NTNU – Trondheim
Norwegian University of
Science and Technology

Reconfigurable Autopilot Design using Nonlinear Model Predictive Control

Application to High Performance and
Autonomous Aircraft

Dzordzoenyenye Kwame Kufoalor

Master of Science in Engineering Cybernetics
Submission date: June 2012
Supervisor: Tor Arne Johansen, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Abstract

The work presented in this thesis examines several aspects of Nonlinear Model Predictive Control (NMPC) that display and confirm its promising potentials as a powerful reconfigurable control scheme. The effects of significant nonlinearities and the intrinsically unstable nature of high performance fighter aircraft, among other challenges, have been shown to be well handled in the NMPC framework. This work illustrates how complex control and stability augmentation measures (which are normally realized through ad hoc mode switching strategies) can be formulated and implemented as NMPC objectives and constraints. Further suggestions on robustness strategies for model/plant mismatch and compensation for coupling effects which are not properly accounted for, have been presented and examined in this work. Results on fault tolerance of NMPC are also presented and discussed in this thesis. In this direction, NMPC has been shown to have unique inherent fault detection capabilities due to its effective utilization of feedback and its internal model predictions. Different types of actuator/control surface failures, including extreme cases of total actuator failure are examined as test cases for the NMPC reconfigurable fault tolerant control scheme developed in this work. The NMPC autopilots are designed for an F-16 fighter aircraft, and the implementation and simulations were done using ACADO nonlinear ptimization solver, interfaced with the MATLAB/Simulink environment.

Preface

This thesis documents the work done in the final semester of the Master's degree programme in Engineering Cybernetics at the Norwegian University of Science and Tecnology (NTNU). The people who have contributed to the production of this thesis, in various of ways, are worth acknowledging.

I would like to thank my supervisor, Professor Tor Arne Johansen for all the valuable discussions, suggestions, corrections, constructive comments, and quick feedback on the work done in this thesis. Significant increase in the quality of results, and in the quality of this thesis in its totality, was achieved due to my supervisor's expert input on Nonlinear Model Predictive Control (NMPC) and Numerical Optimization in general.

Dr. Åge Skullestad at Kongsberg Defence Systems (KDS) was instrumental in the formulation and specification of the objectives of this thesis. He and his colleagues at KDS were also helpful through their comments and questions that stimulated further examination of some aspects of NMPC autopilot implementation, particularly at the initial and the finishing phases of this work.

I also want to thank Philip Maree and Cheerful Esi Kufoalor for proof reading this thesis and giving valuable feedback that further enhanced the comprehension of the contents of this thesis.

My studies, and student life in general, at NTNU have been a pleasant experience due to some awesome friends and colleagues. My special thanks goes to Thomas Ferstad Øvervåg, Svein Napsholm, and Jarle Eilerås, for sharing very interesting moments with me at Gløshaugen. I am also very grateful to Iselin Minde, Kristina Fuglesten, and Toini Sjånes Pedersen for always having a welcoming home for me, especially after very long days at school.

I must also thank my family, especially Cheerful Esi, Edinam Awusi, Nasia Enam, and my parents for their wonderful support and encouragement, throughout my Master's education years.

D. Kwame Kufoalor (Giorgio)
Trondheim, June 2012.

"Aviation is proof, that given the will, we have the capacity to achieve the impossible." — Eddie Rickenbacker

"All things are possible for one who believes." — Jesus Christ

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

A significant amount of research on fault-tolerant control systems is motivated by flight control system designs [5], where the goal is mostly to provide "self-repairing" capability in order to ensure a safe landing in the event of severe faults in the aircraft. A motivational example illustrating the need for reconfigurable control is the fault scenarios and pilot control attempts leading to the crash of flight *EL AL 1862*. The following description of the crash and deductions are based on [6] and [7].

On the 4th of October 1992, flight *EL AL 1862* (a Boeing 747-200F freighter) flying out of Amsterdam International Airport crashed into an eleven-floor appartment building after suffering multiple engine separations on the right wing. The damage to the aircraft resulting from the loss of two engines on the right wing was severe. The right-hand wing leading edge was severely damaged and the attached flaps partially lost. Roll control of the aircraft became limited as a result of the right-hand outboard aileron floating at zero hinge moment, partly lost spoiler system, and failure of outboard trailing edge flaps. The right-hand inboard aileron also became less effective due to disturbed airflow caused by the right-wing damage and loss of a pylon. For unknown reasons the aircraft's lower rudder was lagging behind causing limited directional control.

Even though modern aircrafts are designed to be controllable in the instance of multiple single-wing failures, the damage to the right wing caused by the dropping off of the engines led to significant loss of both lateral and directional control. In the worst case, wheel deflections up to 60° and full pedal deflections were needed to maintain level flight. The final loss of flight *EL AL 1862* was due to a hard-over right roll to 104° and an unrecoverable down pitch of 70°.

The crash of flight *EL AL 1862* serves as a good test case for studying reconfigurable control for several reasons. Firstly, despite the substantial damage and the consequent control limitations, the crew continued flying for almost 15 minutes, giving ample time for automatic failure identification and reconfiguration. Secondly, in spite of the sufficient available time, current emergency procedures could not handle the failure situation. Finally, there was sufficient damage to make the aircraft difficult to fly, thus making it a challenging and interesting problem for reconfiguration.

The increasing complexity of modern and future flight control systems inherently demand advanced and reliable control techniques that will ensure safety and survivability. Several structural design and automatic control methods have been developed over the years to enhance reliability and fault tolerance in safety-critical systems, especially in aircraft where a chain of events triggered by minor faults can be catastrofic. When faults occur, it is very important that system stability is maintained and an acceptable system performance attained. The primary objective of a reconfigurable flight control system will therefore be to redistribute and optimally coordinate control effort among the aircraft's available effective control surfaces and engines [1].

A common feature of most succesful fault tolerant control system designs is the ability to accommodate component failures automatically within a limited time frame. For this reason, most fault tolerant control systems rely on or integrate very effective fault detection and diagnosis (FDD) subsystems. The ability to rapidly detect faults and identify consequent model changes in real time is a key factor since there is usually limited time for a reconfigurable system to react in order to successfully deal with an emergency situation. Implementing an effective Fault Detection and Diagnosis module combined with reconfigurable control module(s) will result in an overall fault tolerant system.

Among the current research interests that are motivated by control surface redundancy and overactuation is Control Allocation (CA). From a practical point of view, Control Allocation forms a natural part of complex control systems, as in modern aircrafts, which have several actuators that are capable of realizing the same commanded control effect(forces and moments) [8]. However, in the past few decades it became apparent that predictive control methods display qualities that could be utilized in more complex, nonlinear applications [9]. Nonlinear Model Predictive Control (NMPC) is characterized by multivariable process modelling, optimization and handling of systems of relatively high complexity. Furthermore, the NMPC framework is capable of effectively accomodating nonlinear faults, ranging from saturations (due to actuator 'jams') to even severe damages that change the aerodynamics of the aircraft.

The description of a reconfigurable controller in a fault tolerant control setting is easily identified to match the inherent properties of NMPC. It can therefore be anticipated that with the increase in research and development on cutting edge strategies, NMPC will become a superior platform for modern and future fault tolerant control systems.

1.2 Problem Statement and Thesis Objectives

The primary objective of this thesis is to study and document Nonlinear Model Predictive Control (NMPC) and its application to autopilot design for an aircraft with highly nonlinear dynamics. Special attention is therefore paid to the stability characteristics of both the aircraft and the NMPC scheme implemented.

Background studies are to be made on the 1-DOF and 3-DOF mathematical models of the aircraft in order to establish a strong knowledge base, and at the same time, evaluate the need for NMPC. For this purpose, the lateral and longitudinal models of the aircraft are simulated in the MATLAB/Simulink environment.

Thereafter, the NMPC autopilots should be designed to satisfactorily control both the lateral and longitudinal motions of the aircraft. The flight envelope considered must be such that both stable and unstable modes of the aircraft are covered. Specific limitations on control surface/actuator deflection limits and servo bandwidth are also to be examined. It is also an objective to study the correlation between control surface limits, servo bandwidth and controllability of the autopilots. As much as possible, comparisons should be made between the performance of the NMPC autopilots and classical autopilots (for e.g. PID control [1]).

Finally, the driving objective of this work is to investigate the reconfigurable properties of NMPC and its potentials as a fault tolerant controller. The goal of the reconfigurable control implementations will be to maintain basic functionality when faults occur. The aircraft should therefore be able to perform turn and climb maneuvres and attain straight level flight at desired set points, also in the event of an actuator/control surface failure.

1.3 Thesis Contributions

This thesis reveals the potentials of NMPC as a stability augmentation controller for inherently unstable systems. Published results [9],[10], that partly study NMPC performance on the same F-16 fighter aircraft model used in this work rely on a stabilizing inner-loop controller, and therefore do not reflect the capabilities of NMPC in this area. This thesis provides suggestions on how classical stability augmentation requirements can be interpreted and implemented as NMPC objectives and constraints. Specifically, effective incorporation of angle of attack and attitude rates into the NMPC objective function, in combination with attitude limits as constraints are suggested and illustrated in chapters 7 and 8.

Robustness strategies for model/plant mismatch and compensation for coupling effects (due to decoupled dynamics approximations) are suggested. Refer to sections 8.4.3, 8.5.2 and chapter 9 for the proposed strategies and discussions on their effect on the flight test results.

This work reveals further superiority of NMPC's minimal reliance on external Fault Detection Systems and Reconfiguration mechanisms, compared to earlier indications in [11], which uses a linear stable aircraft model in the Model Predictive Control framework. The main results on this subject are presented and discussed in sections 8.5 and 9.2, respectively.

The significant advantage of the reduced NMPC tuning task, and increased flexibility, achieved through splitting of the flight dynamics into fast and slow dy-

[1] Proportional-Integral-Derivative control

namics, is also highlighted. The autopilots designed and implemented in this study are based on this splitting strategy. Further research into formulations that support the separation of system dynamics in the NMPC framework is therefore proposed. That is, instead of using separate controllers, one for fast dynamics and the other for slow dynamics, a formulation that incorporates the separation of fast dynamics from slow ones will be preferable (see chapter 7 and sections 8.4, 9.1.1, for complete design, implementation, and discussions on the work done in this area).

Last but not least, an attempt is made to introduce a performance/stability enhancement strategy that suggests a possible relation between the rate of change in state trajectories and NMPC stability in general (see section 3.3.4). This suggestion can be related to the first contribution stated above, since the use of the aircraft's attitude rate limits as constraints resulted in significant enhancement of performance (see section 8.3.3 and the autopilot design summary tables in section 8.4).

1.4 Thesis Outline

This thesis is organised as described below following the introduction chapter:

Chapter 2 gives a presentation of reconfigurable control techniques. It highlights the main components that are involved in the reconfigurable fault tolerant control framework, and concludes by introducing NMPC as a reconfigurable controller. The optimal control allocation formulation based on [12] is also included to serve as a platform for comparison.

In Chapter 3, a solid base for Nonlinear Model Predictive control (NMPC) is formed. Presentations in this chapter cover both theoretical formulations and practical implications of the NMPC scheme in general.

Chapters 4 and 5, can be seen as an aircraft primer, intended to serve as an introduction to modern high performance aircraft from an automatic control perspective. Topics treated include developing the aircraft dynamical model, decoupling into longitudinal and lateral channels, approximations for automatic control, and a brief discussion on stability and control characteristics of an aircraft.

