

Active Fault Tolerant Model Predictive Control of a Turbofan
Engine using C-MAPSS40k

A Thesis

Presented in Partial Fulfillment of the Requirements for the Degree
Master of Science in the Graduate School of The Ohio State
University

By

Deepak Chaitanya Saluru, B.S.

Graduate Program in Aeronautical and Astronautical Engineering

The Ohio State University

2012

Master's Examination Committee:

Dr. Rama Krishna Yedavalli, Advisor

Dr. Igor Adamovich

© Copyright by
Deepak Chaitanya Saluru
2012

Abstract

Aircraft engine control is a crucial component for the safe and stable operation of gas turbine engines which are complex nonlinear systems. As engines have evolved to higher capabilities it is crucial to update the control strategy to ensure maximum functionality of the engine. Current industrial baseline controllers are based in the Proportional-Integral-Derivative (PID) control scheme along with individual limit controllers having critically damped responses housed in the min-max architecture.

In light of the distributed engine control architecture that exploits digital electronics and hence higher on-board computational capabilities, the baseline controller is replaced by a Model Predictive Control (MPC) law with on-line optimization. MPC is a model based control technique that can handle complex constrained dynamics thus allowing the incorporation of component faults in the design process of the controller. Component faults occur during an engine's operation mainly due to fan blade-shroud rubbing, structural wear and tear and foreign object ingestion thus affecting the engine performance.

Simulations on the Linear Time Invariant (LTI) as well as the nonlinear turbofan engine of the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS40k) tool are carried out. In the presence of a component fault, active fault tolerant control using the multi-model MPC approach is applied by switching between the MPC blocks, each using its respective LTI reference model. The control of both

the fan speed as well as the thrust for a demand profile in the Power Level Angle (PLA) is investigated and the MPC performance is compared with that of the PID controller demonstrating the successful replacement of the baseline controller with an on-line fault tolerant MPC. The thrust control approach using MPC consumes lesser fuel when compared with the fan speed control approach.

”For once you have tasted flight, you will always walk the earth with your eyes turned skyward, for there you have been and there you long to return” - Leonardo da Vinci.

Acknowledgments

Firstly, I would like to thank Dr. Yedavalli for providing the opportunity to work in his group and participate in research that is of high importance to the jet engine community. During a time when it was difficult in finding research due to reasons such as a tight budget, different undergraduate background and being a masters student, he showed faith and provided me with a research opportunity that made this thesis possible and also went through the extra trouble obtaining funding in these tough times the Aerospace community is going through.

Secondly, if not for Rohit Belapurkar I would still be sitting and debugging a Simulink model which I was doing in early January. His expertise and vast knowledge in addition to helping me finish my thesis, have enlightened me on what it takes to conduct high quality research. From the start, he had outlined and carefully planned the research project into simpler tasks, helping me implement them with great progress. I have learnt a great deal from him and will take this experience with me a long way in my career wishing him the very best in his.

Thirdly, I would like thank Dr. Jonathan Litt of NASA Glenn Research Center and Dr. Ryan May of ASRC Corp. for there input and guidance regarding the working and science behind C-MAPSS40k. Their prompt and accurate guidance have helped me understand C-MAPSS40k which is a remarkable simulation tool of a turbofan engine.

Next, I would also like to acknowledge the guidance, support and insight received from Dr. Gaitonde and thank him for introducing me to LaTeX. I would also like to thank Dr. Adamovich for being a member of my thesis committee. If it wasn't for Dr. Bons, I would never have realized how much I did not know and how much I had to step up to achieve and overcome the extremely steep learning curve. Most importantly, I'd like to thank Bryson Sullivan, Stuart Benton and Kalyan Goparaju for their help during coursework, guidance regarding graduate research and helping my transition from electronics to Aerospace engineering. If not for Google, MATLAB and LaTeX...I feel any kind of research is impossible. So kudos to the Apollo team for achieving so much without these tools.

To my buddies - Mark Lambach and John Slone, raising canes was fun and it was great to have met you guys and all the other blokes here in OSU. Finally I'd like to thank my parents, especially my dad for his guidance throughout and setting me straight whenever required. OSU has shown me just how difficult and frustrating it can get sometimes and to be persistent with fire in the belly brings me satisfaction which is something I learnt from my role model, the great Ricky Ponting, the greatest and finest batsmen and Australian test captain..... *Ad astra per aspera !!*

Vita

January 3, 1988 Born - Albuquerque, NM, USA

2010 B.S. Electronics and Communications
Engineering, VIT University, India

Publications

Research Publications

Saluru, D., Yedavalli, R., 2012. “Fault Tolerant Model Predictive Control of a Turbofan Engine using C-MAPSS40k”. *AIAA Region III Student Conference*, Ann Arbor, MI (Best Paper Award).

Saluru, D., Yedavalli, R., Belapurkar, R., 2012. “Active Fault Tolerant Model Predictive Control of a Turbofan Engine using C-MAPSS40k”. *5th Annual ASME Dynamic Systems and Control Conference (DSCC)*, Fort Lauderdale, FL, October 17-19.

Saluru, D., Yedavalli, R., Belapurkar, R., “Active Fault Tolerant Model Predictive Control of a Turbofan Engine using C-MAPSS40k”. *AIAA Journal of Guidance, Control and Dynamics* (under review)

Fields of Study

Major Field: Aeronautical and Astronautical Engineering

Studies in:

Aircraft & Spacecraft Control	Prof. Yedavalli
Rocket Propulsion	Prof. Gaitonde
Jet Propulsion, Viscous Flows	Prof. Bons

Table of Contents

	Page
Abstract	ii
Dedication	iv
Acknowledgments	v
Vita	vii
List of Figures	x
1. Introduction	1
2. Engine Control Systems	4
2.1 Introduction	4
2.2 Operation of Engine Control Systems	5
2.3 Evolution of Engine Control Systems	6
2.4 Recent Advancements in Engine Control	7
2.5 Transition in Control Architecture	9
3. Overview of C-MAPSS40k	13
3.1 Introduction	13
3.2 C-MAPSS40k GUI	14
3.2.1 Closed loop interface	14
3.2.2 Linearization routine	14
3.3 C-MAPSS40k Controller Design	15
3.4 C-MAPSS40k Engine	15

4.	Model Predictive Control	20
4.1	Motivation and Problem Formulation	20
4.2	Overview of MPC	21
4.3	MPC Analogy	23
4.4	Recent advancements in MPC	24
4.5	High Speed Applications of MPC	25
4.6	MPC in Nonlinear Applications	26
4.7	MPC in Aerospace Applications	28
4.8	Fault Tolerant Control using MPC	29
4.9	MPC Simulation Tool	31
5.	Linear Simulations of MPC on C-MAPSS40k	32
5.1	Linear Simulations	32
5.1.1	Linearization	32
5.1.2	Realization of MPC	34
5.1.3	Fan Speed Control Simulations	37
5.1.4	Thrust Control Simulations	46
5.2	Nonlinear Simulations	52
5.2.1	Nonlinear Fan Speed Control Simulations	58
5.2.2	Nonlinear Thrust Control Simulations	70
6.	Conclusions and Future Work	83
7.	Appendix I : MPC Source Code	86
	Bibliography	90

List of Figures

Figure	Page
2.1 (a) Point-to-point analog harness interfacing between FADEC and system effectors in a star topology (b) FADEC communication with system effectors over a common digital network in a ring topology. . . .	11
3.1 Closed loop interface of the C-MAPSS40k GUI.	17
3.2 Linearization interface of the C-MAPSS40k GUI.	18
3.3 MIN-MAX architecture of the C-MAPSS40k baseline controller	19
3.4 C-MAPSS40k engine with numbered stations.	19
4.1 Control trajectories in a MPC optimization problem ($N_U < N_P$). . . .	22
4.2 Car driving MPC analogy.	24
5.1 Effect of different prediction horizons with $N_U = 1$	35
5.2 MPC simulation in MATLAB/Simulink for N_f control.	36
5.3 Comparison between MPC and PID for nominal conditions	37
5.4 Comparison between MPC and PID for nominal conditions	38
5.5 Comparison between MPC and PID for nominal conditions	38
5.6 Matched and mismatched conditions between the controller and LTI engine	40
5.7 Matched MPC and PID for a fan efficiency of 80% (FE=-0.2)	41

5.8	Matched MPC and PID for a fan efficiency of 80% (FE=-0.2)	42
5.9	Matched MPC and PID for a fan efficiency of 80% (FE=-0.2)	43
5.10	FTMPC for 22% degradation using a 20% degradation reference model	44
5.11	FTMPC for 22% degradation using a 20% degradation reference model	44
5.12	FTMPC for 22% degradation using a 20% degradation reference model	45
5.13	FTMPC for 18% degradation using a 20% degradation reference model	45
5.14	MPC simulation in MATLAB/Simulink for Thrust control.	47
5.15	Change in thrust due to faults in C-MAPSS40k using fan speed control	47
5.16	Thrust comparison between MPC and PID for nominal conditions. <i>Blue = MPC, Red = PID</i>	47
5.17	Multi-model Thrust control for nominal and 80% fan efficiency.	48
5.18	Multi-model Thrust control for nominal and 80% fan efficiency.	49
5.19	Multi-model Thrust control for nominal and 80% fan efficiency.	49
5.20	Thrust control using a nominal reference model for 78%, 80% and 82% fan efficiencies	51
5.21	Thrust control using a nominal reference model for 78%, 80% and 82% fan efficiencies	51
5.22	Thrust control using a nominal reference model for 78%, 80% and 82% fan efficiencies	52
5.23	Parent layer with baseline and MPC applied to the nonlinear engine.	53
5.24	SMMPC subsystem with multiple MPCs and switch logic.	53
5.25	MPC68 subsystem with MPC designed at PLA 68.	55

5.26	Implementation of switching logic.	55
5.27	Implementation of time based supervisor.	56
5.28	MPC and PID applied to a nominal nonlinear engine.	57
5.29	MPC and PID applied to a nominal nonlinear engine.	57
5.30	MPC and PID applied to a nominal nonlinear engine.	58
5.31	Matched MPC and PID applied to a nonlinear engine for FE=-0.2. .	59
5.32	Matched MPC and PID applied to a nonlinear engine for FE=-0.2. .	60
5.33	Matched MPC and PID applied to a nonlinear engine for FE=-0.2. .	60
5.34	Nonlinear engine without MPC fault recovery for FE=-0.2 at 25 sec.	61
5.35	Nonlinear engine without MPC fault recovery for FE=-0.2 at 25 sec.	62
5.36	Nonlinear engine without MPC fault recovery for FE=-0.2 at 25 sec.	62
5.37	Nonlinear engine with MPC fault recovery for FE=-0.2 at 25 sec. . .	63
5.38	Nonlinear engine with MPC fault recovery for FE=-0.2 at 25 sec. . .	64
5.39	Nonlinear engine with MPC fault recovery for FE=-0.2 at 25 sec. . .	64
5.40	MPC68fe20 applied to a Nonlinear engine with FE=-0.22 at 25 sec. .	65
5.41	MPC68fe20 applied to a Nonlinear engine with FE=-0.22 at 25 sec. .	66
5.42	MPC68fe20 applied to a Nonlinear engine with FE=-0.22 at 25 sec. .	66
5.43	MPC68fe20 applied to a Nonlinear engine with FE=-0.18 at 25 sec. .	67
5.44	MPC68fe20 applied to a Nonlinear engine with FE=-0.18 at 25 sec. .	67
5.45	MPC68fe20 applied to a Nonlinear engine with FE=-0.18 at 25 sec. .	68
5.46	Effect of switched prediction horizons $N_P = 20, 50$	69

5.47	Effect of switched prediction horizons $N_P = 20, 50$	69
5.48	Effect of switched prediction horizons $N_P = 20, 50$	70
5.49	Parent layer with baseline and MPC for thrust control of the nonlinear engine.	71
5.50	Modified SMMPC subsystem with multiple MPCs and switch logic.	72
5.51	Modified MPC68 subsystem with MPC designed at PLA 68.	72
5.52	MPC thrust control of the nominal nonlinear engine.	74
5.53	MPC thrust control of the nominal nonlinear engine.	74
5.54	MPC thrust control of the nominal nonlinear engine.	75
5.55	Unswitched MPC thrust control of the nonlinear engine for FE=-0.2.	76
5.56	Unswitched MPC thrust control of the nonlinear engine for FE=-0.2.	76
5.57	Unswitched MPC thrust control of the nonlinear engine for FE=-0.2.	77
5.58	Switched MPC thrust control for FE=-0.2 at 25 sec.	77
5.59	Switched MPC thrust control for FE=-0.2 at 25 sec.	78
5.60	Switched MPC thrust control for FE=-0.2 at 25 sec.	78
5.61	MPC68fe20 applied to a Nonlinear engine with FE=-0.22 at 25 sec.	79
5.62	MPC68fe20 applied to a Nonlinear engine with FE=-0.22 at 25 sec.	80
5.63	MPC68fe20 applied to a Nonlinear engine with FE=-0.22 at 25 sec.	81
5.64	MPC68fe20 applied to a Nonlinear engine with FE=-0.18 at 25 sec.	81
5.65	MPC68fe20 applied to a Nonlinear engine with FE=-0.18 at 25 sec.	82
5.66	MPC68fe20 applied to a Nonlinear engine with FE=-0.18 at 25 sec.	82

Chapter 1: INTRODUCTION

Turbofan engines are complex nonlinear systems that operate along profiles varying in environmental conditions and Mach number. As the aviation industry grew larger, the operating profiles have become more demanding on jet engines financially and physically. Engine control plays a crucial role in the safe and stable operation of the engine within its operational and physical limits thus extending the operational lifetime of the engine. Operational limits include thrust specific fuel consumption (TSFC), spool acceleration and stall margins of the compressor and turbine while the physical limits are structural due to the pressure and temperature within the engine. As engines have evolved to higher capabilities, the control systems too have evolved from hydro-mechanical to electronic systems, becoming more complex in terms of the architecture, control law, sensors and actuators used.

The main aim of engine control has always been to achieve the desired thrust level which is described in Eq. (1.2) [1] for a turbofan engine

$$\textit{Net Thrust} = \textit{Gross Thrust} - \textit{Momentum Drag} \quad (1.1)$$

$$\tau_{net} = \dot{m}_a[(1 + f)u_e - u] - (P_e - P_a)A_e \quad (1.2)$$

$$f = \dot{m}_f / \dot{m}_a \quad (1.3)$$

where \dot{m}_a [Kg/sec] is the air mass flow rate, f is the fuel-air ratio, u_e [m/sec] is the exit velocity of the gas in the nozzle section, u [m/sec] is the flight velocity, P_e [kPa] is the exit pressure in the nozzle and P_a [kPa] is the ambient pressure outside the engine, A_e [m²] is the exhaust area. For an ideal expansion of gases through the nozzle, the momentum drag equals zero. For analyses purposes, the performance of the engine is expressed as non-dimensional parameters one of them being the Propulsion Efficiency η_p which is the ratio of the thrust power to the rate of production of propellant kinetic energy [1].

$$\eta_p = \frac{\tau}{\dot{m}_a[(1+f)(u_e^2/2) - u^2/2]} \quad (1.4)$$

For $f \ll 1$, this becomes

$$\eta_p = \frac{2u/u_e}{1 + u/u_e} \quad (1.5)$$

Since thrust is not a directly measurable property, the traditional engine control approach has been to control either the speed of the fan section or the engine pressure through the manipulation of the fuel flow rate into the combustion chamber.

This thesis explores the application of model predictive control (MPC) on a high bypass turbofan engine through simulations carried out in the MATLAB/Simulink environment. MPC is a control design technique that computes the control law through the optimization of a quadratic cost function subject to system constraints over a finite horizon. The cost function generally includes the system outputs and the control inputs. MPC has been successful mainly in the chemical and processing industry and is starting to be an attractive control technique in the aerospace industry especially for aircraft control.

The rest of the thesis is organized as follows. Chapter 2 presents an introduction to engine control systems along with the basics of their operation, evolution towards modern day technological existence, recent work and the advancement towards future architectures. Chapter 3 outlines the basics of the C-MAPSS40k simulation tool developed by NASA GRC in terms of its user interface, controller architecture and an overview of the turbofan engine being simulated.

Chapter 4 outlines the basic theoretical background of model predictive control along with a literature review on the theoretical advancements of MPC and its applications in the Automotive and Aerospace industry. The fault tolerant control approach applied in this thesis is introduced in this chapter along with the simulation tool for realizing MPC.

Chapter 5 is the main contribution of this thesis which describes in detail the various MPC simulations carried out. First the design process is discussed followed by the implementation of MPC. The ideal scenario as well as a scenario with the occurrence of a fault in the fan are simulated and the ability of MPC to recover from the induced faults is demonstrated using a multi-model approach. The traditional fan speed control is followed, also highlighting its shortcomings. The thrust control technique is investigated in a simplified manner. These simulations are carried out for both the linear time invariant as well as the nonlinear C-MAPSS40k engine.

Finally in Ch. 6, the conclusions are listed and the future aspects of MPC in engine control are briefly discussed followed by the source code implementation of MPC in MATLAB in Ch. 7.

Chapter 2: ENGINE CONTROL SYSTEMS

2.1 Introduction

Gas turbine engines due to their complex nature in design and varying operational profiles in terms of altitude and Mach number require suitable control that ensure the engine operation is sustained. In reality, no two engines are identical due to differences that arise during their manufacturing process, largely owing them to differences in the properties of the material stock being used. Therefore an engine control design should be able to handle these manufacturing differences and hence should be robust [2]. Over several operational cycles the engine deteriorates due to regular wear and tear of engine components slowly degrading the performance of the gas turbine engine. Certain components such as the fan blades and turbine blades degrade faster than others due to their physical location and this higher rate of health deterioration needs to be accounted for.

In practical scenarios, ingestion of foreign objects such as birds or debris or passage through storm clouds leads to a sudden shift in the engine performance due induced sudden degradation in the components also termed as faults. Hence the control design should be versatile in addition to being robust in the sense that either certain lines

of codes or control gains can be modified easily on-the-fly so as to account for both slow and fast degradation of engine components.

This modification is carried out once the actual fault or kind and amount of degradation is known which usually carried out by the Fault Detection and Isolation (FDI) diagnostics. FDI techniques generally monitor the health states of the engine through feedback measurement and estimation and based on the residues obtained when comparing the current health with that of a new engine are able to point out the fault present and its significance.

2.2 Operation of Engine Control Systems

Jet engines are closed loop systems with automatic control systems that consist of the four main control component namely actuators, controller, sensors and the engine plant. Because the engine operational cycle is mainly thermodynamic in nature, closed loop computation of the fuel flow rate is sufficient to achieve sustained operation over most flight profiles with the variable stator vanes (VSV) and variable bleed valves (VBV) scheduled in an open loop manner based on the spool speed and the Mach number. However, military engines which can traverse complicated flight profiles in a very short duration require the additional manipulation of VBV and VSV positions.

The main control objective in engine control is to ensure that a desired value of thrust, governed by the PLA position, is maintained. However since the thrust cannot be directly measured, the traditional approach has been controlling the fan speed or the engine pressure ratio (EPR) which are measurable and the thrust can hence be estimated using an on-board engine model. These control systems ensure safe and stable engine operation through the following basic function [2]:

- Set point control to evolve the engine performance to the desired values (Fan speed or thrust)
- Limit control that ensure the system does not violate its physical and operational constraints
- Transient control to achieve the desired set-point within a reasonable time frame

It possible sometimes to have the set-point and transient control functions performed by the same control module. Current industrial controllers consist of these modules in a min-max architecture which will be discussed briefly in Ch 3. To achieve the desired performance the positions of actuators such as the fuel metering valve (FMV), VBV and VSV are manipulated. In current industrial standard controllers, while the VBV and VSV positions are based on look-up tables, the FMV position is calculated by the control design which takes into account the fuel flow rate of the existing system. The complexity of jet engines demands thermodynamic properties of the engine such as temperature and pressure to be measured at various stages of the engine in addition to the measurement of the spool speeds.

2.3 Evolution of Engine Control Systems

The very first generation of gas turbine engines used control systems that were hydro-mechanical in nature which helped overcome the complexity and weight of purely mechanical systems. These systems are comprised of two of parts namely the mechanical circuit and the hydraulic circuit. The mechanical circuit connects the cockpit demand control interface to the hydraulic circuit which is made of hydraulic valves and actuators. A control system of this kind is found in the SR-71 "Blackbird"

aircraft. The advent of transistor electronics allowed analog electronic control units to be implemented in the 1970s where the amplitude of the electric control signals are varied to manipulate the actuators. This eventually led towards the Full Authority Digital Engine Control (FADEC) in the 1980s that uses digital electronics and as its name suggests, is completely responsible for the operation of the gas turbine engine and the first FADEC was tested on the PW F100 engine. During this time, the increasing computational power available also made it feasible for extensive engine modeling and simulations to be carried out making it possible to test novel strategies at lower costs providing the foundation for current advanced jet engines such as the PW F135.

The use of digital electronics has not only made the control unit lighter and smaller but has also allowed the feasibility of complex control designs including advanced functionality as seen in the dual channel FADECs that have embedded engine models allowing engine health monitoring and multi-variable control. An example of these FADECs are found in the GE F110, PW F119 and PW F135 engines. A detailed evolution of engine control systems and engine modeling are described in Jaw et al [2].

2.4 Recent Advancements in Engine Control

Concerns over fuel efficiency and the emissions of Nitrous Oxide (NOX) and Carbon dioxide (CO₂) and improvements in the thrust produced have led to advancements in gas turbine engine designs which require advanced robust control techniques to be applied to ensure a stable and sustained jet engine operation. Although PID

based controllers have been performing well, additional design and performance constraints increase the controllers complexity especially when transitioning towards a distributed control architecture.

Over the last decade, the engine control community has published rich literature on various engine control methodologies while also paving the path for future developments. Richter [3] proposes a novel control technique using a mixed H_∞/H_2 multi-objective feedback gain synthesis with sliding mode controllers ensuring the system is within design constraints and the proposed technique is applied on C-MAPSS. In reference [4] Richter provides a comprehensive literature on various control concepts applied to turbofan engines using the C-MAPSS40k simulation tool. These include classical frequency domain methods, Linear Quadratic Regulator (LQR), mixed H_∞/H_2 , simplified H_∞ , Linear Parameter Varying (LPV) methodologies, Sliding Mode Control and Model Predictive Control (MPC). Hacker [5] applies a L_1 adaptive controller to the turbofan engine model in C-MAPSS40k while handling system constraints and nonlinear uncertainties.

Han [6] presents a new parameter-scheduled control design where parameterized linear matrix inequalities (PLMI) are reduced to linear matrix inequalities (LMI) based on a improved convex polyhedron construction algorithm and show satisfactory global control performance. Samar [7] propose a 2 degree-of-freedom (DOF) multi-mode controller designed through discrete time H_∞ optimization and use a simple strategy for anti-windup and bumpless transfer between controllers for the Rolls Royce Spey turbofan engine.

Hu et al. [8] use a fuzzy controller that includes a fuzzy integral mixed controller, smith forecast and compensation with the fuzzy rules optimized through genetic algorithms. Simulations show fast high precision system response without any oscillations and overshoot. Similarly, Shuqing et al. [9] present a control scheme for the whole flight envelope where the control gains are optimized using genetic algorithms for a certain turbofan engine model. Then a neural network approximation is trained to relate the PID control gains with altitude and mach number to obtain a neural network based optimal PID controller. In references [10, 11] Wang et al. develop a 2DOF H_∞ based control law for a complete flight envelope. The envelope is divided into eight subregions with their corresponding controllers which are switched using a bumpless switch logic.

There is also rich literature on the active control of phenomena which are required due to the complex flow patterns present within the engine. Decastro [12] proposes a rate-based model predictive control on a linear parameter varying (LPV) turbofan engine model for active tip clearance control while also suggesting the same method for engine control. The LPV model is obtained by differentiating the linear time invariant (LTI) engine model from C-MAPSS40k. Similarly, Liu et al. [13] demonstrate active compressor stability management using C-MAPSS40k and the simulations show an improvement in the emergency response of the engine.

