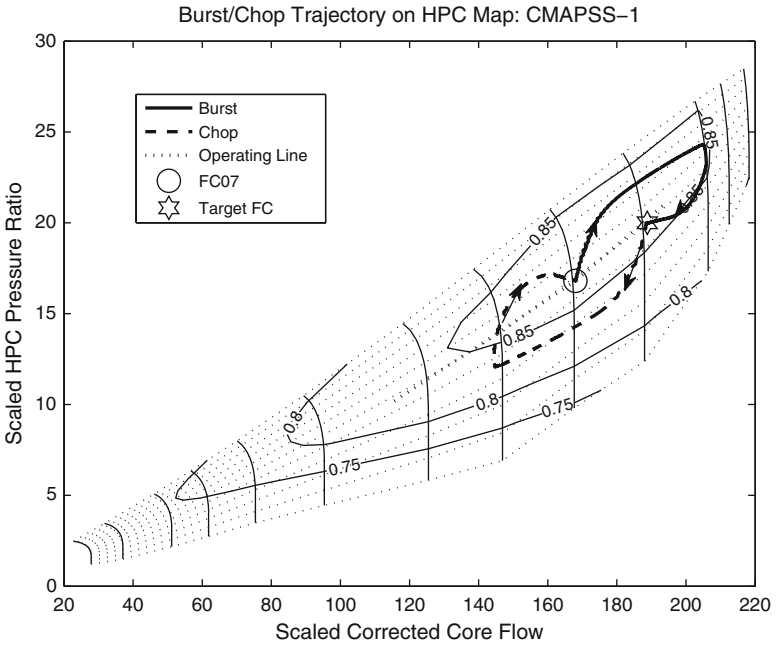

Fig. 8.15 CMAPSS simulation of max–min/SMC with $\mathcal{H}_2/\mathcal{H}_\infty$ tuning: Representation of burst/chop trajectories on HPC map

transient, but is effectively held at the limit of 20 by its regulator. An upper limit of 1.3 was specified for EPR. Since the designer has no control over the sliding mode dynamics for the limit regulators, small oscillations around the limit value may be observed in a nonlinear engine simulation. This is the case with EPR: even if this regulator is forced to remain active at all times, convergence to $s = 0$ is not one-sided, but contains some oscillation. This explains the slight overshoot observed for EPR in Fig. 8.13. Figure 8.14 shows the fuel flow input produced by the control system. No significant chattering is detected, despite the high regulation accuracy of this system. Finally, the burst and chop sequence has been represented in the HPC map in Fig. 8.15.

8.8 Summary

The above CMAPSS implementations demonstrate the effectiveness of the max–min arrangement with SM regulators. The designer can use this technique to achieve a balance between speed of response and allowable limits for critical engine variables. Even with constant control gains, limit relaxation will be reflected in faster responses, and conversely, the main output response will become slower if

limits are made more restrictive. This feature is highly desirable for the development of *resilient aircraft control systems*, where the engines feature aggressive control modes reserved for emergency maneuvers. In emergencies, extending engine life becomes secondary to achieving enhanced thrust response. Indeed, recent aviation safety research [69,72] indicates that thrust response times determine the feasibility of certain emergency maneuvers where the propulsion system is used for flight control. In one scenario where all rudder control has been lost, the pilot commands different levels of thrust to the engines to achieve a yawing moment. Studies indicate that the control system must feature fast thrust response modes to be used in emergencies. In these situations, enhanced response must be favored over engine durability, while still guaranteeing component safety.

On the premise that the standard max–min architecture is used for both normal and enhanced responses, there are essentially two ways of obtaining faster thrust responses: (a): redesigning the regulators for larger closed-loop bandwidths; and (b): relaxing the protective limits on variables which tend to peak as thrust response is made faster. Among the variables displaying such peaking are turbine outlet temperature, which peaks during acceleration, stall margin, which tends to undershoot during acceleration, and combustor pressure, which tends to undershoot during deceleration. Unfortunately, the max-min arrangement with linear regulators introduces an undesirable relationship between design bandwidths, limit settings and the achieved speed of response. This observation was first made by Litt [69] and confirmed by the author in a simulation study [73]. The same study demonstrates that the max-min arrangement with SM regulators removes this limitation.

The technique presented in this chapter requires the specification of augmented state references, which includes target steady values for the actuators, since integral control is used. As mentioned in Sects. 4.7.1 and 6.3, the steady map of the nonlinear engine must be used to pre-calculate such references. This drawback is aggravated when health parameter changes occur, since the steady map depends on these uncertain parameters. Note, however, that real-time implementations of the SM limit regulators do not require state references, due to the definition of the sliding function. Only the limit setting and a real-time output measurement are needed to calculate the sliding function.

Chapter 9
Engine Limit Management with Model Predictive Control

Abstract This chapter introduces the concept of model predictive control and develops basic prediction formulas based on linear state-space models. The constrained optimization problem is formulated using compact matrix formulas suitable for incorporation in Matlab's quadratic program solver. Model predictive control is then applied to the engine control problem to address input and output constraints. The chapter also discusses computational complexity and approaches aimed at its reduction. Matlab code and simulations using the CMAPSS nonlinear engine model are included.

Model predictive control (MPC) is a technique conceived to incorporate constraints explicitly at the outset of the control law derivation process. As its name implies, MPC relies on predictions generated using a mathematical model of the plant. In this regard, the philosophy behind the construction of MPC laws is radically different from traditional error-feedback control approaches. Instead of producing a control action in response to the current and past errors, MPC evaluates the results of applying candidate control input sequences, selecting the sequence that minimizes a performance measure. Only the first term of the sequence is then injected to the plant, and the process is then repeated. The effects of applying a candidate control input sequence must be evaluated through a performance figure whose value depends on the predicted plant response.

System operation under MPC resembles a game of chess, where the player analyzes several moves in advance, using the rules of the game and his/her perception of the opponent's playing strategy as a "model" that allows him/her to make predictions. Piece values and strategic considerations lead the player to determine an optimal course of events for the next few moves. Naturally, the player is allowed only one turn, so only the first term of the optimal sequence is implemented. In a control system, the opponent's move is analogous to the reaction of the plant to a control input. The player must then repeat his/her predictions and select a new optimal move, given the current situation.

An insightful comparison between MPC and error-based feedback (say, PI control) can also be made by analogy with the act of steering a car on a winding road, with limited visibility. Normal driver actions correspond to MPC: the visible

H. Richter, *Advanced Control of Turbofan Engines*, DOI 10.1007/978-1-4614-1171-0_9,

portion of the road's centerline provides predictions used by the brain to make anticipatory corrections to the steering wheel and the speed of the vehicle. Note that visible portion of the road constitutes a *moving*, or *receding horizon*, a key concept associated with MPC. In contrast, error-feedback control is analogous to using the rearview mirror as the only source of visual information: the driver reacts to errors that have already occurred, such as deviating excessively from the centerline.

Because of its anticipatory character, MPC is well suited to handle constraints, since only sequences predicted to satisfy the constraints are considered as suitable control candidates. Since candidates are evaluated through a performance function (typically an LQR-like cost), good transient performance is ensured in addition to constraint satisfaction. Constraints are easily formulated to capture limits on the state variables and output and input magnitudes and rates.

Despite its many attractive features, MPC has a significant shortcoming in its computational burden. The calculations involved in prediction and optimal sequence determination must be completed quickly enough, so that the plant state has not deviated excessively from the value it had at the beginning of prediction. The requirement of fast control computation is not exclusive to MPC, of course. Digital control implementations use sample periods significantly smaller than the fastest time constant found in the design plant. In GTE control, sample times near 0.01 s are typical. The digital control system must complete various operations in less than the sample time: reading sensors, calculating the control law, commanding the control inputs to the actuators and other operations. A safe *latency* time must also be allowed where the control system is in a wait state until the next sample instant comes up. This means that the time allotted for control law calculations is a small percentage of the sample period. As it will be seen below, the MPC calculation involves a numerically intensive quadratic constrained optimization problem, known as a *quadratic program*.

Studies have been conducted to determine the applicability of MPC to the GTE control problem, including some by the author of this book aimed at reducing computational complexity. Examples of this research include the work of Brunell and co-workers [74], De Castro [75], Richter [76], and Van Essen [77].

In this chapter, the standard state-space formulation of MPC for linear plants is presented, maintaining the exposition at an introductory level. Basic examples and Matlab code are provided for nominal linear plants, followed by realistic CMAPSS simulations. Readers interested in a more comprehensive and in-depth exposition of MPC techniques are referred to Camacho and Bordons [78] or Rossiter [79].

9.1 Digital Control Systems and Zero-Order Hold Equivalents

The MPC strategy presented in this chapter is based on a linear *discrete-time* model of the plant. A discrete-time equivalent of the continuous-time linear plant is a set of difference equations describing the changes of the *samples* of the state variable

upon application of a sequence of constant control inputs. The samples of the continuous-time state variable $x(t)$ are denoted $x(k)$, and the sequence of constant control inputs is denoted $u(k)$. Samples are separated in time by the *sampling period* T_s, and the following relationship holds:

$$t = kT_s,$$

where the sample index k is a nonnegative integer. This situation corresponds to a *digital control implementation*, where actuators are updated only at predetermined instants and held constant at all other times. Similarly, sensors are read only at prescribed instants of time. In most implementations, the rate at which sensors are read and actuators updated is the *sample rate*, the reciprocal of the sampling period. In a *multirate* implementation, sensor readouts may occur more frequently than actuator updates, or individual sensors or actuators may be processed at different rates. The action of holding sensor and actuators constant except at sampling instants is termed a *zero-order hold* (ZOH) operation. Ideally, digital-to-analog (DAC) and analog-to-digital (ADC) converters perform the ZOH operation when processing actuators and sensors.

A ZOH equivalent model captures the dynamic relationship between $x(k)$ and $u(k)$, given an underlying continuous-time process with state variable $x(t)$ and input $u(t) = u(kT_s)$. When the continuous-time, n-th order process is given by the state-space model

$$\dot{x} = Ax + Bu, \tag{9.1}$$

$$y = Cx + Du, \tag{9.2}$$

the ZOH equivalent has the following state-space form [80]:

$$x(k+1) = A_d x(k) + B_d u(k), \tag{9.3}$$

$$y(k) = Cx(k) + Du(k), \tag{9.4}$$

where matrices A_d and B_d are determined as follows:

$$A_d = \mathrm{e}^{AT_s}, \tag{9.5}$$

$$B_d = \int_0^{T_s} \mathrm{e}^{As} B \mathrm{d}s. \tag{9.6}$$

Figure 9.1 illustrates the arrangement used in a digital control system and the relationship between continuous and discrete-time signals. A comprehensive study of discrete-time systems and digital controls requires considerable time, and lies outside the scope of this book. Readers are referred to Franklin, Powell, and Workman [81] or Åstrom and Wittenmark [80] for in-depth coverage.

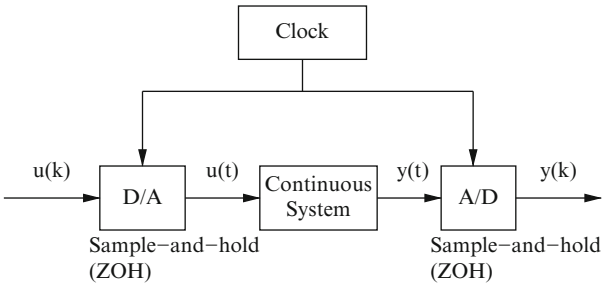

Fig. 9.1 Digitally controlled continuous-time plant

For the purposes of our introductory presentation of MPC, a discussion on discrete-time stability and the calculation of A_d and B_d are sufficient. Consider the following autonomous discrete-time system:

$$x(k+1) = A_d x(k).$$

This system is said to be stable if given any initial condition $x(0)$, the solution satisfies $||x(k)|| \to 0$ as $k \to \infty$. That is, the solution sequences generated by choosing arbitrary initial conditions must all converge to zero. Stability is equivalent to the requirement that all eigenvalues of A_d have magnitudes less than 1. This condition is also commonly expressed as the requirement that all eigenvalues of A_d lie inside the unit circle of the complex plane.

Consider now the control system of (9.3). The concepts of stabilizability and state feedback stabilization are entirely analogous to those studied in Sect. 4.2.2: the system is said to be stabilizable if there exists a state feedback gain K such that $A_d - B_d K$ has all its eigenvalues inside the unit circle. This matrix arises from the state feedback control law $u(k) = -Kx(k)$. A matrix K can be found that places the n eigenvalues of $A_d - B_d K$ at any desired location if the pair (A_d, B_d) is controllable. The definition of controllability and its numerical rank test are the same as for continuous-time systems.

The calculation of A_d and B_d given A, B, and a sample period T_s can be accomplished by various methods. Once A_d has been obtained, B_d is directly evaluated by performing the integration indicated in (9.6). Therefore, only the calculation of A_d must be considered. This matrix is defined through the following *matrix exponential* series

$$e^X = I + X + \frac{1}{2!}X^2 + \frac{1}{3!}X^3 + ...,$$

where X is a square matrix. The matrix exponential should not be confused with the matrix resulting from taking the exponential of the individual entries. In Matlab, the latter is calculated as `e^X`, while the matrix exponential is calculated using `expm(X)`.

One simple way to calculate e^X applies to *nilpotent* matrices. A nilpotent matrix has the property that $X^q = 0$ for some integer $q \geq 2$. In this case, the series definition of e^X has a finite number of nonzero terms and the computation can be done directly. The Laplace formula of (9.7) may also be used when A has low dimensions.

$$e^{At} = \mathcal{L}^{-1}\{(sI - A)^{-1}\} \tag{9.7}$$

For general matrices, the exponential is best calculated numerically, using a linear algebra package such as Matlab. The Cayley–Hamilton theorem may also be used, reducing the computation of e^X to a system of linear equations, see, for instance, Brogan [65].

For our purposes, the ability to obtain A_d and B_d numerically is sufficient. In Matlab, this is accomplished the continuous-to-discrete conversion function `c2d`. As an example, the following code obtains the ZOH equivalent of the CMAPSS-1 or -2 model matrices at FC01:

```
%Assuming that A and B are in the workspace
C=[1 0];
D=0; %C and D need to be specified: use fan speed as the output
sys_CT=ss(A,B,C,D); %create continuous-time system
Ts=1e-3; %choose a sample period
sys_DT=c2d(sys_CT,Ts,'zoh'); %convert to discrete-time using ZOH
[Ad,Bd,Cd,Dd]=ssdata(sys_DT); %extract matrices
%Note that Cd=C and Dd=D
```

The reader can verify that the eigenvalues of A_d are inside the unit circle. This is expected since the system is stable in continuous-time and must remain so in discrete-time. The selection of T_s must consider various factors:

1. The sampling frequency $1/T_s$ has to be large enough to capture continuous-time dynamics with fidelity. As a rule of thumb, $1/T_s$ must be chosen to be at least ten times larger than the highest frequency (pole) present in the plant model.
2. The sampling frequency $1/T_s$ has to be larger than the design bandwidth of the closed-loop system.
3. T_s cannot be smaller than the minimum sample period afforded by the control hardware, including DAC, ADC, and real-time processor.

Real-time propulsion control algorithms are implemented by the full-authority digital engine controller (FADEC), a digital computer and associated electronic hardware mounted on or near the engine. Readers are referred to Jaw and Mattingly [14] for more information on FADEC and its predecessors.

The digital controller supplied with CMAPSS-40k, for instance, uses $T_s = 0.015$ s. The fastest plant pole is -4, found at ground idle conditions, as it can be determined from the data of Appendix C. The sampling frequency of 66.7 Hz is therefore more than ten times larger than the fastest dynamics of the plant (4 rad/s). A 0.5-s settling time specification with 5% overshoot, for example, corresponds to a closed-loop bandwidth of 11.6 rad/s, as calculated with the formulas of Sect. 3.1.3. The sampling period of $T_s = 0.015$ s. is clearly adequate according to the above guidelines.