Chapter 6 and 7 follow with formulations of the autopilots based on both classical control and the NMPC framework, respectively. Together, these two chapters form a solid base for the flight control strategies implemented in this work.

The main results of this work are presented and briefly discussed in chapter 8. Further discussions are covered in chapter 9. A conclusion is given in chapter 10 as a final note for this thesis.

The contents of the Appendix are included in order to make this thesis self contained and to facilitate direct reference to information (or facts) that further elaborate on some aspects of this thesis.

Chapter 2

Reconfigurable, Optimal, and Predictive Control

2.1 Reconfigurable Control Techniques

A *fault tolerant control system* (FTCS) is a closed-loop control system which can tolerate component malfunctions, while maintaining desirable performance and stability properties [1]. According to reference [1] fault tolerant control systems can be classified into two types: *passive* and *active*. In the passive type, controllers are fixed and are designed to be robust against a class of presumed faults, and hence has limited fault-tolerant capabilities. Active fault tolerant control systems on the contrary react to system component failures actively by reconfiguring control actions so that stability and acceptance performance of the whole system can be maintained. Active control in this contest also implies that the controller compensates for the fault's impact by either selecting a pre-computed control law or by synthesizing a new one on-line.

Active fault tolerant control systems can be further divided into four subsystems:

1. A reconfigurable controller
2. A fault detection and identification scheme
3. A controller reconfiguration mechanism
4. A command/reference governor

The critical issue in any active fault tolerant controller is the limited amount of time available for the FDI system and for the control system reconfiguration. Besides having an easily configurable controller and an effective FDI system, an overall reconfiguration mechanism which leads as much as possible to the restoration of system stability and acceptable performance is very crucial. Figure 2.1 shows an overall structure of a typical active fault tolerant control system. It should be noted that, in order to ensure that the closed-loop system tracks a command input trajectory in the event of faults, a reconfigurable feedforward controller is often needed. In addition, a command/reference governor (possibly a reconfigurable guidance system) may also be needed to be able to adjust command inputs or reference trajectories automatically. The main aim of a reference governor will be to take into consideration the degraded system performance after a fault and avoid potential actuator saturation. This work presented in this thesis

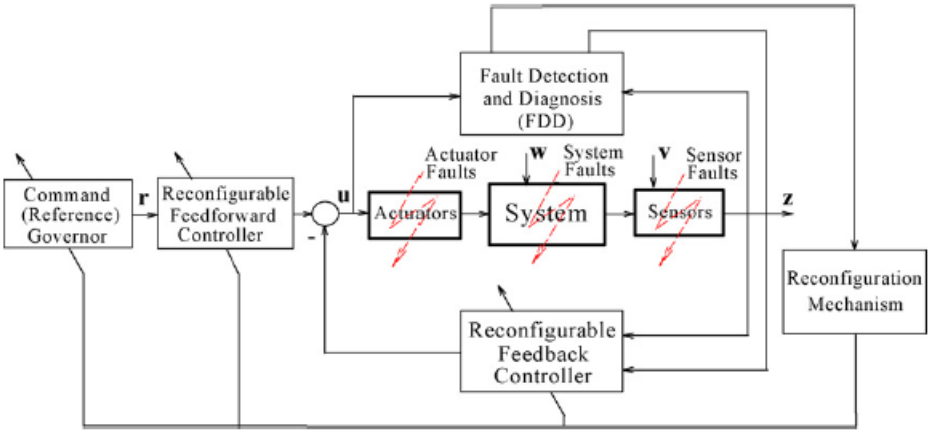


Figure 2.1: A general structure of an *active* fault tolerant control system [1]

focuses mainly on reconfigurable controller design.

There are numerous approaches to reconfigurable control design. A detailed classification of the design approaches can be found in [1], where the methods are grouped based on either the mathematical tool used (e.g Linear Quadratic, Pseudo-inverse, Feedback linearization, Model Predictive Control, Adaptive control, linear Parameter Varying, H_∞, and other robust control techniques), the design approach (pre-computed control law, or on-line automatic redesign), the reconfiguration mechanism (e.g optimization, switching, matching, compensation), or the type of system to be dealt with (linear or nonlinear system). Several of these methods may be combined in practice to achieve the best overall fault tolerant control system. Nevertheless, the performance of the overall system will depend on many factors, such as the availability of the remaining (functional) actuators, redundancy in the system, and the type of control strategies adopted in the reconfigurable controller design. These factors, among other challenges, have led to several research fields among which dynamic redundancy management, control action re-allocation and re-destribution stands out. Redundancy can be seen as a key ingredient in any fault tolerant system since almost all modern military and civil aircrafts have triplex- or quadruplex-redundant actuation systems, flight control computers, databus systems, air data and motion sensor systems [1].

2.2 NMPC as a Reconfigurable Controller

The reconfigurable fault-tolerant control systems review in [1] highlights important limitations on conventional approaches to solving reconfigurable control problems for constrained multivariable systems and systems with significant nonlinearities. In general, how to design fault-tolerant control systems which can work effectively in the entire range of nonlinear systems, and how to distinguish the changes induced by faults from that by operating condition variations still remain to be investigated. In order to handle nonlinear systems beyond using linearized models, control schemes such as feedback linearization, nonlinear dynamic inversion, backstepping, neural networks, adaptive control, and Lyapunov

methods have been developed. However, *effective* design methods for dealing with nonlinear fault tolerant control issues are not yet available [1]. Furthermore, a design strategy to deal with nonlinearities introduced by constraints of input and state/output variables is another challenging issue.

Model predictive control, in general, has been identified as a method that offers good possibilities for reconfiguration and fault-tolerant control [11], [1], [6], [7], [13]. This claim is simply due to the fact that model predictive control is capable of handling most of the challenges for reconfigurable control in a generic and systematic manner. In addition to the challenges mentioned in the previous paragraph, the following are specific and significant difficulties faced when dealing with reconfigurable flight control [13]:

- It is a multivariable problem, with strong cross couplings between modes usually appearing after failures. An aircraft loses its symmetry after surface damage, and conventional simplified control approaches may not be applicable.
- It is a nonlinear problem, which means that the trim values and the linearized models change after failures, requiring the continuous use of nonlinear or adaptive control algorithm.
- The system may be highly unstable, leaving very little time for reconfiguration.
- Actuator authority is limited, and control saturation problems are aggravated when an aircraft experiences control surface damages.

The most significant challenge in designing control laws for a damaged aircraft is, perhaps, the capability of accounting for nonlinearities [13]. It is therefore highly important to conduct active studies and research on the numerous possibilities and inherent properties offered by Nonlinear Model Predictive Control (NMPC) strategies. Positive results in this direction will definitely contribute to finding effective solutions to the very challenging problems of reconfigurable control mentioned above.

The NMPC scheme and its important aspects are discussed in chapter 3. Basically, a defining feature of model predictive control is the repeated constrained optimization of a performance objective over a prediction horizon, and the general structure consists an *optimizer*, a *predictor*, and an *internal model*. NMPC relies on its internal model for predictions and will therefore require a fault model of the system, if the severity of the damage is such that the dynamics of the system change. Changes in the internal model can be done in either an adaptive fashion, or using a multi-model switching scheme, or by relying on an FDI scheme which provides a fault model. For example, changed stability derivatives and control aerodynamic coefficients of a damaged aircraft can be identified online and used by the NMPC controller as internal model for prediction.

Furthermore, if the failure affects the capabilities or performance of the controlled system (i.e. the aircraft), it is possible to change the objectives, or the constraints, or both, accordingly. It may however be a challenging problem to know how to change the problem formulation if a failure occurs [11]. In some

cases, the parameters used in the NMPC algorthm, such as the state and input weightings, or the prediction and control horzon, may be changed.

In addition, actuator faults can be handled naturally in the NMPC framework via changes in the input constraints. The result in this case will be similar to the optimal control allocation approach, where other input channels are used to produce the same effect (see section D.1). It is also interesting to note that, for an over-actuated system, a model predictive controller can be designed so that it exhibits some fault-tolerant properties even in the absence of any knowledge of the failure [11].

Chapter 3

Nonlinear Model Predictive Control (NMPC)

Model Predictive Control (MPC) is an advanced control methodology that uses multivariable process models to predict future system- and control behavior. The formulation of the MPC problem includes constraints on inputs (*manipulated variables*) and states (*controlled variables*). The formulated problem is then solved by using mathematical programming to optimize predicted future performance. Optimization in predictive control also means allowing operations closer to constraints, which in the process industry implies an increase in profitability, while in a general sense enhances performance and fault tolerance.

There are several specific reasons why handling of constraints makes MPC comparably superior to other (especially conventional) control strategies. Traditionally, control system performance requirements are not formulated in a way which reflects the presence of constraints. MPC on the other hand allows complex control objectives to be stated by explicitly incorporating constraints as part of the problem formulation. Another important factor identified [14] is the avoidance of *integrator wind-up* problems, which occur with conventional controllers if long-duration set-point errors cause integrator outputs to exceed the saturation limits. This situation is known to cause overshoots and even instability [15]. The fact that the predictive controller is aware of constraints, particularly actuator sturation constraints, and never generates input signals to voilate them, removes the wind-up problem. In other words, with predictive control, the wind-up problem does not arise.

As mentioned earlier, the predictive controller has an *internal model* (process model) which is used to predict the behavior of the plant (or process). Most widespread MPC is the type termed *linear MPC* - using linear process models for prediction. However, MPC using *nonlinear* process models, termed nonlinear MPC (NMPC), has gained usage over the past 10-15 years [16]. Typical reasons for using NMPC are that the process operates in several steady states with significantly different dynamics, or there are large disturbances that excite nonlinearities. Handling of important nonlinearities, among others, has increased research focus on efficient implementation of nonlinear optimization techniques tailored for NMPC. The choice of NMPC over well known systematic nonlinear control methods such as feedback linearization and constructive Lyapunov-based

methods, is again justified by the fact that they are not developed in order to handle constraints in a systematic manner. In addition, such nonlinear methods depend on complicated design procedures that do not scale well to large systems [17].

The significant factors that have limited the industrial impact of NMPC include challenges of guarateeing a global solution to the resulting nonlinear optimization problem within real-time requirements [17],[18]. The most obvious reason for this limitation is that using nonlinear models may result in a non-convex nonlinear optimization problem, which may in addition have multiple local minima. Solving such nonlinear programming problems is usually time consuming and error prone. The positive developments on efficient NMPC implementation techniques in recent years make further exploration of the powerful and promising potentials of NMPC an important research area.

The work presented in this thesis deviates from the traditional and most common way of NMPC implementation. Predictive control in general is usually implemented on top of traditional local (typically PID) controllers which take care of fast system dynamics and stabilization (if necessary) [14],[9]. This work, however, attempts to replace the low-level controllers by directly controlling the actuators of a high performance, highly nonlinear aircraft with several unstable modes. The NMPC scheme explores both fast inner-loop and relatively slow outer-loop autopilot functionalities, and therefore demonstrates the powerful potentials of predictive control.

This chapter looks at how the NMPC problem is formulated, various techniques used in solving the NMPC problem, and important control issues such as stability, dynamic performance and practical implecations.