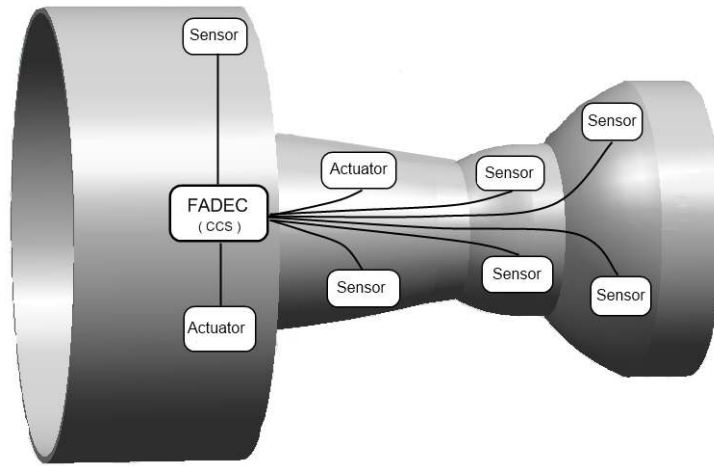
2.5 Transition in Control Architecture

So far, controllers have been designed with the pure intention of accomplishing solely engine functionality within its limits only and not performance [14, 15] while the engines themselves have evolved to higher complexity. However, the mass and heat

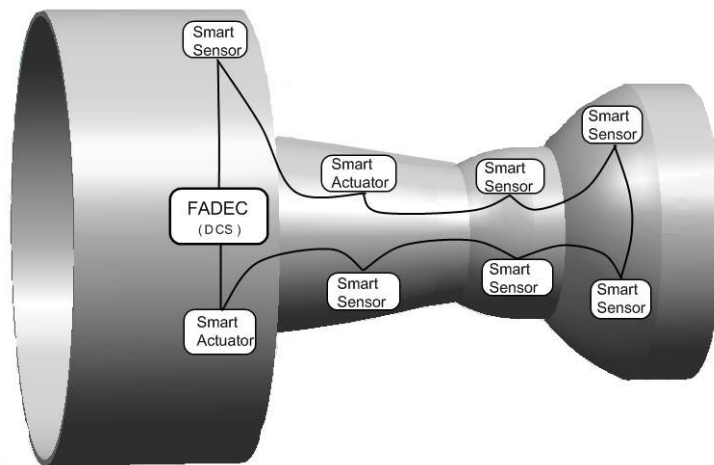
dissipation by the control unit affect the aerodynamic drag of the aircraft indirectly affecting the engine performance. The use of new technologies in the current centralized architecture where the FADEC interfaces directly with the system effectors in a star topology [14] is limited and leads to increased weight and integration complexity. Hence there is the necessity to change the architecture of the control system for a safer and stable sustained operation. Culley et al [14, 15, 16] present a strong case for the Distributed Engine Control (DEC) architecture, outlining a road map for the progressive transition from the centralized architecture to a fully distributed architecture and highlighting the financial and technological benefits of doing so.

A distributed architecture permits the use of off-the-shelf and future commercial technologies [17] on current jet engines namely high speed, high temperature and low power digital integrated circuits while also providing ease of integration and segregation of actuators and sensors over a common digital network interface with a computationally heavy controller thus avoiding compatibility issues and making the control unit lighter in weight. The complexity of a jet engine requires additional robust active control of local phenomena such as combustion and tip clearance whose states are to be relayed with the FADEC. A fully distributed architecture has local processing capabilities [18] allowing for the active control of localized phenomena as mentioned in reference [15] and the efficient integration between the FADEC and sub controllers. A basic schematic of the centralized and distributed architecture is shown in Fig. 2.1 [18].

The advantage with digital networks is their high tolerance towards high temperature signal corruption and significantly lower attenuation per unit length. This allows placing the FADEC outside the harsh operational environment of an engine thereby



(a) Centralized engine control architecture



(b) Fully distributed engine control architecture

Figure 2.1: (a) Point-to-point analog harness interfacing between FADEC and system effectors in a star topology (b) FADEC communication with system effectors over a common digital network in a ring topology.

improving the control system cooling which was previously limited by using the fuel as a coolant. Since data has to be transmitted over digital networks, the choice of the appropriate communication protocol is crucial. Current protocols include the MIL-STD-1553, control area network (CAN), SPIDER, SAFEbus, and TTTech Time Triggered Architecture (TTA) of which time triggered protocols offer performance of high reliability and robustness towards faults [17]. Additionally, the communication protocols used should be able to support Fault Detection and Isolation (FDI) schemes, be highly modular and have a high throughput.

However, constraints on the channel bandwidth limit the performance of digital networks thus affecting the control action. While different protocols present various degrees of performance, the control design and analyses is independent of the protocols being used and should be able to handle all the current protocols in existence. The use of digital networks induces time delays in the network due to serial transmission of packets. Further, packet collisions or node failures result in loss of information. These packet dropouts and time delays are investigated in Yedavalli et al. [17].

If the control design is based on a min-max algorithm, the induced time delays due to a digital network can cause a lag in the min-max switching thus affecting the controller output and hence the engines response. Therefore switching techniques of this kind are best not to be used in systems having time delays and a different control law would allow for better distributed engine control.

Chapter 3: OVERVIEW OF C-MAPSS40K

3.1 Introduction

Every engine manufacturer has their own individual engine simulation environment which differ from each other due to component dynamics, control gains and design specifications. With the increasing need for the application of novel control techniques and a transition in control architecture, technological breakthroughs would be on different platforms and hence relative in nature and would be difficult to validate for industrial application.

In order to normalize these differences NASA Glenn Research Center developed the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) which is a component level physics based simulation that can be used as a reference platform to test control designs on a validated and verified platform. C-MAPSS is a simulation of a large turbofan engine producing a maximum of 90,000 lbf of thrust . C-MAPSS40k is based on a generic high bypass two spool turbofan engine of the 40,000 lbf thrust class engines producing maximum of 35,000 lbf of thrust at take-off conditions [19]. The engine's bypass ratio is 6:1 with an overall pressure ratio (OPR) of 25 [20].

The C-MAPSS simulation series is the successor of the previously designed MAPSS which represents a low bypass military turbofan engine with an afterburner. The

40,000 lbf thrust class engines include the CFM56/F108 series of commercial turbofan engines used mainly on the A320, A340 and the Boeing 737 commercial aircrafts

3.2 C-MAPSS40k GUI

3.2.1 Closed loop interface

The C-MAPSS40k graphical user interface (GUI) allows for simulations over a variety of dynamic operating profiles by specifying different ambient conditions such as altitude, Mach number and power level angle (PLA). Engine health deterioration can be simulated either by specifying overall engine degradation or individual component faults by specifying component health parameters at desired points along the profile [21] as shown in Fig. 3.2.1. The health parameters are specified as a percentage with 0 being nominal, 1 being 100% degradation and -1 being -100% degradation affecting the component properties as shown in Eq. 3.1. The parameters specified should also be rational.

$$\text{component property} = \text{component property}_{\text{nominal}} * (1 + \text{healthparameter}) \quad (3.1)$$

3.2.2 Linearization routine

The linearization tab of the GUI generates linear time invariant (LTI) state space models at user defined equilibrium conditions [20] with choices between a single variable (W_f) and multivariate (W_f , VBV and VSV) control. The linearization routine uses a Jacobian scheme to generate LTI state space models of the form given in Eq. 3.2 and Eq. 3.3 where x is the state vector formed by the fan speed N_f and core speed N_c , u is the input vector. 13 health parameters are provided by the linearization routine in the form of the L and M matrices with h as the health parameter vector that can

be used for fault diagnostics.

$$\dot{x}(t) = Ax(t) + Bu(t) + Lh(t) \quad (3.2)$$

$$y(t) = Cx(t) + Du(t) + Mh(t) \quad (3.3)$$

3.3 C-MAPSS40k Controller Design

C-MAPSS40k has a FADEC-like controller with a min-max architecture, similar to those in the industry [20], consisting of a gain scheduled PID controller for manipulating either the fan speed or the engine pressure. In addition, there are a set of individual limit controllers that have critically damped responses ensuring safe and stable operation of the engine within its operational and physical limits [19]. These limiters typically maintain design limits on the fan speed, core speed, engine pressure and acceleration of the high speed spool. The main PID controller and the individual limiters are arranged in a min-max architecture as shown in Fig. 3.3 [19].

Implementing a controller of this architecture with multiple components in reality is complex and expensive and hence alternative control designs which are easier to implement and offer similar or perhaps better performance need to be considered to address the issues mentioned in Ch 2.

3.4 C-MAPSS40k Engine

As mentioned earlier, C-MAPSS40k uses a nonlinear physics based engine modeled at the component level. Each component is described through the thermodynamic relations of the corresponding stage and the components are modeled in C ensuring faster than real-time simulation. A point to note is that the fan tip is described as the fan from here on (and in C-MAPSS40k) while the fan hub is considered as an

additional stage of the low pressure compressor. A representation of the engine used in C-MAPSS40k is shown in Fig. 3.4 [19].

The application of C-MAPSS40k for engine control purposes is still in its infancy with limited literature [3, 4, 19, 20, 21]. C-MAPSS40k has a realistic FADEC like controller in a min-max architecture and a user friendly GUI. Further advantages include the realization of an actual nonlinear engine with ability to generate linear models as well as the inclusion of actuator and sensor dynamics that follows the full flight envelope due to atmospheric modeling. In addition, the programming of the engine components in C enables the simulation to run faster than real time thus making C-MAPSS40k an attractive tool to utilize its open source nature for either designing new engine controllers or adding additional modules for active phenomena control.

Therefore this thesis which exploits C-MAPSS40k and employs the MPC control technique for engine control significantly contributes to the engine control community and supports the cause for the development of C-MAPSS40k by NASA GRC.



Figure 3.1: Closed loop interface of the C-MAPSS40k GUI.

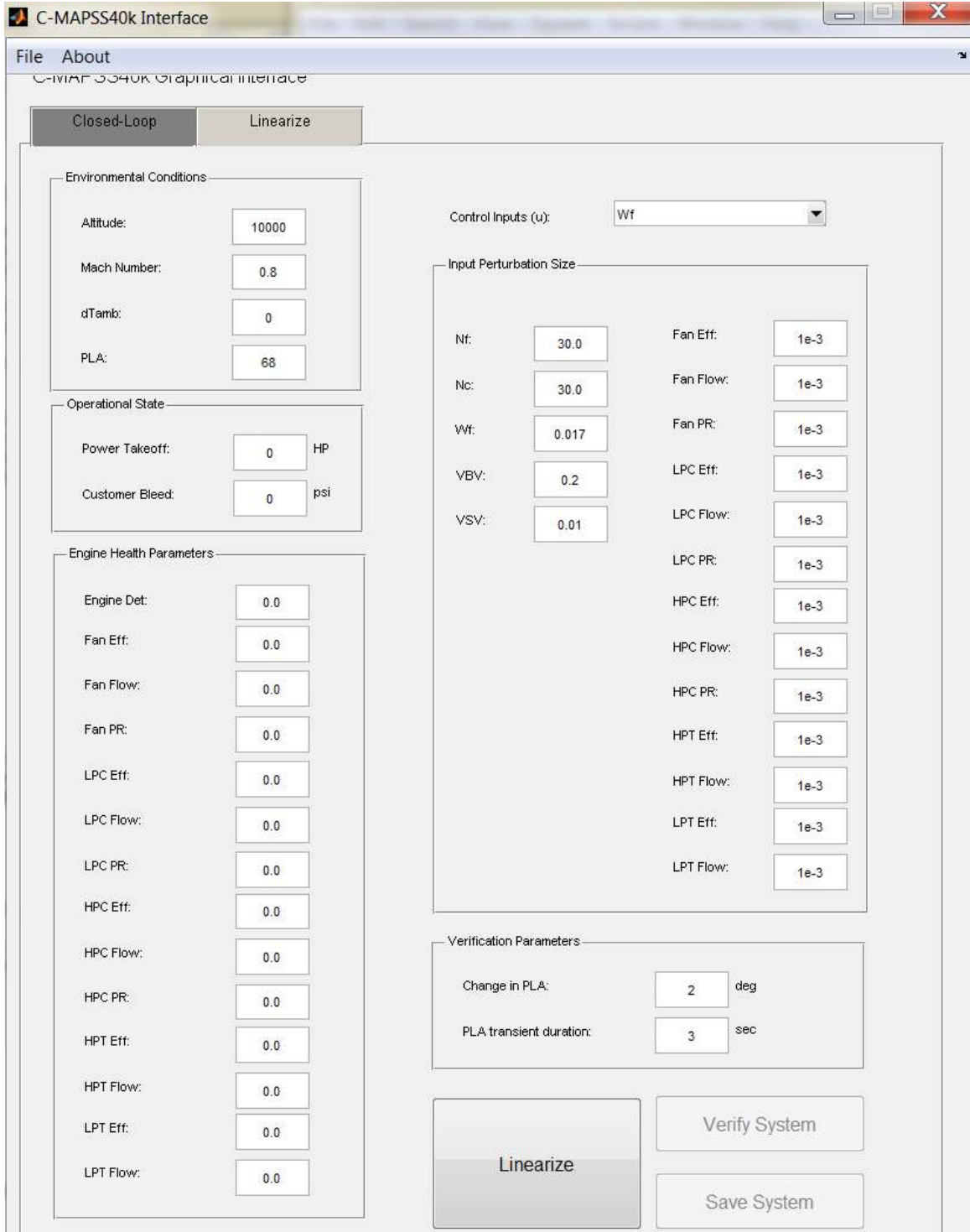


Figure 3.2: Linearization interface of the C-MAPSS40k GUI.

Chapter 4: MODEL PREDICTIVE CONTROL

4.1 Motivation and Problem Formulation

The need to apply a different control technique arises from three main issues namely:

- Avoid the use of multiple individual limit controllers
- Avoid the use of the min-max architecture in a distributed control environment
- Exploit the computational power available in a distributed control environment

In order to address these issues, a model based control technique that incorporates the system constraints into the control design needs to be chosen. Model predictive control (MPC) is one such control design technique which calculates the control law subject to system constraints through the optimization of a quadratic cost function and hence is computationally intensive. MPC has been used successfully in the chemical and process industry and is being studied by the automotive as well as the aerospace industry for applications including gas turbine engines.

Decastro [12] has used MPC for active tip clearance control and has suggested that his proposed technique can be applied for the control of jet engines. Richter [4] has demonstrated the application of MPC on C-MAPSS40k. This thesis extends the

application of MPC on the C-MAPSS40k engine for a profile with varying throttle positions (PLA) in the presence of component faults while simultaneously addressing the traditional fan speed control as well as the thrust control approach in a simplified scenario. In addition to simulations being carried out on a LTI engine model, the application of MPC on the non-linear C-MAPSS40k engine is also demonstrated for a flight profile from take-off to cruise conditions with the occurrence of faults.

4.2 Overview of MPC

Traditional control involves the design of a controller for a given system to get a desired closed loop performance and subsequent integration of limiters to ensure safe operation [22]. The Model Predictive Control (MPC) technique allows the incorporation of system constraints in the optimization problem where generally a set-point is being tracked. Hence, the need to use separate limit controllers is avoided simplifying the control design and its implementation.

MPC is a model based control scheme where a linear model of the plant is used to predict the behavior of the plant denoted by $\hat{y}_{k+i|k}$ ($i = 0, 1, \dots, N_P$), starting at the current time instant, over a finite prediction horizon N_P . In order for the plant to reach the desired set-point $r_{k|k}$, the optimum input $\hat{u}_{k+i|k}$ ($i = 0, 1, \dots, N_U$) trajectory is chosen such that it minimizes the cost function given by Eq. (4.1) subject to the constraints described by Eq. (4.2)-(4.4)

$$\min_{u_k, u_{k+1}, \dots, u_{k+N_U}} J = \sum_{i=1}^{N_P} (r_{k+i} - \hat{y}_{k+i})^T Q (r_{k+i} - \hat{y}_{k+i}) + \sum_{j=1}^{N_U} u_{k+j}^T R u_{k+j} \quad (4.1)$$

Subject to:

$$M_1 \leq y \leq \gamma_1 \quad (4.2)$$

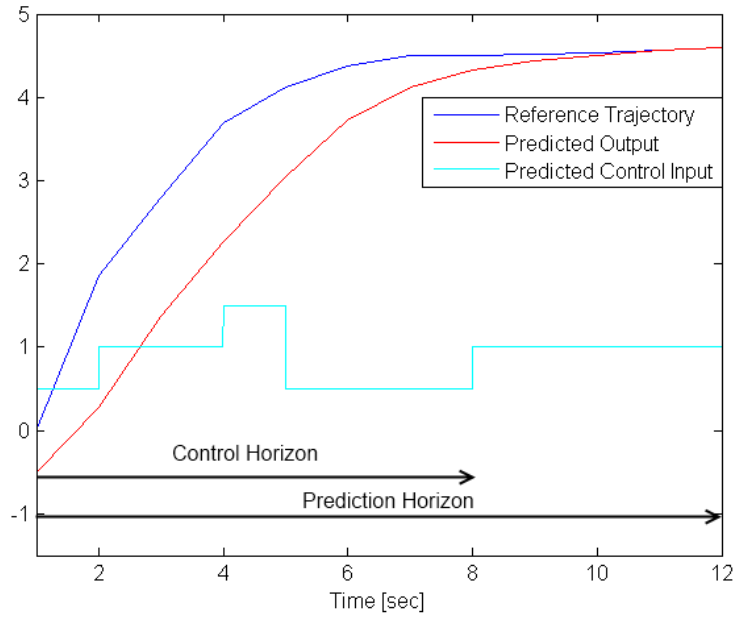


Figure 4.1: Control trajectories in a MPC optimization problem ($N_U < N_P$).

$$M_2 \leq u \leq \gamma_2 \quad (4.3)$$

$$M_3 \leq \Delta u \leq \gamma_3 \quad (4.4)$$

where Q and R are the output and input weighting matrices respectively with appropriate dimensions and the M_i and γ_i are the minimum and maximum constraints on the system. N_U is the control horizon representing the length of the input trajectory and $N_U \leq N_P$. Only the first element of this trajectory is applied to the plant i.e. $u_{k|k} = \hat{u}_{k+0|k}$. At the next sampling interval, the whole process is repeated and hence MPC is also referred to as receding horizon control. Equation (4.1) represents a general cost function used in most aerospace application [23]. The physical interpretation of the horizons in terms of the actual system is shown in Eq. (4.5).

$$Horizon[sec] = Sampling\ Time[sec] * Horizon[-] \quad (4.5)$$

The optimization problem involves finding the minimum of a quadratic function subject to the (in)equality constraints. Therefore the solution is within a convex polytope. It should be noted that the minimum value of the cost function satisfying the constraints need not be the actual minimum value of the quadratic function i.e. the cost after optimization in MPC is generally higher due to imposed constraints. Since Eq. (4.1) is quadratic in nature, the optimization problem becomes a Quadratic Program (QP). An infinite horizon unconstrained MPC is equivalent to a LQR problem [24] and therefore inherits the benefits of optimization control theory.

4.3 MPC Analogy

An even simpler way of understanding MPC is through the many analogies that exist pertaining to action taken after prediction using a certain reference model. One such analogy is that of driving a car depicted in Fig. 4.3 [25] where the MPC equivalent would be the driver predicting/estimating the path ahead over a finite horizon through the windshield and then deciding upon the necessary set of control actions based on the cars limitations and its current conditions and applying only the first of the control actions at that time instance. The PID equivalent would be deciding upon the control action sequence based on past events or rather by looking into the rear view mirror. This is however a harsh method of explaining the PID scheme.

Another analogy which is even more effective in comparison is that of playing chess. Here one predicts the opponent's moves, using the previous games of the

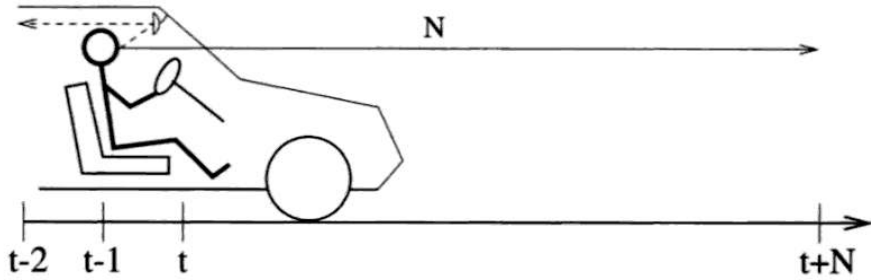


Figure 4.2: Car driving MPC analogy.

opponent as the reference models, over a certain number of moves in the game (finite horizon) and then accordingly plans his/her next few moves (control horizon) taking into consideration the available pieces and moves left (constraints) and plays out only the first move at the current time instant. The chess analogy for the PID control scheme would be to calculate your next move based on only what has occurred previously in the game till the current time instant.

4.4 Recent advancements in MPC

There is rich literature that has been published over the last two decades that describe the mathematical aspects and implementation of MPC. Bemporad [26] presents a survey on MPC regarding uncertainty descriptions and techniques for handling robust constraints. Wang [24] demonstrates the application of MPC using offline computations that exploit discrete Laguerre functions which are used to replace the discrete pulse inputs. Both Camacho [25] and Rossiter [27] present extremely good literature on the theoretical concepts involved in MPC. Model predictive control has reached theoretical maturity and the performance largely depends on the chosen control and

prediction horizons in addition to the weighting matrices as shown by Rossiter [27]. Mayne et al. [28] survey the issues on the stability and optimality on the MPC problem applied to both linear as well as nonlinear systems.

Both Qin [29] and Darby [30] present a survey on the current MPC techniques used in the industry and the challenges that lie ahead. Since MPC is based on iterative on-line optimization, the inherent computational burden previously limited its use to low-bandwidth applications mainly in the chemical and process industry. One of the key issues in current research is aimed at reducing the computational burden as demonstrated by Wang [31]. Alessio [32] presents a survey on explicit MPC, a technique which addresses the computational burden, one of the main drawbacks of MPC. In explicit MPC, the control law is calculated off-line as an explicit piecewise affine function of the reference and state vectors for a certain range of operational conditions thereby reducing the computational time required. Gawthrop [33, 34] present another approach in the implementation of MPC by combining intermittent control and the ability of MPC to handle constraints via quadratic programming.

4.5 High Speed Applications of MPC

Technological advancements in integrated circuits over the last decade have led to improved computational hardware and thus have extended the use of MPC to high-bandwidth applications in the automotive and electronic industry. Kouro et al. [35] have demonstrated the application of Finite Control Set (FCS) MPC to control power converters present in power systems. Similarly Xie et al. [36] present the control of a three-phase inverter system using an improved MPC that predicts at one sampling time instant and translates the voltage into control signals at the next instant thereby

avoiding the switching delay caused by traditional MPC. In addition, Zong et al. [37] demonstrate the application of MPC for the active load management of a distributed power system.

Cairano et al. [38] apply MPC to control magnetically actuated mass spring dampers where first they demonstrate linear MPC followed by the application of a hybrid MPC taking into account the all the system constraints and the electrical dynamics. Bolognani et al. [39] demonstrate MPC as applied to permanent magnet synchronous motor drives. Similarly Santana et al. [40] show the control of speed and rotor flux of an induction motor using MPC where the speed and rotor flux are estimated using an extended Kalman filter.

Scenarios such as the active control of noise and vibration exist where these phenomena need to be kept under certain limits. These limits can be translated into constraints and MPC be exploited as demonstrated by Wills et al. [41]. MPC which traditionally exploits LTI models has also been extended to linear time varying (LTV) systems. Falcone et al. [42] apply a LTV MPC scheme to the active front steering (AFS) system of an autonomous vehicle.

4.6 MPC in Nonlinear Applications

The growing popularity of MPC due its ability to compute a control law that obeys system constraints is seen in its application to large systems as well. To control large transportation networks that include power grids, road traffic and communication networks, Negenborn et al. [43] propose a multi-agent control scheme with coordination among the agents to improve decision making. Each agent employs MPC for the control of a corresponding subsystem. In addition, two architectures for the decision

making process namely the serial and parallel schemes with the serial scheme showing more preferable properties than the parallel scheme.

Similarly Scattolini [44] reviews decentralized, distributed and hierarchical architectures for MPC. A decentralized architecture has individual regulators controlling a corresponding subsystem whereas a distributed architecture would have these individual controllers or agents communicating with each other directly. A hierarchical architecture on the other hand has an additional coordinator unit for the coordination between the agents. Additionally, this architecture can be used in a multilayer scheme where the control action is carried out by a number of agents working at different time scales i.e. system either has a combination of fast and slow dynamics or the optimization and control algorithms operate at different rates. This kind of coordination would also require an appropriate communication protocol which would determine the performance of the entire controller unit.

An alternative approach to using multiple agents parallelly is to switch between the agents or controllers based on an event where each controller is solely responsible for the control law when triggered. Magni et al. [45] present a research finding on switched MPC for a nonlinear thermal system whose algorithm exploits the concept of multiple cost functions. To summarize their algorithm, an appropriate region is selected and optimization is then carried out after having chosen the correct weighting matrices Q and R . The first element of the control trajectory is applied only if switching is feasible.

Similarly, Zanma et al. [46] demonstrate switching MPC for the control of a nonlinear system where the switching algorithm is based on a maximal constrained positively invariant (CPI) set. Mhaskar et al. [47] present a mathematical paper on MPC

for a switched nonlinear system. They propose an algorithm where a controller is designed for each constituent mode and the predictive control of the switching system is carried out upon ensuring the transitions between the modes maintain stability of the system.