If a value for T_s has not been yet determined, it is still possible to find A_d and B_d in terms of T_s by resorting to a symbolic computation package. This procedure is adequate for small values of n. The following Matlab code assumes that the Symbolic Math Toolbox is available.

```
%Assuming that A and B are in the workspace
syms Ts %declare symbolic variable Ts
Ad=expm(A*Ts); %calculate matrix exponential symbolically
vpa(Ad,3) %display results to 3 decimals
Bd=int(Ad*B,0,Ts); %integrate symbolically
vpa(Bd,3) %display results to 3 decimals
```

The reader may verify that substituting $T_s = 0.001$ into the symbolic expressions for A_d and B_d yields the same results as the `c2d` function used earlier.

9.2 Optimal Receding Horizon Control

The linear quadratic control (LQR) approach presented in Sect. 4.4 used the cost function of (4.14) as a measure of optimality. The upper integration limit is ∞, so that the long-term response of trajectories, as well as their transient behavior are captured. Trajectories not converging to zero are thus disallowed, since they would result in infinite cost. The use of infinity as upper integration limit corresponds to an *infinite-horizon* optimization problem. The solution to the discrete-time optimal linear quadratic control problem parallels the continuous-time case discussed in Sect. 4.4. That is, the solution to the infinite-horizon LQR problem is given by the linear state feedback law $u(k) = -Kx(k)$, where the state feedback gain K is found from a discrete-time algebraic Riccati equation.

In contrast, the *finite-horizon* linear quadratic optimal control problem uses a finite time T as upper integration limit. In the general case, the solution is no longer expressible as a feedback law, but rather as an optimal control function defined only between 0 and T. The optimal control function is thus an explicit function of time, in what constitutes an *open-loop* or *feedforward* control solution [82]. Finite horizons are useful, for instance, when the system state must be transferred optimally between two values in a given time, without regard to system behavior beyond the horizon. Optimal spacecraft trajectories may be determined in this fashion. For example, a satellite may be required to transfer to a lower orbit in a prescribed time and by burning fuel in an optimal fashion. Thrusters are turned off once the satellite reaches the lower, faster orbit and the optimization problem no longer applies.

In discrete-time systems, a finite-horizon quadratic cost function aimed at optimal output tracking has the form

$$J = \sum_{i=1}^{i=n_y} e(k+i)^T Q e(k+i) + \sum_{i=0}^{n_u-1} u(k+i)^T R u(k+i), \tag{9.8}$$

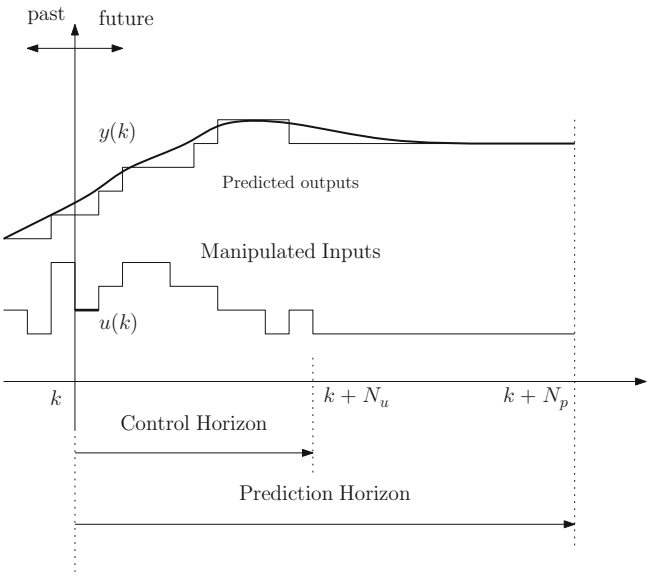

Fig. 9.2 Receding-horizon control schematic

where $r(k)$ is the vector of reference inputs, $e(k) = r(k) - y(k)$ and Q and R are positive-definite weighting matrices. The summation limits n_y and n_u are the *prediction horizon* and the *control horizon*, respectively.

An optimal control sequence $u^*(k+i), i = 0, 1, 2, ...n_u - 1$ is obtained by minimizing J subject to the system dynamics of (9.3) and (9.4), and possibly including input and output constraints. The unconstrained minimization case reduces to a discrete version of the algebraic Riccati equation (see Kirk, [82]), while the constrained case requires an entirely numerical approach. If the engineering problem under consideration truly involves a finite horizon (as in the satellite example), $u^*(k)$ is applied to the plant over the entire control horizon in an open-loop fashion. In contrast, problems requiring continued feedback control action must use either infinite horizons or a *receding-horizon* approach. In the latter case, a subsequence $u^*(k+i), i = 0, 1, 2...n_u^*$ is applied. Once the last element of the subsequence has been applied, the optimization problem is solved again, leading to a new optimal sequence, and the process is repeated indefinitely. Receding-horizon control usually sets the subsequence length to 1, that is, only $u^*(k)$ is applied to the plant. Figure 9.2 illustrates the concept of receding horizon control. Of course, the use of receding-horizon control is reasonable when the optimal solution sequences are not the same each time the problem is solved. This is certainly the case when considering constraints, since the optimal sequence will be strongly influenced by the proximity of the state variables and control inputs to the constraint boundaries at the time of optimization. As it occurs in the GTE problem, design plant matrices

may also be changing in time (recall the LPV approach of Sect. 5.3.) The presence of constraints and plant parameter variations is indeed the main reasons why receding horizon control constitutes a valid choice for GTE control problems.

9.3 Prediction Equations

We begin our development of MPC laws by obtaining formulas to predict state and output sequences over a horizon n_y, starting from a state $x(k)$ and applying a sequence of inputs $u(k+i)$, for $i = 0, 1, 2...n_u - 1$. Using the plant model of (9.3) and (9.4) to perform iterated substitutions, we have:

$$
\begin{aligned}
x(k+1) &= A_d x(k) + B_d u(k) \\
x(k+2) &= A_d x(k+1) + B_d u(k+1) = A_d^2 x(k) + A_d B_d u(k) + B_d u(k+1) \\
x(k+3) &= A_d x(k+2) + B_d u(k+2) = A_d^3 x(k) + A_d^2 B_d u(k) \\
&\quad + A_d B_d u(k+1) + B_d u(k+2) \\
&\vdots = \vdots \\
x(k+n_y) &= A^{n_y} x(k) + A^{n_y-1} B_d u(k) + A^{n_y-2} B_d u(k+1) + ... + B_d u(k+n_y-1).
\end{aligned}
$$

The sequence of outputs generated by (9.4) is likewise predicted as follows:

$$
\begin{aligned}
y(k+1) &= Cx(k+1) = CA_d x(k) + CB_d u(k) + Du(k+1) \\
y(k+2) &= Cx(k+2) = CA_d^2 x(k) + CA_d B_d u(k) + CB_d u(k+1) + Du(k+2) \\
y(k+3) &= Cx(k+3) = CA_d^3 x(k) + CA_d^2 B_d u(k) + CA_d B_d u(k+1) \\
&\quad + CB_d u(k+2) + Du(k+3) \\
&\vdots = \vdots \\
y(k+n_y) &= CA^{n_y} x(k) + CA^{n_y-1} B_d u(k) + CA^{n_y-2} B_d u(k+1) \\
&\quad + ... + CB_d u(k+n_y-1) + Du(k+n_y).
\end{aligned}
$$

Note that the above formulas assume $n_u = n_y$. In matrix form, the prediction equations reduce to

$$\hat{x} = P_{xx}x(k) + H_x\hat{u}, \tag{9.9}$$

$$\hat{y} = Px(k) + H\hat{u}. \tag{9.10}$$

Here, the circumflex notation over x, y, and u is used to denote "prediction" ($\hat{u}$ will be the outcome of an optimization calculation, yet it is still customary to refer to it as the "predicted" control input sequence). Matrices P_{xx}, P, H_x, and H are as follows:

$$P_{xx} = \begin{bmatrix} A_d \\ A_d^2 \\ \vdots \\ A_d^{n_y} \end{bmatrix} \quad P = \begin{bmatrix} CA_d \\ CA_d^2 \\ \vdots \\ CA_d^{n_y} \end{bmatrix}, \tag{9.11}$$

$$H_x = \begin{bmatrix} B_d & 0 & 0 & \dots \\ A_d B_d & B_d & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots \\ A_d^{n_y} B_d & A_d^{n_y-2} B_d & A_d^{n_y-3} B_d & \dots \end{bmatrix}, \tag{9.12}$$

$$H = \begin{bmatrix} CB_d & D & 0 & \dots \\ CA_d B_d & CB_d & D & \dots \\ \vdots & \vdots & \vdots & \vdots \\ CA_d^{n_y} B_d & CA_d^{n_y-2} B_d & CA_d^{n_y-3} B_d & \dots \end{bmatrix}. \tag{9.13}$$

The above prediction formulas are valid for the MIMO case. When the plant output y is a p-by-1 vector and the control input is an m-by-1 vector, P has dimensions pn_y-by-n and H has dimensions pn_y-by-mn_y. The input sequence is organized by stacking the m-by-1 vectors $\hat{u}(k+i)$ vertically, beginning with $i = 0$. The same arrangement is used for $\hat{y}$.

9.4 Incremental MPC Formulation: Unconstrained Case

The benefits of integral control action can be incorporated into the MPC formulation by a discrete version of state augmentation. The control update equation analogous to continuous-time integration is:

$$u(k+1) = u(k) + \Delta u(k). \tag{9.14}$$

The new control input is $\Delta u(k)$, and the augmented plant model becomes:

$$\begin{bmatrix} x(k+1) \\ u(k) \end{bmatrix} = \begin{bmatrix} A_d & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix} + \begin{bmatrix} B_d \\ I \end{bmatrix} \Delta u(k),$$

$$y(k) = [C \quad D] \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix} + D\Delta u(k).$$

Denoting the augmented state vector as $x_a^T(k) = [x^T(k) \;\; u^T(k-1)]$, the augmented model can be compactly represented by:

$$x_a(k+1) = A_{da}x_a(k) + B_{da}\Delta u(k), \tag{9.15}$$

$$y(k) = C_{da}x_a(k) + D\Delta u(k), \tag{9.16}$$

where the definition of matrices A_{da}, B_{da}, and C_{da} is evident. Note that prediction equations (9.9) and (9.10) are directly applicable to the augmented model, using A_{da}, B_{da}, C_{da}, and D_d as plant matrices.

Next, consider a simplified version of cost function (9.8) as follows:

$$J = \sum_{i=1}^{i=n_y} e(k+i)^T e(k+i) + \lambda \sum_{i=0}^{n_u-1} \Delta u(k+i)^T \Delta u(k+i). \tag{9.17}$$

Here a lumped scalar weight λ is used, and $\Delta u(k+i)$ for $i = 0, 1, ..n_u - 1$ are the variables with respect to which optimization is to be carried out. The subsequent development assumes that n_y exceeds n_u by at least one, that is, $n_y \geq n_u + 1$. Matrix H in (9.13) was defined using $n_u = n_y$, however. A modified H is obtained by assuming that $\Delta u(k+i) = 0$ for $i \geq n_u$, implying that control increments are zero between control and prediction horizons. This is equivalent to retaining only the first n_u block-columns of H (that is, the first mn_u columns of H) in (9.13) when calculating predicted outputs.

The predicted sequence $e(k+i)$ for $i = 0, 1, 2...n_y$ can be expressed in terms of the current state and predicted control sequence by using the definition of the error and the prediction equations. Indeed, using $\hat{y}(k+i) = Px_a(k) + H\Delta u(k)$, the first summation in (9.17) becomes

$$\sum_{i=1}^{i=n_y} e(k+i)^T e(k+i) = [r - Px_a(k) - H\Delta\hat{u}]^T [r - Px_a(k) - H\Delta\hat{u}], \tag{9.18}$$

where $\Delta\hat{u}$ and the reference sequence r are organized as follows:

$$\Delta\hat{u} = \left[\begin{array}{l} \left.\begin{array}{l}\Delta\hat{u}_1(k)\\ \Delta\hat{u}_2(k)\\ \vdots\\ \Delta\hat{u}_m(k)\end{array}\right\} \Delta\hat{u}(k) \\ \hline \left.\begin{array}{l}\Delta\hat{u}_1(k+1)\\ \Delta\hat{u}_2(k+1)\\ \vdots\\ \Delta\hat{u}_m(k+1)\end{array}\right\} \Delta\hat{u}(k+1), \\ \hline \vdots \\ \hline \left.\begin{array}{l}\Delta\hat{u}_1(k+n_u-1)\\ \Delta\hat{u}_2(k+n_u-1)\\ \vdots\\ \Delta\hat{u}_m(k+n_u-1)\end{array}\right\} \Delta\hat{u}(k+n_u-1) \end{array}\right], \tag{9.19}$$

$$\Delta r = \left[\begin{array}{l} \left.\begin{array}{l}r_1(k)\\ r_2(k)\\ \vdots\\ r_p(k)\end{array}\right\} r(k) \\ \hline \left.\begin{array}{l}r_1(k+1)\\ r_2(k+1)\\ \vdots\\ r_p(k+1)\end{array}\right\} \Delta r(k+1) \\ \hline \vdots \\ \hline \left.\begin{array}{l}r_1(k+n_y-1)\\ r_2(k+n_y-1)\\ \vdots\\ r_p(k+n_y-1)\end{array}\right\} \Delta r(k+n_y-1) \end{array}\right]. \tag{9.20}$$

The second summation in (9.17) is equivalent to $\lambda(\Delta\hat{u})^T\Delta\hat{u}$. Expanding (9.18) and combining gives:

$$J = \Delta^T\hat{u}[H^TH + \lambda I]\Delta\hat{u} + 2\left[x_a^TP^TH - r^TH\right]\Delta\hat{u} + J_0,$$

where J_0 contains terms that are independent of $\Delta\hat{u}$. In unconstrained minimization, the minimizing argument is unaffected by the constant term J_0. The optimization problem can then be formulated as follows:

minimize over $\Delta\hat{u}$:

$$J' = (\Delta\hat{u})^T\left[H^TH + \lambda I\right]\Delta\hat{u} + 2\left[x_a^TP^TH - r^TH\right]\Delta\hat{u}.$$

The above objective function is quadratic, with $H^TH + \lambda I$ being symmetric and positive-definite. Therefore, it possesses a global minimum that can be found by equating the gradient to zero. The solution for the optimal sequence starting at time k is

$$\Delta\hat{u} = (H^TH + \lambda I)^{-1}H^T(r - Px_a(k)). \tag{9.21}$$

Only the first element of this sequence is to be applied to the plant. Thus, the MPC input has the form

$$u(k) = P_r r - Kx_a,$$

where P_r is an m-by-pn_y matrix resulting from extracting the first m rows of $(H^TH + \lambda I)^{-1}H^T$, and K is an m-by-n matrix resulting from extracting the first m rows of $(H^TH + \lambda I)^{-1}H^TP$.

Clearly, *unconstrained MPC reduces to a state feedback law with reference prefilter*, with constant gains.