3.1 Formulation of the NMPC Problem

A very general and simple formulation of the nonlinear model predictive problem has the control objective of minimizing a *cost function* $J(x,u,t)$, which takes the form:

$$J(x,u,t) = \int_0^T \ell(x(t),u(t),t)\,\mathrm{d}t \tag{3.1}$$

where $\ell(x(t),u(t),t)$ is a non-negetive function, termed the *stage* cost, and $T>0$ is the *horizon*. The stage cost $\ell(.)$ can be chosen as an l_1 or l_2 type cost function [17]

$$\ell(x(t),u(t),t) = \|x(t)-r_x(t)\|_Q^2 + \|u(t)-r_u(t)\|_R^2 \tag{3.2}$$

$$\ell(x(t),u(t),t) = \|Q(x(t)-r_x(t))\|_1 + (\|R(u(t)-r_u(t))\|_1 \tag{3.3}$$

where the properties of the weight matrices $Q \geq 0$ and $R \geq 0$ are essential for performance, and in some cases also stabilty (as presented in section 3.3). The plant to be controlled has a state vector x and an input vector u, and has a nonlinear behavior governed by the vector differential equation

$$\frac{dx}{dt} = f(x,u) \tag{3.4}$$

In addition to the equality constraints introduced by the nonlinear plant model (3.4), general inequality constraints jointly on states and inputs can be defined as

$$g(x(t), u(t), t) \leq 0 \tag{3.5}$$

and finally, the control input vector is constrained to some set $u(t) \in U$, where

$$U = \{u_{min} \leq u(t) \leq u_{max}\} \tag{3.6}$$

3.1.1 The Receding Horizon Principle

The NMPC principle is to repeatedly solve the formulated finite-horizon optimal control problem at each sampling instant, to obtain an optimal sequence of inputs (or control trjectory). Only the first of the optimal input sequence is applied as input to the plant until the next sample instant.The NMPC principle in its basic form is illustrated in figure 3.1. It is interesting to note that time-varying reference trajectories, known disturbances and other external input signals can be accounted for in the NMPC formulation since both the control objective and contraints are time-dependent [17]. The NMPC principle implies discretization of

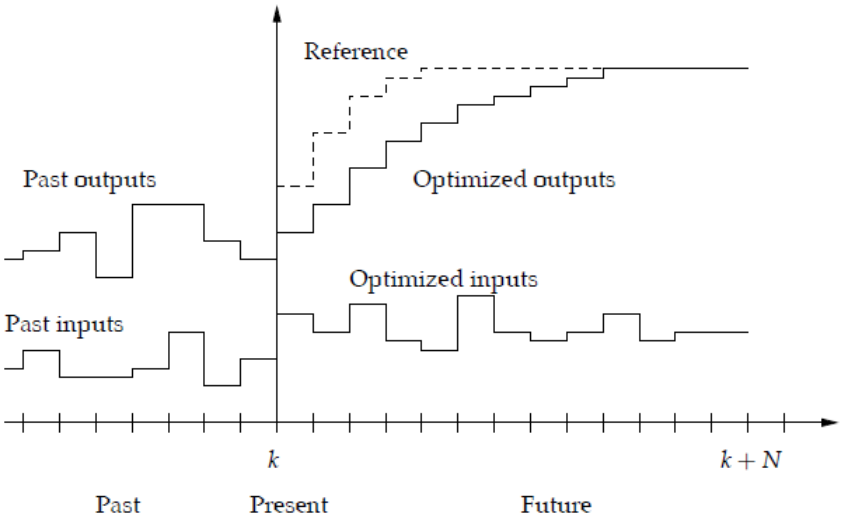


Figure 3.1: The MPC principle

the plant model represented by f in (3.4) and a finite number of decision variables that together define a finite-dimentional optimization problem. In a general and simple form, the resulting discrete time NMPC optimization problem optimizes a truncated objective function (3.7), subject to nonlinear constraints on states and inputs. The horizon T becomes finite, and an approximate solution of (3.4) can be found at N discrete time instants $T_d = t_1, t_2, ..., t_N \subset [0, T]$. The new objective is to minimize the cost function

$$J(x(t_k), u(t_k), t_k) = \sum_{k=0}^{N-1} \ell(x(t_{k+1}), u(t_k), t_k) \tag{3.7}$$

subject to

$$x(t_{k+1}) = f(x(t_k), u(t_k), t_k), \qquad k = 0, ..., N-1 \tag{3.8}$$

$$x(t_0) = x(0), \qquad \text{given} \tag{3.9}$$

$$g(x(t_k), u(t_k), t_k) \leq 0, \qquad t_k \in T_d \tag{3.10}$$

$$u_{min} \leq u(t_k) \leq u_{max}, \qquad t_k \in T_d \tag{3.11}$$

The predicted behavior of the plant in an NMPC setting depends on an assumed input trajectory $\hat{u}(k+i|k), (i = 0, 1, ..., N-1)$ that is to be applied over the prediction horizon. A simple case will be to choose the input trajectory such as to bring the plant output at the end of the horizon to the required *reference* value $r(k+N)$. In this case, several input trajectories can fulfill the requirement. However, the idea here is to select the (optimal) input which promises the best predicted behavior. This can for example imply obtaining the input trajectory which requires the smallest input energy. Once the optimal future input trajectory has been found, only the *first element* of that trajectory is applied as the input signal to the plant. Then the whole cycle of prediction and trajectory determination is repeated, one sampling interval later. Since the prediction horizon remains of the same length as before, but slides along by one sampling interval at each step, the control strategy is termed *receding horizon* [14].

3.1.2 The Reference Trajectory

The *reference* trajectory shown in figure 3.1, defines an important aspect of the closed-loop behavior of the controlled plant. It starts at the current output $y(k)$, and defines an ideal trajectory along which the plant should return to the desired set point. It is usually assumed that the reference trajectory approaches the set-point exponentially from the current value, with the 'time-constant' of the exponential, T_{ref}, defining the speed of response as illustrated in the following example [14]:

Suppose the current error, or the deviation of the current output $y(k)$ from the value of the set-point trajectory $s(k)$ is

$$\epsilon(k) = s(k) - y(k) \tag{3.12}$$

then the reference trajectory is chosen such that the error i steps later, if the output followed it exactly, would be

$$\epsilon(k+i) = e^{-iT_s/T_{ref}}\epsilon(k) \tag{3.13}$$

where T_s is the sampling interval. The reference trajectory will then be defined to be

$$r(k+i|k) = s(k+i) - \epsilon(k+i) \tag{3.14}$$

$$= s(k+i) - e^{-iT_s/T_{ref}}\epsilon(k) \tag{3.15}$$

The reference trajectory can however be alternatively defined by other convenient functions, and can as well be as simple as a straight line from the current output and meeting the set-point trajectory after a specified time.

3.1.3 Control Input Parameterization

The presented NMPC formulation uses future $u(t_k)$ as optimization variables, although it can in some cases be a good idea to use "perturbations" $c(t_k)$ to some (possibly nonlinear) feedback control law $\alpha(x(t_k))$ as optimization variables. In other words, future control moves can be parameterized as $u(t_k) = \alpha(t_k) + c(t_k)$ [16]. Cases when this might be smart are when the process model has unstable modes being stabilized by the feedback $\alpha(x(t_k))$. A more general and practical approach for parameterizing the input signal is to assume that $u(t_k)$ is piecewise constant with regular sampling interval t_s such that T is an integer multiple of t_s, and parameterized by a vector $U \in \mathbb{R}^p$ such that $u(t) = \mu(t, U) \in \mathbb{R}^r$ is piecewise continuous [17]. Moreover, a general choice is a piecewise constant control input $\mu(t, U) = U_k$ for $t_k \leq t \leq t_{k+1}$.

The reformulations made to the original optimal control problem, (3.1) to (3.6), are generally required in order for numerical integration to be applicable. Various techniques and further reformulations introduced for numeric solution by nonlinear optimization solvers are presented in the following section.

3.2 Solving the NMPC Problem

The NMPC formulation presented in section 3.1 usually involves more equations to be satisfied than the number of available variables. In cases of this nature, it is generally impossible to find an exact solution, and some kind of approximate solution will be necessary. Most of the existing approaches to solving the NMPC problem are based on Sequential Quadratic Programming (SQP) methods, which make quadratic approximations to the objective function and linear approximations to the constraints, and iteratively solve a Quadratic Program (QP) to find the search direction leading to the optimal solution [16],[19].

3.2.1 Sequential Quadratic Programming

The basic idea of SQP is presented in [11] by supposing that a general constrained optimization problem is to be solved, of the form:

$$\min_{\theta}\{V(\theta) : H_i(\theta) = 0, \Omega_j(\theta) \leq 0\} \tag{3.16}$$

where $\{H_i(.)\}$ and $\{\Omega_j(.)\}$ are sets of nonlinear functions, and that an iterate θ_k has been obtained. The SQP makes a quadratic approximation to $V(\theta_k)$:

$$q_k(d) = \frac{1}{2} d^T \nabla^2_{\theta\theta} L(\theta_k, \lambda_k) d + \nabla V(\theta_k)^T d \tag{3.17}$$

where $L(\theta_k, \lambda_k) = V(\theta) + \sum_i \lambda_i H_i(\theta) + \sum_j \lambda_j \Omega_j(\theta)$ is the Lagrangian. The next iterate, θ_{k+1}, is given by

$$\theta_{k+1} = \theta_k + \alpha_k d_k, \quad \alpha_{\in}[0,\ 1] \tag{3.18}$$

where the search direction d_k is found by solving the quadratic programming problem which results from minimizing $q_k(d)$, subject to local linear constraints:

$$\min_{d}\{q_k(d) : H_i(\theta_k) + \nabla H_i(\theta_k)^T d = 0, \Omega_j(\theta_k) + \nabla \Omega_j(\theta_k)^T d \leq 0\} \tag{3.19}$$

Several variations of this basic SQP idea are now employed in numerical solvers to achieve efficient solution to the NMPC problem. Different approaches are used in for example estimating the Lagrange multipliers, λ_k, or the Hessian matrix $\nabla^2_{\theta\theta} L(\theta_k, \lambda_k)$.

In order to use numerical solvers to find a solution to the nonlinear optimization problem, two main reformulation approaches are introduced in the following section.

3.2.2 Numerical Optimal Control Formulation

The Sequential approach (also called *single shooting* or *reduced space*) separates the QP part of the SQP optimization into two. The model is first simulated using the current value of the control trajectory parameters U_k as optimization variables, thus removing the intermediate states $(x(t_1), x(t_2), ..., x(t_N))$ as optimization variables, by substitution into the cost and constraint functions. The resulting reduced size QP problem is then solved iteratively. An obvious advantage is achieved through the rather few optimization variables used, which easily lead to feasible solutions (w.r.t. the nonlinear plant model) at each iteration. However, the optimization algorithm in a sequential setting has very limited control over the simulation part, making systems with unstable modes a challenge. This is mainly because the states tend to diverge for many choices of input variables. Little control over the simulation part also means that the sequential approach may use more or less arbitrary and seperate optimization solvers, which may in some cases be simple and convenient when compared to the simultaneous approach presented next.