4.7 MPC in Aerospace Applications

The success of MPC in automotive applications has attracted the aerospace community in recent years for its application. Gavilan et al.[48] present the application of MPC for a spacecraft rendezvous using a chance constrained approach. In this technique, disturbances are incorporated into the system constraints using a probabilistic formulation which are then translated into a set of algebraic equations. The control law is computed such that a desired probability for the constraints is satisfied. Lopes et al.[49] demonstrate attitude control and stabilization of a VTOL (Vertical Take-off and Landing) quad-rotor aircraft using MPC which utilizes a LTI model for computation.

Model predictive control has also been successful for non-linear systems as demonstrated by Slegers et al.[50] for UAVs (Unmanned Air Vehicles) with 6DOF. As mentioned earlier, the application of MPC using a rate based approach for the tip clearance control of a turbofan engine is described in DeCastro [12]. The model used for computation is a linear parameter varying (LPV) model which is differentiated so that non-homogeneous terms are canceled out in a Taylor series expansion. DeCastro also suggests this method be used for the control of gas turbine engines.

Kestner [23] describes a detailed application of MPC to a tip jet reaction drive used in a helicopter. The distributed architecture employed here has individual controllers

for the right and left engines and the tip jet. Richter[51] proposes a multiplexed approach for a multivariate MPC law of a jet engine. In his approach, the three control variables are updated one at a time in a cyclical manner easing the computation burden. van Soest et al. [52] demonstrate the control of a nonlinear re-entry flight by combining feedback linearization (FBL) with MPC. In FBL a nonlinear state feedback is exploited that cancels the nonlinearities of a system. Alexis et al. [53] demonstrate the experimental application of switched MPC for the attitude control of a quadrotor helicopter. Their prediction is based on piecewise affine (PWA) models and the switching is governed by the rotation angle rates of the helicopter.

While Richter[4] demonstrates MPC for an aircraft engine, the approach used to calculate the input trajectories is analytical or off-line. Turbofans engines are highly complex systems with much faster dynamics than an aircraft and so far MPC has not been implemented practically for engine control due to limited on-board capabilities. The distributed control architecture allows computationally heavy control techniques like MPC to exploit the high speed processing capabilities available on-board the aircraft thus making on-line computation, a faster than real time optimization feasible. The advantage of on-line optimization is that the current state of the system is considered thus avoiding conservative control approaches and allowing for more robust control.

4.8 Fault Tolerant Control using MPC

In realistic flight scenarios, there is the possibility of component faults occurring within the engine due to regular component wear and tear, structural and thermal stresses and ingestion of foreign objects such as birds, debris and sand particles. While

regular wear and tear lead to slow health deterioration over engine lifetime, a fault is an abrupt or sudden deterioration in components due to stress or more likely foreign object ingestion.

There are two main methods for automated fault tolerant control (FTC) [54]. In passive methods, a robust controller is designed which can tolerate faults and compute a control law that accommodates the faults. The second technique is active FTC which assumes FDI i.e. Fault Detection and Isolation. Appropriate fault compensation/recovery is then carried out. Fault detection and isolation (FDI), forms a strong basis for engine health management and is of keen research interest among engine manufacturers. Once a fault has been detected and identified, appropriate action is taken to ensure safe and stable operation of the system.

This fault recovery process can be easily incorporated into MPC by modifying the cost function, system constraints or the reference model [55] since a fault is a disturbance or an uncertainty larger in magnitude. Prakash et al. [56] integrates model based FDI with MPC where both schemes exploit Kalman filters. Both Camacho et al. [55] and Prakash et al. propose multi-model MPC for fault tolerant control where each reference model corresponds to a fault that is likely to occur. The multi-model approach which is an active FTC technique is followed in this paper. Since only a certain number of faults can be accounted for, the occurrence of an unmodeled fault should be compensated by a reference model that corresponds to a fault which correlates strongly with the unmodeled fault. This avoids redundancy and over modeling of faults.

Fault tolerant model predictive control has been successful in aircraft dynamics where faults occur in the presence of external disturbances that either damage the

actuators and sensors or cause instability to the aircraft. Maciejowski [57] presents a case study of flight 1862 and shows that the crash could have been avoided using fault tolerant MPC (FTMPC). An inner MPC loop is used to track a reference updating itself based on a FDI module. Joosten et al. [54] present FTMPC using dynamic inversion which is the most basic form of FBL.

Almeida [58] demonstrates a FTMPC technique in the presence of actuator faults by redistributing the control effort among the healthy actuators enabling the controller to still track the reference. The aircraft model of the controller is modified to perform fault recovery along with the modification of the cost function. Similar research is presented by Zhang et al. [59] although with a different inner loop architecture and realization.

4.9 MPC Simulation Tool

With the increasing availability of high computational power and the widespread application of MPC, research groups have also been involved in translating the breakthroughs in MPC into applicable software simulation tools such as those developed by Bemporad et al. [60] for extending the research and application of MPC. A more recent version of the MPC simulation tool (Version 3.2) in MATLAB developed by Bemporad is used for the simulations carried out in this thesis. This tool uses a cost function similar to Eq 4.1 where the model used for computation is a LTI state space model. The MPC can be either designed via a GUI or through the command window. The implementation of MPC is discussed in detail in Ch. 5.

Chapter 5: LINEAR SIMULATIONS OF MPC ON C-MAPSS40K

This chapter describes the main contribution of this thesis including a detailed discussion on the realization of model predictive control and its application to a turbofan engine engine for different scenarios. First a discussion on the linear simulations is presented followed by the application of MPC on the nonlinear C-MAPSS40k engine.

5.1 Linear Simulations

5.1.1 Linearization

In Sec 5.1 a discrete LTI state space model of the C-MAPSS40k engine is used for the plant model as well as the MPC reference model. In order to demonstrate the constrained performance of MPC over a time varying demand profile in terms of varying power level angle (PLA), the nonlinear C-MAPSS40k engine is linearized at operating conditions at which the baseline limit controllers are active i.e. at these conditions the engine would violate its limits in the absence of the limiters. The nominal linearization operating conditions are:

- Altitude: 10,000 ft (3048 m)
- Mach: 0.8

- PLA: 68°

The demand profile for the simulations to follow is a PLA transient of 58-80-63 denoting a flight from cruise conditions to a maximum thrust condition and then back to cruise conditions at slightly higher thrust. Hence it is very crucial to choose the most accurate PLA linearization point or a set of them for a given demand profile. In the linear simulations to follow, only the one linearization point mentioned above is chosen. Based on the environmental conditions, C-MAPSS40k translates a PLA demand to a demand in the fan speed N_f *dmd*. Linearizing the nonlinear engine at the specified operating point mentioned above, a continuous linear time invariant state space equation described by Eq 5.1 and Eq 5.2 are obtained where A_c , B_c , C_c and D_c are of the appropriate dimensions.

$$\dot{x}(t) = A_c x(t) + B_c u(t) \quad (5.1)$$

$$y(t) = C_c x(t) + D_c u(t) \quad (5.2)$$

The resultant continuous time linear state space equations are discretized with a zero order hold at a sampling rate of 0.015 sec to obtain a discrete state space representation of the form

$$x_{k+1} = A_d x_k + B_d u_k \quad (5.3)$$

$$y_k = C_d x_k + D_d u_k \quad (5.4)$$

where $x \in R^n$ and $u \in R^m$ are the state and input vectors and A_d , B_d , C_d and D_d are of the appropriate dimensions. The fan speed N_f and the core speed N_c form the state vector x while the fuel flow rate W_f forms the input vector u .

5.1.2 Realization of MPC

For a discrete state space system, the output at time instant k is independent of the input at time instant k (i.e. u_k). The input can be described as

$$u_k = u_{k-1} + \Delta u_k \quad (5.5)$$

and hence Eq. (5.3) and Eq. (5.4) are augmented as shown by Richter [51] using Eq. (5.5) to form the discrete state space system given by (5.6) and (5.7) which can be rewritten as described by Eq. (5.8) and Eq. (5.9).

$$\begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix} = \begin{bmatrix} A_d & B_d \\ 0 & I_m \end{bmatrix} \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix} + \begin{bmatrix} B_d \\ I_m \end{bmatrix} \Delta u_k \quad (5.6)$$

$$y_k = [C_d \quad D_d] \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix} \quad (5.7)$$

$$\bar{x}_{k+1} = A_g \bar{x}_k + B_g \Delta u_k \quad (5.8)$$

$$y_k = C_g \bar{x}_k \quad (5.9)$$

where A_g , B_g and C_g represent the augmented state space matrices. Therefore, from Eq. (5.8), Δu_k is the control variable to be optimized by the MPC in Eq. 4.1. The influence of Δu_k on the output can be neglected as Δu_k is very small. The outputs are in the order N_f , N_c , W_f , T25, T50, Ps3, P50, net thrust, LPC_SM and HPC_SM. As mentioned previously, the performance of MPC largely depends on the control and prediction horizons N_U and N_P respectively (i.e. tuning parameters) and the output and input weighting matrices Q and R respectively. Figure 5.1 shows the

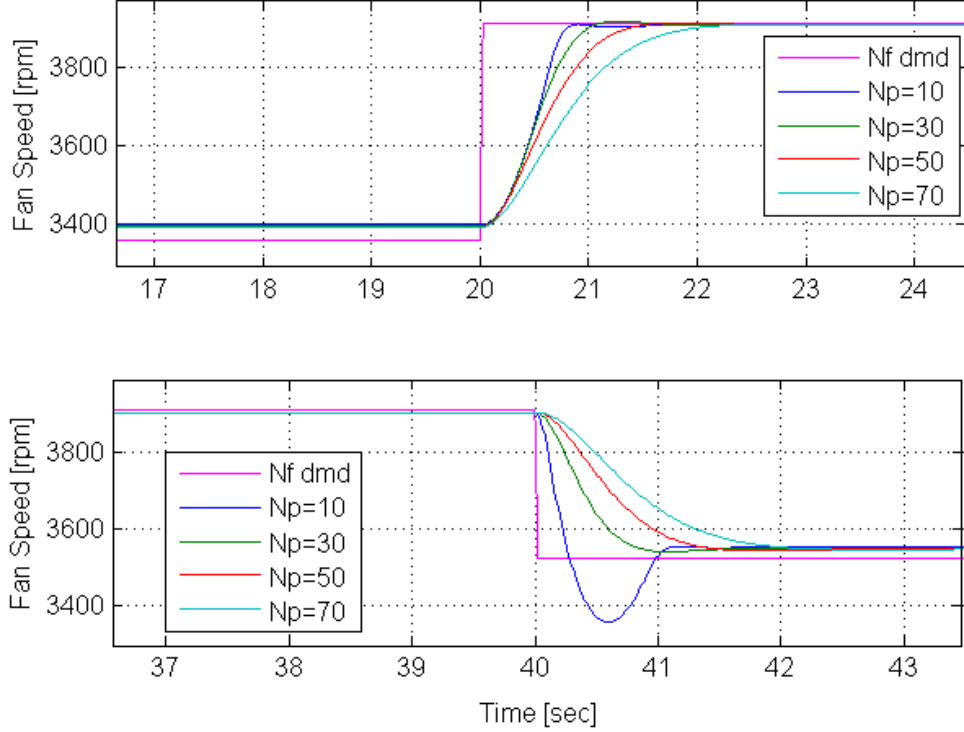


Figure 5.1: Effect of different prediction horizons with $N_U = 1$.

effects of different prediction horizons on the behavior of MPC for a fixed value of N_U . This concept is also discussed in the literature by Rossiter [27] however for a different physical example. As N_P increases, so does the rise time as well as the simulation time. However the occurrence of overshoot during transients is avoided. For the rest of simulations carried out in the MATLAB/Simulink environment, the horizons are chosen as $N_U = 1$ and $N_P = 50$. The weighting matrices chosen are diagonal and positive definite and such that the MPC performance is sustained in the presence of faults or health deterioration. Finally the hard constraints considered for MPC are the limits imposed on the C-MAPSS40k engine namely:

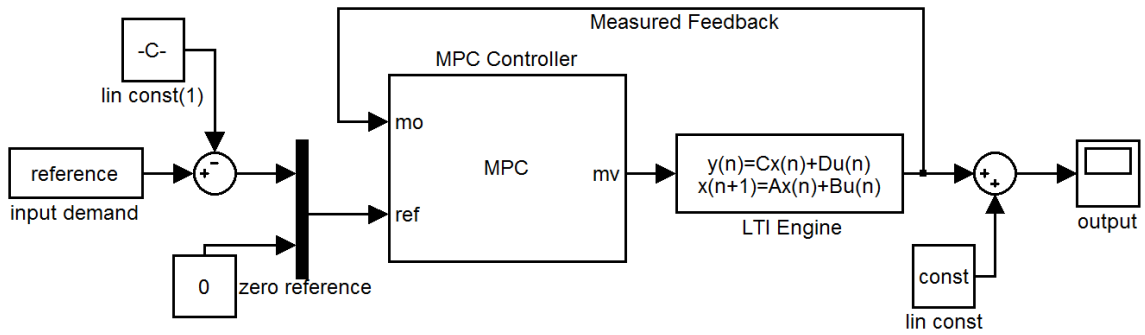


Figure 5.2: MPC simulation in MATLAB/Simulink for N_f control.

- N_f max: 4200 rpm
- N_c max: 12200 rpm
- Ps3 max: 433 psi (2.985 MPa)
- Ps3 min: 49 psi (0.337 MPa) for simulation conditions
- T50 max: 1500 °R (833 °K)

The MATLAB source code for the creation of the MPC object is given in Ch. 7 for a nominal case that exploits the MPC toolbox. The Simulink block diagram is shown in Fig. 5.2. For the simulations to follow, the flight conditions (FlightA) used are :

- Altitude: 10,000 ft (3048 m)
- Mach: 0.8
- PLA dmd: 58° - 68° - 63°
- Duration: 60 sec

5.1.3 Fan Speed Control Simulations

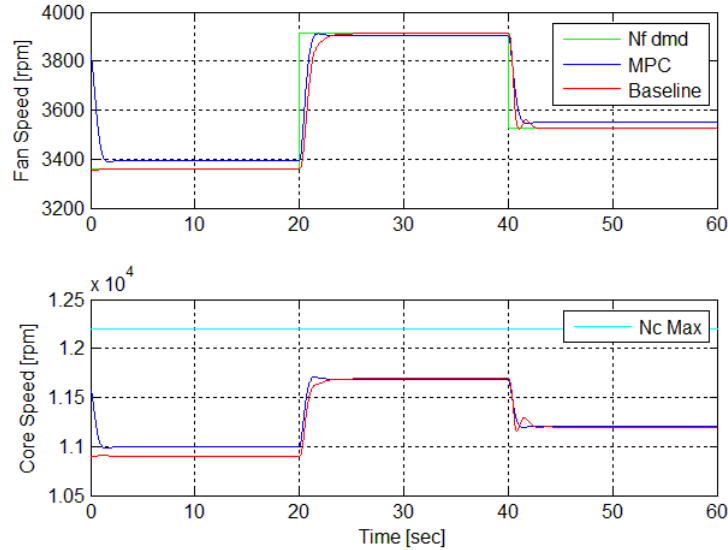


Figure 5.3: Comparison between MPC and PID for nominal conditions

This section describes the traditional fan speed control approach using MPC. A 60 sec simulation for the chosen conditions takes about 3.3 sec using the baseline PID of C-MAPSS40k and 3.6 sec using MPC i.e. the simulations carried out are faster than real time. Fortunately, the feedback variables in Fig. 5.2 comprising the state variables of C-MAPSS40k and $Ps3$ are measurable. The demand profile for fan speed is used as the main set-point or reference by the MPC i.e. a traditional engine control approach of controlling solely the fan speed and ensuring other engine parameters are within bounds of good performance.

Since the linearization point (trim values) is taken as the pseudo origin, the trim fan speed is subtracted from the fan speed demand and the complete set of trim values

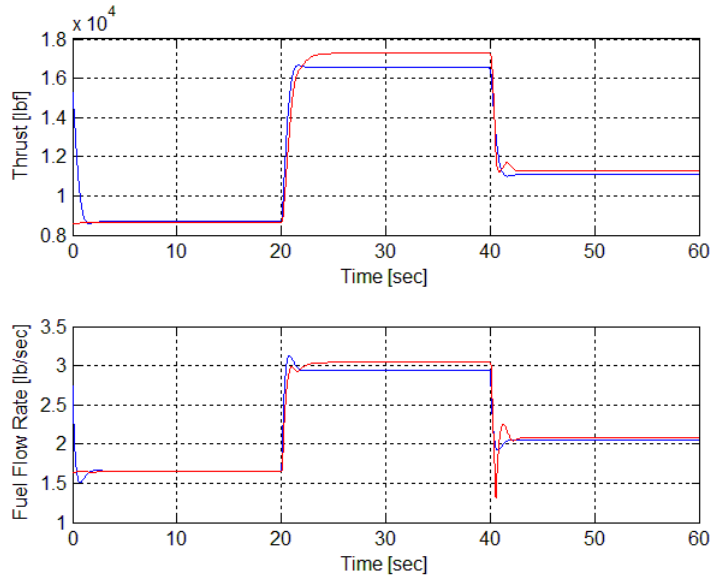


Figure 5.4: Comparison between MPC and PID for nominal conditions

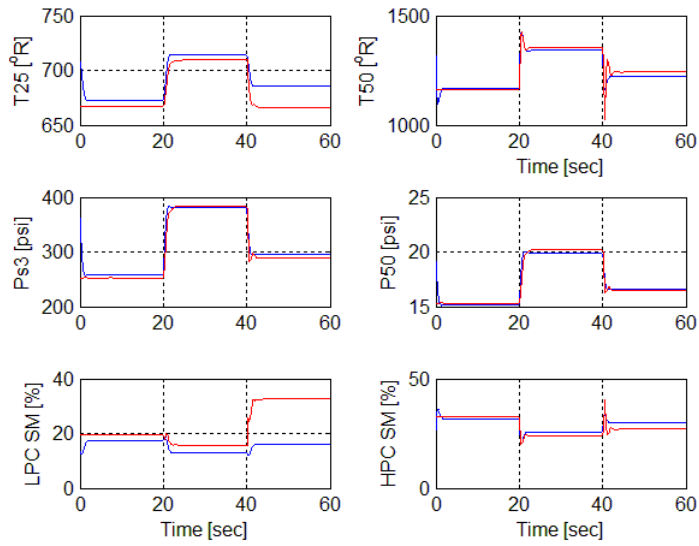


Figure 5.5: Comparison between MPC and PID for nominal conditions

are added to outputs of the linear engine to go back to the actual origin. Also, since this is a simulation for tracking the fan speed demand only, it is sufficient to provide the reference for fan speed while setting the reference for the rest of the outputs to zero.

For a flight conditions FlightA in the absence of faults i.e. a nominal case, Fig. 5.3 - 5.5 shows a comparison between the performances, that are in agreement with each other, of the baseline PID applied to the nonlinear C-MAPSS40k engine and MPC where both the reference and plant model are based on the 68 PLA LTI engine. Of course, the rise time of MPC can be made faster by tuning the horizons however, at the cost of set-point overshoot and violation of the core acceleration limits.

As is noticed in Fig. 5.4, the spike in the fuel flow rate during throttling down is not present due to the MPC. Additional engine parameters such as the temperature and pressure at different stations along the engine as described in Fig. 3.4 along with the stall margins of the low pressure and high pressure compressors are compared between the baseline and MPC in Fig. 5.5. This difference arising between the responses of the baseline and MPC in the stall margin of the low pressure compressor is mainly due to the fact that MPC is applied to a LTI engine.

Fault Tolerant Simulations

The health matrices obtained from C-MAPSS40k are important for FDI purposes and engine health diagnostics [20] but can be neglected during fault recovery as the system dynamics in the presence of faults are modeled during linearization at faulty operating conditions. The multi-model approach introduced in Sec 4.8 is used in this thesis where a reference model is based on a certain component fault and is applied on an engine having the same fault assuming proper FDI has been carried

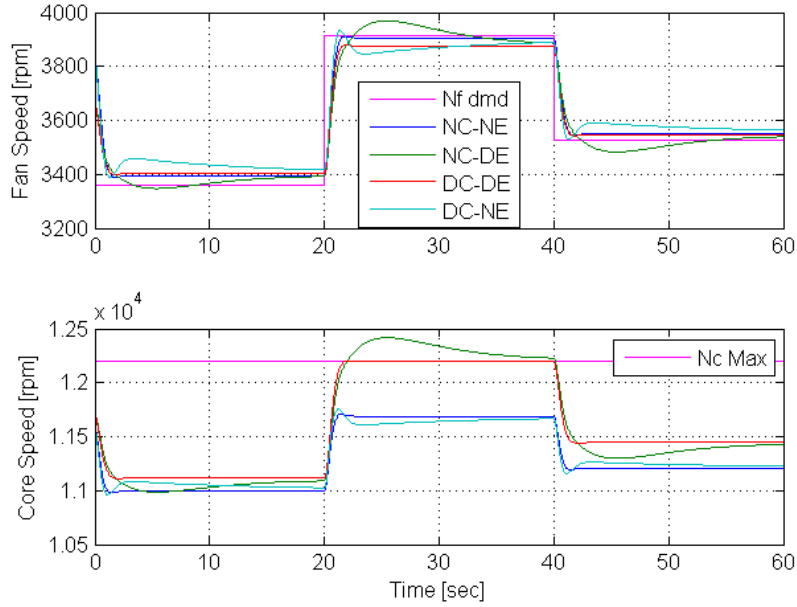


Figure 5.6: Matched and mismatched conditions between the controller and LTI engine

out beforehand i.e. a 20% degradation in the fan efficiency ($FE=-0.2$) of an engine will require a controller whose reference model is based on the same fault of similar magnitude. This is shown in Fig. 5.6 where the best performance is achieved when there is a match between the plant and the linear reference model used by MPC i.e. the Nominal Controller - Nominal Engine (NC-NE) case and Degraded Controller - Degraded Engine (DC-DE) case. When a nominal reference model is used in the presence of faults (NC-DE) i.e. a mismatch, the constraints are violated as is in the case of N_c . Similarly, there is a loss in performance when a degraded reference model is used in the absence of faults (DC-NE). It is assumed that the switching between reference models is taken care off beforehand by appropriate FDI schemes in order to achieve the best performance.

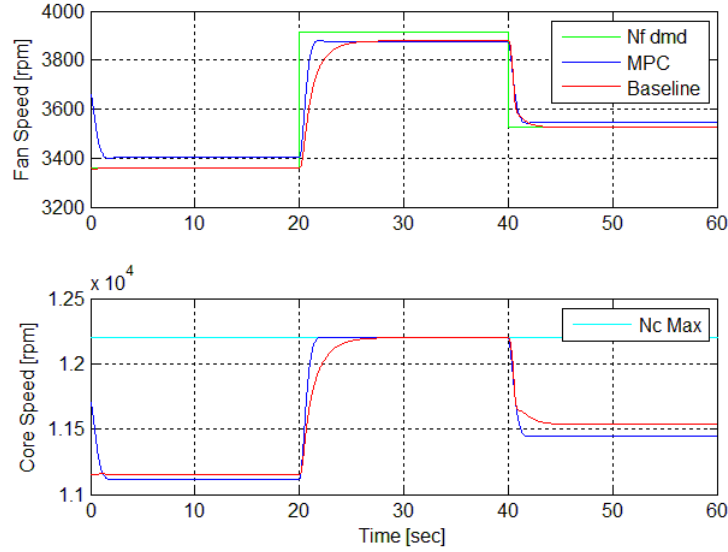


Figure 5.7: Matched MPC and PID for a fan efficiency of 80% (FE=-0.2)

Figures 5.7 - 5.9 show simulations for an engine for the flight conditions FlightA with a 20% degradation in fan efficiency (FE). The effects of a fault are clearly seen between the matched nominal case and the matched degraded case through a shift in engine parameters when compared with Fig. 5.3. The matched MPC compares well with the baseline PID ensuring constrained performance. In addition, the maximum temperature limit at the low pressure turbine exit is 1500° R [19] which the PID controller violates slightly, however this constraint is met by MPC. These simulations show the effectiveness of MPC in handling dynamic system constraints and faults. The input weighting scalar and output weighting vector used for the nominal and 20% fan efficiency degradation are 10 and $[100 \ 19 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0]$. The output weighting matrices Q and R are diagonal with the main diagonal elements defined by the weighting vectors. So, in this case Q and R are of the form:

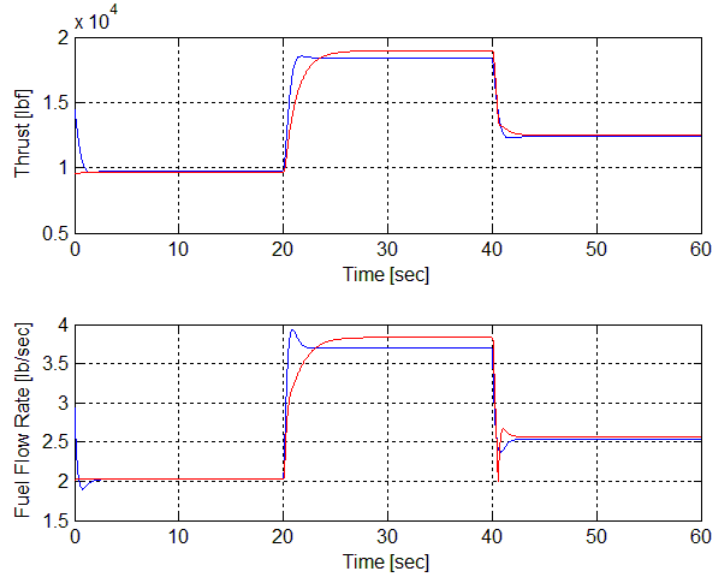


Figure 5.8: Matched MPC and PID for a fan efficiency of 80% (FE=-0.2)

$$Q = \text{diag} [100 \ 19 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0] \quad (5.10)$$

$$R = \text{diag} [10] \quad (5.11)$$

The steady state error in T25 and the stall margins is mainly due to the fact that MPC is being applied to a LTI engine model whereas the baseline controller is being applied to the nonlinear engine. Also, manipulating only one variable, W_f in this case, for an output vector of higher dimension can lead to a sacrifice in few outputs. Therefore, a multivariate control approach is preferred but this leads to much heavier computation.