9.4.1 Example

Consider the following arbitrary system matrices:

$$A_d = \begin{bmatrix} 0 & 1 & 3 \\ 1 & 1 & 0 \\ -1 & 0 & 2 \end{bmatrix}, \quad B_d = \begin{bmatrix} 1 & 2 \\ 2 & 0 \\ 0 & -1 \end{bmatrix}.$$

Suppose two outputs are defined as

$$y = C_d x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \end{bmatrix}.$$

Here, $n = 3$, $m = 2$ and $p = 2$. As an example, take $n_y = 3$ and $n_u = 2$. Matrices H and P, as well as the equivalent feedback gain and prefilter gain are readily computed using the following iterative Matlab code:

```
%Plant matrices
Ad = [0 1 3;1 1 0;-1 0 2];Bd = [1 2;2 0;0 -1];
C = [1 0 0;0 1 -1];D = [0 0;0 0];
n=size(Ad,1); %state dimension
m=size(Bd,2); %input dimension
p=size(C,1); %output dimension
%Horizons
ny=3;nu=2;
%Compute P
P=C*Ad;
for i=1:ny-1,
  P=[P;C*Ad^(i+1)];
end
%Compute H
H=zeros(p*ny,m*nu);
for i=1:ny,
    for j=1:i,
        H(1+(i-1)*p:i*p,1+(j-1)*m:j*m)=C*Ad^(i-j)*Bd;
    end
    H(1+(i-1)*p:i*p,j*m+1:(j+1)*m)=D;
end
%Retain only the first nu blocks
H=H(:,1:nu*m);
%Extract prefilter and state feedback gain
lambda=2; %example
Pr=inv(H'*H+lambda*eye(m*nu))*H';
Pr=Pr(1:m,:);
K=inv(H'*H+lambda*eye(m*nu))*H'*P;
K=K(1:m,:);
```

The reader can check that the eigenvalues of $A_d - B_d K$ are $0.0772 \pm 0.1146i$ and 0.224, located within the unit circle. The above code can be used to experiment with the effects of n_u and n_y on the closed-loop pole locations.

Although unconstrained MPC and state feedback with prefilter have similar laws, their implementations are slightly different. In state feedback, $r(k)$ is multiplied by the prefilter gain at each sampling instant. In MPC, r is a vector of *predicted* references, and its value must be available over the prediction horizon. The dimensions of P_r are m-by-pn_y, while the reference prefilter gain used in linear state feedback has dimensions m-by-p.

Except for the possibility of tracking changes in the reference inputs by means of prediction, there is little reason to use unconstrained MPC. There is no clear relationship between n_y and n_p and the resulting closed-loop pole locations, making other linear approaches preferable. The advantages of MPC become evident only when constraints are contemplated, as developed in the following section.

9.5 Incremental MPC Formulation: Constrained Case

Suppose now that the control input and a set of outputs are required to remain within prescribed limits at all times. The cost function to be minimized is given by (9.17).

Because the objective function is quadratic, the minimizing argument is independent of the constant J_0, even in the constrained case. Thus, the optimization problem takes the form

minimize over $\Delta\hat{u}$

$$J' = (\Delta\hat{u})^T[H^T H + \lambda I]\Delta\hat{u} + 2[x_a^T P^T H - r^T H]\Delta\hat{u}$$

subject to

$$\underline{U} \leq u(k+i) \leq \overline{U} \text{ for } i = 0, 1...n_u - 1 \text{ (input constraints)}$$
$$\underline{Y} \leq y(k+i) \leq \overline{Y} \text{ for } i = 1...n_y \text{ (output constraints).}$$

Here, $\overline{U}$, $\underline{U}$, $\overline{Y}$, and $\underline{Y}$ are vectors containing the bounds for the individual input and output channels. Input and output constraints must be expressed in terms of the optimization degrees of freedom $\Delta\hat{u}$ before a numerical solution to the problem can be programmed. Since $u(k+i) = u(k-1+i) + \Delta u(k+i)$, the input constraint for $i = 0$ becomes:

$$\underline{U} - u(k-1) \leq \Delta u(k) \leq \overline{U} - u(k-1).$$

Advancing k by one step gives:

$$\underline{U} - u(k) \leq \Delta u(k+1) \leq \overline{U} - u(k).$$

Replacing $u(k)$ with $u(k-1) + \Delta u(k)$ and re-arranging yields:

$$\underline{U} - u(k-1) \leq \Delta u(k+1) + \Delta u(k) \leq \overline{U} - u(k-1).$$

Advancing k by one step once more and performing the same substitution for $u(k)$ gives:

$$\underline{U} - u(k-1) \leq \Delta u(k+3) + \Delta u(k+2) + \Delta u(k+1) \leq \overline{U} - u(k-1).$$

A clear pattern arises, allowing us to write the entire set of input constraints in matrix form as follows:

$$\begin{bmatrix} I & 0 & 0 & ... \\ I & I & 0 & ... \\ \vdots & \vdots & \vdots & \vdots \\ I & I & I & .... \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+n_u-1) \end{bmatrix} \leq \begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix} (\overline{U} - u(k-1))$$

and

$$-\begin{bmatrix} I & 0 & 0 & \dots \\ I & I & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots \\ I & I & I & \dots \end{bmatrix}\begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+n_u-1) \end{bmatrix} \leq -\begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix}(\underline{U} - u(k-1)).$$

Introduce the following definitions:

$$C_c = \begin{bmatrix} I & 0 & 0 & \dots \\ I & I & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots \\ I & I & I & \dots \end{bmatrix}, \quad L = \begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix}. \tag{9.22}$$

The complete set of input constraints can now be compactly represented as:

$$C_c \Delta\hat{u} \leq \bar{d}_u(k), \tag{9.23}$$

$$-C_c \Delta\hat{u} \leq \underline{d}_u(k), \tag{9.24}$$

where $\bar{d}_u(k) = L(\overline{U} - u(k-1))$ and $\underline{d}_u(k) = -L(\underline{U} - u(k-1))$.

A reasonable approach to output constraint handling is the requirement that *predicted* outputs satisfy the constraints over the prediction horizon. Using $\hat{y}(k) = Px_a(k) + H\Delta u(k)$, output constraints reduce to:

$$H\Delta\hat{u} \leq \bar{d}_y(k), \tag{9.25}$$

$$-H\Delta\hat{u} \leq \underline{d}_y(k), \tag{9.26}$$

where vectors $\bar{d}_y(k)$ and $\underline{d}_y(k)$ are defined as

$$\bar{d}_y(k) = \begin{bmatrix} \overline{Y} \\ \overline{Y} \\ \vdots \\ \overline{Y} \end{bmatrix} - Px_a(k), \quad \underline{d}_y(k) = -\begin{bmatrix} \underline{Y} \\ \underline{Y} \\ \vdots \\ \underline{Y} \end{bmatrix} + Px_a(k). \tag{9.27}$$

The bound vectors in the above equations contain n_y blocks $\overline{Y}$ or $\underline{Y}$, each one having dimensions p-by-1. Input and output constraints can now be combined in a single inequality on $\Delta\hat{u}$, resulting in the following optimization problem:

minimize over $\Delta\hat{u}$

$$J' = (\Delta\hat{u})^T[H^T H + \lambda I]\Delta\hat{u} + 2[x_a^T P^T H - r^T H]\Delta\hat{u}$$

subject to

$$M\Delta\hat{u} \leq d(k),$$

where M and $d(k)$ are defined as

$$M = \begin{bmatrix} C_c \\ -C_c \\ H \\ -H \end{bmatrix}, \quad d = \begin{bmatrix} \bar{d}_y(k) \\ \underline{d}_y(k) \\ \bar{d}_u(k) \\ \underline{d}_u(k) \end{bmatrix}. \tag{9.28}$$

This constrained optimization problem constitutes a *quadratic program*, for which efficient numerical solution routines exist. In Matlab (Optimization Toolbox), the `quadprog` routine is available to handle this optimization. Note that the constraint definition involves a constant matrix M and a variable vector $d(k)$. The latter must be recomputed at each sampling instant. Note also that selected components of M and d may be omitted to reflect the absence of certain constraints.

9.6 Example: Linearized CMAPSS-40k Plant at Ground Idle

An example is now presented that illustrates the preparatory steps required to apply MPC to the GTE problem, followed by a linearized discrete-time simulation entirely programmed in a Matlab script included in Appendix D. The objective is to produce a fast change in fan speed while maintaining fuel flow, VSV, and VBV within acceptable bounds. In addition, two outputs, namely T_{48} and SmHPC are to be kept between allowable limits.

Matrices A and B corresponding to ground idle conditions are given in Appendix C. Output matrices for ΔT_{48} and Δ SmHPC are also listed. This example uses $n_y = 7$ and $n_u = 3$. Since the linearized model is incremental, the bounds on inputs and outputs are given accordingly.

In practical situations such as the one represented in this example, the controlled variables may not be subject to constraints, and the constrained outputs may not need to be controlled. In this example, three actuators are used to control a single output, namely fan speed. Two other variables, namely HPT outlet temperature and HPC stall margin are considered as constrained outputs. This implies that different sets of H and P matrices must be used, according to whether an output is controlled or constrained. The code listed in Appendix D illustrates how a mixed

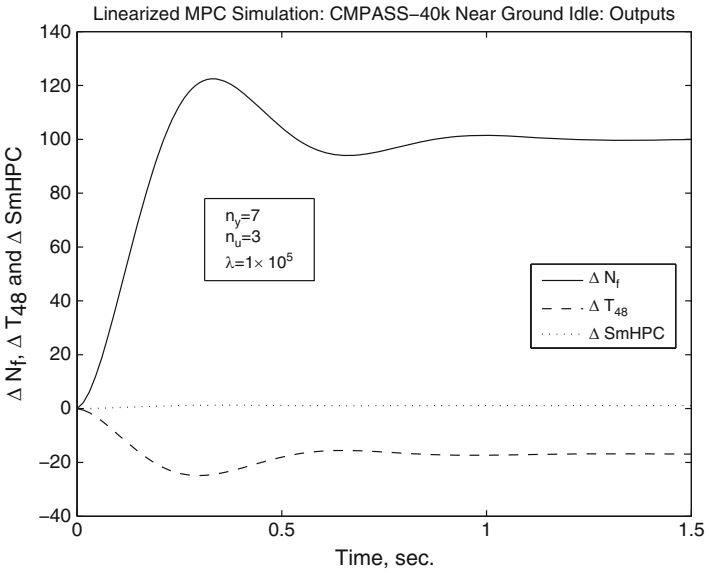

Fig. 9.3 Output Response of CMAPSS-40k engine linearized near ground idle: model predictive control with large control weighting

set of constraint handling and tracking requirements is incorporated into the MPC formulation.

Figures 9.3 and 9.4 show the input and output trajectories with $\lambda = 1 \times 10^5$. Due to the high value of control weighting, the actuators do not attain their maximum allowable values and the settling time required to attain a setpoint for ΔN_f of 100 rpm is roughly 1 s. An overshoot of 22% is obtained. A second simulation using $\lambda = 0.01$ shows that the tradeoff between response speed and control effort can easily be manipulated with a single tuning parameter. Indeed, Figs. 9.5 and 9.6 show that the settling time is significantly reduced, while the overshoot becomes zero. This is achieved through periods of maximal control effort in W_F and VSV. Note that the large shaft accelerations observed in Fig. 9.5 are possible only in idealized, linear simulations. The weighting parameter λ must be in the order of magnitude of 1×10^5 for a nonlinear CMPASS simulation to be feasible, as shown in the next example.

9.7 Example: Nonlinear Engine Simulation: CMAPSS-40k

Deployment of MPC to a high-fidelity simulator may be done at various levels of sophistication and complexity. For instance, the engine may be linearized numerically at every simulation step, as an effective way of dealing with plant parameter

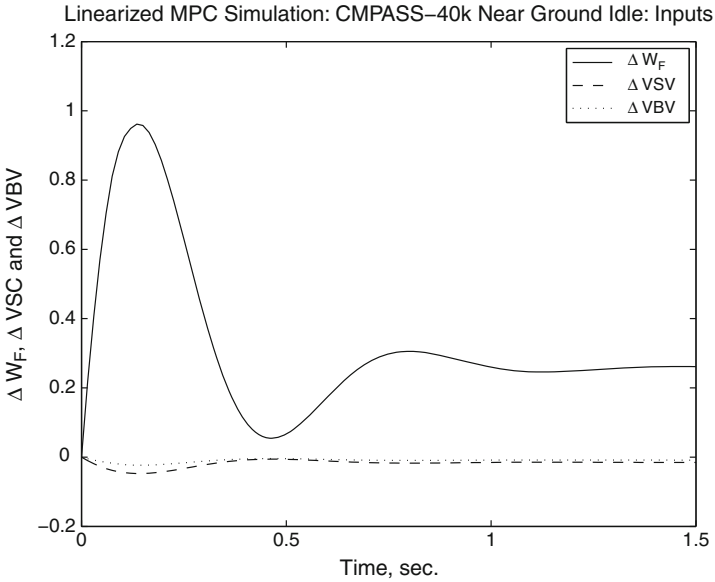

Fig. 9.4 Control inputs of CMAPSS-40k engine linearized near ground idle: model predictive control with large control weighting

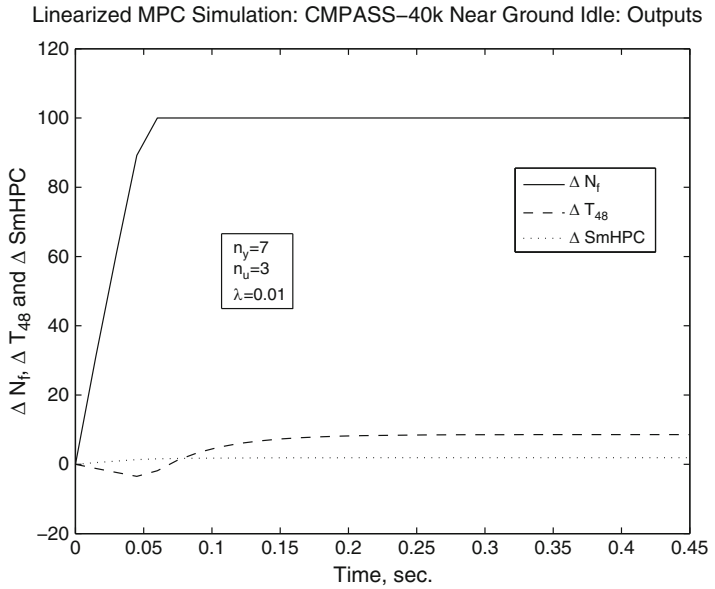

Fig. 9.5 Output response of CMAPSS-40k engine linearized near ground idle: model predictive control with small control weighting

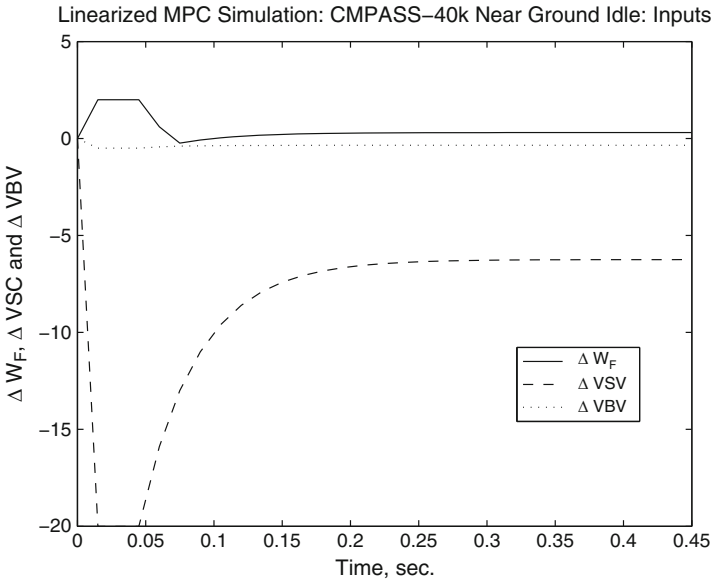

Fig. 9.6 Control inputs of CMAPSS-40k engine linearized near ground idle: model predictive control with small control weighting

changes. Alternatively, the LPV decomposition of Sect. 5.3 could be used to reduce the computational load. In this example, a fixed plant model is used throughout simulation. Offsets in the regulated and constrained outputs are expected, since predictions are based on an approximate model. The objectives are the same as in the previous linearized example: control fan speed while maintaining T_{48} and SmHPC within the following bounds: $-150°R \leq \Delta T_{48} \leq 300°R$ and $-10\% \leq \Delta$SmHPC $\leq 20\%$. Bounds are also specified for the three control inputs as follows: 0.05 pps $\leq W_F \leq 5$ pps, $-50° \leq$ VSV $\leq -6°$, and $0.01 \leq$ VBV ≤ 1 (measured as a fraction of fully-open). A fan speed increment demand of 500 rpm was used throughout the example.