The Simultaneous approach (*full space*) uses both the control trajectory parameters and states as optimization variables. The solution to the differential equations and optimization is obtained simultaneously, and for this purpose differential equations are discretized in time and enter the optimization problem as additional nonlinear equality constraints. The solution in this case will not generally be feasible (considering the presence of nonlinear equality constraints) before the algorithm converges. Obviously, a much larger number of optimization variables is involved, leading to a comparably bigger problem size. However, the cost and constraint function evaluation is much simpler, and sparsity of the resulting nonlinear programming problem can be exploited to achieve an efficient solution. In addition, nonlinear optimization problems are generally non-convex, and the convergence and success of the optimization algorithm depends largely on the initial guess provided for the solution. The availability of a good initial guess for the state trajectory is an advantage that can be exploited by the simultaneous approach. A continuous update of the initial states can also serve as a means of detecting divergence of the predicted state trajectories. Typical simultaneous approaches use *collocation* methods to parameterize/discretize the differential equations, or the well known *Direct Multiple Shooting* which is of particular interest to the work presented in this thesis.

Multiple shooting can be viewed as a combination of the sequential appraoch and the simultaneous approach in the sense that it divides the horizon into a few 'sub-horizons' where the model is simulated (as in the sequential approach)

on each sub-horizon. It then uses the nonlinear equality constraints (similar to the simultaneous approach) such that the state at the end of each sub-horizon matches the first state at the next sub-horizon. As a consequence of this approach, the number of optimization variables are reduced (considering each sub-horizon) while at the same time allowing more control over the simulation. Moreover, the resulting nonlinear program takes a special sparse structure which can be utilized for an efficient solution.

Understanding the underlying factors and basic differences between these optimization approaches is vital for making a good choice of options available in numerical solvers and optimization tools specially tailored for NMPC problems. It is however important to note that the methods presented above result in the same optimal control trajectory (neglecting errors due to discretization and numerical approximations) [17]. Numerical properties and computational complexity can therefore be identified as the main difference between the mentioned approaches. Other important aspects including feasibility, choice of numerical methods, computational cost, and factors that contribute to an efficient solution are treated in [17], [20], [21], [19] and their references.

3.3 Stability and Dynamic Performance

Feasibility of the on-line computation, stability, and performance issues are well covered and treated extensively for linear systems (see for example [14] and references therein). Stability (even nominal), in general, is not guaranteed for NMPC schemes which mainly replace the linear process model with a nonlinear model [16]. However, several approaches have been proposed introducing necessary modifications to the basic NMPC formulation to ensure stability (see references [20], [22], [23], [24]). For stability, it is necessary (not sufficient) that unstable modes are detectable through the cost function, as for linear MPC. However, since it is generally not easy to check detectability (or observability) for nonlinear systems, one approach that implies detectability is to choose a cost function with magnitude greater than the sum of the squared norms of both states and controls, that is:

$$\ell(x, u, t) \geq \varepsilon(\|x\|^2 + \|u\|^2), \quad \varepsilon > 0 \tag{3.20}$$

In this way the states and control actions become directly observable through the cost function and it follows intuitively that minimization of the cost function will influence the states that are controllable.

It is a common practice to augment unstable systems with stabilizing inner-loops before applying an MPC scheme. Some practical reasons for employing a stabilizing inner loop in an MPC frame work are presented in [9], where possible sampling rate reduction advantage is worth mentioning. The sampling rate of the NMPC scheme can be reduced since the inner-loop will be handling the high bandwidth disturbances and tracking requirements with its smaller time implementation. It is, however, important to explore the numerous potentials of NMPC, including its inherent reconfigurable properties, to examine both stability and dynamic performance capabilities, when applied to systems with highly nonlinear and unstable modes.

3.3.1 The NMPC Stability Problem

The main aim of most stabilizing NMPC approaches is to show that the finite horizon NMPC strategies used in practice can guarantee stability of the resulting closed-loop system. It is at this stage clear that the NMPC principle leads to closed-loop control since each new optimized control trajectory is based on the most recent state information. The numerical optimal control problem solved at each sampling instant, on the other hand, provides essentially an open-loop trajectory. The key problem with a finite horizon results from the difference between the predicted open-loop and the resulting closed-loop behavior [20], [17]. The problem is even worse with unstable plants models since their behavior can rapidly diverge from the real plant behavior, and numerical errors in the simulation can be rapidly amplified by the instability [14]. The result is a complete loss of the model's accuracy as a predictor of plant behavior. In this case, the easiest way of stabilizing the model is to introduce measured states (actual past plant outputs) and past inputs as initial values (whenever available) to solve the numerical optimal control problem. This approach does not necessarily lead to a stable closed loop, implying that other specific closed-loop stabilizing strategies must be implemented to guarantee stability.

The importance of understanding how the design parameters (horizon, weight matrices, terminal cost and certain constraints) should be chosen to avoid an unstable NMPC, especially for open loop unstable systems, are highlighted in [17]. Based on the proper choice of design parameters, a general framework to design stabilizing nonlinear model predictive controllers is presented in [23]. An outline of these results are presented here as a basis for the stability analysis of NMPC used in the work presented in this thesis.

3.3.2 Stability Preserving NMPC formulations

The most common stabilizing NMPC methods implement sufficiently large horizon or a terminal cost and a terminal constraint to increase the chances of stability. The use of an infinite horizon means that the open-loop input and state trajectories computed as the solution of the NMPC optimization problem at a specific sampling instant are equal to the closed-loop trajectories of the nonlinear system. It also implies that the remaining part of the trajectories at the next sampling instant are optimal. Moreover, convergence of the closed-loop is obtained [20]. However, the use of an infinite horizon easily leads to an infinite dimensional problem which is difficult (if not impossible) to solve or implement in practice. Research on more practical approaches (using finite horizon) led to the introduction of either a *terminal cost* $S(x(T), T)$ or a *terminal constraint* or both into the NMPC problem formulation. The use of a terminal constraint in most stabilizing NMPC formulations states that the state vector at the end of the horizon, $x(t_N)$ should be inside a set Ω. The reformulated NMPC problem in a general form

entails minimizing the new cost function

$$J(x(t_k), u(t_k), t_k) = \sum_{k=0}^{N-1} \ell(x(t_{k+1}), u(t_k), t_k) + S(x(t_N), T) \tag{3.21}$$

subject to

$$x(t_{k+1}) = f(x(t_k), u(t_k), t_k), \qquad k = 0, ..., N-1 \tag{3.22}$$
$$x(t_0) = x(0), \qquad \text{given} \tag{3.23}$$
$$g(x(t_k), u(t_k), t_k) \leq 0, \qquad t_k \in T_d \tag{3.24}$$
$$u_{min} \leq u(t_k) \leq u_{max}, \qquad t_k \in T_d \tag{3.25}$$
$$x_N \in \Omega \tag{3.26}$$

The variants that implement only a terminal cost can be justified by the use of the implicit requirement that $x(t_N) \in \Omega$ is satisfied for every initial state x in a given compact set, and this is automatically satisfied if N is chosen sufficiently large. It is generally necessary to have a terminal constraint (even if it can be omitted in the on-line computations for sufficiently large N) if the system is nonlinear or if the system is linear and constrained but unstable [24]. Also, feasibility and reduction in the cost function compared to the control trajectory computed at the previous sample is sufficient for asymptotic stability of NMPC, provided the terminal constraints are included in the formulation [17],[25]. Furthermore, for a constrained nonlinear system, stabilizing ingredients found in [22],[26], and based on the accounts of [24],[16], suggest the use of a local control law $u(t_k) = \alpha(x(t_k)) = Kx$ to stabilize the linearized system

$$x(t_{k+1}) = Ax(t_k) + B(t_k)u \tag{3.27}$$

and the choice of Ω to satisfy the set constraint $\Omega \subset X$ and $\alpha(x(t_k)) \subset U$. Where X and U are the system constraints imposed on the state and control sequences respectively. By choosing Ω, according to [22], to be a level set of the terminal cost $S(.)$, where $S(.) = (1/2)x^T Px$ is a Lyapunov function for the linearized system (3.27) and satisfying the Lyapunov equation

$$S(Ax + B\alpha(x)) - S(x) + \bar{\ell}(x, \alpha(x)) = 0 \tag{3.28}$$

for all x where $\bar{\ell}(x, u) = \beta\ell(x, u)$ and $\beta \in (1, \infty)$. Replacing $\ell(.)$ by $\bar{\ell}(.) > \ell(.)$ provides sufficient margin that ensures that $f(x, \alpha(x)) \in \Omega$, $\forall x \in \Omega$ and $\dot{S}(x, \alpha(x)) + \ell(x, \alpha(x)) \leq 0$ are satisfied when Ω is a sufficiently small level set of $S(.)$. In the conditions above, it is required that Ω is forward invariant under $\alpha(.)$ (i.e. $x(t_k) \in \Omega \implies x(t_{k+1}) \in \Omega$) and that $S(.)$ is a local Lyapunov function. In addition, the forward invariant requirement automatically holds if Ω is defined as $\{x : S(x) \leq c\}$, for some $c > 0$. The aim of these conditions (applied to a regulation problem with origin at zero) is to obtain a terminal cost, $S(.) > 0$, that overbounds the rest of the horizon, that is,

$$S(x(t_N)) \geq \sum_{k=N}^{\infty} \ell(x(t_k), \alpha(x(t_k))) \tag{3.29}$$

The above statement is true if $S(.)$ fulfills

$$S(x(t_{k+1})) - S(x(t_k)) \leq -\ell(x(t_k), \alpha(x(t_k))), \quad \forall_x(t_k) \in \Omega, \tag{3.30}$$

by adding together (3.30) for $k = N, N+1, ..., \infty$ and using that $S(x(t_\infty)) = 0$.

To sum up, the resulting NMPC optimization problem (3.21) ensures that the closed-loop system $x(t_{k+1}) = f(x(t_k), \alpha(x(t_k)))$ converges to zero as $k \to \infty$ if its initial state lies in Ω. In other words, stability is guaranteed if the terminal cost $S(.)$ and a local control law $\alpha(.)$ are chosen such that (3.30) holds on a terminal set constraint Ω where the control law $\alpha(.)$ is unconstrained. The results achieved using the above approach are usaully refered to as *quasi-infinite* since most of them attempt to construct a finite horizon objective function that overbounds the infinite horizon objective function.

3.3.3 A general Stabilizing NMPC design approach

It should be noted that the use of local linearization is an attempt to introduce a fairly practical stabilizing NMPC approach, since finding a terminal cost such that (3.30) holds for all $x(t_k)$ is difficult, and often impossible for complex nonlinear systems subject to constraints. The idea of local stabilization, however, limits most *quasi-infinite* approaches to only certain class of nonlinear systems and control objectives, as identified in reference [23]. A practical challenge arises when dealing with systems for which the resulting linearization is not stabilizable. The same applies to nonlinear systems that cannot be stabilzed by a continuous feedback (such as nonholonomic systems). This also leads naturally to the question of how well most existing stabilizing NMPC schemes relate to fast trajectory tracking and time-varying reference problems.

Another motivation for further research is the fact that most practitioners of MPC methods know that for some systems, by an appropriate choice of some parameters of the objective function and horizon (usually obtained by trial-and-error and some imperical rules), it is possible to obtain stabilizing trajectories without imposing demanding artificial constraints. An encouraging development in this direction combines the results of [23] with a formulation along the lines of the ideas of [22] to obtain nonlocal results, as presented in [20].