As shown previously, the best performance from MPC is achieved during a match between the reference model and the engine model. However, it is to model every

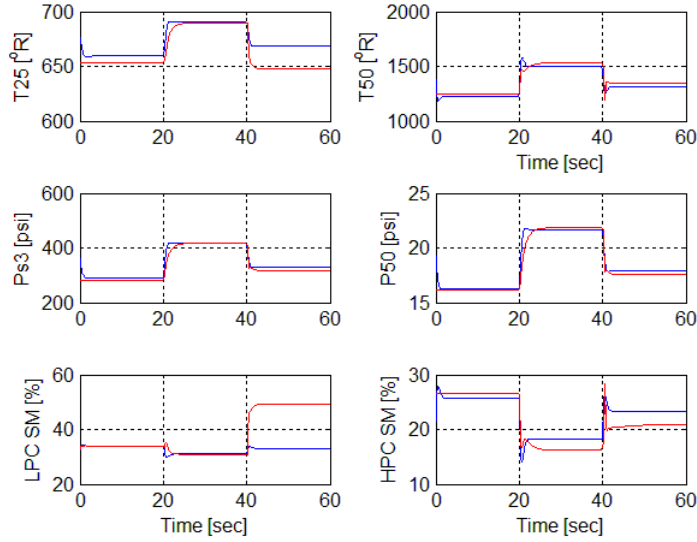


Figure 5.9: Matched MPC and PID for a fan efficiency of 80% (FE=-0.2)

amount of degradation can be tedious and redundant. Figures 5.10 - 5.13 demonstrate the ability of MPC based on a faulty reference model of 80% fan efficiency (FE=80% or FE=-0.2) to handle unmodeled faults. For an illustration, faults that are within $\pm 2\%$ of the modeled faults are used. The matched cases between the reference model and the LTI engine include 20% (fe20match) and 22% (fe22match) deterioration in the fan efficiency in Fig. 5.10. For the matched case with FE=78%, the N_c weighting scalar in Eq. (5.10) is changed to 29 from 19 while calibrating with C-MAPSS40k in order to achieve reasonable performance.

The performance of the mismatched case where a reference model based on FE=80% is applied to an engine with FE=78% (fe20c-fe22e) is similar to the matched FE=78% (fe22 match) case. To achieve this, the output weighting vector for this mismatched case is adjusted to that of the FE=78% reference model i.e. a 22% degradation level is

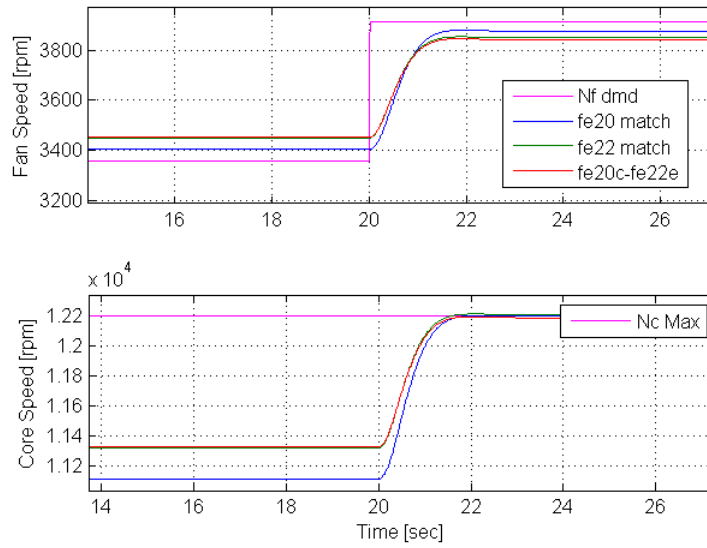


Figure 5.10: FTMPC for 22% degradation using a 20% degradation reference model

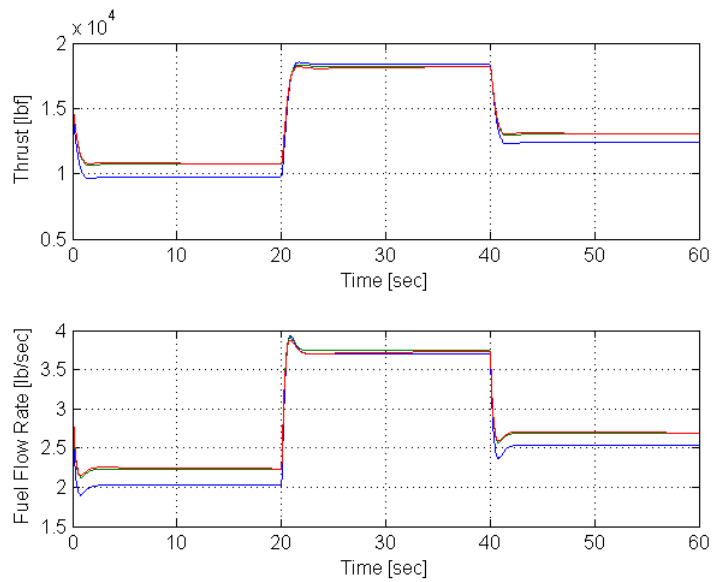


Figure 5.11: FTMPC for 22% degradation using a 20% degradation reference model

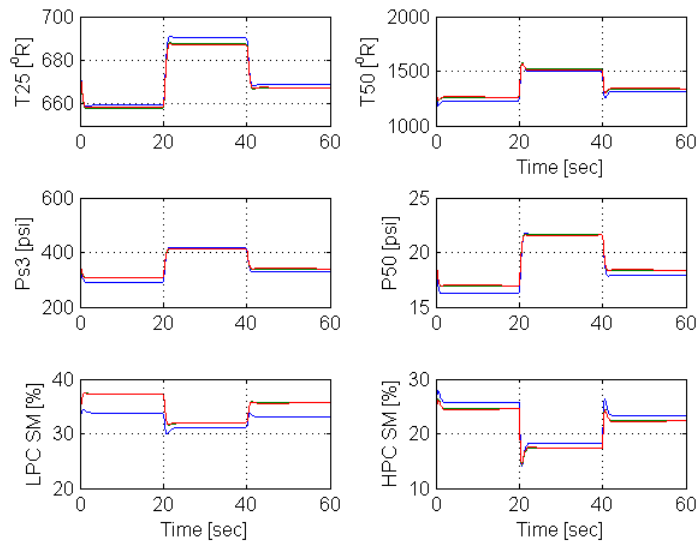


Figure 5.12: FTMPC for 22% degradation using a 20% degradation reference model

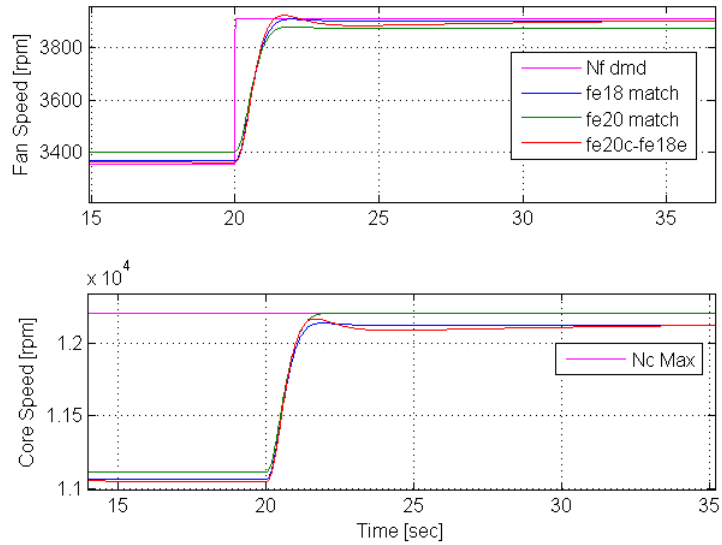


Figure 5.13: FTMPC for 18% degradation using a 20% degradation reference model

handled well by a 20% degradation reference model by simply tuning the appropriate output weights.

Similarly Fig. 5.13 shows the ability of the MPC based on a FE=80% reference model to handle an 18% degradation in fan efficiency (fe20c-fe18e) by tuning the N_c scalar weight to 1 in Eq. (5.10). The tuning of only the N_c scalar weight is sufficient since the core is most likely to violate system constraints more often than other engine parameters and the actual matched performance can be achieved without actually designing a MPC controller for those particular fault magnitudes thus avoiding modeling redundancy.

5.1.4 Thrust Control Simulations

In the control of fan speed or engine pressure, while their demand profiles are satisfied in the presence of component faults ensuring stable operation, there is nevertheless a compromise in the engine performance namely thrust [61, 62]. When comparing Fig. 5.3 and Fig. 5.7 or looking at the matched nominal and degraded cases in Fig. 5.6, the fan speed is compromised due to a fault in the fan i.e. the actual fan speed is lower than the desired fan speed due to a fault. Therefore, the exact demand set-point is not satisfied along with a deviation in the thrust produced from nominal values as shown in Fig. 5.15 which is mainly due to the rise in temperature of the combustion chamber.

Since the goal is to maintain desired thrust response under all conditions, this traditional approach needs to be shed and methods to control the thrust directly need to be implemented. But as mentioned earlier, thrust cannot be measured directly and to overcome this, model based control techniques are used where an on-board engine

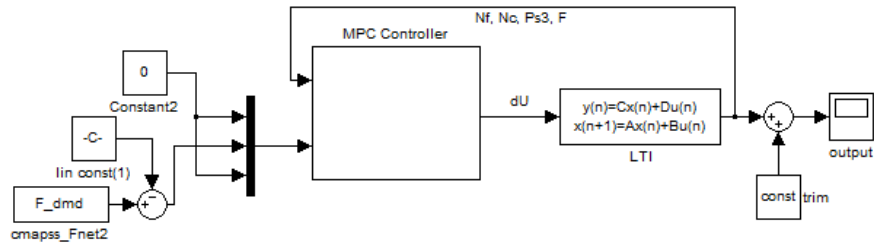


Figure 5.14: MPC simulation in MATLAB/Simulink for Thrust control.

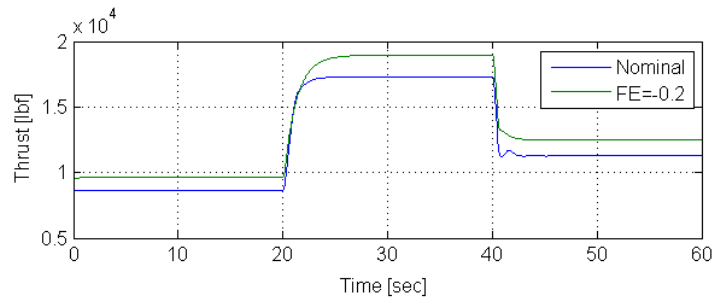


Figure 5.15: Change in thrust due to faults in C-MAPSS40k using fan speed control

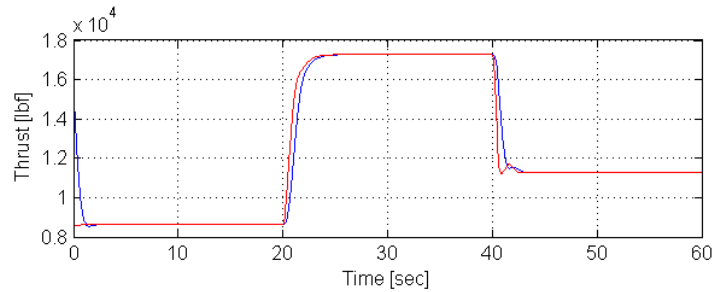


Figure 5.16: Thrust comparison between MPC and PID for nominal conditions. *Blue = MPC, Red = PID*

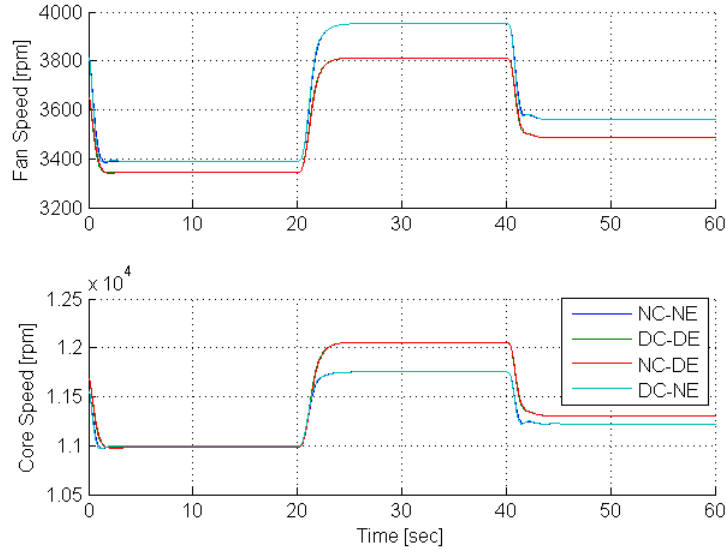


Figure 5.17: Multi-model Thrust control for nominal and 80% fan efficiency.

model is used to estimate the thrust produced [61] generally with the assistance of estimation filters [63]. However, if the linear engine model being used has the thrust as the output with an accurate relation between the thrust produced and the state vectors, then the thrust produced can be used in the feedback loop. In the following thrust control simulations, the simulation model is similar to Fig. 5.2 with fan speed demand replaced with thrust demand and the thrust produced is used the fourth feedback variable. Also, the trim value that is subtracted from the thrust demand is the trim thrust value with all the other output references set to zero.

Since a LTI engine model is used as the plant whose outputs readily includes thrust, thrust estimation techniques such as those proposed in references in [63, 64, 65] can be avoided since the estimated thrust is almost identical to the actual thrust produced. MPC being a model based technique uses a reference model whose output

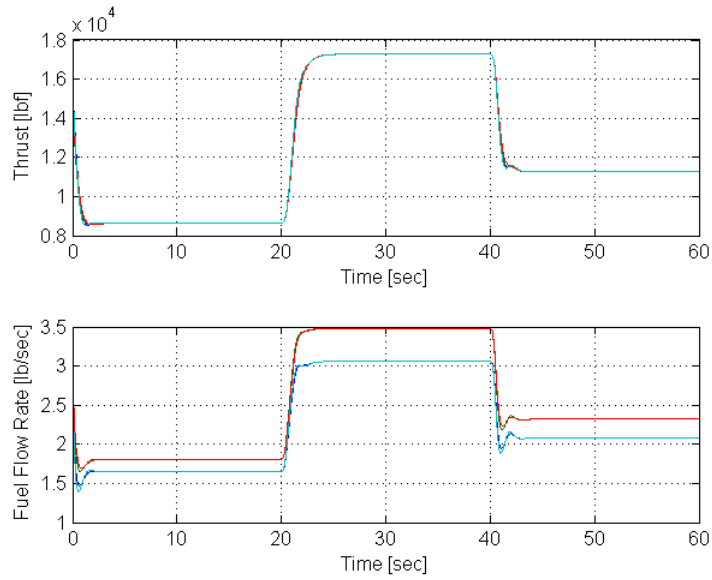


Figure 5.18: Multi-model Thrust control for nominal and 80% fan efficiency.

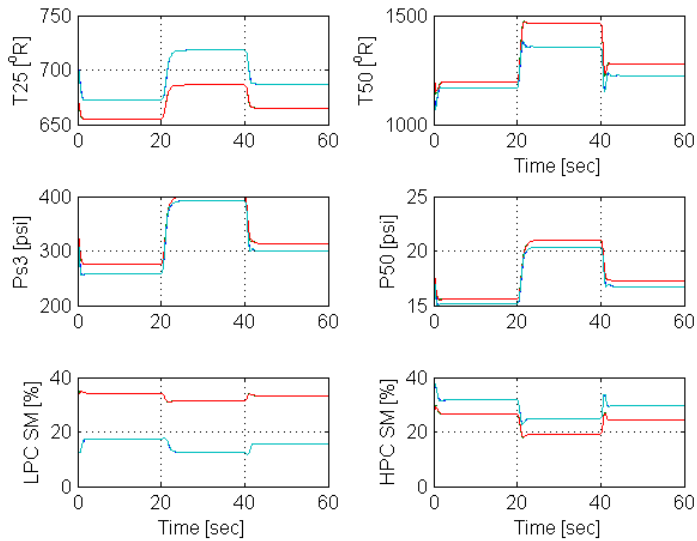


Figure 5.19: Multi-model Thrust control for nominal and 80% fan efficiency.

includes the thrust produced, which can be weighted accordingly. Since this is a thrust control scheme, the input reference to MPC is a thrust profile, which here is the thrust produced by the non-linear C-MAPSS40k engine at the nominal operating conditions specified in Sec. 5.1.2 which can be viewed as the nominal thrust demand. Following the same modeling procedure described in Sec. 5.1.2, Fig. 5.16 shows a comparison between the thrust reference obtained from C-MAPSS40k and the MPC simulation demonstrating the ability of MPC to follow the input thrust profile. The same input weighting scalar as in Eq. (5.11) is used but the output weighting matrix is modified as shown in Eq. (5.12) with a higher importance on thrust.

$$Q = \text{diag} [10 \ 19 \ 1 \ 0 \ 1 \ 1 \ 0 \ 100 \ 0 \ 0] \quad (5.12)$$

The multi-model approach used in the fan speed control simulations is applied for the control of thrust as shown in Fig. 5.17 - 5.19 for the nominal conditions and for a 20% degradation in the fan efficiency (FE=-0.2). Interestingly, the performance depends only on the current health condition of the LTI engine irrespective of the reference model used. In all the cases, the thrust produced remains the same maintaining the engine constraints while the rest of the engine parameters change.

Therefore the modeling of faulty conditions can be avoided without any loss in performance (thrust). This is further demonstrated in Fig. 5.20 - 5.22 where a MPC with a reference model based on nominal operating conditions is applied to a LTI engine with a magnitude in degradation of 18%, 20% and 22% in fan efficiencies. In addition, N_c never reaches its physical limits during thrust control at FE=-0.2 as was the case during fan control.

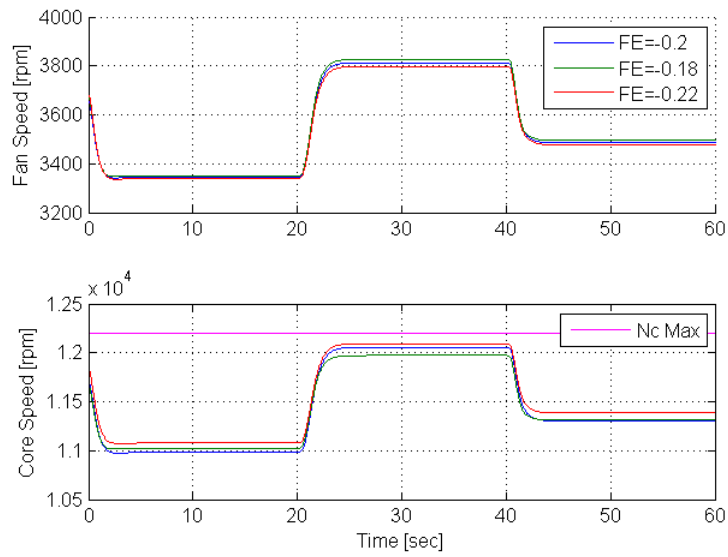


Figure 5.20: Thrust control using a nominal reference model for 78%, 80% and 82% fan efficiencies .

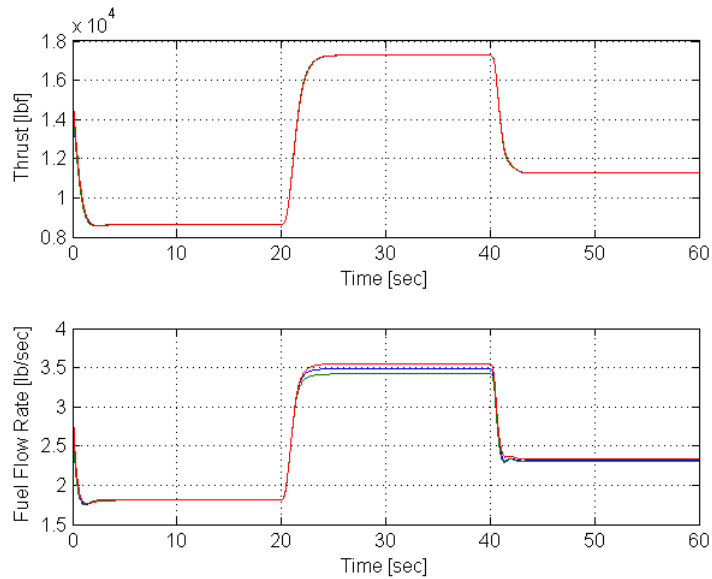


Figure 5.21: Thrust control using a nominal reference model for 78%, 80% and 82% fan efficiencies .

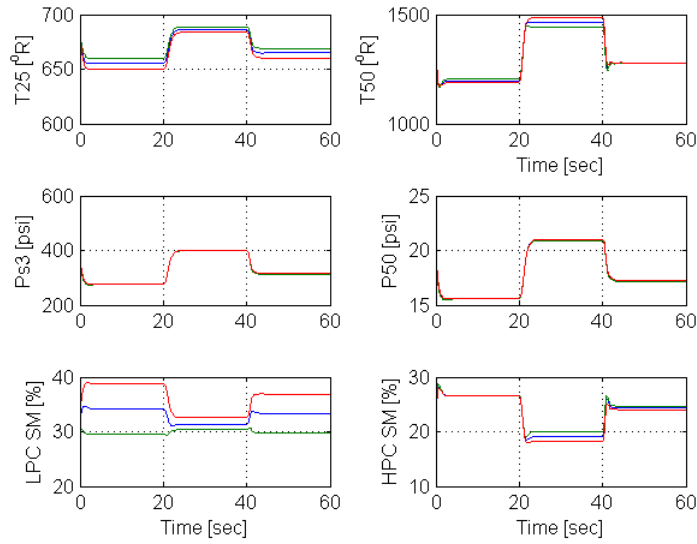


Figure 5.22: Thrust control using a nominal reference model for 78%, 80% and 82% fan efficiencies .

Therefore implementing thrust control not only avoids additional modeling of faulty conditions but also ensures that the actual desired performance (thrust) is maintained at desired levels at all times.

5.2 Nonlinear Simulations

Following the implementation of MPC on LTI engine models, this section describes the application of MPC to the actual nonlinear C-MAPSS40k engine to obtain the true comparison between the performance of the baseline PID controller and MPC that continues to exploit a LTI reference model.