Very little tuning iterations are required to obtain acceptable responses. The designer adjusts only λ and the control and prediction horizons. Figure 9.7 shows the effect of changing n_y for fixed values of n_u and λ. It is observed that performance improvement is initially obtained by increasing n_y, but further changes have no effect beyond $n_y = 10$. Similarly, the effect of changing n_u for fixed values of n_y and λ is shown in Fig. 9.8. Again, responses improve as n_u is increased, but no detectable improvement occurs beyond $n_u = 3$. Although the threshold values of n_y and n_u are tailored to this example, the qualitative behavior is typical of MPC. In other words, the optimal trajectories quickly converge to the infinite-horizon solutions.

Changes in λ have a strong influence on the simulation outcome. Low values tend to produce faster responses with actuator saturation, and very high values may

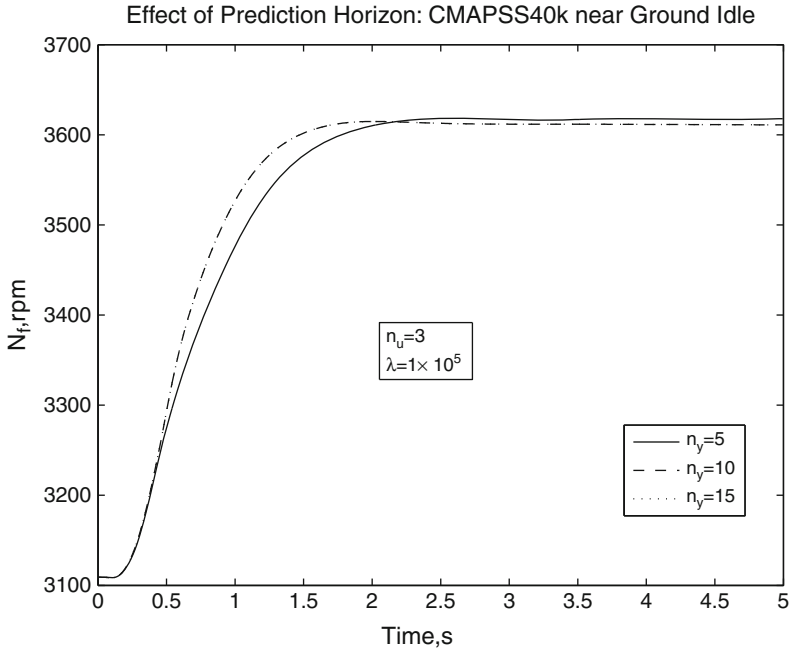

Fig. 9.7 Effect of prediction horizon: MPC in CMAPSS-40k

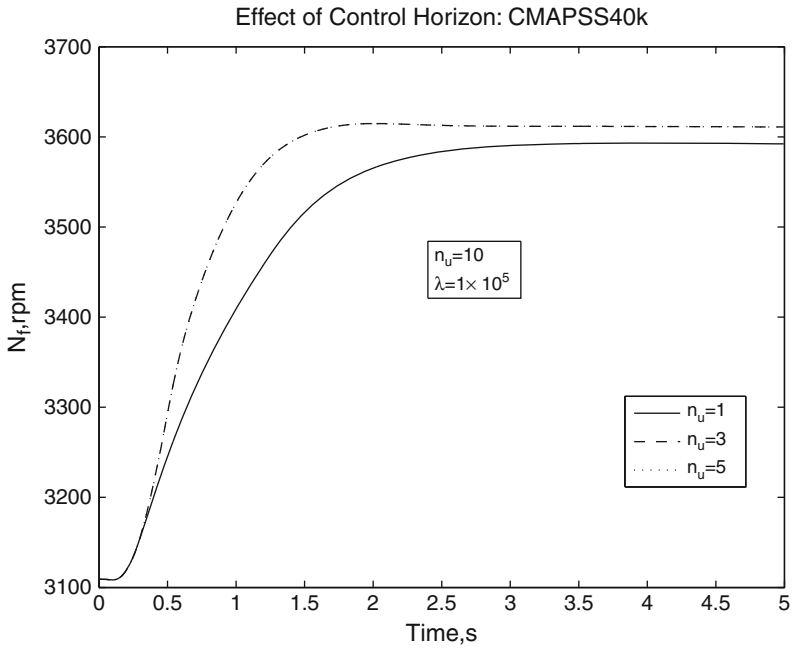

Fig. 9.8 Effect of control horizon: MPC in CMAPSS-40k

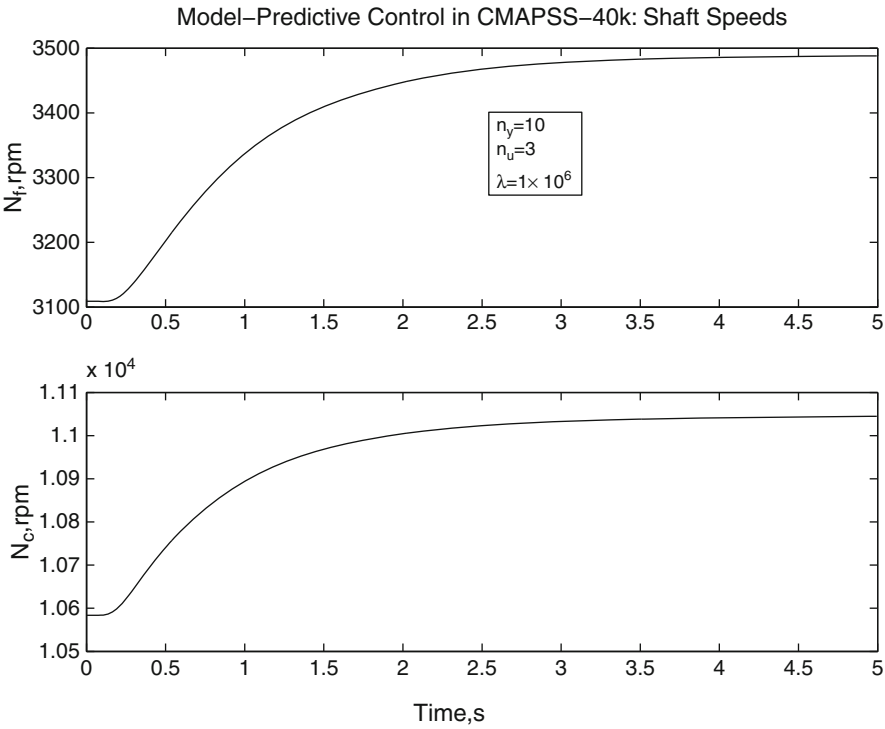

Fig. 9.9 Shaft speed responses: MPC in CMAPSS-40k

penalize control activity excessively, resulting in poor transient response. A value of $\lambda = 1 \times 10^6$ was selected by trial-and-error for the remaining simulations, along with $n_y = 10$ and $n_u = 3$. Figure 9.9 shows the responses of fan and core speeds. The settling time for fan speed is near 3 s, and there is a noticeable offset from the incremental requirement of 500 rpm. Since constrained output prediction relies on the C and D matrices of the linearized model, significant deviations between actual and predicted outputs occur. Because of this, if the intended output bounds are used directly in the MPC calculations, output constraint violations are observed in this example. The bounds may be made more conservative in an effort to satisfy output constraints in real-time operation or realistic simulations. In this example, the incremental upper output bounds were taken as $150°R$ and 10% for T_{48} and SmHPC, respectively. The incremental lower output bounds were taken as $-75°R$ and -5%. Figure 9.10 shows that T_{48} and SmHPC remain within their originally intended incremental bounds. Figure 9.11 shows the optimal control input trajectories. Only VBV reaches its bound (0.1%), due to the high control penalty used in this design.

Efficient ways are available to refine predictions by adding a generalized disturbance term to the prediction equations. State estimation techniques can be

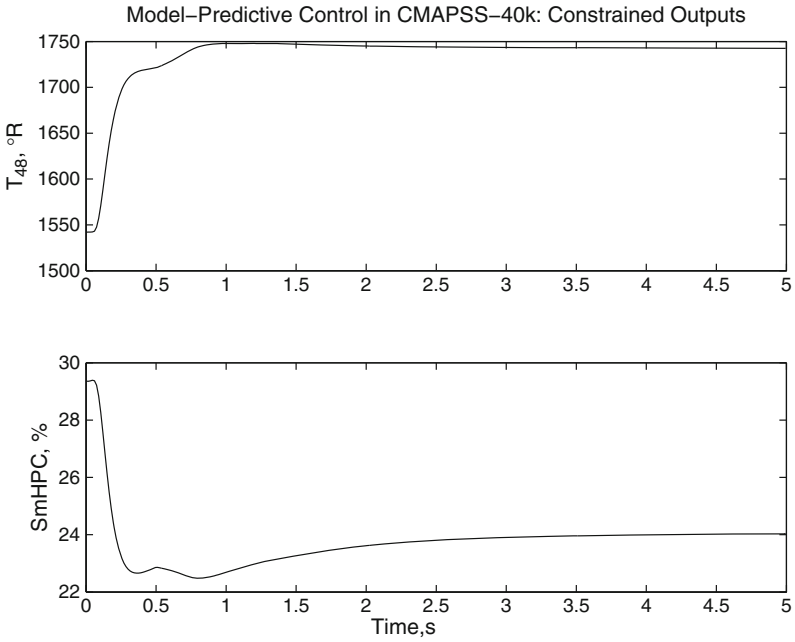

Fig. 9.10 Constrained output responses: MPC in CMAPSS-40k

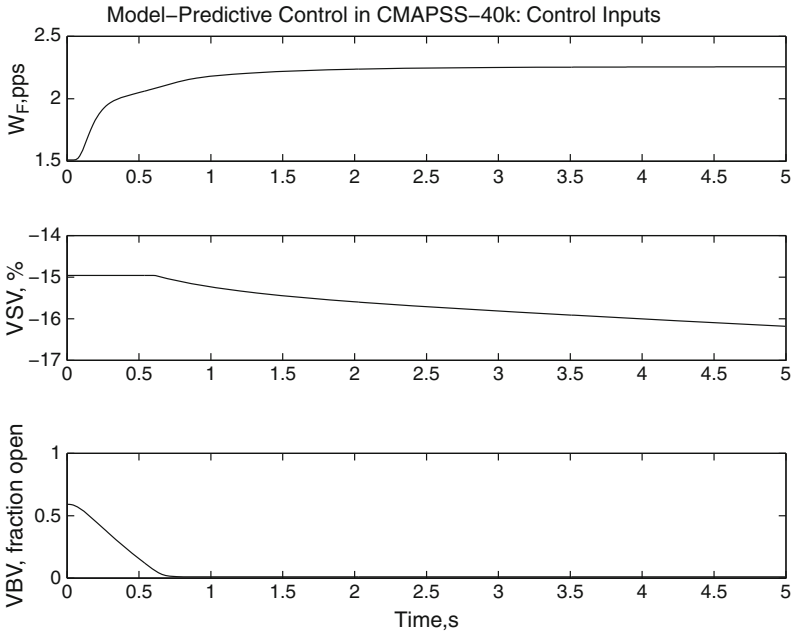

Fig. 9.11 Control input histories: MPC in CMAPSS-40k

used to generate online estimates of the generalized disturbance; see, for instance, Camacho [78] and Rossiter [79]. In the GTE problem, correcting terms must be generated by means of the engine steady map, which introduces additional complexity in the MPC law. One significant advantage of MPC is that it exploits the availability of three actuators to achieve control of a single relevant variable, namely fan speed, without the "square plant" limitations frequently found in traditional linear MIMO control. In the unconstrained case, a single control input (fuel flow) suffices to control a single output. When constraints are contemplated, the extra degrees of freedom in control input improve the feasibility of achieving the simultaneous regulation and limit protection tasks.

9.8 Addressing Computational Burden

Prior to the 1990s, MPC applications were confined to "slow" systems such as oil refining and other chemical processes, due to the large amount of time required to perform numerical optimizations. Currently, due to rapid progress in developing fast and cheap computing power, MPC is being considered as a serious candidate for faster processes such as mechanical systems and aircraft engine controls. Theoretical MPC research in the 2000s was directed at complexity reduction by piecewise implementation [83], multiplexed schemes [76, 84–86], and hybrid approaches that can optimize over logical or integer variables. For a survey containing industrial applications, see [87].

The availability of fast and cheap real-time processing has motivated work on MPC applied to aircraft engines [74–76, 88]. However, the FADEC on-board processors used in GTE control systems are still not powerful enough to accommodate the execution of the complex algorithm in real time. In the following sections, a brief overview is offered on MPC variants specifically designed to reduce the computational cost.

9.8.1 Explicit MPC Implementations

In an explicit MPC implementation, the solution of the optimization problem inherent to predictive control is performed offline. The solution consists in the computation of regions of the state space and corresponding control gains to be used in the actual real-time implementation. An explicit MPC implementation is essentially a multidimensional look-up table for the control gain. This approach requires that the plant model and objective function be reducible to a multiparametric (mp) programming problem. That is, the optimization problem should have the initial state appearing as a parameter in a linear fashion. The paper by Bemporad, Borrelli, and Morari [89] develops a performance criterion based on the sum of either the ∞ norm or 1-norm of the input command and the deviation of the state from its desired

value. By taking the ∞ norm over space and the 1-norm over time, it is possible to reduce the problem to an mp case. The solution offered by using the mixed $1/\infty$ norm is attractive for relatively small linear MPC problems. This method saves computing time by offering precomputed solutions at each time step. In the case of large plants, using this approach would impose high demands in memory because of the high-number of state variables. In addition, the explicit implementation is not well suited to plants exhibiting high variability, and the possibility of reducing the problem to multiparametric programming is difficult to establish.

9.8.1.1 1-Norm Criterion with End Condition

Dynamic matrix control (DMC) involves a system description in terms of step response parameters. This description is used in a receding-horizon optimization to generate a sequence of controls that minimizes the deviations between the actual and desired step responses. DMC is traditionally used in conjunction with a quadratic objective function (QDMC). As seen in [90], an objective function based on the 1-norm can be used along with an end condition. This approach is computationally less intensive than QDMC. By introducing an end-condition in the performance index, it is possible to obtain a stable and high performance control system even when using input/output constraints and short prediction horizons. The control law is calculated by solving an online linear program, which is less complex than a quadratic one.

9.8.1.2 Mixed-Norm Approaches

A cost function is introduced in [91] that allows the formulation of a robustly stable MPC problem solvable by a linear program. Using a $1/\infty$ norm performance index for this cost allows precomputation of the solution, so that the linear program does not have to be solved online. The objective of predictive control is to compute the future control sequence $u(k), u(k+1), ..., u(k+N_u)$ in such a way that the optimal j step-ahead prediction $y(k+j|k)$ are driven close to $r(k+j)$ for the prediction horizon. The above approaches focus on changing the objective function so that the optimization problem becomes linear. In some cases, this allows part or all of the optimization process to be moved offline.

9.8.2 Multiplexed Control

A *multiplexed control implementation* denotes an arrangement in which a group of actuators is updated sequentially and cyclically, as opposed to simultaneously. In this technique, only a group of actuators are updated every sampling instant keeping all other actuators held at their previous values. The group of actuators being

updated is given by a predetermined schedule. A multiplexing schedule specifies the instants at which each actuator is commanded to the value dictated by the control law. Multiplexed control has been contemplated as an effective way to reduce the complexity of MPC [76, 84–86].

Application of multiplexing to receding-horizon strategies is straightforward. At each time step, a constrained MPC problem of reduced dimension is solved, corresponding to a subset of all available actuators. Only the first value of the calculated control sequence is applied, while the remaining actuators are kept at their previous values. The problem is solved again at subsequent time steps, but using the appropriate subset of actuators. In [85], two related algorithms are presented, assuming full state information. In one of the schemes, all actuators are optimized at once, but under the constraint that their increment will be nonzero only every m time steps, where m is the number of actuators. This scheme does not reduce the dimensionality of the constrained optimization routine, so it is of little advantage in terms of computational efficiency. In the second scheme, only one actuator at a time is optimized. Information about the previously predicted values of the other actuators is included in the optimization. The approach amounts to treating the nonoptimized actuators as known disturbances. The predicted values of the actuators will never be realized, however, due to the receding nature of the control algorithm.