The results in [23] proposes a rather general framework of NMPC for systems satisfying *mild* hypotheses and at the same time offering an increased flexibility in the choice of design parameters to reduce the constraints of the optimal control problem. In brief, stability of the closed-loop system can be guaranteed by choosing the *design parameters* (in this context: time horizon T, 'running' cost ℓ, terminal cost S and terminal constraint set Ω) to satisfy the following stability conditions [23]:

1. The set Ω is closed and contains the origin.

2. The function $\ell(x(t_k), u(t_k), t_k)$ is continuous, $\ell(0, 0, .) = 0$, and $\exists$ a continuous positive definite and radially unbounded function $M : \mathbb{R}^n \to \mathbb{R}_+$ s.t. $\ell(x, u, t) \geq M(x), \forall (t, u) \in \mathbb{R} \times \mathbb{R}^m$. Moreover, the "extended velocity set" $\{(v, \bar{\ell}) \in \mathbb{R}^n \times \mathbb{R}_+ : v = f(x, u, t), \bar{\ell} \geq \ell(x, u, t), u \in U(t)\}$ is convex $\forall (t, x)$.

3. The function S is positive semi-definite and continuously differentiable

4. The time horizon T is such that, the set Ω is reachable in time T from any initial state and from any point in the generated trajectory: that is, $\exists$ a set X containing X_0 such that for each pair $(t_0, x_0) \in \mathbb{R} \times X$ $\exists$ a control $u : [t_0, t_0 + T] \to \mathbb{R}^m$ satisfying $x(t_0 + T; t_0, x_0, u) \in \Omega$.
   Also, for all control functions u in the conditions above
   $x(t; t_0, x_0, u) \in X, \forall t \in [t_0, t_0 + T]$

5. There exists a scalar $\varepsilon > 0$ such that for each time $t \in [T, \infty)$ and each $x(t) \in \Omega$, we can choose a control function $\tilde{u} : [t, t + \varepsilon] \to \mathbb{R}^m$, with $\tilde{u}(t_k) \in U(t_k), \forall t_k \in [t, t + \varepsilon]$ satisfying
   $\dot{S}(x(t), t) = S_t(x(t), t) + S_x(x(t), t) \cdot f(x(t), \tilde{u}(t), t) \leq -\ell(x(t), \tilde{u}, t)$ and
   $x(t + r; t, x(t), \tilde{u}) \in \Omega, \forall r \in [0, \varepsilon]$

The above stability conditions firstly guarantee the existence of solutions to the optimal control problem, and secondly ensures that the closed-loop trajectory is actually driven to the origin. The latter is analogous to requiring that S is a control Lyapunov function. A distinguishing feature in this approach is condition 5 is only required to be satisfied on a subset Ω, which gives a considerable freedom of choice. Also, an appropriate choice of Ω makes it easier to choose the remaining design parameters.

The stability conditions are greatly simplified by making some standard choices of part of the design parameters. Condition 1 is necessary to guarantee the existence of a solution in the open-loop optimal control problem, while the first part of conditions 1 and 2 are trivially satisfied by choosing the usual quadratic objective function $\ell(x, u) = x^T Q x + u^T R u$, with $Q > 0$ and $R \geq 0$ and $S(x) = x^T P x$, with $P \geq 0$. The second part of condition 2 is a well-known requirement for the existence of solution in optimal control problem with integral cost, and it is automatically satisfied if ℓ is convex and f depends linearly on u, or if ℓ does not depend on u. Condition 4 is necessary to guarantee the existence of an admissible control sequence of the optimal control problem, and for instance, by choosing Ω as the whole space $\mathbb{R}^n$ makes condition 4 trivially satisfied. Moreover, Ω can be chosen to be a set of points satisfying condition 5, especially for more complex systems where it might be difficult to find parameters satisfying the stability conditions with a large set as $\Omega = \mathbb{R}^n$. Also, the task of choosing S to satisfy condition 5 can be simpified if the set Ω is restricted to be just a subset of $\mathbb{R}^n$ containing the origin, for instance, a linear subspace, a closed ball centred at the origin, or a set of trajectories approaching the origin.

In addition to the stability conditions, [23] offers a list of hypotheses that involve only the data of the system model, and can be used as initial test on the adequacy of the method.

3.3.4 Extentions to the general stabilizing approach

The results of [23] and the deductions made above motivate a practical approach to establishing the existence of a decreasing Lyapunov-like function. Furthermore, using the idea that feasibility implies stability [25], an effort to achieve an admissible control sequence in the sense of condition 4 above, will have a stabilizing effect on the NMPC. In the attempt to find the set Ω, as points satisfying

condition 5 for a fairly complex nonlinear system (such as a fighter aircraft), the following ideas are proposed for regulation towards time-varying references (which are expected in rapid maneuvres made by fighter aircrafts and missiles).

The following approach also reduces the very complex nature of the procedures that use the complete plant model $f(.)$ directly in their attempts to fulfill stability conditions similar to 5 above. It also avoids the use of the linearized plant model around some equilibrium points at which the plant must be stabilizable.

To simplify the introduction to the idea it is assumed that the plant model function $f(.)$ is *not* affine in the controls u, and a quadratic objective formulation is chosen as follows:

$$J(.) = \sum_{i=0}^{N-1} \underbrace{\left(\|x_{k+i|k} - r_{k+i|k}\|_Q^2\right)}_{\ell} + \underbrace{\|x_{k+N|k} - r_{k+N|k}\|_P^2}_{S}, \quad i = 0, 1, ..., N-1 \tag{3.31}$$

The linear dependence of $f(.)$ on u makes the "extended velocity" set convex, according to condition 2 of the general stabilizing NMPC approach in section 3.3.3. It is therefore further assumed that a solution exists for the resulting optimal control problem. However, the implementation of an objective function with no cost on inputs may lead to aggressive control, which in some practical cases results in the loss of stability. It should be therefore noted that the use of (3.31) as the objective function in this derivation leads to an implementation strategy that will be shown later as applicable for cost functions that have penalties and constraints on inputs. The error dynamics of the system takes the continuous form:

$$\dot{e}(t) = \dot{x}(t) - \dot{r}(t) \tag{3.32}$$

where $e(t) = x(t) - r(t)$, and for a constant reference, $e(t) = x(t) - r$:

$$\dot{e}(t) = \dot{x}(t) = f(x(t), u(t), t) \tag{3.33}$$

For simplicity let $e(t) = e$, $x(t) = x$ and

$$S(x) = \frac{1}{2} e^T P e, \quad P = P^T > 0, \tag{3.34}$$

such that $S(x)$ is a control Lyapunov function candidate.

Furthermore:

$$\dot{S}(x) = e^T P \dot{e} = e^T P \dot{x} \tag{3.35}$$

The main idea in this proposal is to influence the rate (as in condition 5) at which the states are driven towards inner level sets of S by introducing a new *terminal rate constraint* based on the idea of finding the steepest descent. That is, $\dot{x}$ in (3.35) is desired such that

$$\dot{x} = -\Gamma \nabla S(x) \triangleq \mathcal{S}(x), \quad \Gamma = \Gamma^T > 0 \tag{3.36}$$

where Γ is a scaling matrix which can be seen as the *rate control gain*, and

$$\nabla S(x) = \frac{\partial e^T}{\partial x} P e = P e \tag{3.37}$$

The choice (3.36) made for $\dot{x}$ does not replace the plant dynamics, but leads to an additional implementable constraint specification. Since the inverse of a positive definite matrix is also positive definite, P can be chosen such that $\Gamma^{-1} = P$, implying that

$$\nabla S(x) = \Gamma^{-1} e \tag{3.38}$$

and

$$\dot{x} = -\Gamma\Gamma^{-1} e = -e \tag{3.39}$$

Inserting equation (3.39) into (3.35) yields

$$\dot{S}(x) = e^T P \dot{x} = -e^T P e \leq 0 \tag{3.40}$$

The quadratic 'running' cost also takes the simple form

$$\ell(x) = e^T Q e, \quad Q = Q^T > 0 \tag{3.41}$$

Resulting in the condition

$$\dot{S}(x) = -e^T P e \leq -e^T Q e, \quad P \geq Q > 0 \tag{3.42}$$

The result (3.42) is comparable to $\dot{S} = S_t(.) + S_x(.) \cdot f(.) \leq -\ell(.)$ in condition 5. The use of (3.36) instead of $f(x(t), \tilde{u}(t), t)$ (see condition 5) suggests a way of *reshaping* the boundary behaviour of the dynamics of the system being controlled. Further investigations will therefore be required to assess the reliability of this choice. The idea here is to find suitable *stability-governing functions* $\mathcal{S}$ (dependent on x and possibly on u) that will induce a continual descent path for the control error.

The anticipated reduction in control error is immediately evident in the solution of the resulting rate constraint (3.39):

$$\dot{e} = -e \tag{3.43}$$

$$\implies e(t) = \mathrm{e}^{-t} e(0) \tag{3.44}$$

which can be written as

$$\begin{aligned} &|e(t)| = \mathrm{e}^{-t} |e(0)| \\ \implies &|e(t)| \leq \rho_s |e(0)|, \quad \rho_s = \mathrm{e}^{-t}, \quad \text{or } 0 < \rho_s < 1 \end{aligned} \tag{3.45}$$

An implementation strategy can be further deduced as follows:

$$\begin{aligned} &-\rho_s |e(0)| \leq e(t) \leq \rho_s |e(0)| \\ \implies &r - \rho_s |e(0)| \leq x(t) \leq \rho_s |e(0)| + r \end{aligned} \tag{3.46}$$

and applying the condition that $P \geq Q > 0$.

This interesting result depicts the induced reduction in control error portrayed by equation (3.42). In other words, introducing an error reduction rate constraint adds a possible descent path to the optimization problem, and can be used to govern the search of feasible (and possibly optimal) solutions. Equation (3.45) can also be seen as a dynamic (or *contracting*) constraint that induces a stabilizing effect to the NMPC. The implementation of (3.39) or (3.45) will result in a dynamic but persistent guidance of the states toward their reference trajectories,

rather than forcing the predicted states to lie in a terminal set Ω at the end the horizon.

Remark 1:
The above stabilizing approach can be adopted as a systematic way of relaxing the stability enforcing terminal constraints in practice. However, the choice of the *shaping* (or stability-governing) function (3.36) and the objective function (with no penalty on controls) may lead to a rather aggressive control. It is therefore necessary to extend the above results for quadratic 'running' costs of the form,

$$\ell(x) = e_x^T Q e_x + e_u^T R e_u, \quad Q = Q^T > 0,\ R = R^T > 0 \tag{3.47}$$

and equation (3.42) becomes

$$\dot{S}(x) = -e_x^T P e_x \leq -e_x^T Q e_x - e_u^T R e_u \tag{3.48}$$

$$\implies -e_x^T (P - Q) e_x \leq -e_u^T R e_u \tag{3.49}$$

The result in (3.49) requires further analysis before any conclusions can be drawn. Differences in scaling between states and inputs is one the reasons for requiring further work. However, an interesting observation is made when the weighting matrices are chosen such that (3.49) is always true, and that the largest error-values ($|e_L|$) become the dominating factors in (3.49). This assumption suggests that $|e_{x_L}| \geq |e_{u_L}|$, which will *not* be true when $|e_{x_L}| = 0$ while $|e_{u_L}| > 0$ (a rather common scenario, since control input/effort is often needed to maintain 'zero' error in the system states). For this reason, a new variable $\gamma_s > 0$ can be introduced such that $\gamma_s + |e_{x_L}| \geq |e_{u_L}|$, and implying that

$$\begin{aligned} -(\gamma_s + |e_{x_L}|) \leq e_{u_L} \leq (|e_{x_L}| + \gamma_s), \quad e_{u_L} = u_L - u_{r_L} \\ u_{r_L} - (\gamma_s + |e_{x_L}|) \leq u_L \leq (|e_{x_L}| + \gamma_s) + u_{r_L} \end{aligned} \tag{3.50}$$

where u_L is the input resulting in the largest control input error e_{u_L} at a given time instance, and u_{r_L} is the corresponding reference for the largest control input error. For the situation when $|e_x| \approx 0$:

$$u_{r_L} - \gamma_s \leq u_L \leq \gamma_s + u_{r_L} \tag{3.51}$$

A reasonable choice of γ_s will be $\gamma_s \geq u_{max}$, yielding

$$u_{r_L} - u_{max} \leq u_L \leq u_{max} + u_{r_L}, \tag{3.52}$$

which will already be fulfilled if the control inputs are constrained, and since a very common objective is to minimize the use of control effort (suggesting that $u_{r_L} = 0$ or chosen such that $u_{r_L} \ll u_{max}$).