Figure 5.23 shows the parent C-MAPSS40k Simulink layer with both the baseline PID and the switched multiple MPC (SMMPC) subsystem along the with the time based supervisor whose implementation is shown in Fig. 5.27. Using the manual

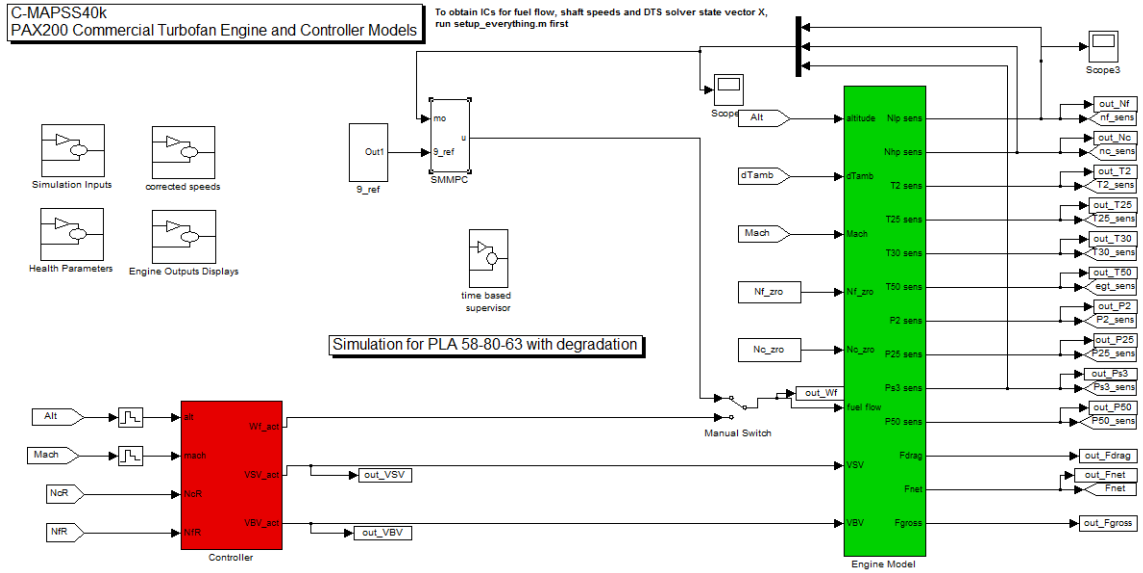


Figure 5.23: Parent layer with baseline and MPC applied to the nonlinear engine.

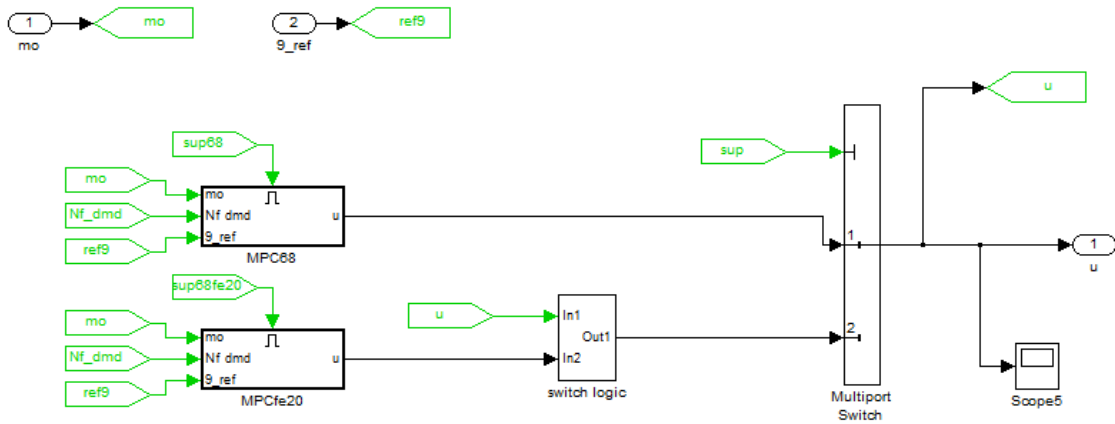


Figure 5.24: SMMPC subsystem with multiple MPCs and switch logic.

switch, the control law from either the baseline or MPC can be selected. The time based supervisor in Fig. 5.27 is used to trigger, based on time, a corresponding MPC block in Fig. 5.24 and then select its respective control law. The 9_ref subsystem in Fig. 5.23 is a multiplexed input of nine zeros for the outputs of the engine excluding the fan speed.

The SMMPC subsystem has two MPC blocks both designed at PLA 68. However, the MPCfe20 block is designed at a fan efficiency of 80% i.e. a 20% degradation in the fan efficiency. Hence this block is triggered only in the scenario where a fault in the fan should occur otherwise the nominal model based MPC68 would be ON. In Sec. 5.1.2, Δu was the optimized control input being applied to the LTI engine.

However in the nonlinear simulation, the nonlinear engine takes u as the control input. Therefore using Eq. (5.5), u is calculated as shown below and the implementation is shown in Fig. 5.25.

$$u_0 = \Delta u_0 \tag{5.13}$$

$$u_1 = u_0 + \Delta u_1 \tag{5.14}$$

$$u_k = u_{k-1} + \Delta u_k \tag{5.15}$$

On the assumption of the presence of FDI, when a fault is detected, the appropriate MPC is triggered and the sudden switching in the control law can cause a bump in the response of the engine. Hence a bumpless switch logic 'Inertia Delayed to Soften the Switch' proposed by Wang [11] is used in these simulations for a smooth transition between the different MPC control laws. The switching logic from Wang [11] is modified in this work as

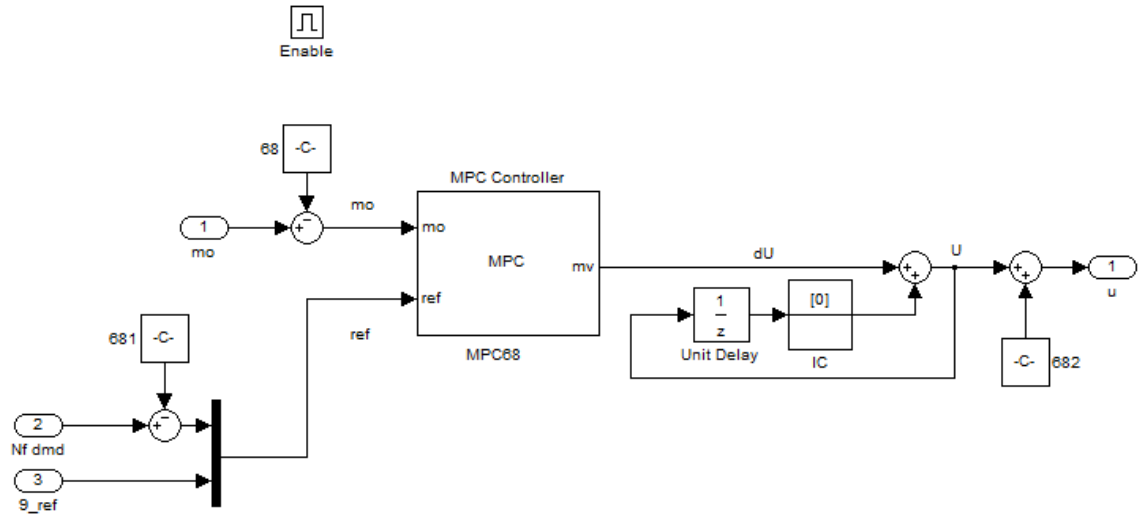


Figure 5.25: MPC68 subsystem with MPC designed at PLA 68.

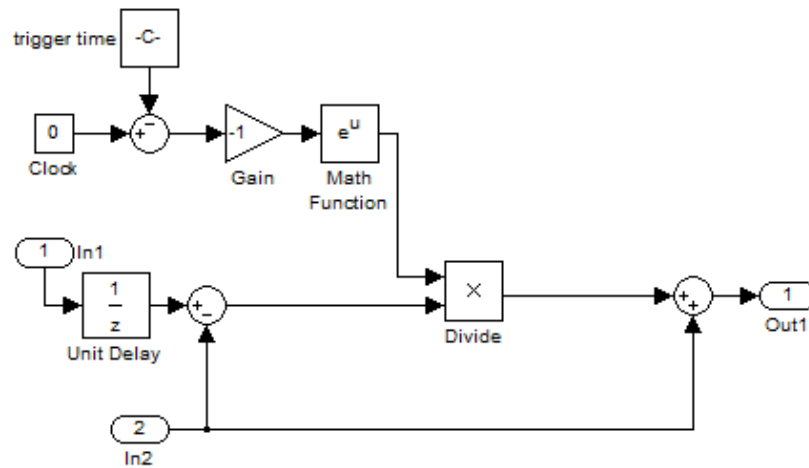


Figure 5.26: Implementation of switching logic.

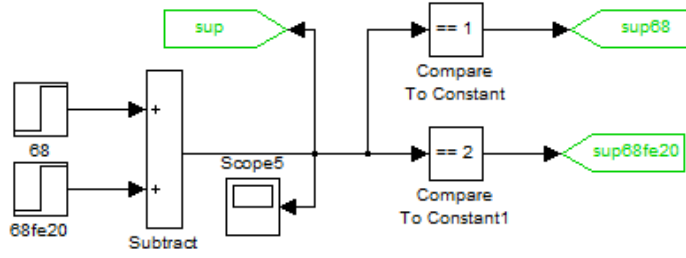


Figure 5.27: Implementation of time based supervisor.

$$u_k = u_b + e^{-t}(u_{k-1} - u_b) \quad (5.16)$$

where u_b is the control input from the new triggered MPC and t is the total working period of the new triggered MPC from the triggered time instant to the current time instant. As a result, at the trigger time instant $t = 0$ the current control law is that from the previous time instant i.e. $u_k = u_{k-1}$. As t increases, the effect of u_b increases exponentially till the current law is solely dictated by it thus achieving a smooth exponential transition in control law. The implementation of the switch logic is shown in Fig. 5.26. Therefore the actual switching would be from the MPC68 at nominal conditions to the MPC68fe20 for a permanent damage in the fan. Just as mentioned in Sec. 5.1.3, the trim values are subtracted at appropriate locations in Fig. 5.25 since the MPC uses a LTI reference model for computation of the control law.

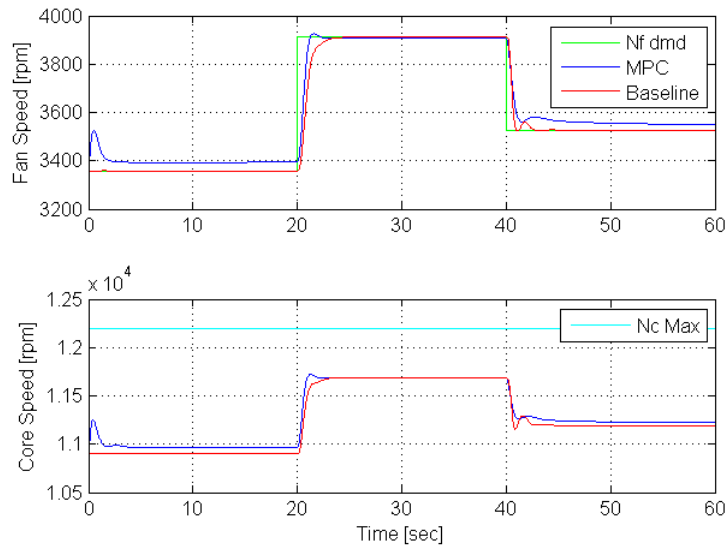


Figure 5.28: MPC and PID applied to a nominal nonlinear engine.

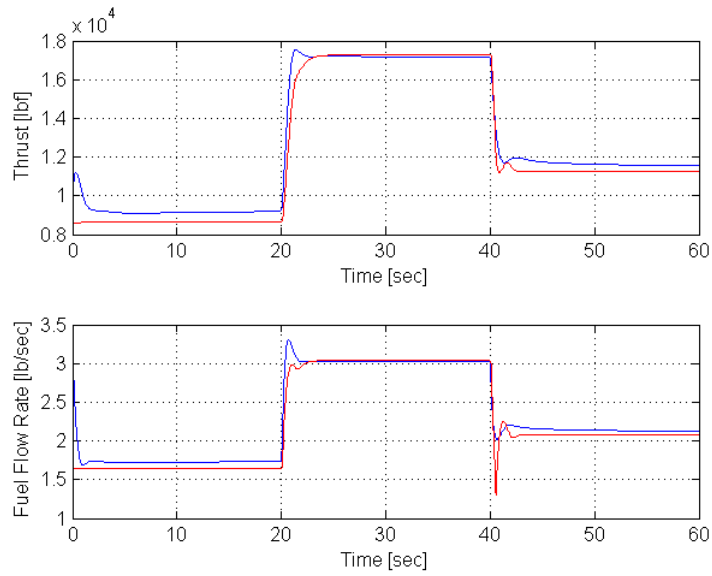


Figure 5.29: MPC and PID applied to a nominal nonlinear engine.

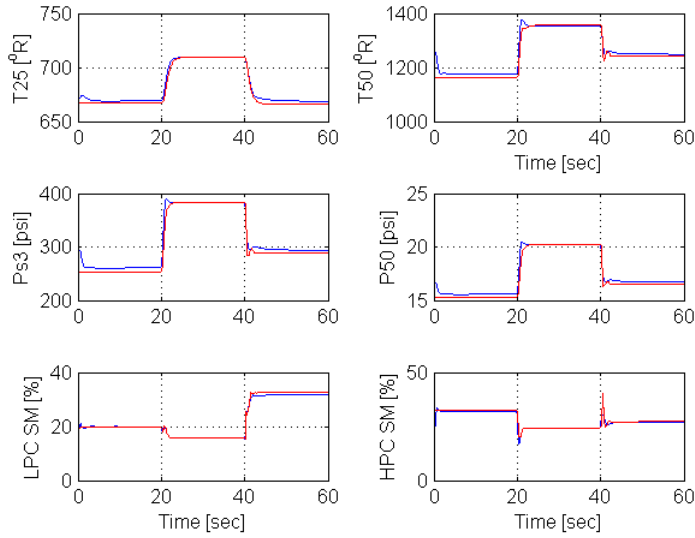


Figure 5.30: MPC and PID applied to a nominal nonlinear engine.

5.2.1 Nonlinear Fan Speed Control Simulations

Figures 5.23 - 5.25 are specifically designed for the control of the fan speed. In this section the simulations for the application of MPC to the nonlinear C-MAPSS40k engine for the control of the fan speed are discussed in detail for the flight conditions FlightA. For the same PLA demand profile and the ambient conditions specified in the Sec. 5.1.3, MPC designed at 68 PLA i.e. a linear reference model based MPC is applied to nonlinear C-MAPSS40k engine and the performance is compared with that of the baseline controller as shown in Fig. 5.28 - 5.30 for the nominal case. Also the same weighting matrices as described in Eq. (5.10) and Eq. (5.11) are used here.

The engine response due to MPC compares well with that of the baseline PID. In addition, the spikes found in the baseline control law during the PLA transient from 80 to 63 is absent in the case of the MPC. Another point to note is that the

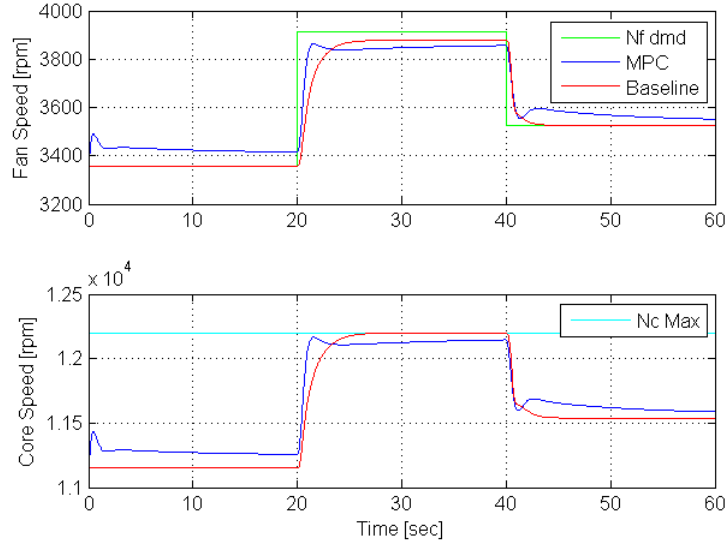


Figure 5.31: Matched MPC and PID applied to a nonlinear engine for FE=-0.2.

discrepancy in the stall margin of the low pressure compressor found during the linear engine simulation in Sec. 5.1.3 is not found here since the actual physical engine is being simulated.

Figures 5.31 - 5.33 show the case when there is a 20% degradation in the fan efficiency at the start of the simulation. Assuming proper FDI prior to the start of the simulation, the MPC block MPC68fe20 is triggered and the performance is compared with that of the baseline controller. For a degradation in fan efficiency of the nonlinear engine, Eq. (5.10) is modified to

$$Q = \text{diag} [100 \ 23 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0] \quad (5.17)$$

so that the response of N_c is appropriate and within reasonable bounds. The performance of the MPC technique once again compares well with that of the baseline

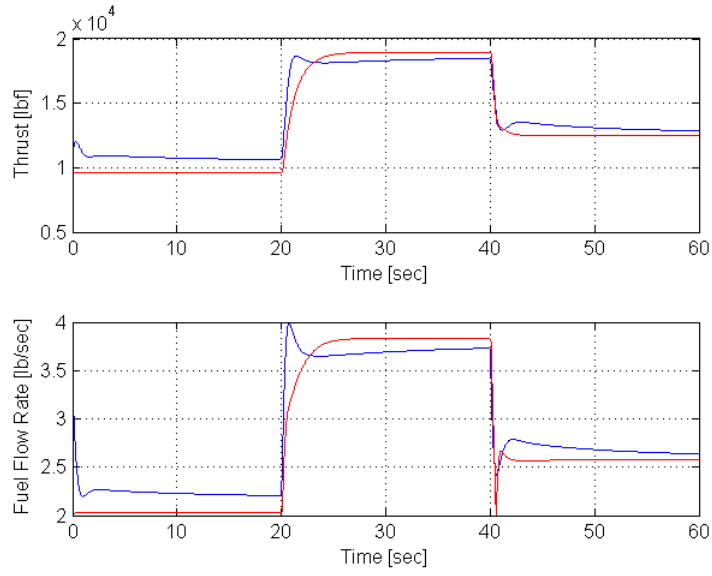


Figure 5.32: Matched MPC and PID applied to a nonlinear engine for $FE=-0.2$.

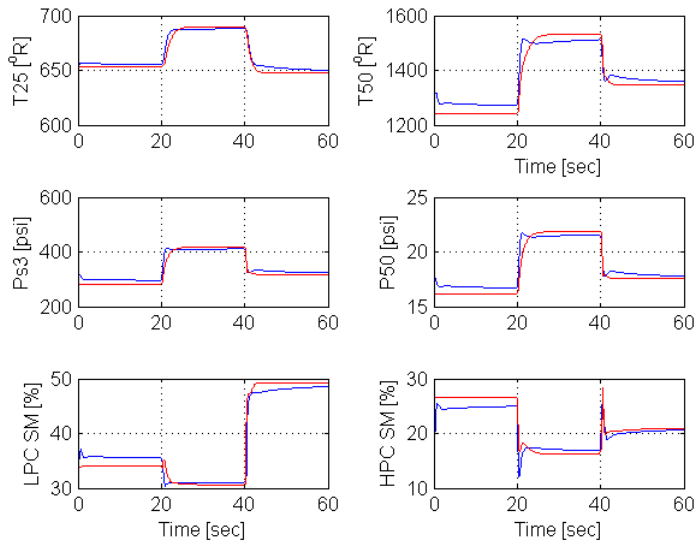


Figure 5.33: Matched MPC and PID applied to a nonlinear engine for $FE=-0.2$.

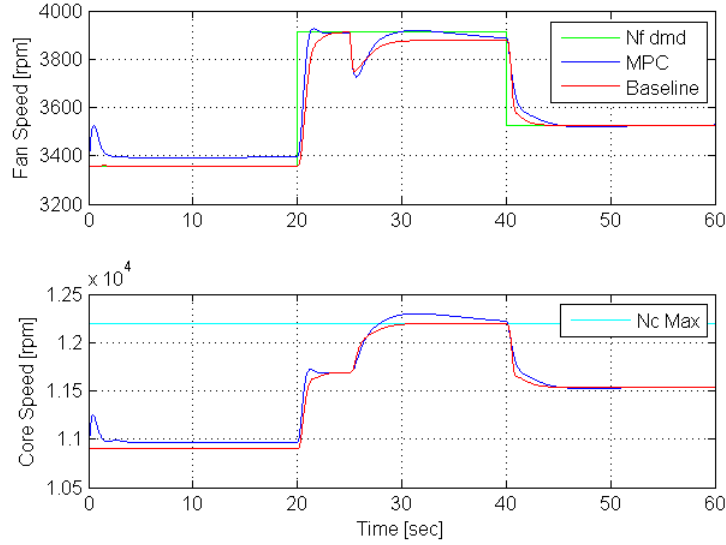


Figure 5.34: Nonlinear engine without MPC fault recovery for $FE=-0.2$ at 25 sec.

controller. N_c does not operate at its physical limits hence the engine is not taken to its extremities. Also the fuel flow rate computed by the MPC does not have the spike which is present due to the baseline controller and the responses of the temperature, pressure and stall margins are in accordance with both the control laws. However just as in Sec. 5.1.3, the temperature limit at the low pressure turbine of $1500^\circ R$ is maintained by the matched MPC.

In Fig 5.34 - 5.36, a permanent fault where the fan efficiency drops by 20% is introduced at 25 sec and no action is taken by the MPC i.e. the nominal reference model based MPC continues to operate in the presence of the fault occurring at 25 sec. The change in engine response is clearly seen at 25 sec due to the occurrence of the fault. Since no fault recovery action is carried out by MPC, the N_c response violates its physical limits and so does the exit temperature of the low pressure turbine T_{50} .

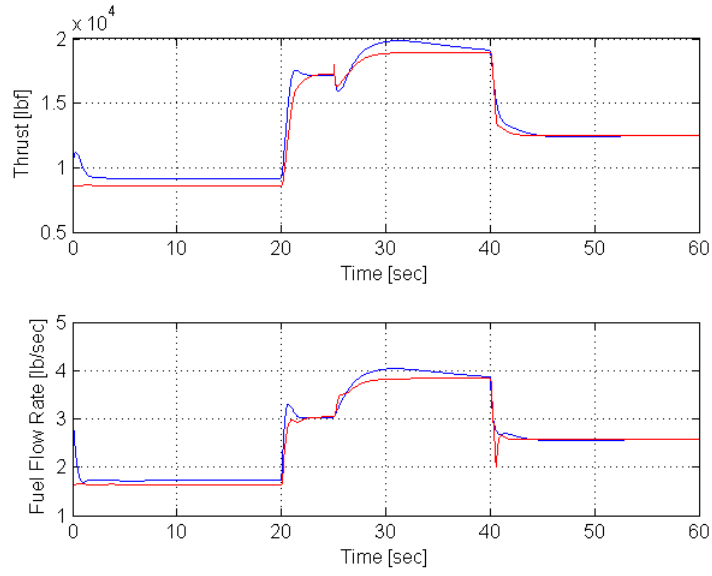


Figure 5.35: Nonlinear engine without MPC fault recovery for FE=-0.2 at 25 sec.

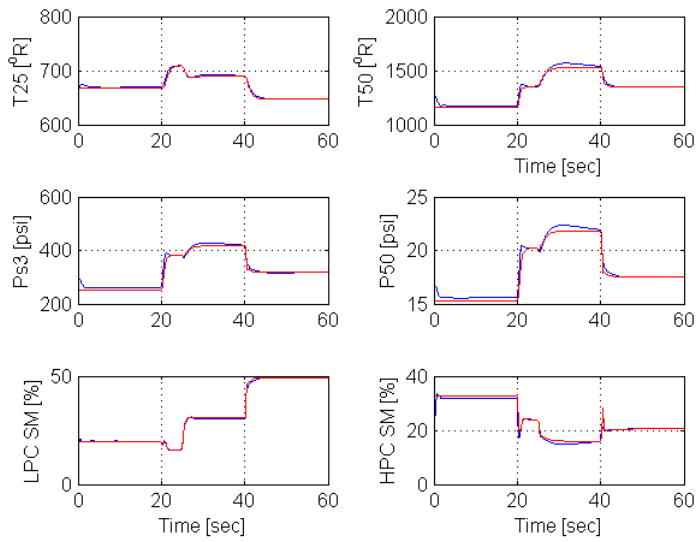


Figure 5.36: Nonlinear engine without MPC fault recovery for FE=-0.2 at 25 sec.

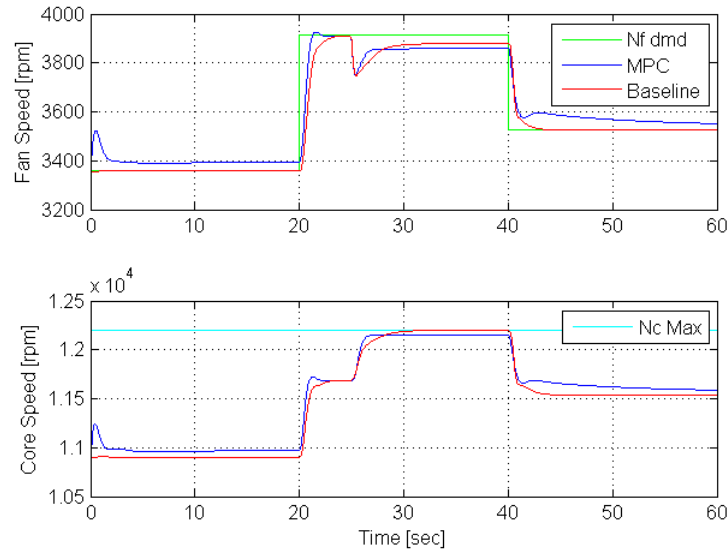


Figure 5.37: Nonlinear engine with MPC fault recovery for $FE=-0.2$ at 25 sec.