Multiplexing is an effective way to reduce the complexity (i.e., dimensionality) of MPC calculations. At sampling instant k, linearization and discretization are performed to derive a single-input linear model, where the input is selected from among the set of all actuators according to a predefined, cyclic schedule. All other inputs are assumed constant and equal to the values they had at sampling instant $k-1$. The single-input linear model is used in the MPC optimization of Sect. 9.5, resulting in an optimal move sequence for the selected input. The first element of the sequence is applied to the corresponding plant actuator, while the other actuator commands are held at their previous values. The operation is repeated at the following sampling instants for the remaining inputs one by one and according to the schedule, completing what will be termed an *update cycle*. Update cycles are repeated indefinitely, until the control system is stopped. The update cycle is illustrated in Fig. 9.12. Note that the effective sample rate has been reduced to the original one divided by the number of actuators. However, the rate at which *some* actuator is being updated is the same as the original. The computational advantage of the multiplexed implementation lies in that all QP routines are now performed over just one degree of freedom. It is a well-established fact that *the time required to solve a QP problem grows with the cube of the number of inputs* [92], while the sample rate reduction is only linear in the number of inputs. Therefore, the time savings earned by multiplexing may even allow to increase the original sample rate to help recover any lost performance due to slower sampling. This is especially true for the disturbance rejection properties (a faster rate helps reduce the effects of disturbance in the intersample). This possibility is heavily dependent on the problem at hand, since the other computational costs need to be taken into account.

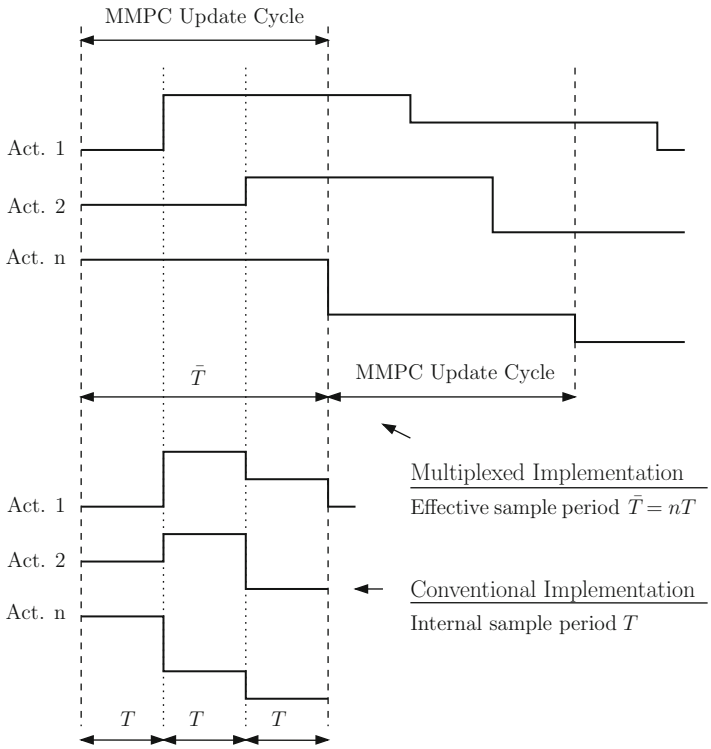

Fig. 9.12 Multiplexed control updates

A study conducted by the author [76] demonstrated the feasibility of using multiplexed MPC to control a large commercial turbofan engine. Dramatic computation time savings were achieved without significant performance losses.

The main concern in a multiplexed MPC implementation is stability. The closely related periodic estimation problem has been analyzed by DeNicolao [93]. The concept is to seek conditions under which a periodic extension of the first few values of the optimal gain sequence can be stabilizing when used as periodic feedback. These conditions are obtained by considering the cyclomonotonicity properties of the discrete periodic Riccati equation (DPRE). In [94], a similar problem is considered and sufficient conditions for stability are derived. The reader interested in MPC stability results in a broad sense is referred to Sect. 7.8 in Camacho [78] as a starting point. A rigorous study of MPC stability for nonlinear systems was pioneered by Mayne and co-workers [95–97].

Afterword

A comparison among the various approaches to control seems in order. A recurrent problem associated with state-based feedback implementations is the need to provide accurate state references. Steady references must be obtained by evaluation of the nonlinear engine's steady map. Time-varying references require a reference trajectory generator such as the one used in SM control. Reference state generation by means of a linearized model is largely inadequate, however, and nonlinear reference models reflecting engine dynamics are prohibitively complex. Besides, such a model cannot capture the effects of uncertain inputs. The beneficial robustness properties of SM control are then less effective, since states are being controlled toward values that do not result in the desired output setpoints.

The sliding mode approaches presented in this book use constant gains. Despite its robustness properties, a constant-gain SM controller cannot be expected to deliver the same transient response qualities across the whole flight envelope. In this respect, the standard gain-scheduled SISO fan speed regulator can outperform a SM controller. When engine limits are considered, however, sliding modes introduce significant advantages over linear regulation, regardless of whether constant or scheduled gains are used. Using scheduled, or even better, adaptive gains in conjunction with SM regulators in a max–min architecture is a promising possibility. SISO versions of SM control for second-order systems are available, which require references for the output only, not for the states. Moreover, adaptive versions have been developed. Preliminary work by the author and his graduate students suggest that adaptive SM control in combination with the max–min architecture represents a marked improvement over the standard approach to engine control: robust output regulation and limit management are achieved with low-complexity controllers, without the need for scheduling tables.

In regards to engine limit protection, MPC is as effective as the max–min arrangement with SM regulators, provided the quadratic program remains feasible and its computation can be completed in a timely manner. Computational burden has so far prevented MPC from being deployed to FADEC systems.

H. Richter, *Advanced Control of Turbofan Engines*, DOI 10.1007/978-1-4614-1171-0,

Advances in adaptive control are being made to introduce robustness properties and fast adaptation rates. Application of these novel theories to the GTE control problem have shown encouraging results [98, 99].

Appendix A
Time-Optimal Control of Fan Speed

Classical optimal control can be used to derive insight into the theoretical limits of response speed. The minimum-time problem, also known as time-optimal control problem, although not directly implementable in the nonlinear engine, provides valuable information regarding the theoretical limits of engine response speed as a function of constraints in input variables. In this section, the controller yielding the minimum transfer time to a steady operating condition is developed, assuming that upper and lower limits are placed in the fuel flow.

The linearized engine model with fuel flow input and fan speed output has a constant structure across flight conditions, namely a transfer function with one real zero and two poles:

$$\frac{\Delta N_f(s)}{\Delta W_F(s)} = \frac{k(s+a)}{(s+p_1)(s+p_2)} \tag{A.1}$$

where ΔN_f and ΔW_F represent the deviations in fan speed and fuel flow from the steady point at which linearization was performed. The values of k, a, p_1 and p_2 vary according to flight condition. The following analysis assumes that p_1 and p_2 are real. This assumption is satisfied by the linearized models arising from all 14 flight conditions in CMAPSS-1, and for most flight conditions in CMAPSS-40k.

Due to the relatively simple form of transfer function (A.1), the derivation of the time-optimal law is possible by analytical means. As it has been well established [82], the optimal control strategy to minimize the transition time between two states under bounded control is a *bang-bang* law. That is, either the maximum or the minimum control is used, the transitions being dictated by the current location of the state relative to a *switching surface*. Theory shows that the maximum number of switchings between minimum and maximum control equals the order of the system minus one, provided the system is linear and time-invariant. Therefore, at most one switching is used in the time-optimal law for transfer function (A.1). The switching surface is given by a curve in the plane of the states.

H. Richter, *Advanced Control of Turbofan Engines*, DOI 10.1007/978-1-4614-1171-0,

A.1 Time-Optimal Regulator

The time-optimal regulator is designed to transfer the state from an arbitrary initial condition to the origin. Its construction is a necessary step in obtaining the time-optimal setpoint tracker. Let u denote the input, i.e., $U(s) = W_F(s)$. Suppose that fuel flow is bounded by maximum and minimum values as follows:

$$U_{\min} \leq u \leq U_{\max}$$

Suppose that the initial state is given by x_0. The switching curve is given by the set of points x_0 which result in the system reaching the origin in some time t under constant control. The switching curve, thus, has two branches, one corresponding to $u = U_{\min}$ and the other corresponding to $u = U_{\max}$.

The switching curve may be constructed by several methods. For very simple systems, the state equations are solved analytically for constant input. Time is then eliminated from the equations and the resulting locus forced to pass through the origin.

When time cannot be eliminated from the state solutions due to algebraic complexity, a numerical routine may be written to obtain the desired locus. Explicit state solutions for constant control are still required.

As a third alternative, time may be eliminated from the state equations *before* solving them. An ordinary linear differential equation is then solved to give the desired switching curve.

In this section, the last two methods were used. An observable canonical realization of (A.1) is given by the following state-space matrices:

$$A = \begin{bmatrix} -(p_1 + p_2) & 1 \\ -p_1 p_2 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} k \\ ka \end{bmatrix} \tag{A.2}$$

State x_1 in this realization equals the output ΔN_f. Note that the second state is *not* equal to the acceleration.

The general solution of a linear state-space system is given by

$$x(t) = \mathrm{e}^{\mathrm{A}t} x_0 + \int_0^t \mathrm{e}^{\mathrm{A}\tau} B u(t-\tau) \mathrm{d}\tau \tag{A.3}$$

The matrix exponential, known as *transition matrix* $\phi(t) = \mathrm{e}^{\mathrm{A}t}$, may be obtained by a variety of methods, as discussed in Sect. 9.1. The Laplace formula in (9.7) was used in this case. Performing the indicated operations the transition matrix is found to be

$$\phi(t) = \begin{bmatrix} -\frac{p_1}{p_2-p_1}\mathrm{e}^{-p_1 t} & -\frac{p_2}{p_1-p_2}\mathrm{e}^{-p_2 t} & \frac{1}{p_2-p_1}\mathrm{e}^{-p_1 t} & +\frac{1}{p_1-p_2}\mathrm{e}^{-p_2 t} \\ -\frac{p_1 p_2}{p_2-p_1}\mathrm{e}^{-p_1 t} & -\frac{p_1 p_2}{p_1-p_2}\mathrm{e}^{-p_2 t} & \frac{p_2}{p_2-p_1}\mathrm{e}^{-p_1 t} & +\frac{p_1}{p_1-p_2}\mathrm{e}^{-p_2 t} \end{bmatrix} \tag{A.4}$$

The integration indicated in (A.3) is carried out using $u(t) = U$, where U is some constant. Calling $R(t, U) = \int_0^t \mathrm{e}^{A\tau} BU \mathrm{d}\tau$, we get

$$\begin{aligned} R_1(t, U) &= kU((\mathrm{e}^{-p_1 t} - 1)(1 - a/p_1) + (\mathrm{e}^{-p_2 t} - 1)(a/p_2 - 1))/(p_2 - p_1) \\ R_2(t, U) &= kU((\mathrm{e}^{-p_1 t} - 1)p_2(1 - a/p_1) + (\mathrm{e}^{-p_2 t} - 1)p_1(a/p_2 - 1))/(p_2 - p_1) \end{aligned}$$

The solution to the state equations under constant input can then be rewritten as

$$x(t) = \phi(t)x_0 + R(t, U)$$

To obtain the switching curve numerically, simply assign values to t and solve

$$0 = \phi(t)x_0 + R(t, U)$$

for x_0, using the two constant values for U. Note that the above equation is linear in x_0 and that $\phi(t)$ is invertible provided there are no poles at the origin. Also note that time may not be eliminated from (A.3). However, an implicit, closed-form, equation for the switching curve may be obtained by integration of the state differential equations. This form is preferable for control implementation purposes. The state equations of the linearized model in observable canonical form are:

$$\dot{x}_1 = -(p_1 + p_2)x_1 + x_2 + ku \tag{A.5}$$

$$\dot{x}_2 = -p_1 p_2 x_1 + aku \tag{A.6}$$

Time may be formally eliminated by using the chain rule to write

$$\frac{\mathrm{d}x_2}{\mathrm{d}x_1} = \frac{-p_1 p_2 x_1 + aku}{-(p_1 + p_2)x_1 + x_2 + ku} \tag{A.7}$$

Equation (A.7) can be solved for constant u. The process is lengthy, as it entails three transformations and several indefinite integrations. The solution technique used in this work can be found in [100]. The implicit solution is given by

$$\begin{aligned} s(x_1, x_2) &= \ln|z_1| - \ln C + \frac{1}{p_1 - p_2} \\ &\quad \times \left[(p_1 + p_2) \ln \left| \frac{z - p_1}{z - p_2} \right| - p_1 \ln|z - p_1| + p_2 \ln|z - p_2| \right] = 0 \end{aligned} \tag{A.8}$$

where $z_1 = x_1 - \alpha$, $z = \frac{x_2 - \beta}{z_1}$, with $\alpha = aku/(p_1 p_2)$ and $\beta = ku((p_1 + p_2)a - p_1 p_2)/(p_1 p_2)$. The values of C are obtained by setting $x_1 = 0$ and $x_2 = 0$ with each value of control bound. The switching curve is then defined as $s(x_1, x_2) = 0$.

A.2 Example

To illustrate on the shape of the switching curve, consider the linearized model obtained at sea level, standard day, and zero Mach number, with TRA=80 degrees. Nominal fan speed at those conditions is 2,224 rpm, corresponding to a steady fuel flow of 5.511 pps. These conditions are labeled as FC10 in [2] and Table 2.3. Figure A.1 shows the calculated switching curve. Suppose the initial state is $x_0 = [100\ 0]^{\mathrm{T}}$. Since the state of the linearized model is incremental, this implies that the initial fan speed is 2224 + 100 rpm. The switching curve and sample trajectory were obtained using a value of $U_{\max} = -U_{\min} = 5$. The upper-right branch of the switching curve is the locus of the points from which the origin is reached using $u = U_{\min}$, while the lower-right branch corresponds to $u = U_{\max}$. Minimum control is selected at the initial time, which brings the state to the switching curve at point $[-126.2\ -1352]^{\mathrm{T}}$, at $t = 0.126$. Upon reaching the switching curve, the control is switched to the maximum value, which is held constant until the state reaches the origin. The total time for the maneuver is approximately 0.24 s.