Remark 2:
Another important issue that needs further investigation is *feasibility* in the suggested design approach. On this note, the gradient function used in the illustration above is intended to motivate the selection of an appropriate stability-governing function $\mathcal{S}$ that is convex over the space of $x(t)$ within the approximation limits of an SQP solution framework.

Remark 3:

Possibilities of introducing *slack variables*[1] to enhance feasibility can also be considered in this design framework. The use of slack variables may lead to loss of stability, and therefore requires further scrutiny.

Remark 4:
The choice of Γ can also be modified such that some flexibility of *rate control* (in addition to ρ_s) is implemented.

Remark 5:
Finally, since this approach mainly leads to a boundary specification (which becomes narrower over the prediction horizon) on the state and control errors, its implementation should be regarded as a means of enhancing stability, rather than guaranteeing stability.

3.4 Robustness and Practical Implications

3.4.1 Dealing with uncertainties

Robustness from both theoretical and practical points of view covers both the inherent properties of NMPC and specific NMPC designs that take uncertainties and disturbances into account. The inherent robustness property of NMPC originates from the close relation of NMPC and optimal control [20], where closed-loop stabilty and performance are inherently maintained in the presence of uncertainties. This observation is mostly applicable to unconstrained systems, which are easily studied by examining the descent property of a Lyapunov function for the nominal closed-loop system. However, when state and control constraints are involved, it is necessary, in addition, to ensure that disturbances do not cause voilation of the constraints [24]. An extra level of complexity of NMPC must therefore be introduced since most of the existing NMPC schemes are based on the assumption that no model/plant mismatch or unknown disturbances are present.

Most robust NMPC schemes that directly account for uncertainties/disturbances are based on a min-max formulation, in which the cost function takes worst case uncertainty/disturbance into account. The resulting min-max problem can be formulated as follows:

$$\min_{u(\,\cdot\,)} \max_{\Delta \in D} = \int_0^T \ell(x(t), u(t), t)\,\mathrm{d}t + S(x(T), T) \tag{3.53}$$

subject to

$$\dot{x}(t) = f_\Delta(x(t), u(t), t) \tag{3.54}$$

where D is the set of uncertainties and f_Δ is the system realization including uncertainties. A big challenge arises when stability constraints are also included since the resulting open-loop min-max optimization problem may not yield feasible solutions, considering all possible uncertainty/disturbance scenarios. The min-max formulation is also noted to provide good performance for the worst-case scenario,

[1] Slack variables are variables used to soften constraints. They are defined in such a way that they are non-zero only if the constraints are voilated. Their non-zero values are heavily penalized in the cost function, so that the optimizer has a strong insentive to keep them at zero if possible [14].

while poor results are typically obtained for the usual nominal operational situations [20], [17]. These drawbacks and other important reasons have led to the proposal of many schemes that aim at finding an acceptable compromise between performance and robustness. A usual objective function considered is [20]:

$$J = (1 - w)P + wR \tag{3.55}$$

where P and R are the performance and robustness terms respectively, and w is the quantifying parameter. Overview of several robust NMPC strategies, including min-max NMPC (and its variants), stochastic NMPC, and mechanisms to avoid steady state errors, are covered in [20], [24], [17] and their references. It must be noted that the results of many of the research works done on NMPC robustness are *conceptual* controllers that work in principle but are too complex to implement[24]. The main reason for the nonpractical nature of most of the proposed solutions to the robustness issues is that the resulting decision variables become infinite in dimension.

3.4.2 Practical challenges and tuning

An important feature of NMPC which naturally draws both theoretical and practical attention, is the internal model required and state feedback assumed by most NMPC formulations. Since all states are usually not measured, a state estimator/observer usually becomes a necessity. The design of observers for NMPC and its accompanying issues must also be considered in practical implementations. In addition, if no first principle model is available, it is often difficult (sometimes impossible) to obtain good nonlinear models based on identification techniques. In such cases other control strategies like linear MPC become better alternatives [20]. On the other hand, the possibility to directly use a detailed nonlinear first principles model is advantageous since the performance of the closed loop can be increased significantly without much tuning. Even though a detailed first principle model may lead to better performance, practical solutions often require a compromise, mostly due to the large computational burden involved in the resulting optimization.

Optimization algorithms developed for NMPC generally deal with non-convex problems, implying that only local, rather than global, solutions will be generally available. In practice, the optimization solvers assume some form of local convexity and guarantee convergence only to good initial guesses for the solution. Using good initial guesses can be regarded as a *warm start* strategy, which can be as simple as using the previous solution of the optimization problem. As a potentially efficient warm start technique, some optimization solvers consider the internal data of the optimization algorithm such as initial estimates of the Hessian approximation (in case exact Hessians are not computed), or initial estimates of factorizations of the Hessian (approximation). The accuracy of the information available for initialization, among other things, determine how beneficial the warm start strategy is to the optimization solver [17].

It may also be difficult to establish a nonconservative hard bound on the number of iterations required for convergence of the nonlinear optimization problem

that must be solved at each sampling instant. Lack of adequate computational resources can therfore limit the performance of NMPC, if good termination criteria are not employed. In such cases sub-optimal NMPC approaches that employ computationally efficient and robust implementations, rather than focusing on optimality, are prefered.

Another key aspect of NMPC is the understanding of how design parameters such as the prediction horizon, the time constant of the reference trajectory, and weight matrices are tuned. Satisfactory tuning may not be easy to achieve, and in worst cases, an unfortunate choice of the design parameters can result in a closed-loop unstable NMPC. As a starting point, conventional LQR tuning guidelines can be adapted for NMPC tuning [17]. In terms of effects of control weighting, the increasing of the weights R on the control moves relative to the weights Q on the tracking errors has the effect of reducing control activity. Basically, a stable plant can be expected to obtain a stable closed-loop by increasing the control weighting sufficiently, whereas an unstable plant can end up with an unstable feedback loop, if the elements of R are increased too much [14].

An important note on tuning, and a common strategy used to optimize the conditioning of the optimization problem, is *scaling* of the NMPC internal model variables. For tuning purposes, the model variables may be scaled such that equal errors on each of them are equally important. In that case equal weights can be assigned to the objective function error variables, as a good starting point for tuning. In addition, some optimization algorithms are sensitive to poor scaling [2], especially when *scale invariance* is not incorporated into all aspects of the algorithm. In such cases, scaling of model variables will be necessary. Scaling can also be regarded as a strategy used to increase the robustness of some controllers to modelling errors. The reason is that scaling reduces the sensitivity of the output trajectories to model errors [11].

The important challenges of NMPC discussed in this section can be seen in most cases as solvable, and the expected improvement of control performance in the use of nonlinear models provide good motivation for further research and studies on several unresolved issues (regarding robustness, output feedback, state estimation, etc.). These issues are however not slowing down the rapid growth of industrial application of NMPC [20], [14].

[2]For instance, in an unconstrained optimization, a problem is said to be poorly scaled if changes to a function variable x in a certain direction produce much larger variations in the function value f than do changes to x in another direction [27].

Chapter 4

Flight Dynamics and Control

Motion through the earth's atmosphere entails aerodynamic forces and moments. Some understanding of coordinate transformation, geodesy and the earth's gravitation is also reqiured for navigation around the earth. Coordinate transformations and the use of appropriate reference frames are vital parts of the aircraft simulation environment used in this project. This chapter introduces the relevant parts of flight dynamics and notations used in this work. Aerodynamic modeling on which most automatic feedback flight control is based is also briefly presented. Finally, the relation between the control system model and actuator input distribution and effectiveness is introduced.

4.1 Coordinate frames and Notations

Terrestrial navigation of aircraft involves the use of reference frames and coordinate transformations. The main reference frames for aircraft navigation and control are denoted by the following abbreviations and notations: ECI $\{i\}$, ECEF $\{e\}$, NED $\{n\}$ and BODY $\{b\}$. The ECI (*Earth-centered inertial*) reference frame is a nonrotating coordinate system in which Newton's laws of motion apply. The origin of ECI is located at the center of the earth. The ECEF (*Earth-centered, Earth-fixed*) frame also has its origin fixed to the center of the Earth, but its axes rotate relative to the inertial frame ECI. A geographic coordinate system has its axes aligned North, East, and Down (NED), or East, North, and Up (ENU), where "up" or "down" points normal to the earth's surface. The NED frame moves with the vehicle and has its origin defined relative to the earth's reference ellipsoid. The BODY frame represents a vehicle-fixed coordinate system which has its origin coinciding with the origin of the NED frame. The origin can be chosen as the vehicle center of mass, and the coordinate axes aligned with the vehicles reference directions. Vehicle *stability-axes* and *wind-axes* coordinate systems are also used as body-fixed frames in aircraft applications.

The aircraft equations of motion presented in the next section require coordinate rotation matrices between ECI, ECEF and NED. A plane rotation around the z-axis is made between ECEF and ECI, using the following rotation matrix:

$$R_i^e = \begin{bmatrix} \cos\mu & \sin\mu & 0 \\ -\sin\mu & \cos\mu & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

where R_i^e represents a rotation from ECI to ECEF, and μ is the rotation angle. After the above rotation has been made, the transformation from ECEF to NED can be achieved as derived in [2] and shown below:

$$R_e^n = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos\ell & \sin\ell & 0 \\ -\sin\ell & \cos\ell & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

$$R_e^n = \begin{bmatrix} -\sin\phi\cos\ell & -\sin\phi\cos\ell & \cos\phi \\ -\sin\ell & \cos\ell & 0 \\ -\cos\phi\cos\ell & -\cos\phi\sin\ell & -\sin\phi \end{bmatrix} \tag{4.3}$$

where ℓ is the terrestrial *longitude* and ϕ is the geodetic *latitude*.