This is intuitive since there is a model mismatch in the presence of faults. Hence, it is important to use an accurate FDI algorithm to govern the switching of the MPC blocks ensuring constrained engine performance.

In Fig 5.37 - 5.39, on the contrary, assuming appropriate FDI has located and identified the fault, the MPC control block is switched from MPC68 to MPC68fe20 in Fig. 5.24 at 25 sec to ensure the engine operates within its physical constraints. The difference in the engine response is now clearly seen when compared to Fig 5.37 - 5.39. N_c is within its limits and so is $T50$. Also the response of MPC is faster than the baseline in terms of attaining the steady state value in the neighborhood of the fault. This is probably due to the switching logic used during the transition from the nominal reference model based MPC to the degraded reference model based MPC and hence also achieving a smooth response in the presence of a fault.

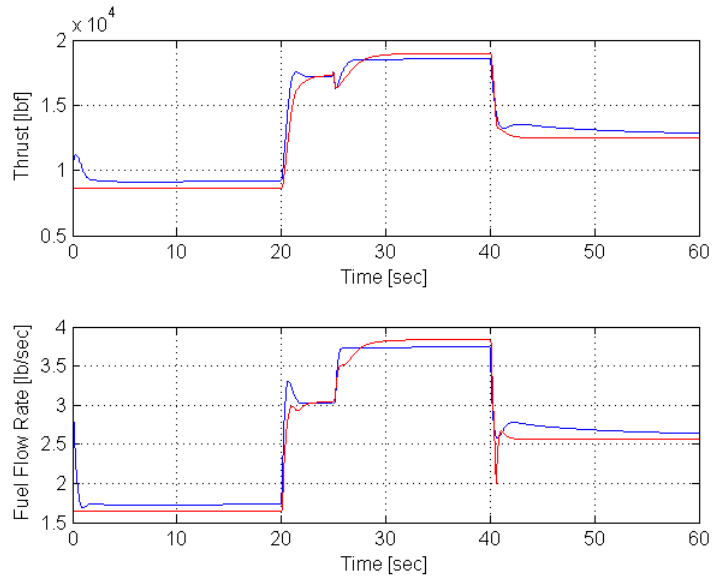


Figure 5.38: Nonlinear engine with MPC fault recovery for FE=-0.2 at 25 sec.

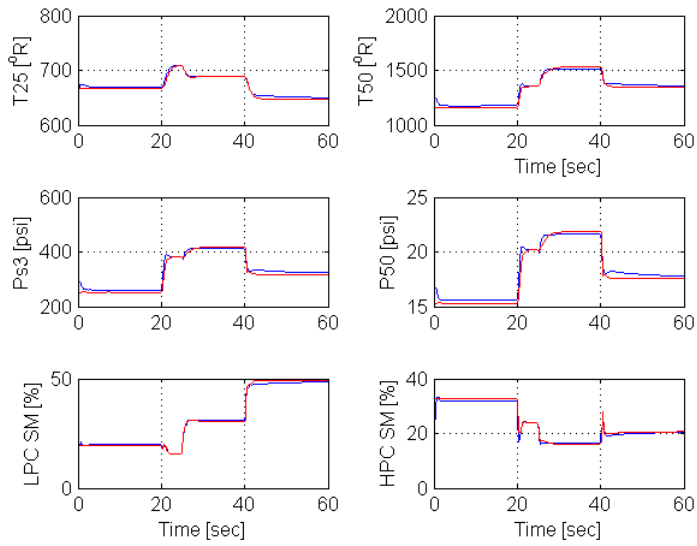


Figure 5.39: Nonlinear engine with MPC fault recovery for FE=-0.2 at 25 sec.

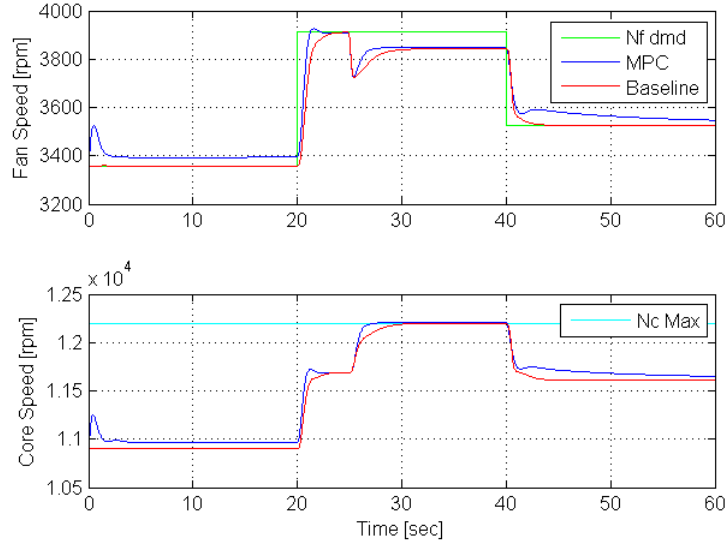


Figure 5.40: MPC68fe20 applied to a Nonlinear engine with FE=-0.22 at 25 sec.

As demonstrated in Sec. 5.1.3, the ability of MPC to handle faults within $\pm 2\%$ of the modeled faulty reference model is shown next but on a nonlinear engine. In Fig. 5.40 - 5.42 for FlightA, a fault is induced at 25 sec such that there is a 22% degradation in the fan efficiency and the fault recovery is carried out by switching to the MPCfe20 block at 25 sec. The weighting scalar for N_c used in Eq. (5.17) is modified to 25 from 23.

As seen clearly, the performance of the MPC agrees with that of the baseline controller for a degradation of 22% in every aspect. The same procedure is carried out next but for a degradation in the fan efficiency of 18% as shown in Fig. 5.43 - 5.45. The weighting scalar for N_c used in Eq. (5.17) is modified to 3 from 23. Once again, the MPC based on a 20% degradation model can successfully handle a 18% degradation in the fan efficiency of the nonlinear engine.

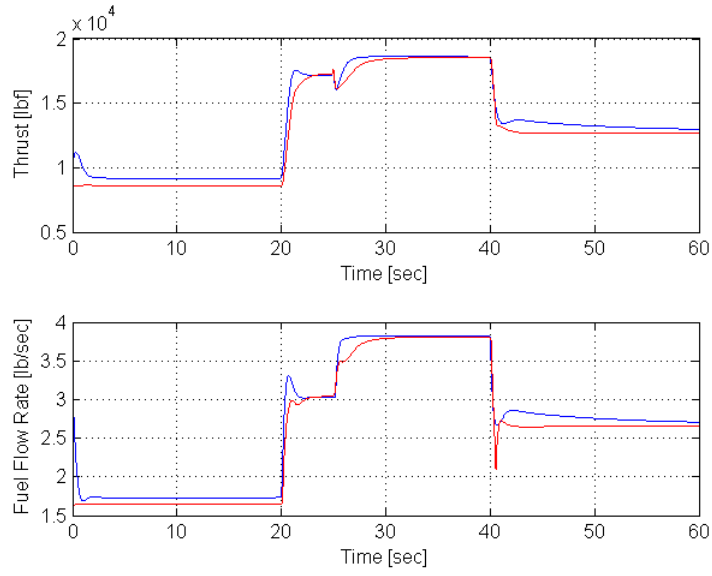


Figure 5.41: MPC68fe20 applied to a Nonlinear engine with FE=-0.22 at 25 sec.

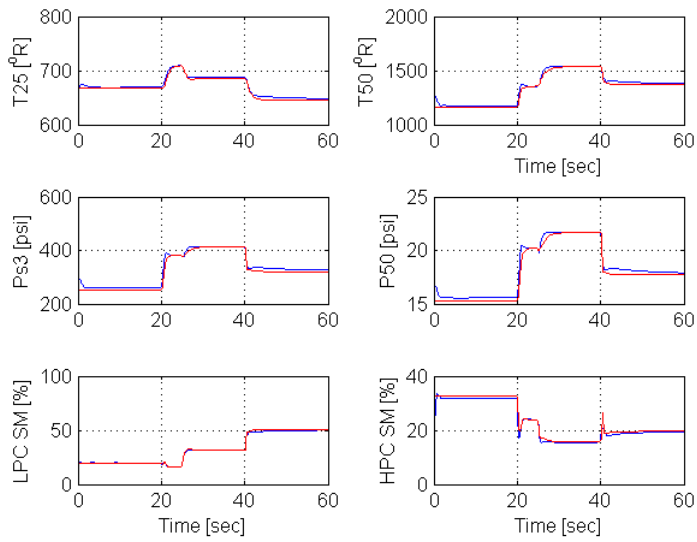


Figure 5.42: MPC68fe20 applied to a Nonlinear engine with FE=-0.22 at 25 sec.

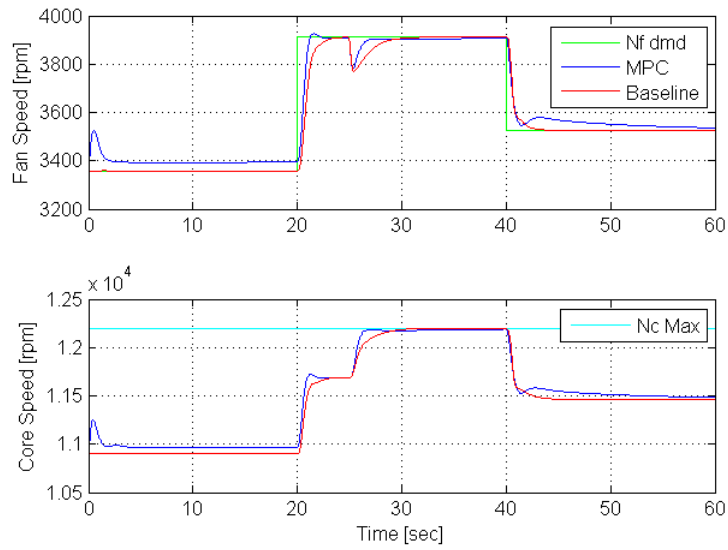


Figure 5.43: MPC68fe20 applied to a Nonlinear engine with FE=-0.18 at 25 sec.

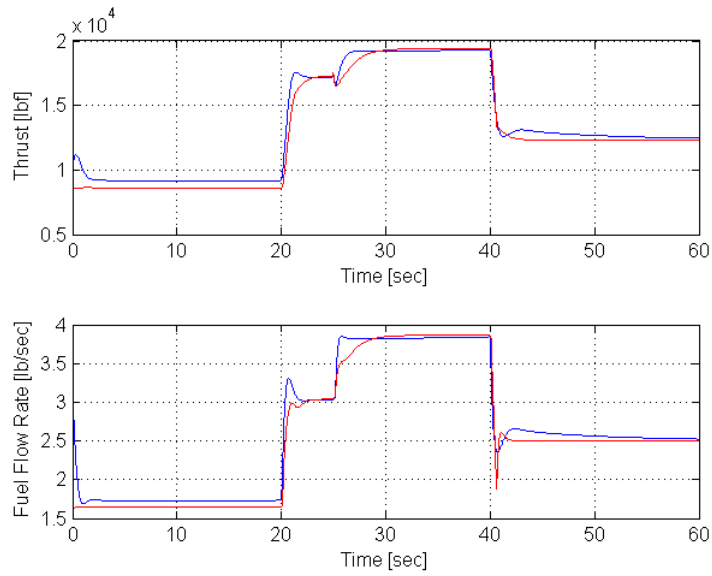


Figure 5.44: MPC68fe20 applied to a Nonlinear engine with FE=-0.18 at 25 sec.

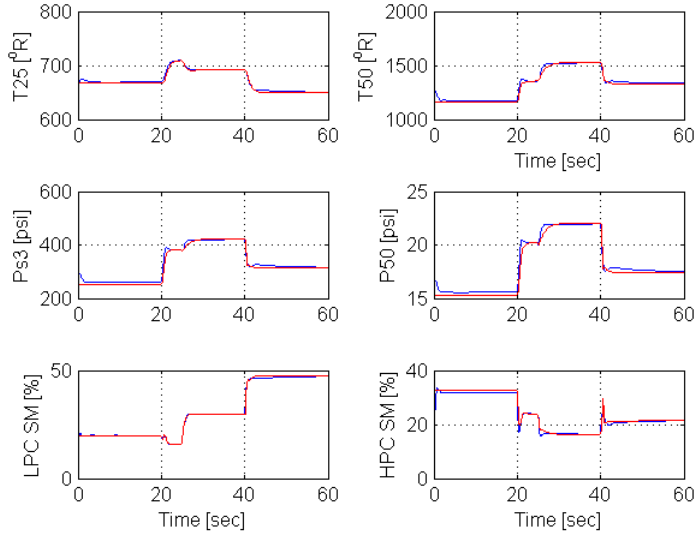


Figure 5.45: MPC68fe20 applied to a Nonlinear engine with FE=-0.18 at 25 sec.

Although the simulation time is 60 sec, since the computationally heavy MPC is being applied to a nonlinear physical engine, the time taken to compute this simulation scenario is 143 sec. This time can be reduced by either increasing the computational power or by improving the MPC algorithm.

To address this issue, an on-line change in the prediction horizon is experimented with and is shown in Fig. 5.46 - 5.48 for the flight conditions FlightA where the simulation starts with a prediction horizon $N_P = 20$ and changes to $N_P = 50$ during the rise in PLA and back to $N_P = 20$ during the steady state PLA of 80.

The same process is repeated during the fall in PLA from 80 to 63. This procedure takes approximately 60 sec and hence can be approximated to that of a real time simulation. However the engine response achieved has small peaks during the switching process between the two MPC blocks (MPC68) with prediction horizons

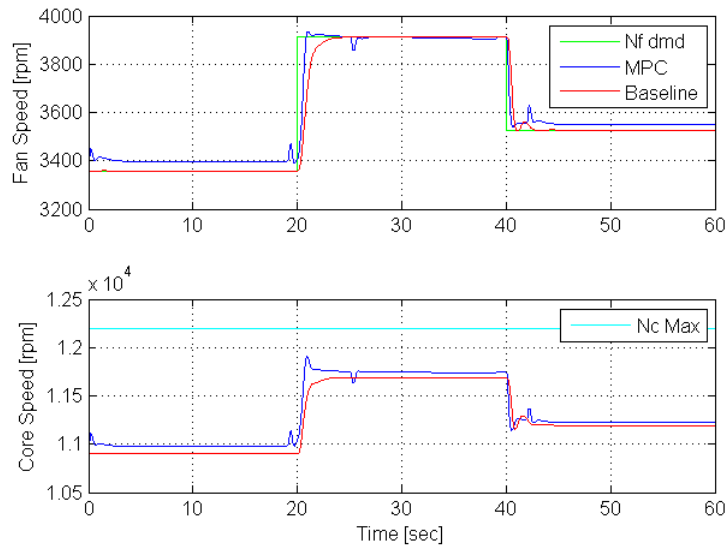


Figure 5.46: Effect of switched prediction horizons $N_P = 20, 50$.

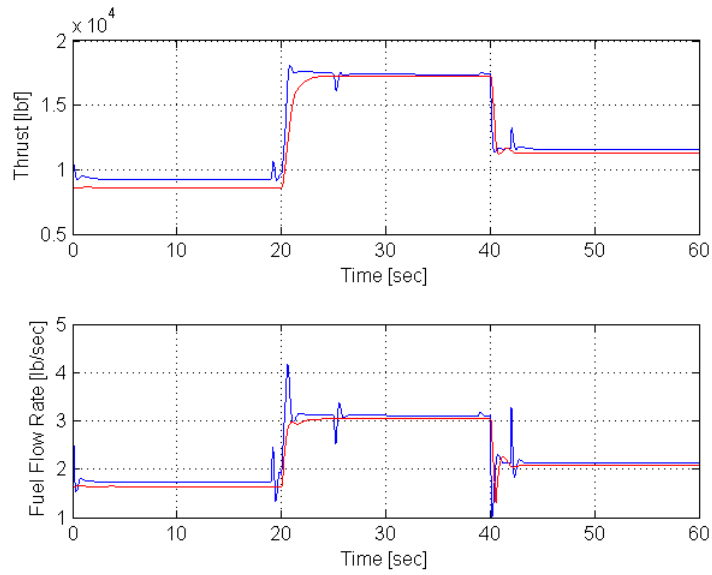


Figure 5.47: Effect of switched prediction horizons $N_P = 20, 50$.

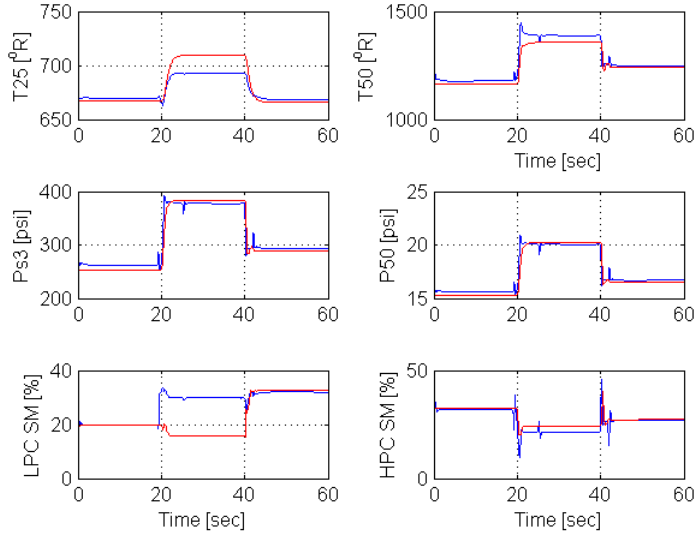


Figure 5.48: Effect of switched prediction horizons $N_P = 20, 50$.

of 20 and 50 and this is possibly a reflection on the difference in the optimization process in Eq. (4.1) which arises due to the switching between different prediction horizons.

5.2.2 Nonlinear Thrust Control Simulations

As demonstrated earlier at the end of Sec. 5.1.3, during the traditional fan speed control approach, there is an increase in the thrust produced by the engine mainly due to the rise in engine temperature when a fault in the fan section occurs. This rise in temperature can take the low pressure turbine (LPT) blades to its temperature limits which deteriorates their health at a faster rate. Therefore the control of Thrust is investigated on the nonlinear C-MAPSS40k engine as done in Sec. 5.1.4.

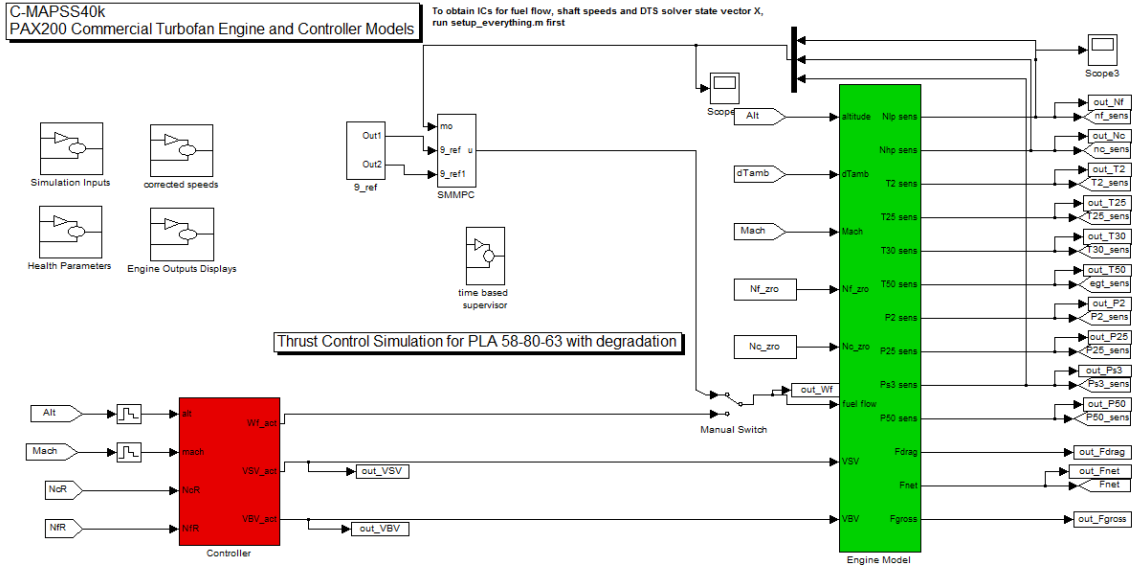


Figure 5.49: Parent layer with baseline and MPC for thrust control of the nonlinear engine.

In C-MAPSS40k, the thrust is calculated using Eq. (1.2) and the nonlinear engine represents the actual gas turbine engine. Since the thrust produced cannot be measured directly using sensors, the thrust is not part of the measured feedback loop in Fig. 5.23. The thrust is controlled using the measured parameters i.e. fan speed N_f , core speed N_c and the chamber pressure $Ps3$ in the feedback loop just as in Sec. 5.2.1. The LTI reference model used by the MPC can predict the thrust produced during the optimization process and hence this is similar to a model based thrust estimation using an on-board model.

Figures 5.23 - 5.25 are modified to Fig. 5.49 - 5.51 for the simulation of the thrust control approach where the fan speed reference is now to set to zero. In order to provide the pilot with the thrust corresponding to the PLA positions, the thrust produced by the nonlinear engine using the baseline controller for the nominal

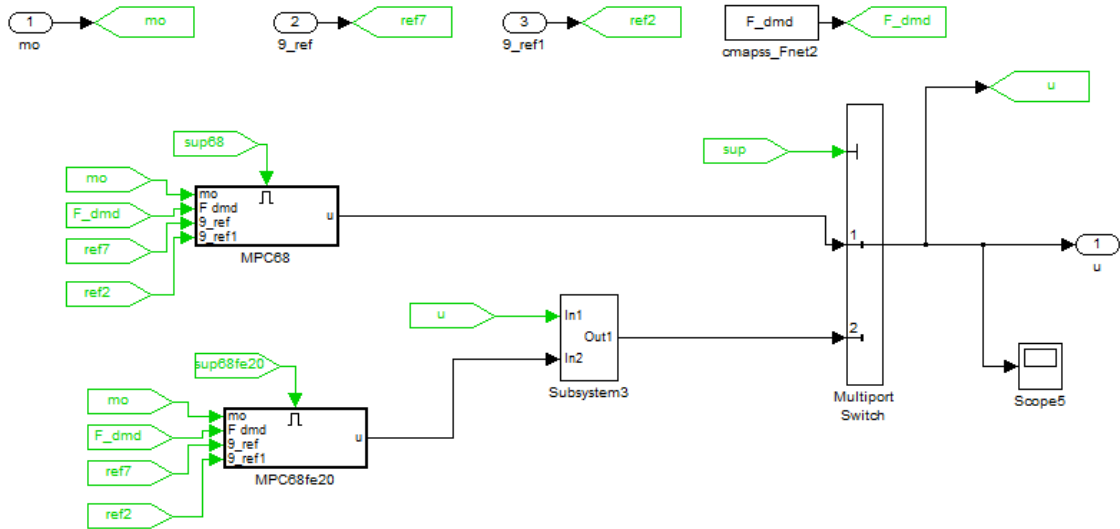


Figure 5.50: Modified SMMPC subsystem with multiple MPCs and switch logic.

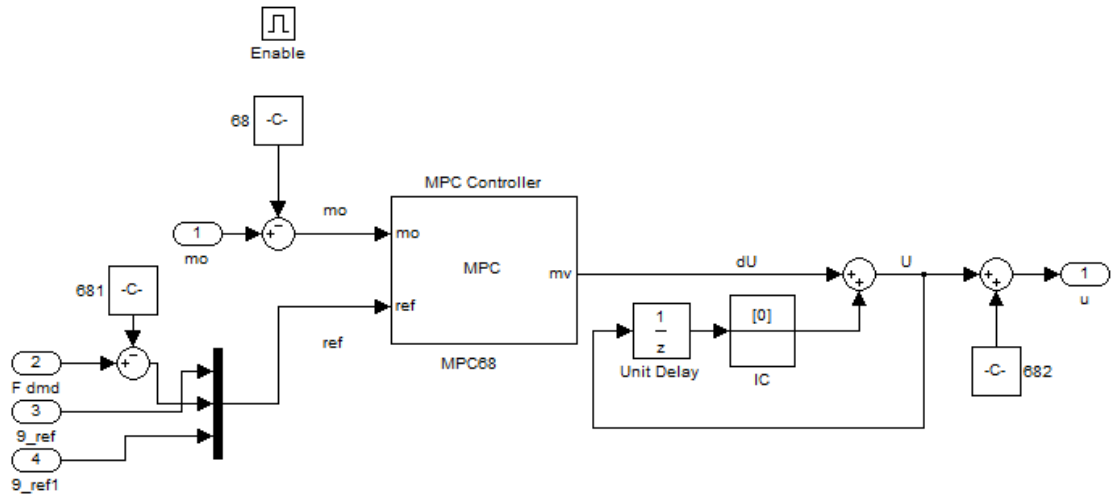


Figure 5.51: Modified MPC68 subsystem with MPC designed at PLA 68.