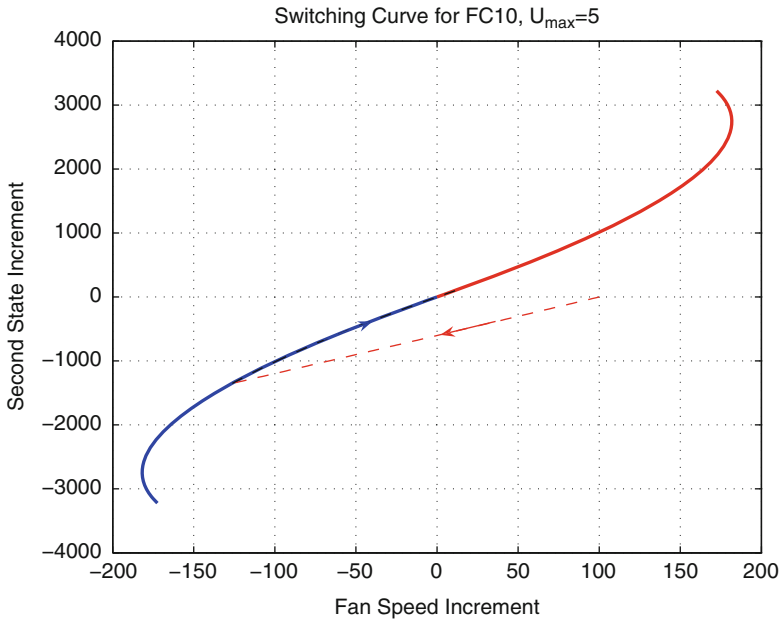

Fig. A.1 Switching curve and sample optimal trajectory

A.3 Minimum-Time Setpoint Tracker

The above regulator is able to transfer the system from any initial state to the origin. We are interested in the transfer from any initial state to an arbitrary steady point. Suppose the target setpoint is defined by states $\bar{x}_1$ and $\bar{x}_2$. Define the setpoint tracking errors as

$$e_1 = \bar{x}_1 - x_1 \tag{A.9}$$

$$e_2 = \bar{x}_2 - x_2 \tag{A.10}$$

If the $[\bar{x}_1 \;\; \bar{x}_2]^{\mathrm{T}}$ defines a steady (equilibrium) point, then the following relations obtained from (A.5) and (A.6) must hold:

$$u_0 = \frac{p_1 p_2 \bar{x}_1}{ak} \tag{A.11}$$

$$x_2 = ku_0 \left(\frac{(p_1 + p_2)a}{p_1 p_2} - 1 \right) \tag{A.12}$$

where u_0 is the required steady input. Using the definitions and the above steady relationships, the tracking error dynamics become

$$\dot{e}_1 = -(p_1 + p_2)e_1 + e_2 + k\tilde{u} \tag{A.13}$$

$$\dot{e}_2 = -p_1 p_2 e_1 + ak\tilde{u} \tag{A.14}$$

where $\tilde{u} = u_0 - u$. That is, when the desired setpoint is an equilibrium point, the tracking error dynamics coincide with those of the plant and the switching curve derivations are valid relative to $\tilde{u}$. Therefore, the time-optimal control for setpoint tracking in terms of $\tilde{u}$ is given by

$$\tilde{u} = \begin{cases} u_0 - \tilde{u}_{\max}, & (e_2 \geq 0 \text{ and } s(e_1, e_2, \tilde{u}_{\min}) < 0) \text{ or } (e_2 < 0 \text{ and } s(e_1, e_2, \tilde{u}_{\min}) > 0) \\ u_0 - \tilde{u}_{\min}, & (e_2 \geq 0 \text{ and } s(e_1, e_2, \tilde{u}_{\min}) > 0) \text{ or } (e_2 < 0 \text{ and } s(e_1, e_2, \tilde{u}_{\min}) < 0) \\ 0, & e_1 = 0 \text{ and } e_2 = 0 \end{cases} \tag{A.15}$$

At implementation time, zero control is used within a small neighborhood of the origin. Since $u = u_0 - \tilde{u}$, the values of $\tilde{u}_{\min}$ and $\tilde{u}_{\max}$ are readily calculated so that u effectively attains its maximum and minimum values during implementation.

It is important to note that $u_{\max}$ must be chosen to be equal or larger than the steady input required to maintain the desired steady output, that is $u_{\max} \geq u_0$ when $u_0 > 0$, and, similarly, $u_{\min} \leq u_0$ when $u_0 < 0$. The time-optimal solution approaches an open-loop step response as the control bounds approach the required steady values.

A.4 Example: Setpoint Tracking

For comparison purposes, the speed transfer corresponding to an increment of 5° in TRA discussed in [69] is simulated under time-optimal control. Unlike [69], no integrator is placed in front of the plant. None is required to achieve zero steady error under bang-bang control and idealized linear plants. A TRA increment of 5° corresponds to a fan speed increase demand of $\bar{x}_1 = 37.6$ rpm. Using (A.11) and (A.12), the values of the second equilibrium state and the input are $\bar{x}_2 = 245.84$ and $u_0 = 0.264$. If the time-optimal law is applied with $u_{\max} = u_0$, we obtain an open-loop step response (no switching) Fig. A.2 shows the response resulting from using $u_{\max} = -u_{\min} =$. As it can be seen, the transfer is completed in 0.48 s. The speed overshoot is about 20%. Of particular importance is the response of the high-pressure turbine outlet temperature, T48. The maximum excursion is about 88°R, which added to the baseline of 1941°R applicable to FC10 gives a maximum value of 2029°R, which is significantly lower than the nominal limit of 2072°R. It can also be observed that the fan acceleration does not exceed the 500 rpm/s limit.

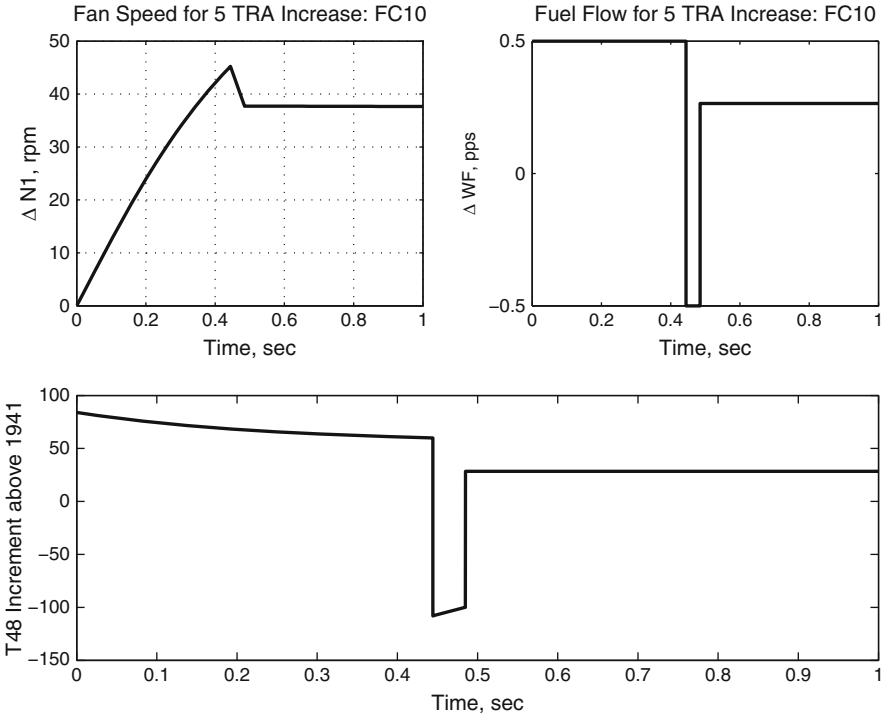

Fig. A.2 Time-optimal response to 5 TRA increase

Incorporation of engine output limits in a classical optimal control framework is cumbersome and leads to numerically intensive solutions. However, the presence of such limits will result in slower transfer times for the fan speed. Therefore, the basic analysis of this section provides an absolute lower-bound on response times, given a set of fuel flow rate bounds.

Appendix B
Representative Linear Model Matrix Listings: 90k Engine

Linearized Models

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

State Vector: $x = [\Delta N_f \ \Delta N_c]^{\mathrm{T}}$
Control Input: $u = \Delta W_F$
Outputs: $y = Cx + Du = [\Delta T_{48} \ \Delta SmHPC]^{\mathrm{T}}$
Find D for each output in the table.

FC01	EPR	T48	PS30	WF/PS30
$A = \begin{bmatrix} -3.8557 & 1.4467 \\ 0.4690 & -4.7081 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0000 \\ 0.0002 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0573 \\ -0.3224 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0514 \\ 0.1912 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0046 \\ -0.0172 \end{bmatrix}$
WF: $B = \begin{bmatrix} 230.6739 \\ 653.5547 \end{bmatrix}$	0.0236	146.3700	20.1147	5.0778

FC02	EPR	T48	PS30	WF/PS30
$A = \begin{bmatrix} -4.1804 & 1.5321 \\ 0.3244 & -4.9290 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0001 \\ 0.0002 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0361 \\ -0.3162 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0542 \\ 0.1969 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0047 \\ -0.0172 \end{bmatrix}$
WF: $B = \begin{bmatrix} 231.8138 \\ 674.5305 \end{bmatrix}$	0.0230	140.0339	20.0431	4.9098

H. Richter, *Advanced Control of Turbofan Engines*, DOI 10.1007/978-1-4614-1171-0,

FC03	EPR	T48	PS30	WF/PS30
$A = \begin{bmatrix} -4.0334 & 1.4777 \\ 0.5872 & -4.6338 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0000 \\ 0.0002 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0492 \\ -0.3267 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0469 \\ 0.1886 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0043 \\ -0.0174 \end{bmatrix}$
WF: $B = \begin{bmatrix} 225.5204 \\ 627.2142 \end{bmatrix}$	0.0222	147.3667	19.5623	5.0602

FC04	EPR	T48	PS30	WF/PS30
$A = \begin{bmatrix} -3.7401 & 1.4001 \\ 0.4752 & -4.5586 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0000 \\ 0.0002 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0620 \\ -0.3214 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0523 \\ 0.1851 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0049 \\ -0.0172 \end{bmatrix}$
WF: $B = \begin{bmatrix} 231.5508 \\ 657.3084 \end{bmatrix}$	0.0246	151.4478	20.1995	5.2629

FC05	EPR	T48	PS30	WF/PS30
$A = \begin{bmatrix} -2.9378 & 1.0937 \\ 0.4149 & -3.4793 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0000 \\ 0.0002 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0572 \\ -0.3081 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0399 \\ 0.1401 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0048 \\ -0.0170 \end{bmatrix}$
WF: $B = \begin{bmatrix} 239.7745 \\ 688.5648 \end{bmatrix}$	0.0352	202.1763	20.9649	7.1323

FC06	EPR	T48	PS30	WF/PS30
$A = \begin{bmatrix} -2.9150 & 1.0362 \\ 0.7871 & -3.4432 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0000 \\ 0.0003 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0496 \\ -0.3524 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0089 \\ 0.1395 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0012 \\ -0.0195 \end{bmatrix}$
WF: $B = \begin{bmatrix} 239.3185 \\ 723.6188 \end{bmatrix}$	0.0520	234.8718	21.1914	8.4488

FC07	EPR	T48	PS30	WF/PS30
$A = \begin{bmatrix} -1.7435 & 0.7462 \\ 0.5080 & -2.1737 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0000 \\ 0.0002 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0244 \\ -0.2665 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0017 \\ 0.0855 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0004 \\ -0.0189 \end{bmatrix}$
WF: $B = \begin{bmatrix} 287.6845 \\ 891.1333 \end{bmatrix}$	0.0461	410.4741	25.5719	16.1801

FC08	EPR	T48	PS30	WF/PS30
$A = \begin{bmatrix} -1.7852 & 0.6006 \\ 0.4678 & -2.1335 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0000 \\ 0.0004 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0427 \\ -0.3263 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0062 \\ 0.0845 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0014 \\ -0.0192 \end{bmatrix}$
WF: $B = \begin{bmatrix} 252.1854 \\ 788.2555 \end{bmatrix}$	0.1046	389.0245	22.7781	14.4589

FC09	EPR	T48	PS30	WF/PS30
$A = \begin{bmatrix} -1.2320 & 0.3757 \\ 0.2003 & -1.4073 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0001 \\ 0.0004 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0143 \\ -0.2806 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0117 \\ 0.0540 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0037 \\ -0.0173 \end{bmatrix}$
WF: $B = \begin{bmatrix} 255.8006 \\ 789.9624 \end{bmatrix}$	0.1396	545.2447	22.8232	20.2336

FC10	EPR	T48	PS30	WF/PS30
$A = \begin{bmatrix} -3.6284 & 1.5373 \\ 0.9017 & -4.6475 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0000 \\ 0.0002 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0058 \\ -0.3480 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0178 \\ 0.1947 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0018 \\ -0.0192 \end{bmatrix}$
WF: $B = \begin{bmatrix} 247.3701 \\ 685.2015 \end{bmatrix}$	0.0204	167.9945	20.9823	5.9613

FC11	EPR	T48	PS30	WF/PS30
$A = \begin{bmatrix} -3.3384 & 1.4517 \\ 0.8547 & -4.2881 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0000 \\ 0.0001 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0334 \\ -0.3062 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0036 \\ 0.1730 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0004 \\ -0.0190 \end{bmatrix}$
WF: $B = \begin{bmatrix} 262.2267 \\ 773.8465 \end{bmatrix}$	0.0170	193.0274	22.7029	7.1444

FC12	EPR	T48	PS30	WF/PS30
$A = \begin{bmatrix} -2.6591 & 1.3162 \\ 0.5251 & -3.2409 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0000 \\ 0.0001 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0807 \\ -0.3435 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0067 \\ 0.1558 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0008 \\ -0.0194 \end{bmatrix}$
WF: $B = \begin{bmatrix} 274.6438 \\ 814.3750 \end{bmatrix}$	0.0145	236.4380	24.2831	9.0547

FC13	EPR	T48	PS30	WF/PS30
$A = \begin{bmatrix} -2.1668 & 1.2163 \\ 0.5305 & -2.7527 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0000 \\ 0.0001 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0561 \\ -0.3109 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0079 \\ 0.1363 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0012 \\ -0.0201 \end{bmatrix}$
WF: $B = \begin{bmatrix} 293.1481 \\ 867.6547 \end{bmatrix}$	0.0118	307.2016	26.4699	12.3309

FC14	EPR	T48	PS30	WF/PS30
$A = \begin{bmatrix} -1.8470 & 0.7489 \\ 0.0996 & -1.4302 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0000 \\ 0.0000 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0252 \\ -0.1936 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0054 \\ 0.0792 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0010 \\ -0.0144 \end{bmatrix}$
WF: $B = \begin{bmatrix} 305.0075 \\ 973.0158 \end{bmatrix}$	0.0093	423.5049	30.4577	18.5904

Appendix C
Representative Linear Model Matrix Listings: 40k Engine

Linearized Models

$$\begin{aligned} \dot{x} &= Ax + Bu + \Gamma w \\ y &= Cx + Du + \Lambda w \end{aligned}$$

State Vector: $x = [\Delta N_f \ \Delta N_c]^{\mathrm{T}}$
Control Input: $u = [\Delta W_F \ \Delta VSV \ \Delta VBV]^{\mathrm{T}}$
Outputs: $y = [\Delta T_{48} \ \Delta SmHPC]^{\mathrm{T}}$
Find the entries of D for each input/output pair in the tables.