In addition to the earth-related coordinate transformations stated above, the aerodynamic forces and moments on an aircraft are produced by relative motion with respect to the air and depends on the orientation of the aircraft with respect to the airflow [2]. To be able to specify the aerodynamic forces and moments, two orientation angles (with respect to the relative wind) are needed. The aerodynamic angles used are the *angle of attack* (α) and the *sideslip* angle (β), defined as:

$$\alpha := \arctan\left(\frac{W}{U}\right) \tag{4.4}$$

$$\beta := \arcsin\left(\frac{V}{V_T}\right) \tag{4.5}$$

where

$$V_T = \sqrt{U^2 + V^2 + W^2} \tag{4.6}$$

is the *total speed* of the aircraft, consisting of the longitudinal velocity U, lateral velocity V, and vertical velocity W. The angles α and β define the rotations necessary to transform BODY coordinates to the aircraft's *stability-axes* and *wind-axes* coordinates respectively. The relation between the *body-fixed* coordinate systems are shown in figure 4.1. The rotation matrices from BODY to stability, and stability to wind axes are

$$R_b^s = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \tag{4.7}$$

$$R_s^w = \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.8}$$

and the combined rotation from BODY to wind becomes

$$R_b^w = \begin{bmatrix} \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \\ -\cos\alpha\sin\beta & \cos\alpha & -\sin\alpha\sin\beta \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \tag{4.9}$$

By applying the rotation matrix (4.9) on the wind axes velocity vector $[V_T, 0, 0]^T$, the relationship between the velocities in the BODY and wind axes can be written

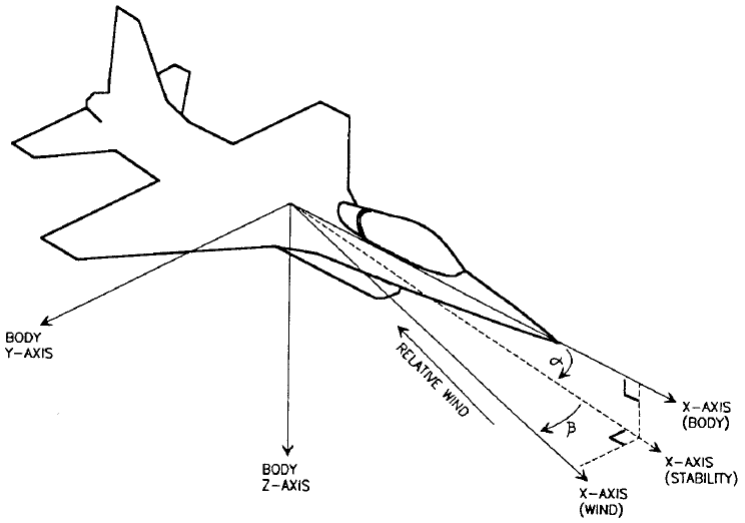


Figure 4.1: Illustration of aircraft body-fixed coordinate systems [2]

as:

$$\begin{aligned} U &= V_T \cos\alpha \cos\beta \\ V &= V_T \sin\beta \\ W &= V_T \sin\alpha \cos\beta \end{aligned} \tag{4.10}$$

It is worth noting that aerodynamic forces such as *lift*, *drag* and *cross-wind* are naturally defined with wind-axes coordinates and therefore make calculations in the wind-axes frame desirable. Stability-axes coordinate system on the other hand is used for analysing the effect of perturbations from study-state flight [2]. In addition, the position of the aircraft are described in relative to the inertial reference frame while the linear and angular velocities are conveniently expressed in the body-fixed coordinate system. The need for flexibility and the convenience of calculations in different coordinate systems make the above coordinate transformations a natural part of the broad field of Guidance, Navigation and Control.

4.2 Equations of motion, forces and moments

The mathematical representation of flight dynamics consists of rigid-body kinetics, aerodynamic forces and moments, and kinematic equations.

4.2.1 Rigid-body kinetics

The aircraft model can be written in matrix form [3], expressed in the BODY reference frame $\{b\}$ as:

$$\boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \tag{4.11}$$

$$\boldsymbol{\nu} = [U, V, W, P, Q, R]^T \tag{4.12}$$

$$\boldsymbol{\eta} = [X_E, Y_E, Z_E, \Phi, \Theta, \Psi]^T \tag{4.13}$$

where U, V, W represent the longitudinal, lateral and vertical velocities respectively, and P, Q, R represent the roll, pitch and yaw rates respectively. The X_E, Y_E, Z_E coordinates represent the Earth-fixed position of the aircraft, and the Euler angles Φ, Θ, Ψ represent the attitude of the aircraft. M_{RB} and $C_{RB}(\nu)$ take the following form:

$$\boldsymbol{M}_{RB} = \begin{bmatrix} m\boldsymbol{I}_{3\times3} & \boldsymbol{O}_{3\times3} \\ \boldsymbol{O}_{3\times3} & \boldsymbol{I}_{CG} \end{bmatrix} \tag{4.14}$$

$$\boldsymbol{C}_{RB}(\boldsymbol{\nu}) = \begin{bmatrix} m\boldsymbol{S}(\boldsymbol{\nu}_2) & \boldsymbol{O}_{3\times3} \\ \boldsymbol{O}_{3\times3} & -\boldsymbol{S}(\boldsymbol{I}_{CG}\boldsymbol{\nu}_2) \end{bmatrix} \tag{4.15}$$

where $\boldsymbol{\nu}$ has been further divided into $\boldsymbol{\nu}_1 := [U, V, W]^T$ and $\boldsymbol{\nu}_1 := [P, Q, R]^T$. The above equations were derived with the assumption that the BODY coordinate system origin is in the aircraft's center of gravity (CG). The inertia tensor used in equation (4.14) above is defined as (assuming $I_{xy} = I_{yz} = 0$, which corresponds to xz-plane symmetry):

$$\boldsymbol{I}_{CG} = \begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{xz} & 0 & I_z \end{bmatrix} \tag{4.16}$$

$\boldsymbol{\tau}$ in equation (4.11) is a generalized vector that includes aerodynamic and control forces formulated in [3] as:

$$\boldsymbol{\tau}_{RB} = -\boldsymbol{g}(\boldsymbol{\eta}) + \boldsymbol{\tau} \tag{4.17}$$

$$\boldsymbol{\tau}_{RB} = \begin{bmatrix} X & Y & Z & L & M & N \end{bmatrix}^T \tag{4.18}$$

where $\boldsymbol{g}(\boldsymbol{\eta})$ is obtained from the gravitational force $\boldsymbol{f}_G = [0, 0, \mathrm{m}g]^T$ which acts in the aircraft's CG:

$$\boldsymbol{g}(\boldsymbol{\eta}) = -(\boldsymbol{R}_b^n)^T \begin{bmatrix} \boldsymbol{f}_G \\ \boldsymbol{0}_{3\times1} \end{bmatrix} = \begin{bmatrix} \mathrm{m}g\sin(\Theta) \\ -\mathrm{m}g\cos(\Theta)\sin(\Phi) \\ -\mathrm{m}g\cos(\Theta)\cos(\Phi) \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.19}$$

g is the gravity constant and m is the mass of the aircraft. Equation (4.19) consists of the components of the aircraft's weight vector $\boldsymbol{W}$, which takes the form:

$$\begin{aligned} W_x &= -\mathrm{m}g\sin(\Theta) \\ W_y &= \mathrm{m}g\cos(\Theta)\sin(\Phi) \\ W_z &= \mathrm{m}g\cos(\Theta)\cos(\Phi) \end{aligned} \tag{4.20}$$

The state vector (4.18) for forces and moments acting on the aircraft consists of longitudinal force X, transverse force Y, vertical force Z, roll moment L, pitch momment M and yaw moment N. The resulting aircraft model in component

form and corresponding to (4.11) becomes:

$$\begin{aligned} \mathrm{m}(\dot{U} + QW - RV + g\sin(\Theta)) &= X \\ \mathrm{m}(\dot{V} + UR - WP - g\cos(\Theta)\sin(\Phi)) &= Y \\ \mathrm{m}(\dot{W} + VP - QU - g\cos(\Theta)\cos(\Phi)) &= Z \\ I_x\dot{P} - I_{xz}(\dot{R} + PQ) + (I_z - I_y)QR &= L \\ I_y\dot{Q} + I_{xz}(P^2 - R^2) + (I_x - I_z)PR &= M \\ I_z\dot{R} - I_{xz}\dot{P} + (I_y - I_x)PQ + I_{xz}QR &= N \end{aligned} \tag{4.21}$$

4.2.2 Aerodynamic Forces

In addition to the weight vector $\boldsymbol{W}$, the external forces acting on the aircraft depend on the aerodynamic force vector $\boldsymbol{R}$ and thrust vector $\boldsymbol{E}$. Assuming that the thrust produced by the engine, F_T, acts parallel to the aircraft's BODY x-axis, the components of $\boldsymbol{E}$ becomes:

$$\begin{aligned} E_x &= X_T = F_T \\ E_y &= Y_T = 0 \\ E_z &= Z_T = 0 \end{aligned} \tag{4.22}$$

where the subscript T indicates thrust effects. The components of $\boldsymbol{R}$ along the body-axes can also be written as

$$\begin{aligned} R_x &= X_A \\ R_y &= Y_A \\ R_z &= Z_A \end{aligned} \tag{4.23}$$

where the subscript A indicates aerodynamic effects. The size of the aerodynammic forces X_A, Y_A and Z_A is determined by the amount of air diverted by the aircraft in different directions. The amount of air diverted by the aircraft mainly depends on the total velocity (or Mach number M), density of the airflow ρ, the geometry of the aircraft, the orientation of the aircraft relative to the airflow (α and β), the control surface deflections δ_s, the angular rates (P, Q, R), altitude (h), and the propulsion system effects (T_c). There are other variables such as the time derivatives of the aerodynamic angles, but these effects are less prominent, since it is assumed that the aircraft is a rigid body. Based on the above dependencies, the aerodynamic forces can be modelled in the following general form:

$$\begin{aligned} X_A &= \bar{q}SC_X(\alpha, \beta, M, h, \delta_s, T_c, \dots) \\ Y_A &= \bar{q}SC_Y(\alpha, \beta, M, h, \delta_s, T_c, \dots) \\ Z_A &= \bar{q}SC_Z(\alpha, \beta, M, h, \delta_s, T_c, \dots) \end{aligned} \tag{4.24}$$

S represents the wing total area, and $\bar{q}$ is the dynamic pressure defined by

$$\bar{q} := \frac{1}{2}\rho V_T^2 \tag{4.25}$$

The coefficients C_X, C_Y and C_Z are usually obtained from wind tunnel data and flight tests. Combining the equations in (4.23) with the thrust components (4.22),

the force equations of (4.21) can be rewritten as:

$$\begin{aligned} \mathrm{m}(\dot{U} + QW - RV + g\sin(\Theta)) &= X_A + F_T \\ \mathrm{m}(\dot{V} + UR - WP - g\cos(\Theta)\sin(\Phi)) &= Y_A \\ \mathrm{m}(\dot{W} + VP - QU - g\cos(\Theta)\cos(\Phi)) &= Z_A \end{aligned} \tag{4.26}$$

4.2.3 Aerodynamic Moments

The external moments that affect the aircraft are those that are due to aerodynamics ($M_A = [\ell, m, n]^T$) and engine angular momentum ($M_T = H_{eng}$). As a result, the moments are:

$$\begin{aligned} L &= \ell \\ M &= m - R\,H_{eng} \\ N &= n + Q\,H_{eng} \end{aligned} \tag{4.27}$$

where the engine angular momentum (H_{eng}) is assumed to be parallel to the BODY x-axis of the aircraft. The aerodynamic moments can be expressed in a similar way as the aerodynamic forces in equation (4.24):

$$\begin{aligned} \ell &= \bar{q}SbC_{\ell_T}(\alpha, \beta, M, h, \delta_s, T_c, \dots) \\ m &= \bar{q}S\bar{c}C_{m_T}(\alpha, \beta, M, h, \delta_s, T_c, \dots) \\ n &= \bar{q}SbC_{n_T}(\alpha, \beta, M, h, \delta_s, T_c, \dots) \end{aligned} \tag{4.28}$$

where b is the wing span, and $\bar{c}$ represents the *mean aerodynamic chord* of the aircraft's wing. Inserting equation (4.27) into the moment equations of (4.21), yields the complete body-axis moment equautions:

$$\begin{aligned} I_x\dot{P} - I_{xz}(\dot{R} + PQ) + (I_z - I_y)QR &= \ell \\ I_y\dot{Q} + I_{xz}(P^2 - R^2) + (I_x - I_z)PR &= m - R\,H_{eng} \\ I_z\dot{R} - I_{xz}\dot{P} + (I_y - I_x)PQ + I_{xz}QR &= n + Q\,H_{eng} \end{aligned} \tag{4.29}$$

Further elaborations on aerodynamic force and moment coeffeicients are presented in the following section.