FlightA conditions is provided as the nominal thrust reference to the MPC. The 9_ref subsystem is now modified to provide two sets of zero references - one of 7 and the other of 2 - for the outputs of the nonlinear engine excluding the thrust.

For the following simulations, the same flight profile FlightA is used with Eq. (5.12) modified to

$$Q = \text{diag} [1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1000 \ 0 \ 0] \quad (5.18)$$

Figures 5.52 - 5.54 show a comparison between the MPC used for the thrust control and the baseline used for the fan speed control of the nonlinear engine for the nominal FlightA conditions. Since the nominal thrust produced by the nonlinear engine using the baseline (for fan speed control) is used as the thrust reference for the MPC for the control of thrust, the actual comparison in performance is seen in Fig. 5.53. The thrust produced due to MPC follows its reference closely.

Although the resulting fan speed due to MPC exceeds the N_f demand for fan speed control, the task of MPC now is to control the thrust only ensuring the other parameters do not exceed their limits. As seen throughout, the control law computed by MPC does not have the spikes during the fall of PLA. For the nominal conditions, the advantage of thrust control is not emphasized enough as the drawback of fan speed control seen under the influence of a fault in the fan.

A fault in the fan is now induced at 25 sec. In Sec. 5.1.4 the multi-model approach was redundant and a MPC using a LTI reference model obtained at nominal conditions was sufficient, in the presence of faults, to maintain the desired thrust. In Fig. 5.55 - 5.57 however this is not the case as the thrust increases from the desired

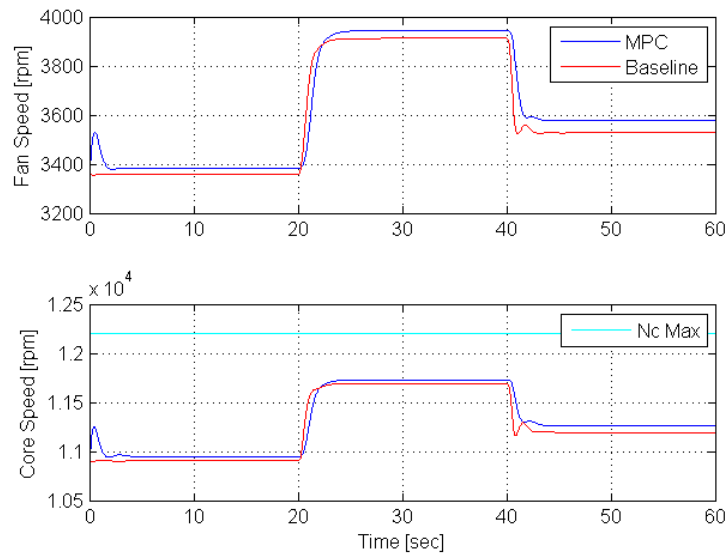


Figure 5.52: MPC thrust control of the nominal nonlinear engine.

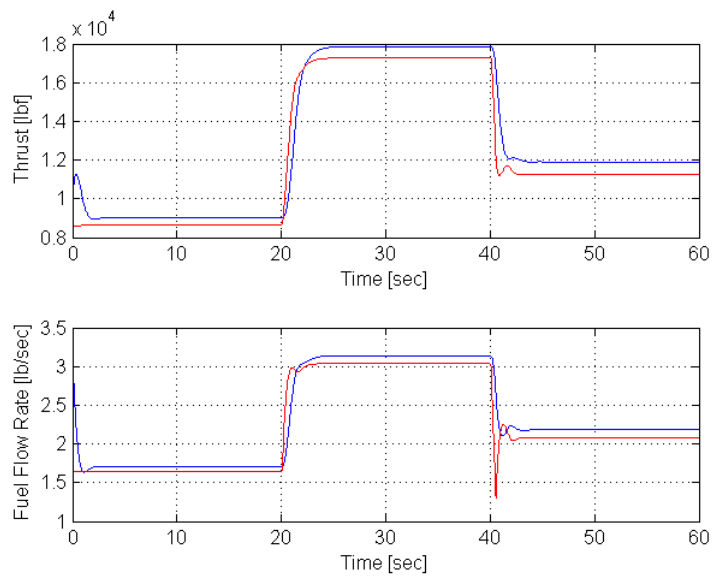


Figure 5.53: MPC thrust control of the nominal nonlinear engine.

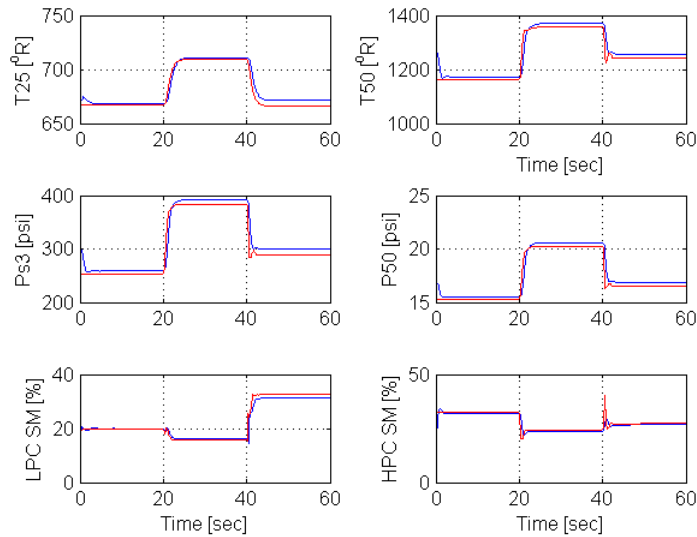


Figure 5.54: MPC thrust control of the nominal nonlinear engine.

value at 25 sec due to the presence of a fault in the fan even though the nominal thrust produced is used as the reference.

Hence, at 25 sec the MPC blocks are switched from MPC68 to MPC68fe20 in Fig. 5.50 using the time based supervisor described in Sec. 5.2.1. This is shown in Fig. 5.58 - 5.60 where the performance of the baseline PID at $FE=-0.2$, baseline PID at nominal conditions for fan speed control and the switched MPC for thrust control are all compared. In Fig. 5.59, the nominal thrust is used as the reference for MPC as well as the standard for comparison. Till 25 sec, the Nom PID and Baseline match as they are identical. Due to the induced fault, the baseline deviates from Nom PID and produces thrust that is higher than desired. However, when the MPC is switched at 25 sec, the thrust recovers to the desired nominal value which is the main advantage of employing thrust based engine control. Additionally, the fuel consumed by the

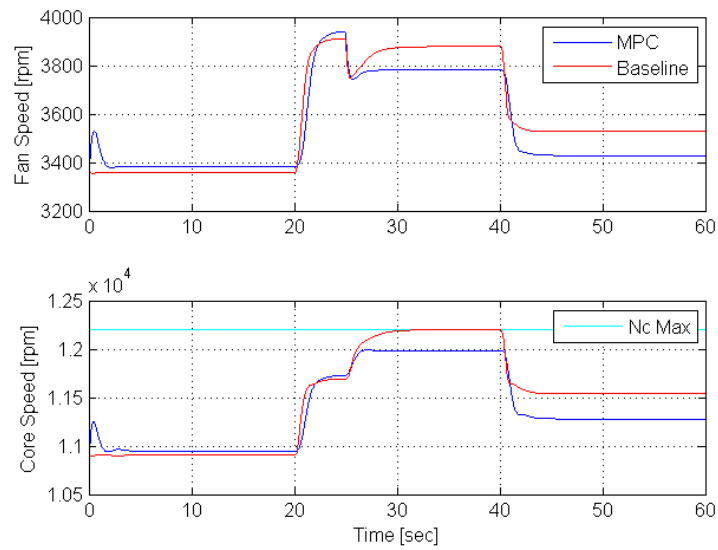


Figure 5.55: Unswitched MPC thrust control of the nonlinear engine for FE=-0.2.

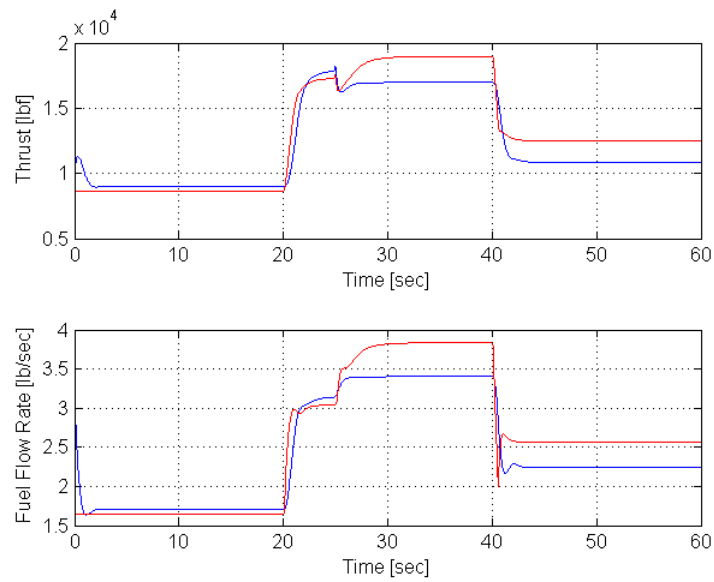


Figure 5.56: Unswitched MPC thrust control of the nonlinear engine for FE=-0.2.

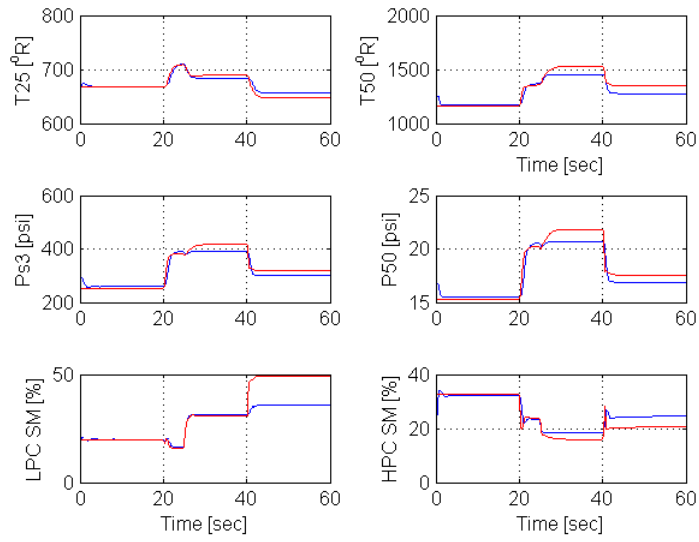


Figure 5.57: Unswitched MPC thrust control of the nonlinear engine for $FE=-0.2$.

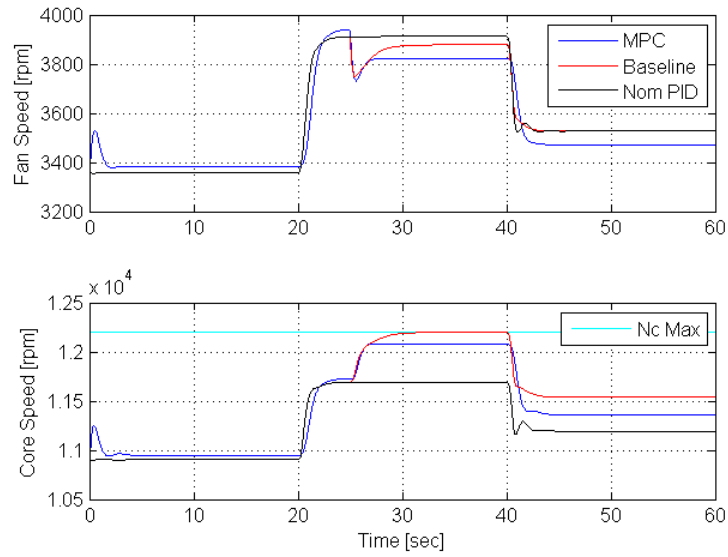


Figure 5.58: Switched MPC thrust control for $FE=-0.2$ at 25 sec.

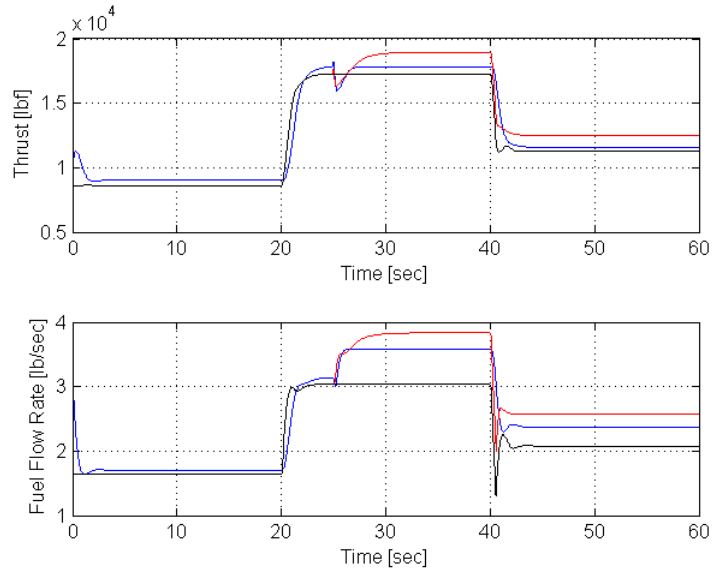


Figure 5.59: Switched MPC thrust control for FE=-0.2 at 25 sec.

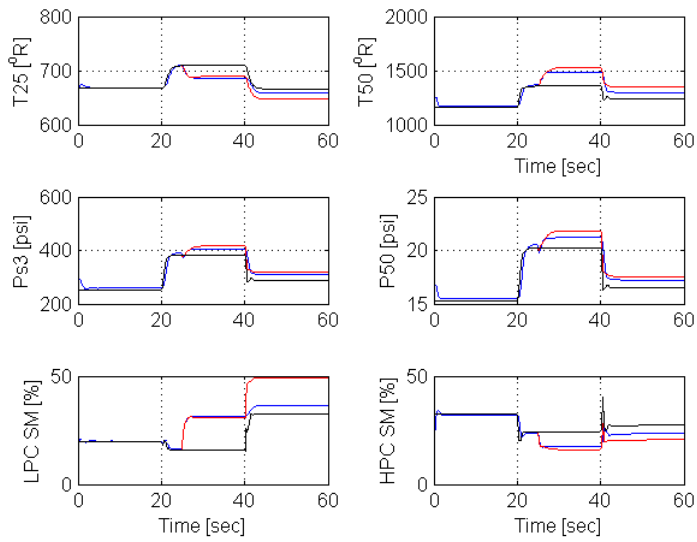


Figure 5.60: Switched MPC thrust control for FE=-0.2 at 25 sec.

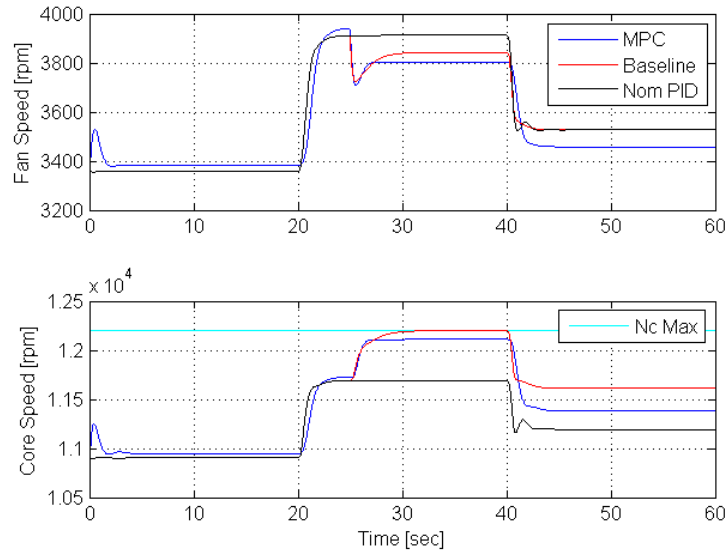


Figure 5.61: MPC68fe20 applied to a Nonlinear engine with FE=-0.22 at 25 sec.

engine due to the MPC is reduced by 3.4% when compared with the baseline PID in the presence of a fault in the fan.

As seen in Fig. 5.60, the temperature limit at the low pressure turbine exit is once again maintained by the MPC while the baseline PID exceeds it slightly. The stall margin of the low pressure compressor due to MPC is closer to that of the nominal baseline at the fall of PLA and hence achieving a higher compressor efficiency than the baseline at FE=-0.2. The time taken to complete a 60 sec thrust control simulation of a nonlinear engine is approximately 130 sec.

Next, the case to avoid modeling faults that are within $\pm 2\%$ of the modeled faulty reference model is made for the thrust control approach. In Fig. 5.61 - 5.63 for FlightA, a fault is induced at 25 sec such that there is a 22% degradation in the fan efficiency and the fault recovery is carried out by switching to the MPCfe20 block

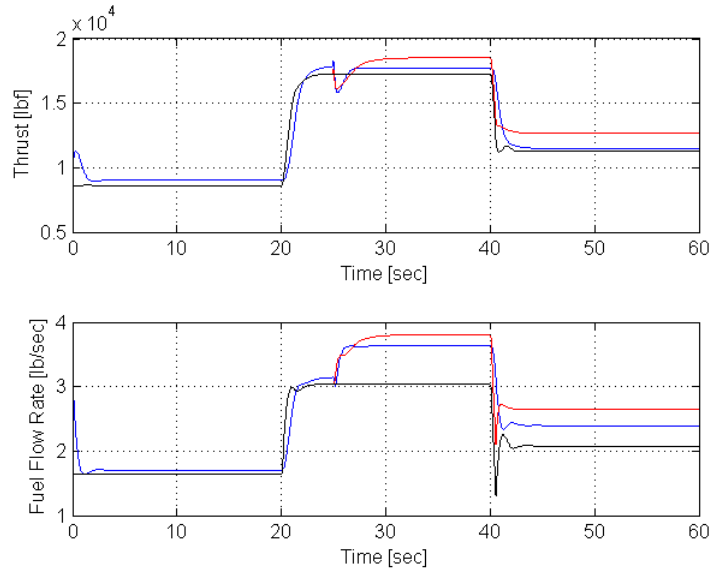


Figure 5.62: MPC68fe20 applied to a Nonlinear engine with $FE=-0.22$ at 25 sec.

at 25 sec. The weighting scalar for N_c used in Eq. (5.18) is used as is. The results discussed in the previous simulations hold true even here. The ability of the MPC to maintain the desired thrust upon switching to the 20% degradation reference model for a 22% is shown.

Similarly for a degradation of 18% in the fan efficiency, the application of MPC68fe20 is shown in Fig. 5.64 - 5.66. The weighting scalar for N_c used is identical to that in Eq. (5.18). Once again, the MPC based on a 20% degradation model can successfully handle a 18% degradation maintaining the desired thrust and thus avoiding modeling redundancy.

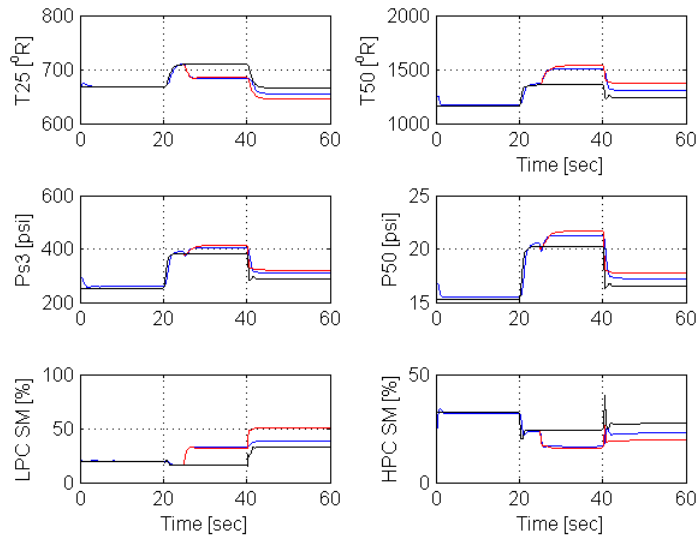


Figure 5.63: MPC68fe20 applied to a Nonlinear engine with FE=-0.22 at 25 sec.

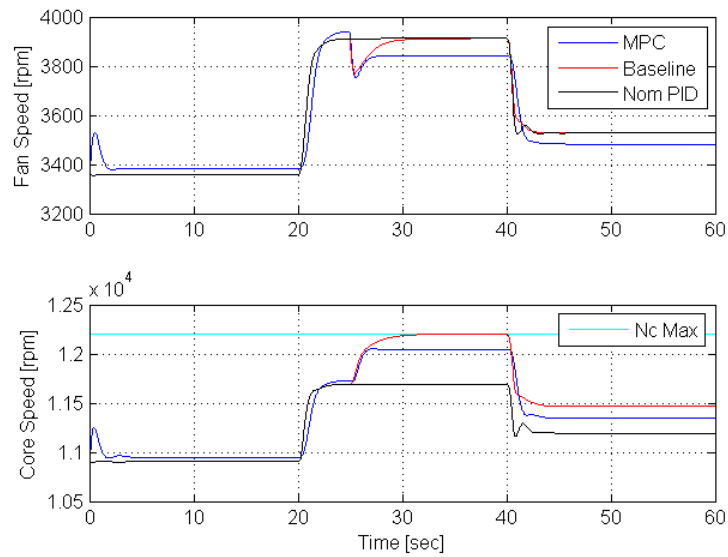


Figure 5.64: MPC68fe20 applied to a Nonlinear engine with FE=-0.18 at 25 sec.

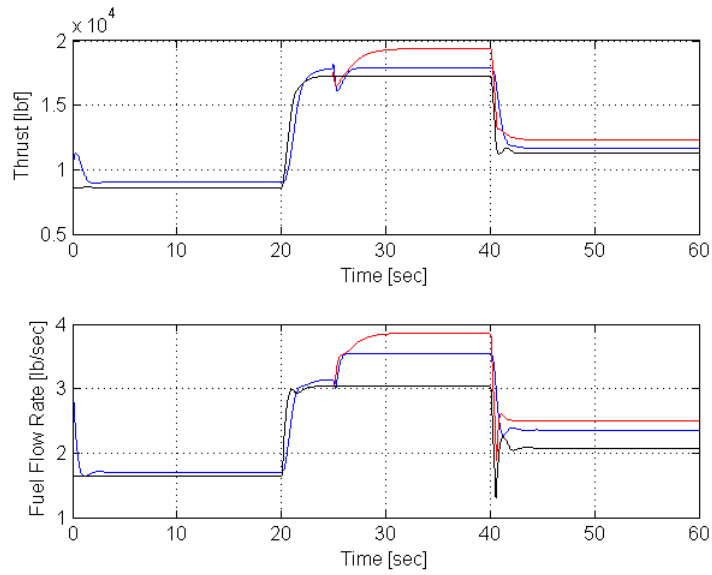


Figure 5.65: MPC68fe20 applied to a Nonlinear engine with FE=-0.18 at 25 sec.

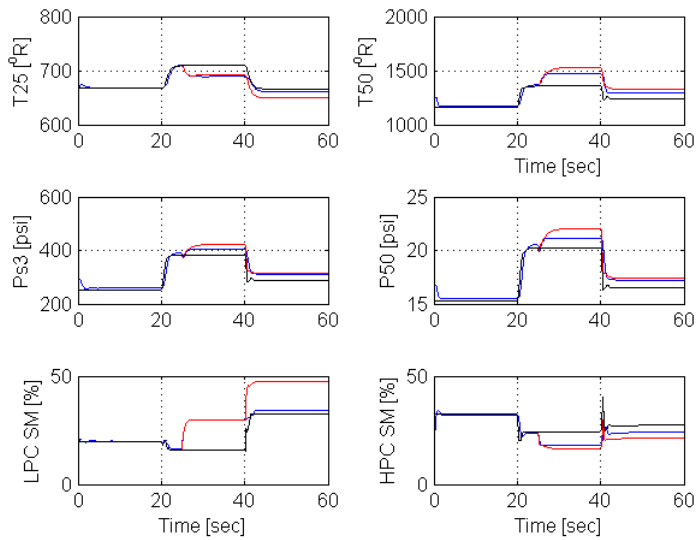


Figure 5.66: MPC68fe20 applied to a Nonlinear engine with FE=-0.18 at 25 sec.