A: Ground Idle $A = \begin{bmatrix} -3.3808 & 1.2954 \\ 0.4444 & -3.0501 \end{bmatrix}$	T48 $C^{\mathrm{T}} = \begin{bmatrix} -0.0191 \\ -0.1178 \end{bmatrix}$	SM-HPC $C^{\mathrm{T}} = \begin{bmatrix} 0.0158 \\ -0.0037 \end{bmatrix}$
WF: $B = \begin{bmatrix} 667.8408 \\ 1333.9594 \end{bmatrix}$	289.0525	−10.9483
VSV: $B = \begin{bmatrix} -39.2134 \\ 117.2730 \end{bmatrix}$	0.1332	0.1837
VBV: $B = \begin{bmatrix} -14.2485 \\ -26.8107 \end{bmatrix}$	1.2568	−0.4766

H. Richter, *Advanced Control of Turbofan Engines*, DOI 10.1007/978-1-4614-1171-0,

$$\Gamma^{\mathrm{T}} = \begin{bmatrix} 3229.35 & 1.44 \\ -5799.41 & -22.52 \\ -4131.20 & -23.42 \\ 2617.58 & -3623.81 \\ -857.69 & 763.08 \\ -749.29 & 609.21 \\ -972.53 & 5988.77 \\ 3004.78 & -6507.42 \\ 265.74 & -1023.23 \\ -1402.83 & 8408.06 \\ 576.60 & -2749.05 \\ 3706.27 & 72.97 \\ -2727.40 & 4456.86 \end{bmatrix} \qquad \Lambda^{\mathrm{T}} = \begin{bmatrix} -0.11 & 0.00 \\ 1.70 & -0.00 \\ 1.76 & -0.00 \\ -417.24 & -11.38 \\ -59.25 & 20.54 \\ -74.03 & 19.72 \\ -468.85 & 16.05 \\ -385.27 & -4.06 \\ 41.60 & -2.16 \\ -633.12 & 0.01 \\ 161.07 & 136.04 \\ -5.49 & 0.00 \\ -335.61 & 0.05 \end{bmatrix}$$

B: Flight Idle $A = \begin{bmatrix} -1.1549 & 0.6616 \\ -0.0159 & -0.7976 \end{bmatrix}$	T48 $C^{\mathrm{T}} = \begin{bmatrix} 0.0303 \\ -0.1536 \end{bmatrix}$	SM-HPC $C^{\mathrm{T}} = \begin{bmatrix} 0.0009 \\ 0.0184 \end{bmatrix}$
WF: $B = \begin{bmatrix} 683.7410 \\ 1895.4487 \end{bmatrix}$	702.9648	−41.7502
VSV: $B = \begin{bmatrix} -47.9399 \\ 57.0275 \end{bmatrix}$	13.0597	−1.5734
VBV: $B = \begin{bmatrix} -2.6723 \\ -22.7373 \end{bmatrix}$	−1.2138	−0.0640

$$\Gamma^{\mathrm{T}} = \begin{bmatrix} 774.61 & 0.14 \\ -1619.52 & -8.48 \\ -5595.28 & -28.85 \\ 549.65 & -310.76 \\ -314.65 & 145.97 \\ -749.83 & 334.51 \\ -323.11 & 1936.69 \\ 1353.04 & -1484.78 \\ 109.25 & -517.99 \\ -461.76 & 2840.79 \\ 225.65 & -1269.65 \\ 1069.22 & 10.19 \\ -834.76 & 1069.20 \end{bmatrix} \qquad \Lambda^{\mathrm{T}} = \begin{bmatrix} -0.04 & -0.00 \\ 2.55 & 0.00 \\ 8.66 & 0.00 \\ -158.40 & 6.91 \\ 13.63 & 0.05 \\ 0.30 & 1.78 \\ -324.31 & 17.08 \\ -366.88 & 43.97 \\ 72.22 & -3.22 \\ -459.36 & 0.00 \\ 186.01 & 146.49 \\ -1.65 & 0.00 \\ -172.90 & 0.03 \end{bmatrix}$$

C: Approach	T48	SM-HPC
$A = \begin{bmatrix} -1.8587 & 0.8003 \\ 0.1120 & -1.4456 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.0164 \\ -0.1012 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0134 \\ -0.0017 \end{bmatrix}$
WF: $B = \begin{bmatrix} 758.8063 \\ 1522.1852 \end{bmatrix}$	521.6510	−24.0067
VSV: $B = \begin{bmatrix} -4.7317 \\ 14.9885 \end{bmatrix}$	−0.8507	0.0558
VBV: $B = \begin{bmatrix} -50.5556 \\ -50.8004 \end{bmatrix}$	9.4784	−2.7488

$$\Gamma^{\mathrm{T}} = \begin{bmatrix} 1346.32 & 0.48 \\ -2202.88 & -13.81 \\ -4815.21 & -33.99 \\ 1232.31 & -922.65 \\ -253.04 & 122.88 \\ -175.93 & 150.89 \\ -348.79 & 3056.55 \\ 1915.89 & -2952.48 \\ -23.71 & -500.22 \\ -602.69 & 4501.85 \\ 294.39 & -1467.15 \\ 1906.22 & 17.62 \\ -1831.97 & 2520.80 \end{bmatrix} \quad \Lambda^{\mathrm{T}} = \begin{bmatrix} -0.06 & -0.00 \\ 1.69 & 0.04 \\ 4.15 & 0.11 \\ -259.24 & -3.95 \\ -22.74 & 10.74 \\ -78.80 & 22.12 \\ -393.71 & 16.44 \\ -355.64 & 5.27 \\ 46.08 & -2.16 \\ -560.19 & 0.01 \\ 158.06 & 141.35 \\ -2.19 & 0.00 \\ -313.68 & 0.05 \end{bmatrix}$$

D: Max Cruise	T48	SM-HPC
$A = \begin{bmatrix} -0.7922 & 0.1767 \\ -0.2641 & -0.7087 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.1844 \\ 0.0828 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0169 \\ -0.0040 \end{bmatrix}$
WF: $B = \begin{bmatrix} 639.9616 \\ 1304.5674 \end{bmatrix}$	705.5011	−23.6348
VSV: $B = \begin{bmatrix} -2.1429 \\ 15.2779 \end{bmatrix}$	−3.4862	0.0948
VBV: $B = \begin{bmatrix} -11.7834 \\ 8.2245 \end{bmatrix}$	9.5838	−0.6111

$$\Gamma^{\mathrm{T}} = \begin{bmatrix} 1306.60 & 0.03 \\ -2871.81 & -0.40 \\ -920.00 & -0.07 \\ 547.54 & -746.63 \\ 26.20 & -613.61 \\ -84.34 & -12.52 \\ -337.36 & 2536.27 \\ 763.42 & -2005.82 \\ 74.17 & -303.30 \\ -528.99 & 3666.80 \\ 209.50 & -1090.18 \\ 1751.01 & 9.85 \\ -1263.63 & 2124.35 \end{bmatrix} \quad \Lambda^{\mathrm{T}} = \begin{bmatrix} -0.01 & 0.00 \\ 0.07 & -0.00 \\ 0.01 & -0.00 \\ -107.87 & -15.33 \\ -548.10 & 36.52 \\ -78.55 & 7.95 \\ -469.94 & 14.17 \\ 10.11 & -6.02 \\ 39.85 & -1.46 \\ -657.71 & 0.01 \\ 172.76 & 124.88 \\ -1.76 & 0.00 \\ -381.05 & 0.05 \end{bmatrix}$$

E: Max Climb	T48	SM-HPC
$A = \begin{bmatrix} -2.7539 & 0.8918 \\ -0.1772 & -2.1688 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.1126 \\ -0.0653 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0160 \\ -0.0041 \end{bmatrix}$
WF: $B = \begin{bmatrix} 640.6942 \\ 1253.2251 \end{bmatrix}$	302.9325	−10.0433
VSV: $B = \begin{bmatrix} -29.5063 \\ 104.8643 \end{bmatrix}$	−3.0793	0.2092
VBV: $B = \begin{bmatrix} -49.6910 \\ 17.1234 \end{bmatrix}$	10.7990	−0.8963

$$\Gamma^{\mathrm{T}} = \begin{bmatrix} 2949.29 & 2.26 \\ -5718.23 & -36.75 \\ -3996.15 & -35.99 \\ 2282.37 & -2581.93 \\ -275.23 & 426.38 \\ -815.03 & 1019.16 \\ -922.30 & 6381.18 \\ 2633.97 & -5595.40 \\ 227.18 & -442.27 \\ -1408.07 & 9066.52 \\ 681.88 & -2913.75 \\ 3970.59 & 94.82 \\ -2802.84 & 4729.29 \end{bmatrix} \quad \Lambda^{\mathrm{T}} = \begin{bmatrix} -0.18 & 0.00 \\ 2.86 & -0.00 \\ 2.80 & -0.00 \\ -382.19 & -13.07 \\ -268.89 & 25.74 \\ -93.43 & 18.23 \\ -519.78 & 14.77 \\ -277.55 & -8.18 \\ -45.47 & -1.18 \\ -705.31 & 0.01 \\ 133.56 & 130.33 \\ -7.38 & 0.00 \\ -367.94 & 0.04 \end{bmatrix}$$

F: Max Takeoff	T48	SM-HPC
$A = \begin{bmatrix} -2.8410 & 0.6988 \\ -0.6479 & -1.7524 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} -0.1460 \\ -0.0350 \end{bmatrix}$	$C^{\mathrm{T}} = \begin{bmatrix} 0.0155 \\ -0.0028 \end{bmatrix}$
WF: $B = \begin{bmatrix} 640.7678 \\ 1236.6792 \end{bmatrix}$	278.9871	−9.0952
VSV: $B = \begin{bmatrix} -27.3716 \\ 103.1969 \end{bmatrix}$	−3.8705	0.1971
VBV: $B = \begin{bmatrix} -28.2962 \\ 15.2226 \end{bmatrix}$	6.5006	−0.4516

$$\Gamma^{\mathrm{T}} = \begin{bmatrix} 3144.46 & 1.57 \\ -5319.68 & -25.39 \\ -4707.37 & -33.80 \\ 1933.24 & -1705.43 \\ 285.62 & -982.04 \\ -131.33 & -80.66 \\ -933.15 & 6577.19 \\ 2534.64 & -5371.77 \\ 620.07 & -799.68 \\ -1456.94 & 9484.27 \\ 309.42 & -2823.41 \\ 4260.76 & 89.07 \\ -3040.31 & 4821.35 \end{bmatrix} \qquad \Lambda^{\mathrm{T}} = \begin{bmatrix} -0.11 & 0.00 \\ 1.83 & -0.00 \\ 2.44 & -0.00 \\ -277.43 & -11.18 \\ -318.87 & 24.77 \\ -104.35 & 17.10 \\ -496.25 & 14.59 \\ -205.78 & -6.94 \\ -6.20 & -1.25 \\ -684.59 & 0.01 \\ 127.78 & 129.33 \\ -6.43 & 0.00 \\ -348.04 & 0.04 \end{bmatrix}$$

Appendix D
Matlab Code for Linearized MPC Simulation

```
%MPC example
%CMAPSS-40k linearized at Ground Idle
%3 constrained inputs
%constraints on 2 outputs: T48 and SM-HPC
%Plant matrices for Ground Idle assumed available
%in the workspace as Aa,Ba,Ca,Da
%Plant matrices
A=Aa;B=Ba;
Cconstr=Ca; D=Da;%to be used for constraints
sysCT=ss(A,B,Cconstr,D);
Ts=0.015;
%Discretize
sysDT=c2d(sysCT,Ts,'zoh');
[Adu,Bdu,Cdu_constr,Ddu]=ssdata(sysDT);
%Augment
Ad=[Adu Bdu;zeros(3,2) eye(3)];Bd=[Bdu;eye(3)];
Cd_constr=[Cdu_constr Ddu];Dd_constr=Ddu;
Cd=[1 0 0 0 0]; %to be used for control
n=size(Ad,1); %state dimension
m=size(Bd,2); %input dimension
p=size(Cd,1); %controlled output dimension
%Horizons
ny=7;nu=3;
%Compute P for control
P=Cd*Ad;
for i=1:ny-1,
  P=[P;Cd*Ad^(i+1)];
end
%Compute P for constraints
Pc=Cd_constr*Ad;
for i=1:ny-1,
  Pc=[Pc;Cd_constr*Ad^(i+1)];
end
%Compute H for control
H=zeros(p*ny,m*nu);
for i=1:ny,
    for j=1:i,
```

H. Richter, *Advanced Control of Turbofan Engines*, DOI 10.1007/978-1-4614-1171-0,

```
        H(1+(i-1)*p:i*p,1+(j-1)*m:j*m)=Cd*Ad^(i-j)*Bd;
    end
end
%Retain only the first nu blocks
H=H(:,1:nu*m);
%Compute H for constraints
pc=size(Cd_constr,1); %constrained output dimension
Hc=zeros(pc*ny,m*nu);
for i=1:ny,
    for j=1:i,
        Hc(1+(i-1)*pc:i*pc,1+(j-1)*m:j*m)=Cd_constr*Ad^(i-j)*Bd;
    end
    Hc(1+(i-1)*pc:i*pc,j*m+1:(j+1)*m)=Dd_constr;
end
%Retain only the first nu blocks
Hc=Hc(:,1:nu*m);
%Compute Cc
Cc=zeros(m*nu,m*nu);
for i=1:nu,
    for j=1:i,
        Cc(1+(i-1)*m:i*m,1+(j-1)*m:j*m)=eye(m);
    end
end
%Optimization Weight
lambda=0.01;
r0=100; %controlled output reference
r=100*ones(ny,1);
%Offline computations
Ly=eye(pc);
for i=1:ny-1,
    Ly=[Ly;eye(pc)];
end
y_max=Ly*[300;20]; %Incremental upper-bounds: T48 and SmHPC
y_min=Ly*[-150;-10]; %Incremental lower-bounds: T48 and SmHPC
Lu=eye(m);
for i=1:nu-1,
    Lu=[Lu;eye(m)];
end
u_max=[2;15;0.4]; %Incremental upper-bounds: WF,VSV and VBV
u_min=[-1;-20;-0.5];%Incremental lower-bounds: WF,VSV and VBV
%Initialization
S=H'*H+lambda*eye(m*nu);
%Initial conditions
xa=zeros(n,1);
f=H'*(P*xa-r);
dyp=y_max-Pc*xa;
dym=-y_min+Pc*xa;
uprev=zeros(m,1);
dup=Lu*(u_max-uprev);
dum=-Lu*(u_min-uprev);
M=[Hc;-Hc;Cc;-Cc];
d=[dyp;dym;dup;dum];
xnow=zeros(2,1); %Plant state initialization
ynext=[1 0;Ca]*xnow;
```

```
u=uprev; %Control initialization
simhor=30; %simulation horizon
for k=1:simhor;
   ctrlvec=quadprog(S,f,M,d);
   u_apply=ctrlvec(1:m); %extract first term of optimal sequence
   u=[u uprev+u_apply];  %store control history
   xnext(:,k)=Adu*xnow+Bdu*u(:,end); %update plant
   ynext=[ynext [1 0;Ca]*xnext(:,k)]; %store output history
   xnow=xnext(:,k);
   uprev=u(:,end);
   xa=[xnow;uprev];
   %online MPC reformulation
   f=H'*(P*xa-r);
   dyp=y_max-Pc*xa;
   dym=-y_min+Pc*xa;
   dup=Lu*(u_max-uprev);
   dum=-Lu*(u_min-uprev);
   d=[dyp;dym;dup;dum];
end
t=Ts*[0:simhor];
%Plotting
plot(t,ynext(1,:),'k')
hold on
plot(t,ynext(2,:),'k--')
plot(t,ynext(3,:),'k:')
ylabel('\Delta N_f,\Delta T_{48} and \Delta SmHPC','FontSize',14)
xlabel('Time, sec.', 'FontSize',14)
title('Linearized MPC Simulation: CMPASS-40k Near Ground Idle:
   Outputs', 'FontSize',14)
legend('\Delta N_f','\Delta T_{48}','\Delta SmHPC')
figure(2)
plot(t,u(1,:),'k')
hold on
plot(t,u(2,:),'k--')
plot(t,u(3,:),'k:')
ylabel('\Delta W_F, \Delta VSV and \Delta VBV','FontSize',14)
xlabel('Time, sec.', 'FontSize',14)
title('Linearized MPC Simulation: CMPASS-40k Near Ground Idle:
   Inputs', 'FontSize',14)
legend('\Delta W_F','\Delta VSV','\Delta VBV')
```

Glossary

$\dot{m}$	Mass flow rate
$\dot{m}_{cr}$	Corrected mass flow rate
η_i	Ideal cycle efficiency
$\arg\min_{x \in S} \{ f(x) \}$	Value of x taken from set S which minimizes $f(x)$
F_n	Net thrust
M	Mach number
N_c	Core speed
N_f	Fan speed
$N_{f,cr}$	Corrected fan speed
W_F	Fuel flow rate
CMAPSS	Commercial Modular Aeropropulsion System Simulation
EPR	Engine Pressure Ratio
FADEC	Full-Authority Digital Engine Controller
FMV	Fuel Metering Valve
GTE	Gas Turbine Engine
HPC	High-Pressure Compressor
HPT	High-Pressure Turbine
LMI	Linear Matrix Inequality
LPC	Low-Pressure Compressor
LPT	Low-Pressure Turbine
LPV	Linear Parameter-Varying
LQR	Linear Quadratic Regulator
MRAC	Model Reference Adaptive Control
PD	Proportional-Derivative
PI	Proportional-Integral
PLA	Power Lever Angle
SISO	Multi-Input, Multi-Output
SISO	Single-Input, Single Output
SM	Stall margin
SMC	Sliding Mode Control