Chapter 6: CONCLUSIONS AND FUTURE WORK

The replacement of the min-max architecture based PID baseline controller with a constrained on-line MPC for a throttling demand profile between PLAs 58° and 80° on a LTI and nonlinear turbofan engine model is shown without any compromise in performance when compared with the baseline controller thereby simplifying the engine control design scheme. The active fault tolerant method of using the multi-model approach where the reference model and the engine model are matched shows the best performance in the presence of a fault in the fan and the performance achieved is in accordance with that of the baseline controller of C-MAPSS40k.

The ability of MPC to handle faults that deviate within $\pm 2\%$ of the fault modeled reference model is demonstrated, assuming proper FDI, by adjusting the output weighting matrix accordingly thus avoiding redundant control design. The exact deviation of faults that can be handled by a reference model has to be investigated. Finally assuming proper FDI during the simulation a bumpless transfer in the nonlinear engine performance, while switching between multiple MPCs, is achieved while maintaining engine constraints when a fault in the fan is induced during the simulation. The output weighting matrix is re-tuned accordingly to achieve constrained performance from the nonlinear engine.

Although component faults are handled, the actual engine performance is still affected. Therefore in order to maintain the desired engine thrust, the affinity towards the traditional fan speed control is shed and model based thrust control using MPC is investigated. For the linear engine simulations only, since the LTI plant has an accurate relation between the thrust produced and the state vector, the thrust obtained is used in the feedback which is not the case in the nonlinear simulations.

The multi-model approach is shown to be redundant for thrust control in the linear engine simulations as MPC based on a nominal reference model is sufficient in handling component faults and while the rest of the engine parameters deviate from nominal values due to a fault in the fan, the goal of maintaining desired thrust is achieved. However for the thrust control simulations on the nonlinear engine, the multi-model approach is shown to be required. The total fuel consumed by the engine due to MPC is lesser by 3.4% than that of the baseline (that is used for N_f control) over the entire profile. Also a higher compressor efficiency during the fall in PLA is achieved by using MPC for thrust control. For the linear and nonlinear simulations, faults that deviate within $\pm 2\%$ of the fault modeled reference model can be handled affectively ensuring the engine is within its limits.

In light of the distributed control architecture taking effect in the near future, the computationally heavy MPC technique is a good choice to implement for next generation engine control. However the robustness of MPC and performance of FDI under the influence of time delays due to digital networks needs to be investigated. The high dependence of MPC on the linear reference model which might include modeling errors effects the overall MPC performance due to error propagation in the quadratic optimization. Therefore nonlinear MPC might be better however designing

the MPC based on a nonlinear reference model has its own complexities. Further, effective ways to reduce the computational time during the nonlinear simulations needs to be investigated. On-line changes in the prediction horizon might be an option and requires further mathematical investigation for the understanding the stability of MPC by doing so.

Chapter 7: APPENDIX I : MPC SOURCE CODE

```
1 %% MPC Fan Control for nonlinear engine
2 %% Programmed by Deepak C. Saluru
3
4 %% Simulation Parameters
5 tic
6 Ts=0.015; %sampling time of LTI cmapps engine
7 PredictionHorizon =50;
8 ControlHorizon = 1;
9
10 % load constraints from C-MAPSS40k
11 %Nfmax,Ncmax,Ps3max,Ps3min,T50max,dWfmin68,dWfmax68,Wfmin68,Wfmax68
12
13 %% switching parameters
14 % t168s=0;
15 % t168=19;
16 % t268s=25;
17 % t268=39;
18 % t368s=42;
19
20 t168=0;
21 t168fe20=125;%25;
22
23 %% 68 PLA LTI Nominal Engine Model
24 % Alt: 10000ft
25 % MN: 0.8
26 % PLA: 68
27 % Power T/O:0HP, Bleed: 0psi
28
29 % load linearization trim values lin_const68n
30
31
32 Ac68n=[ -3.0120    1.0332
33         0.1038   -2.5697];
34 Bc68n= 1.0e+003.*[    0.6419
35                   1.3448];
36 Cc68n=[    1.0000    0
```

```

37         0      1.0000
38     0.1383  -0.0433
39    -0.0934  -0.1092
40     0.0549   0.0845
41     0.0022   0.0030
42    10.3743   1.6029
43    -0.0721   0.0481
44     0.0167  -0.0012];
45 Dc68n=1.0e+003.*[
46         0
47         0
48     0.0006
49     0.2307
50     0.0271
51     0.0012
52     1.1357
53    -0.0006
54    -0.0107];
55
56 [Ad68n Bd68n Cd68n Dd68n]=c2dm(Ac68n,Bc68n,Cc68n,Dc68n,Ts,'zoh');
57 [n n]=size(Ac68n);
58 [n m]=size(Bc68n);
59 Ag68n=zeros(n+m,n+m);
60 Ag68n(1:n,1:n+m)=[Ad68n,Bd68n];
61 Ag68n(n+1:n+m,n+1:n+m)=eye(m);
62 Bg68n=Ag68n(:,n+m);
63 [nc mc]=size(Cc68n);
64 Cg68n=eye(nc+1,mc+1);
65 Cg68n(4:nc+1,:)= [Cd68n(3:nc,:) Dd68n(3:nc,:)];
66 Dg68n=zeros(nc+1,m);
67
68 Nfmax68 = Nfmax-lin-const68n(1);
69 Ncmax68 = Ncmax-lin-const68n(2);
70 Ps3max68=Ps3max-lin-const68n(6);
71 Ps3min68=Ps3min-lin-const68n(6);
72 T50max68=T50max-lin-const68n(5);
73 dWfmin68=-0.6;
74 dWfmax68=0.06;
75 Wfmin68=0.2;
76 Wfmax68=[];
77
78 Enginen=ss(Ag68n,Bg68n,Cg68n,Dg68n,Ts);
79 Model.Plant=Enginen; %<-----
80 Model.Plant.InputName = {'dWf'};
81 Model.Plant.OutputName = {'Nf','Nc','Wf','T25','T50','Ps3',...
82     'P50','Fn','LPC_SM','HPC_SM'};
83 Model.Plant.StateName = {'Nf','Nc','U(k-1)'};
84 LTIEngine.InputGroup.MV = 1;
85 Model.Plant.OutputGroup = {[1:2,6],'Measured';[3:5,7:10],'Unmeasured'};
86 clear MV OV
87 MV = struct('Min',dWfmin68,'Max',dWfmax68,'RateMin',[],'RateMax',[]);

```

```

88 for i=1:10
89     OV(i) = struct('Min', [], 'Max', []);
90 end
91 OV(1) = struct('Min', [], 'Max', Nfmax68);
92 OV(2) = struct('Min', [], 'Max', Ncmax68);
93 OV(3) = struct('Min', Wfmin68, 'Max', Wfmax68);
94 OV(5) = struct('Min', [], 'Max', T50max68);
95 OV(6) = struct('Min', Ps3min68, 'Max', Ps3max68);
96
97 Weights = struct('Input', 10, 'InputRate', 1, 'Output', [100 19 1 0 1 1 0 ...
    0 0 0]);
98 MPC68=mpc(Model, Ts, PredictionHorizon, ControlHorizon);
99 set(MPC68, 'Weights', Weights, 'MV', MV, 'OV', OV);
100 MPC68.Weights.ECR=10;
101
102 MPC68s=mpc(Model, Ts, 20, ControlHorizon);
103 set(MPC68s, 'Weights', Weights, 'MV', MV, 'OV', OV);
104 MPC68s.Weights.ECR=10;
105
106 %% 68 PLA LTI Degraded Engine Model
107 % Alt: 10000ft
108 % MN: 0.8
109 % PLA: 68
110 % Power T/O:0HP, Bleed: 0psi
111 % Health: Fan Eff=-0.2,
112
113 % load linearization trim values lin_const68n for FE=-0.2
114
115 Ac68fe20=[ -3.1892    0.4728
116   -1.1528   -1.3364];
117 Bc68fe20= 1.0e+003.*[    0.5891
118   1.1850 ];
119 Cc68fe20=[    1.0000    0
120     0    1.0000
121    0.1320   -0.0305
122   -0.2444    0.0066
123    0.1185    0.0227
124    0.0048    0.0009
125   12.6779    0.5765
126   -0.0593    0.0246
127    0.0176   -0.0014 ];
128 Dc68fe20=1.0e+003.*[    0
129     0
130    0.0011
131    0.2244
132    0.0257
133    0.0013
134    1.2506
135   -0.0009
136   -0.0085 ];
137

```

```

138 [Ad68fe20 Bd68fe20 Cd68fe20 ...
      Dd68fe20]=c2dm(Ac68fe20,Bc68fe20,Cc68fe20,Dc68fe20,Ts,'zoh');
139 [n n]=size(Ac68fe20);
140 [n m]=size(Bc68fe20);
141 Ag68fe20=zeros(n+m,n+m);
142 Ag68fe20(1:n,1:n+m)=[Ad68fe20,Bd68fe20];
143 Ag68fe20(n+1:n+m,n+1:n+m)=eye(m);
144 Bg68fe20=Ag68fe20(:,n+m);
145 [nc mc]=size(Cc68fe20);
146 Cg68fe20=eye(nc+1,mc+1);
147 Cg68fe20(4:nc+1,:)= [Cd68fe20(3:nc,:) Dd68fe20(3:nc,:)];
148 Dg68fe20=zeros(nc+1,m);
149
150 Nfmax68fe20 = Nfmax-lin_const68fe20(1);
151 Ncmax68fe20 = Ncmax-lin_const68fe20(2);
152 Ps3max68fe20=Ps3max-lin_const68fe20(6);
153 Ps3min68fe20=Ps3min-lin_const68fe20(6);
154 T50max68fe20=T50max-lin_const68fe20(5);
155 dWfmin68fe20=-0.6;
156 dWfmax68fe20=0.06;
157 Wfmin68fe20=0.2;
158 Wfmax68fe20=[];
159
160 Enginen=ss(Ag68fe20,Bg68fe20,Cg68fe20,Dg68fe20,Ts);
161 Model.Plant=Enginen; %<-----
162 Model.Plant.InputName = {'dWf'};
163 Model.Plant.OutputName = {'Nf','Nc','Wf','T25','T50','Ps3',...
164   'P50','Fn','LPC_SM','HPC_SM'};
165 Model.Plant.StateName = {'Nf','Nc','U(k-1)'};
166 LTIEngine.InputGroup.MV = 1;
167 Model.Plant.OutputGroup = {[1:2,6],'Measured';[3:5,7:10],'Unmeasured'};
168 clear MV OV
169 MV = ...
      struct('Min',dWfmin68fe20,'Max',dWfmax68fe20,'RateMin',[],'RateMax',[]);
170 for i=1:10
171   OV(i) = struct('Min',[],'Max',[]);
172 end
173 OV(1) = struct('Min',[],'Max',Nfmax68fe20);
174 OV(2) = struct('Min',[],'Max',Ncmax68fe20);
175 OV(3) = struct('Min',Wfmin68fe20,'Max',Wfmax68fe20);
176 OV(5) = struct('Min',[],'Max',T50max68fe20);
177 OV(6) = struct('Min',Ps3min68fe20,'Max',Ps3max68fe20);
178
179 Weights = struct('Input',10,'InputRate',1,'Output',[100 23 1 0 1 1 0 ...
      0 0 0]);
180 MPC68fe20=mpc(Model,Ts,PredictionHorizon,ControlHorizon);
181 set(MPC68fe20,'Weights',Weights,'MV',MV,'OV',OV);
182 MPC68fe20.Weights.ECR=10;
183
184 %%
185 toc

```

Bibliography

- [1] Hill, P., and Peterson, C., 1992. *Mechanics and Thermodynamics of Propulsion*, 2nd ed. Addison-Wesley.
- [2] Jaw, L., and Mattingly, J., 2009. *Aircraft Engine Controls*. AIAA Education Series.
- [3] Richter, H., and Litt, J., 2011. “A Novel Controller for Gas Turbine Engines with Aggressive Limit Management”. *47th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*.
- [4] Richter, H., 2012. *Advanced Control of Turbofan Engines*. Springer.
- [5] Hacker, J., 2011. “L1 Adaptive Control of Uncertain Nonlinear Systems with Dynamic Constraints: As Applied to Commercial Aircraft Engines”. PhD thesis, University of Connecticut.
- [6] Han, X., and Li, H., 2011. “Controller design for polynomial linear parameter varying model of turbofan engine”. *International Conference on Electronics, Communications and Control (ICECC)*, pp. 1961–1964.
- [7] Samar, R., and Postlethwaite, I., 2010. “Design and Implementation of a Digital Multimode H infinity Controller for the Spey Turbofan Engine”. *Journal of Dynamic Systems, Measurement, and Control*, **132**(1).
- [8] Hu, Y., Yang, Y. C., Zhang, S. Y., and Yu, X. X., 2011. “Study of Fuzzy Control for Turbofan Engine Based on Improved Genetic Algorithms Optimization”. *Applied Mechanics and Materials*, **148 - 149**, pp. 1072–1076.
- [9] Shuqing, L., Shengxiu, Z., and Yinan, L., 2011. “Neutral Network Based Optimal PID Controller over Whole Envelope for an Aero-engine”. *Journal of Projectiles, Rockets, Missiles and Guidance*.
- [10] Wang, H., Ouyang, L., Wang, D., and Liu, L., 2010. “Multivariable robust control design of a turbofan engine for full flight envelope operation”. *IEEE International Conference on Information and Automation (ICIA)*, pp. 2121 – 2125.

- [11] Wang, H., and Wang, D., 2011. “Turbofan aero-engine full flight envelope control using Hinf loop-shaping control technique”. *International Journal of Advanced Mechatronic Systems*, **3**(1), pp. 65–76.
- [12] DeCastro, J., 2007. “Rate-Based Model Predictive Control of Turbofan Engine Clearance”. *Journal of Propulsion and Power*, **23**(4), pp. 804–813.
- [13] Liu, Y., Dhingra, M., and Prasad, J. V. R., 2009. “Benefits of Active Compressor Stability Management on Turbofan Engine Operability”. *Journal of Engineering for Gas Turbines and Power*, **131**(4).
- [14] Culley, D., Pauszewski, P., and Smith, B., 2009. The Case for Distributed Engine Control in Turbo-Shaft Engine Systems. NASA TM-215654.
- [15] Culley, D., and Garg, S., 2009. Distributed Control Architecture for Gas Turbine Engine. More intelligent gas turbine engines, RTO Technical Report, TR-AVT-128.
- [16] Culley, D., 2010. Transition in Gas Turbine Control System Architecture: Modular, Distributed, and Embedded. NASA TM-216806.
- [17] Yedavalli, R., Belapurkar, R. K., and Behbahani, A., 2009. “Design of Distributed Engine Control Systems for Stability under Communication Packet Dropouts”. *Journal of Guidance Control and Dynamics*, **32**(5).
- [18] Belapurkar, R., Yedavalli, R., Pauszewski, P., and Greene, W., 2010. “Stability Analysis of ARINC 825-based Partially Distributed Aircraft Engine Control with Transmission Delays and Packet Dropouts”. *46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*.
- [19] Csank, J., May, R., Litt, J., and Guo, T., 2010. “Control Design for a Generic Commercial Aircraft Engine”. *46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*.
- [20] May, R., Csank, J., Litt, J., Lavelle, T., and Guo, T., 2010. “A High-Fidelity Simulation of a Generic Commercial Aircraft Engine and Controller”. *46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*.
- [21] May, R., Csank, J., Litt, J., and Guo, T., 2010. Commercial Modular Aero-Propulsion System Simulation 40k (C-MAPSS40k) Users Guide. NASA TM-216831.
- [22] Maciejowski, J., Goulart, P., and Kerrigan, P., 2007. “Constrained Control Using Model Predictive Control”. *Lecture Notes in Control and Information Sciences*, **346**, pp. 273–291.

- [23] Kestner, B., 2009. “Model Predictive Control (MPC) Algorithm For Tipjet Reaction Drive Systems”. PhD thesis.
- [24] Wang, L., 2009. *Model predictive control system design and implementation using MATLAB*. Springer.
- [25] Camacho, E., 2004. *Model Based Control*. Springer.
- [26] Bemporad, A., and Morari, M., 1999. “Robust model predictive control: A survey”. *Robustness in Identification and Control*, **245**, pp. 207–226.
- [27] Rossiter, J., 2003. *Model Based Predictive Control - A Practical Approach Control Series*. CRC Press.
- [28] Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Sokaert, P., 2000. “Constrained model predictive control: Stability and optimality”. *Automatica*, **36**(6), pp. 789–814.
- [29] Qin, S. J., and Badgwell, T. A., 2003. “A survey of industrial model predictive control technology”. *Control Engineering Practice*, **11**, pp. 733–764.
- [30] Darby, M. L., and Nikolaou, M., 2012. “MPC: Current Practice and Challenges”. *Control Engineering Practice*, **20**(4), pp. 328–347.
- [31] Wang, Y., and Boyd, S., 2010. “Fast Model Predictive Control Using Online Optimization”. *IEEE Transactions On Control Systems Technology*, **18**(2), pp. 267–278.
- [32] Alessio, A., and Bemporad, A., 2009. “A Survey on Explicit Model Predictive Control”. In *Nonlinear Model Predictive Control*, L. Magni, D. Raimondo, and F. Allgwer, eds., Vol. 384 of *Lecture Notes in Control and Information Sciences*. Springer Berlin / Heidelberg, pp. 345–369.
- [33] Gawthrop, P., and Wang, L., 2007. “Intermittent model predictive control”. *Proceedings of the Institution of Mechanical Engineers Part I Journal of Systems and Control Engineering*, **221**, pp. 1007–1018.
- [34] Gawthrop, P., and Wang, L., 2009. “Constrained intermittent model predictive control”. *International Journal of Control*, **82**(6), pp. 1138–1147.
- [35] Kouro, S., Corts, P., Vargas, R., Ammann, U., and Rodriguez, J., 2009. “Model Predictive Control - A Simple and Powerful Method to Control Power Converters”. *IEEE Transactions On Industrial Electronics*, **56**(6), pp. 1826–1838.
- [36] Xie, N., Wang, K., and Lv, Y., 2011. “Improved Model Predictive Control for Three-Phase Inverters”. *3rd International Conference on Communication Software and Networks (ICCSN)*, *IEEE*, pp. 412 – 416.

- [37] Zong, Y., Kullmann, D., Thavlov, A., Gehrke, O., and Bindner, H. W., 2011. “Active Load Management in an Intelligent Building Using Model Predictive Control Strategy”. *IEEE Trondheim PowerTech*, pp. 1–6.
- [38] Cairano, S. D., Bemporad, A., Kolmanovsky, I., and Hrovat, D., 2007. “Model predictive control of magnetically actuated mass spring dampers for automotive applications”. *International Journal of Control*, **80**(11), pp. 1701–1716.
- [39] Bolognani, S., Bolognani, S., Peretti, L., and Zigliotto, M., 2009. “Design and Implementation of Model Predictive Control for Electrical Motor Drives”. *IEEE Transactions On Industrial Electronics*, **56**, pp. 1925–1936.
- [40] de Santana, E. S., Bim, E., and do Amaral, W. C., 2008. “A Predictive Algorithm for Controlling Speed and Rotor Flux of Induction Motor”. *IEEE Transactions On Industrial Electronics*, **55**(12), pp. 4398–4407.
- [41] Wills, A. G., Bates, D., Fleming, A. J., Ninness, B., and Moheimani, S. O. R., 2008. “Model Predictive Control Applied to Constraint Handling in Active Noise and Vibration Control”. *IEEE Transactions On Control Systems Technology*, **16**(1).
- [42] Falcone, P., Borrelli, F., Tseng, H. E., Asgari, J., and Hrovat, D., 2008. “Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation”. *Int. J. Robust Nonlinear Control*, **18**, p. 862875.
- [43] Negenborn, R., Schutter, B. D., and Hellendoorn, J., 2008. “Multi-agent model predictive control for transportation networks: Serial versus parallel schemes”. *Engineering Applications of Artificial Intelligence*, **21**, pp. 353–366.
- [44] Scattolini, R., 2009. “Architectures for distributed and hierarchical Model Predictive Control A review”. *Journal of Process Control*, **19**, pp. 723–731.
- [45] Magni, L., Scattolini, R., and Tanelli, M., 2008. “Switched model predictive control for performance enhancement”. *International Journal of Control*, **81**(12), pp. 1859–1869.
- [46] Zanma, T., Yamakaji, A., and Ishida, M., 2009. “Model Predictive Control Based Switching of Multimodal Controllers in Switched Systems”. *Electrical Engineering in Japan*, **166**, pp. 420–427.
- [47] Mhaskar, P., El-Farra, N. H., and Christofides, P. D., 2005. “Predictive Control of Switched Nonlinear Systems With Scheduled Mode Transitions”. *IEEE Transactions On Automatic Control*, **50**(11).

- [48] Gavilan, F., Vazquez, R., and Camacho, E., 2012. “Chance-constrained model predictive control for spacecraft rendezvous with disturbance estimation”. *Control Engineering Practice*, **20**(2), pp. 111–122.
- [49] Lopes, R., Santana, P., Borges, G., and Ishihara, J., 2011. “Model Predictive Control applied to tracking and attitude stabilization of a VTOL quadrotor aircraft”. *21st International Congress of Mechanical Engineering*.
- [50] Slegers, N., Kyle, J., and Costello, M., 2006. “Nonlinear Model Predictive Control Technique for Unmanned Air Vehicles”. *Journal Of Guidance, Control, And Dynamics*, **29**(5), pp. 1179–1188.
- [51] Richter, H., and Singaraju, A., 2008. “Multiplexed Predictive Control of a Large Commercial Turbofan Engine”. *Journal of Guidance, Control and Dynamics*, **31**(2), pp. 273–281.
- [52] van Soest, W. R., Chu, Q. P., and Mulder, J. A., 2006. “Combined Feedback Linearization and Constrained Model Predictive Control for Entry Flight”. *Journal Of Guidance, Control, And Dynamics*, **29**(2), pp. 427–434.
- [53] Alexis, K., Nikolakopoulos, G., and Tzes, A., 2011. “Switching Model Predictive Attitude Control For A Quadrotor Helicopter Subject To Atmospheric Disturbances”. *Control Engineering Practice*, **19**, pp. 1195–1207.
- [54] Joosten, D., van den Boom, T., and Lombaerts, T., 2008. “Fault-tolerant control using dynamic inversion and model-predictive control applied to an aerospace benchmark”. *Proceedings of the 17th World Congress The International Federation of Automatic Control Seoul, Korea*, **17**.
- [55] Camacho, E., Alamo, T., and de la Pena, D. M., 2010. “Fault-tolerant model predictive control”. *IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1 – 8.
- [56] Prakash, J., Narasimhan, S., and Patwardhan, S., 2005. “Integrating Model Based Fault Diagnosis with Model Predictive Control”. *Ind. Eng. Chem. Res*, **44**, pp. 4344–4360.
- [57] Maciejowski, J., and Jones, C., 2003. “MPC Fault Tolerant Flight Control Case Study: Flight 1862”. *IFAC Safeprocess Proceedings*.
- [58] de Almeida, F. A., and Leiling, D., 2010. “Fault-Tolerant Model Predictive Control with Flight-Test Results”. *Journal Of Guidance, Control, And Dynamics*, **33**(2), pp. 363–375.

- [59] Zhang, Y., Jiang, J., and Theilliol, D., 2008. “Incorporating Performance Degradation in Fault Tolerant Control System Design with Multiple Actuator Failures”. *International Journal of Control, Automation, and Systems*, **6**.
- [60] Bemporad, A., Ricker, N., and Owen, J., 2004. “Model Predictive Control New tools for design and evaluation”. *Proceeding of the 2004 American Control Conference Boston, Massachusetts*.
- [61] Litt, J. S., and Aylward, E. M., 2003. Adaptive Detuning of a Multivariable Controller in Response to Turbofan Engine Degradation. NASA TM-212723.
- [62] Litt, J., Parker, K., and Chatterjee, S., 2003. Adaptive Gas Turbine Engine Control for Deterioration Compensation Due to Aging. NASA TM-212607.
- [63] Henriksson, M., TomasGronstedt, and Breitholtz, C., 2011. “Model-based on-board turbofan thrust estimation”. *Control Engineering Practice*, **19**(6), pp. 602–610.
- [64] Chatterjee, S., and Litt, J. S., 2003. Online Model Parameter Estimation of Jet Engine Degradation for Autonomous Propulsion Control. NASA TM-212608.
- [65] Henriksson, M., and Breitholtz, C., 2004. “Estimation of thrust and mass flow in a jet engine”. *IEEE International Conference on Control Applications, Taipei, Taiwan*, **1**(5), pp. 229–235.