H. Richter, *Advanced Control of Turbofan Engines*, DOI 10.1007/978-1-4614-1171-0,

SmHPC	HPC Stall Margin
TRA	Throttle Resolver Angle
VBV	Variable Bleed Valve
VSV	Variable Stator Vane
ZOH	Zero-Order Hold

References

1. G. Van Wylen, R. Sontag, *Fundamentals of Classical Thermodynamics* (Wiley, New York, 1986)
2. D. Frederick, J. DeCastro, J. Litt, User's Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS). Tech. rep., NASA TM-2007-215026, Glenn Research Center, Cleveland, Ohio (2007)
3. Rev. D SAE AS-755, *Aircraft Propulsion System Performance Station Designation and Nomenclature* (SAE International, USA, 2004)
4. General Electric Aeroengines. The GE90 engine family. http://www.geaviation.com/engines/commercial/ge90/. Accessed May 2010
5. SNECMA. M88-2 technical sheet. http://www.snecma.com/IMG/pdf/M88-2_ang-2.pdf. Accessed May 2010
6. The Aviation Herald. Incident: US Airways A320 near Newark on Jan 13th 2009, compressor stall. http://avherald.com/h?article=413aaea4&opt=0. Accessed June 2010
7. F. Moore, E. Greitzer, A theory of post-stall transients in axial compression systems: Part I – development of equations. Trans. ASME, J. Eng. Gas Turbines Power **108**, 68 (1986)
8. J. Baillieul, S. Dahlgren, B. Lehman, Nonlinear control designs for systems with bifurcations with applications to stabilization and control of compressors. In *Proc. 34th Conference on Decision and Control* (1995), pp. 3062–3067
9. E. Greitzer, Surge and stall in axial flow compressors: Part I – theoretical compression system model. Trans. ASME, J. Eng. Power **98**(2), 190 (1976)
10. J. Simon, L. Valavani, A Lyapunov-based nonlinear control scheme for stabilizing a basic compression system using a close-coupled control valve. In *Proc. 1991 American Control Conference* (1991), pp. 2398–2406
11. D. Mavris, R. Denney, Optimal robust matching of engine models to test data. Tech. rep., Final Report, Air Force Office of Scientific Research, Grant N. FA9550-06-1-0299 (2009)
12. J. Rayleigh, R. Lindsay, *The Theory of Sound, Volume 2*. Dover Classics of Science of Mathematics (Dover, UK, 1976)
13. A. Annaswamy, A. Ghoniem, Active control of combustion instability: Theory and practice. IEEE Control Systems Magazine **22**(6), 37 (2002)
14. L. Jaw, *Aircraft Engine Controls : Design, System Analysis, and Health Monitoring* (AIAA Education Series, 2009)
15. V. Arkov, et al., System identification techniques applied to aircraft gas turbine engines. Annual Reviews in Control **24**, 67 (2000)
16. L. Ljung, *System Identification: Theory for the User* (Prentice-Hall, Upper Saddle River, NJ, 1987)
17. A. Sage, J. Melsa, *System Identification* (Academic International Press, New York, 1971)

H. Richter, *Advanced Control of Turbofan Engines*, DOI 10.1007/978-1-4614-1171-0,

18. G. Kulikov, H. Thompson, *Dynamic Modeling of Gas Turbines - Identification, Simulation, Condition Monitoring and Optimal Control* (Springer, London, 2004)
19. A. Volponi, et al., The use of Kalman filter and neural network methodologies in gas turbine performance diagnostics: A comparative study. ASME Journal of Engineering for Gas Turbines and Power **125**, 917 (2003)
20. T. Kobayashi, D. Simon, Evaluation of an enhanced bank of Kalman filters for in-flight aircraft engine sensor fault diagnostics. Journal of Engineering for Gas Turbines and Power **127**, 497 (2005)
21. T. Kobayashi, D. Simon, Hybrid neural-network genetic-algorithm technique for aircraft engine performance diagnostics. Journal of Propulsion and Power **21**, 751 (2005)
22. D. Simon, D. Simon, Constrained Kalman filtering via density function truncation for turbofan engine health estimation. Int. J. Syst. Sci. **41**(2), 159 (2010)
23. R. May, J. Csank, J. Litt, T. Guo, Commercial Modular Aero-Propulsion System Simulation 40k (CMAPSS-40k) User's Guide. Tech. rep., NASA TM-2010-216831, Glenn Research Center, Cleveland, Ohio (2010)
24. M. Henriksson, T. Grönstedt, Reduced order thrust estimation for a jet engine. In *Proc. ASME Turbo Expo 2005: Power for Land, Sea, and Air* (2005)
25. J. Litt, An optimal orthogonal decomposition method for Kalman filter-based turbofan engine thrust estimation. In *Proc. 50th ASME International Gas Turbine and Aeroengine Congress and Exposition* (2005)
26. R. Dorf, R. Bishop, *Modern Control Systems* (Pearson, UK, 2008)
27. J. Edmunds, Control system design and analysis using closed-loop Nyquist and Bode arrays. Int. J. Contr. **30**(5), 773 (1979)
28. M. Green, D. Limebeer, *Linear Robust Control* (Prentice-Hall, Upper Saddle River, NJ, 1995)
29. S. Skogestad, I. Postlethwaite, *Multivariable Feedback Control-Analysis and Design* (Wiley, New York, 2005)
30. L. Zhou, J. Doyle, K. Glover, *Robust Optimal Control* (Prentice-Hall, Upper Saddle River, NJ, 1995)
31. S. Boyd, L. El Ghaoui, E. Feron, V. Balakrishnan, Linear matrix inequalities in system and control theory. In *Studies in Applied Mathematics*, vol. 15 (Society for Industrial and Applied Mathematics, 1994)
32. G. Strang, *Introduction to Linear Algebra* (Wellesley-Cambridge, Cambridge, UK 1993)
33. M. Johansson, Piecewise linear control systems. In *Lecture Notes in Control and Information Sciences*, vol. 284 (Springer, London, 2003)
34. G. Goodwin, S. Graebe, M. Salgado, *Control System Design* (Prentice-Hall, Upper Saddle River, NJ, 2001)
35. P. Dorato, C. Abdallah, V. Cerone, *Linear Quadratic Control: An Introduction* (Krieger Publishing Company, USA, 2000)
36. H. Kwakernaak, R. Sivan, *Linear Optimal Control Systems* (Wiley, New York, 1972)
37. L. El Ghaoui, S. Niculescu, Advances in linear matrix inequality methods in control. In *Advances in Design and Control* (Society for Industrial and Applied Mathematics, 2000)
38. J. Shamma, M. Athans, Gain scheduling: Potential hazards and possible remedies. In *Proc. 1991 American Control Conference* (1991), pp. 516–521
39. D. Simon, Propulsion Diagnostic Method Evaluation Strategy ProDiMES User's Guide. Tech. rep., NASA TM-2010-215840, Glenn Research Center, Cleveland, Ohio (2010)
40. G. Wolodkin, G.J. Balas, W. Garrard, Application of parameter-dependent robust control synthesis to turbofan engines. AIAA J. Guid. Contr. Dynam. **22**(6), 833 (1999)
41. G. Golub, C.F. Van Loan, *Matrix Computations* (Johns Hopkins University Press, USA, 1996)
42. E. Kamen, P. Khargonekar, On the control of linear systems whose coefficients are functions of parameters. IEEE Trans. Automat. Contr. **AC-29**(1), 25 (1984)
43. P. Apkarian, P. Gahinet, A convex characterization of gain-scheduled H_∞ controllers. IEEE Trans. Automat. Contr. **AC-40**(5), 853 (1995)

44. A. Packard, M. Kantner, Gain scheduling the LPV way. In *Proc. 35th Conference on Decision and Control* (1996), pp. 3938–3941
45. G. Balas, Linear parameter-varying control and its application to a turbofan engine. Int. J. Robust Nonlinear Contr. **12**(9), 763 (2002)
46. D. Henrion, M. Šebek, V. Kučera, Positive polynomials and robust stabilization with fixed-order controllers. IEEE Trans. Automat. Contr. **48**(7), 1178 (2003)
47. W. Gilbert, et al., Polynomial lpv synthesis applied to turbofan engines. Contr. Eng. Pract. **18**, 1077 (2010)
48. K. Åstrom, B. Wittenmark, *Adaptive Control* (Prentice-Hall, Upper Saddle River, NJ, 1994)
49. P. Ioannou, J. Sun, *Robust Adaptive Control* (Prentice-Hall, Upper Saddle River, NJ, 1995)
50. B. Anderson, Failures of adaptive control theory and their resolution. Comm. Inform. Syst. **5**(1), 1 (2005)
51. M. Grimble, Implicit and explicit LQG self-tuning controllers. Automatica **20**, 661 (1984)
52. A. Feuer, A. Morse, Adaptive control of single-input, single-output linear systems. IEEE Trans. Automat. Contr. **23**(4), 557 (1978)
53. V. Utkin, *Sliding Modes in Control Optimization* (Springer, London, 1992)
54. H. Khalil, *Nonlinear Systems* (Prentice-Hall, Upper Saddle River, NJ, 1996)
55. C. Edwards, S. Spurgeon, *Sliding Mode Control: Theory and Applications* (Taylor and Francis, New York, 1998)
56. V. Utkin, K. Young, Methods for constructing discontinuity planes in multidimensional variable structure systems. Autom. Rem. Contr. **39**, 1466 (1978)
57. T. Kuo, et al., Adaptive sliding mode control with pid tuning for uncertain systems. Eng. Lett. **16**(3) (2008)
58. A. Glattfelder, W. Schaufelberger, *Control systems with input and output constraints* (Springer, London, 2003)
59. H. Richter, A multi-regulator sliding mode control strategy for output-constrained systems, Automatica **47**, 2251–2259 (2011)
60. F. Blanchini, Set invariance in control. Automatica **35**, 1747 (1999)
61. M. Nagumo, Über die Lage der Integralkurven gewöhnlicher Differentialgleichungen. Proc. Physico-Math. Soc. Jpn **24**, 272 (1942)
62. B. O'Dell, E. Misawa, Semi-ellipsoidal controlled invariant sets for constrained linear systems. ASME J. Dyn. Syst. Meas. Contr. **124**, 98 (2002)
63. A. Bemporad, Reference governor for constrained nonlinear systems. IEEE Trans. Automat. Contr. **43**(3), 415 (1998)
64. H. Spang, H. Brown, Control of jet engines. Contr. Eng. Pract. **7**, 1043 (1999)
65. W. Brogan, *Modern Control Theory* (Prentice-Hall, Upper Saddle River, NJ, 1991)
66. K. Åstrom, T. Hägglund, *PID Controllers: Theory, Design and Tuning* (Instrument Society of America, 1995)
67. M. Branicky, Analysis of continuous switching systems: Theory and examples. In *Proc. American Control Conference* (1994), pp. 3110–3114
68. A. Foss, A criterion to assess stability of a lowest-wins control strategy. IEE Proc. D **128**(1), 1 (1981)
69. J. Litt, D. Frederick, T.H. Guo, The case for intelligent propulsion control for fast engine response. In *Proc. AIAA Infotech Aerospace Conference, AIAA paper AIAA-2009-1876* (2009)
70. J. Slotine, W. Li, *Applied Nonlinear Control* (Prentice-Hall, Upper Saddle River, NJ, 1990)
71. A. Martucci, A. Volponi, Fuzzy fuel flow selection logic for a real-time embedded full authority digital engine control. ASME J. Eng. Gas Turbines Power **125**(4), 909 (2003)
72. J. Litt, D. Frederick, T. Guo, The case for intelligent propulsion control for fast engine response. Tech. rep., NASA TM-2009-215668, Glenn Research Center, Cleveland, Ohio (2009)
73. Richter, H. and Litt, J.S., A novel controller for gas turbine engines with aggressive limit management, Joint Propulsion Conference, San Diego, California, August 2011.
74. B. Brunell, R. Bitmead, A. Connolloy, Nonlinear model predictive control of an aircraft gas turbine engine. In *Proc. 41 IEEE Conference on Decision and Control* (2002), pp. 4649–4651

75. J. DeCastro, Rate-based model predictive control of turbofan engine clearance. In *42 AIAA/ASME/ASME/ASEE Joint Propulsion Conference and Exhibit* (2006)
76. H. Richter, A. Singaraju, J. Litt, Multiplexed predictive control of a large commercial turbofan engine. AIAA J. Guid. Dynam. Contr. **31**(2), 273 (2008)
77. H.A. van Essen, H.C. de Lange, Nonlinear model predictive control experiments on a laboratory gas turbine installation. J. Eng. Gas Turbines Power-Trans. Asme **123** (2001)
78. E. Camacho, C. Bordons, *Model Predictive Control* (Springer, London, 2003)
79. J. Rossiter, *Model-Based Predictive Control* (CRC Press, Boca Raton, FL, 2003)
80. K. ÅAstrom, B. Wittenmark, *Computer-Controlled Systems: Theory and Design* (Prentice-Hall, Upper Saddle River, NJ, 1996)
81. G. Franklin, D. Powell, M. Workman, *Digital Control of Dynamic Systems* (Addison-Wesley, USA, 1992)
82. D. Kirk, *Optimal Control Theory* (Prentice-Hall, Upper Saddle River, NJ, 1970)
83. A. Bemporad, et al., The explicit quadratic regulator for constrained systems. Automatica **38**(1), 3 (2002)
84. R. Cagienard, et al., Move blocking strategies in receding horizon control. In *Proceedings of the 43rd IEEE Conference on Decision and Control* (2004)
85. K. Ling, et al., Multiplexed model predictive control. In *Proceedings of the 16th IFAC World Congress* (2005)
86. J. Sun, et al., A stable block model predictive control with variable implementation horizon. In *Proceedings of the 2005 American Control Conference* (2005)
87. S. Qin, T. Badgwell, An overview of industrial model predictive control technology. In *Fifth AIChE International Conference on Chemical Process Control* (1997), pp. 232–256
88. J. Fuller, A. Kumar, R. Millar, Adaptive model-based control of aircraft propulsion systems: Status and outlook for naval aviation applications. In *ASME Paper GT2006-90241* (2006)
89. A. Bemporad, F. Borrelli, M. Morari, Model predictive control based on linear programming – the explicit solution. IEEE Trans. Automat. Contr. **47**(12), 1974 (2002)
90. H. Genceli, M. Nikolau, Robust stability analysis of constrained l1 norm model predictive control. AIChE J. **39**(12), 1954 (1993)
91. E. Kerrigan, J. Maciejowski, Robustly stable feedback min-max model predictive control. In *Proceedings of the 2003 American Control Conference* (2003)
92. C. Rao, et al., Application of interior-point methods to model predictive control. J. Optim. Theory Appl. **99**(3), 723 (1998)
93. G. De Nicolao, Cyclomonotonicity and stabilizability properties of solutions of the difference periodic Riccati equation. IEEE Trans. Automat. Contr. **37**(9), 1405 (1992)
94. W. Yan, R. Bitmead, Periodic receding horizon lq regulators for discrete systems. In *Proceedings of the 30th IEEE Conference on Decision and Control* (1991)
95. D. Mayne, H. Michalska, Receding horizon control of nonlinear systems. IEEE Trans. Automat. Contr. **35**(7), 814 (1990)
96. D. Mayne, et al., Constrained model predictive control: Stability and optimality. Automatica **36**(6), 789 (2000)
97. H. Michalska, D. Mayne, Robust receding horizon of constrained linear systems. IEEE Trans. Automat. Contr. **38**(11), 1623 (1993)
98. N. Hovakimyan, C. Cao, $\mathcal{L}_1$ *Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation (Advances in Design and Control)* (Society for Industrial and Applied Mathematics, 2010)
99. A. Thompson, J. Hacker, C. Cao, Adpative engine control in the presence of output limits. In *Proc. AIAA Infotech Aerospace Conference* (2010)
100. L. Råde, B. Westergren, *Mathematics Handbook for Science and Engineering* (Springer, London, 2004)

Index

H. Richter, *Advanced Control of Turbofan Engines*, DOI 10.1007/978-1-4614-1171-0,