4.2.4 Aerodynamic force and moment coefficients

The forces and moments acting on a complete aircraft can be defined in terms of dimentionless aerodynamic coefficients. The aircraft aerodynamic coefficients are functions of the aerodynamic angles (α and β), Mach number (M), and altitude (h). In addition, control surface deflections (δ_s) and propulsion system effects (T_c) cause changes in the coefficients. The dependence of an aerodynamic coefficient can therefore be written as

$$C = C(\alpha, \beta, M, h, \delta_s, T_c) \tag{4.30}$$

To model aerodynamic effects when an aircraft maneuvers, a differential equation model of the aerodynamic force or moment is required. A linear approximation of the differential equation model can be achieved by considering the aerodynamic forces and moments to be linearly proportional to the angular rates that produced them. The resulting coefficients of proportionality are known as the *aerodynamic*

derivatives [2].

For control allocation purposes and specifically considering control surfaces, the aircraft dynamic equations involving rotational rates are of particular interest. From the equations involving the turn rates (P, Q, R) an explicit relationship between turn rates and torques (ℓ, m, n) applied to the aircraft can be obtained [28]. However, the rolling, pitching and yawing moments of the aircraft and how they are influenced by control effects can be conveniently described using dimensionless derivatives as presented in [2].

The rolling, pitching and yawing moments are related to their coefficients through the following formulations expressed in the aircraft's BODY coordinates, as introduced in section 4.2.3:

$$\ell = \bar{q}SbC_\ell \tag{4.31}$$
$$m = \bar{q}S\bar{c}C_m \tag{4.32}$$
$$n = \bar{q}SbC_n \tag{4.33}$$

where b is the wing span, S is the wing total area, $\bar{c}$ represents the *mean aerodynamic chord* of the aircraft's wing, and $\bar{q}$ is the dynamic pressure. The baseline rolling-moment coefficient is primarily a function of sideslip, angle of attack and Mach number, and can be written as $C_\ell(\beta, \alpha, M)$. Rolling moments are created by sideslip, by the control action of ailerons and rudder, and as damping moments resisting rolling and yawing motion. The rolling moment model of a high performance aircraft is given in [2] as

$$\begin{aligned} C_\ell &= C_\ell(\alpha, \beta, M) + \Delta C_{\ell_{\delta a}}(\alpha, \beta, M, \delta_a) + \Delta C_{\ell_{\delta r}}(\alpha, \beta, M, \delta_r) \\ &\quad + \frac{b}{2V_T}[C_{\ell_p}(\alpha, M)P + C_{\ell_r}(\alpha, M)R] \end{aligned} \tag{4.34}$$

where C_{ℓ_r} is the *roll-damping derivative*, and the quantities $Pb/(2V_T)$ and $Rb/(2V_T)$ can be thought of as dimensionless roll and yaw rates respectively. Linearized versions of the rolling moment dependence on β and a control surface deflection can be achieved by linearizing around the origin, yielding:

$$C_\ell(\alpha, \beta, M) \approx C_{\ell_\beta}(\alpha, M) \times \beta \tag{4.35}$$
$$\Delta C_{\ell_{\delta s}}(\alpha, \beta, M, \delta_s) \approx C_{\ell_{\delta s}}(\alpha, \beta, M) \times \delta_s \tag{4.36}$$

where C_{ℓ_β} is the *dihedral derivative* that determines static stability in roll, and $C_{\ell_{\delta s}}$ consists of roll *control derivatives*.

Similar formulations can be made for pitching and yawing moment coefficients as follows:

$$\begin{aligned} C_m &= C_m(\alpha, M, h, \delta_F, T_c) + \Delta C_{m_{\delta e}}(\alpha, M, h, \delta_e) + \frac{\bar{c}}{2V_T}[C_{m_q}Q + C_{m_{\dot{\alpha}}}\dot{\alpha}] \\ &\quad + \frac{x_R}{\bar{c}}C_L + \Delta C_{m_{thrust}}(\delta_t, M, h) + \Delta C_{m_{gear}}(h) \end{aligned} \tag{4.37}$$
$$\begin{aligned} C_n &= C_n(\alpha, \beta, M, T_c) + \Delta C_{n_{\delta r}}(\alpha, \beta, M, \delta_r) + \Delta C_{n_{\delta a}}(\alpha, \beta, M, \delta_a) \\ &\quad + \frac{b}{2V_T}[C_{n_p}(\alpha, M)P + C_{n_r}(\alpha, M)R] \end{aligned} \tag{4.38}$$

The corresponding linearized coefficients for control surfaces are

$$\Delta C_{m_{\delta s}}(\alpha, M, h, \delta_s) \approx C_{m_{\delta s}}(\alpha, M, h) \times \delta_s \tag{4.39}$$

$$\Delta C_{n_{\delta s}}(\alpha, \beta, M, \delta_s) \approx C_{n_{\delta s}}(\alpha, \beta, M) \times \delta_s \tag{4.40}$$

and the linearized yawing moment dependence on β is

$$C_n(\alpha, \beta, M, T_c) \approx C_{n_\beta}(\alpha, M, T_c) \times \beta \tag{4.41}$$

The baseline pitching moment coefficient consists of both low-speed and high-speed aircraft variables and can therefore be correspondingly expressed as $C_m(\alpha, M, T_c)$ and $C_m(\alpha, M, h)$. δ_F represents the wing-flap deflection, which can be treated as a separate increment. C_{m_q} is the pitch damping derivative, and the purpose of the $(x_R C_L)/\bar{c}$ is to correct for any x-displacement (x_R) of the aircraft's center of mass from the aerodynamic data reference position. The *acceleration derivative* $C_{m_{\dot{\alpha}}}$ appearing in (4.37) is used to model the delay in the change in sidewash felt at the tail of the fuselage and wings. The effects of engine thrust vector not passing through the aircraft center of mass and landing-gear moment effects are also included in the last two terms. In the yawing moment coefficient, C_{n_r} is the *yaw-damping derivative*, and C_{n_β} represents the *yaw stiffness derivative*.

4.2.5 Kinematics

The kinematics involve the aircraft motion when the mechanisms (such as forces and moments) causing the motion are not regarded. Kinematic equations are therefore primarily used for translation, rotation and attitude representation. The kinematic equations of a body-fixed coordinate system $\{b\}$ with respect to NED $\{n\}$ can be expressed in terms of Euler angles as in [3]:

$$\begin{bmatrix} \dot{X}_E \\ \dot{Y}_E \\ \dot{Z}_E \end{bmatrix} = \boldsymbol{R}_b^n \begin{bmatrix} U \\ V \\ W \end{bmatrix} = \boldsymbol{R}_{z,\Psi} \boldsymbol{R}_{y,\Theta} \boldsymbol{R}_{x,\Phi} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \tag{4.42}$$

Expanding equation (4.42) gives:

$$\begin{bmatrix} \dot{X}_E \\ \dot{Y}_E \\ \dot{Z}_E \end{bmatrix} = \begin{bmatrix} c\Psi & -s\Psi & 0 \\ s\Psi & c\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\Theta & 0 & s\Theta \\ 0 & 1 & 0 \\ -s\Theta & 0 & c\Theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\Phi & -s\Phi \\ 0 & s\Phi & c\Phi \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \tag{4.43}$$

$$= \begin{bmatrix} c\Psi c\Theta & -s\Psi c\Phi + c\Psi s\Theta s\Phi & s\Psi s\Phi + c\Psi c\Phi s\Theta \\ s\Psi c\Theta & c\Psi c\Phi + s\Phi s\Theta s\Psi & -c\Psi s\Phi + s\Theta s\Psi c\Phi \\ -s\Theta & c\Theta s\Phi & c\Theta c\Phi \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \tag{4.44}$$

The result in (4.44) consists of equations that are also refered to as the aircraft's navigation equations. The aircraft's attitude is determined by:

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} \dot{\Phi} \\ 0 \\ 0 \end{bmatrix} + \boldsymbol{R}_{x,\Phi}^T \begin{bmatrix} 0 \\ \dot{\Theta} \\ 0 \end{bmatrix} + \boldsymbol{R}_{x,\Phi}^T \boldsymbol{R}_{y,\Theta}^T \begin{bmatrix} 0 \\ 0 \\ \dot{\Psi} \end{bmatrix} \tag{4.45}$$

which gives:

$$\begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} 1 & s\Phi t\Theta & c\Phi t\Theta \\ 0 & c\Phi & -s\Phi \\ 0 & s\Phi/c\Theta & c\Phi/c\Theta \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \tag{4.46}$$

where s, c, t represent sin, cos and tan respectively.

4.3 Nonlinear Control Model of a High Performance aircraft

The equations of motion derived in the previous sections can be gathered into a complete system of first order differential equations, categorized into two groups:

Force and Moment equations:

$$\dot{U} = RV - QW - g\sin\Theta + \frac{1}{\mathrm{m}}(X_A + F_T) \tag{4.47}$$

$$\dot{V} = -RU + PW + g\sin\Phi\cos\Theta + \frac{1}{\mathrm{m}}Y_A \tag{4.48}$$

$$\dot{W} = QU - PV + g\cos\Phi\cos\Theta + \frac{1}{\mathrm{m}}Z_A \tag{4.49}$$

$$\dot{P} = (c_1R + c_2P)Q + c_3\ell + c_4(n + Q\,H_{eng}) \tag{4.50}$$

$$\dot{Q} = c_5PR - c_6(Q^2 - R^2) + c_7(m - R\,H_{eng}) \tag{4.51}$$

$$\dot{R} = (c_8P - c_2R)Q + c_4\ell + c_7(n + Q\,H_{eng}) \tag{4.52}$$

where

$$\Gamma c_1 = (I_y - I_z)I_z - I_{xz}^2, \quad \Gamma c_2 = (I_y - I_y + I_z)I_{xz}, \quad \Gamma c_3 = I_z$$
$$\Gamma c_4 = I_{xz}, \quad c_5 = \frac{1}{I_y}(I_z - I_x), \quad c_6 = \frac{I_{xz}}{I_y}, \quad c_7 = \frac{1}{I_y}$$
$$\Gamma c_8 = I_x(I_x - I_y) + I_{xz}^2, \quad \Gamma c_9 = I_x, \quad \Gamma = I_xI_z - I_{xz}^2$$

and

Kinematic and Navigation equations:

$$\dot{\Phi} = P + \tan\Theta(Q\sin\Phi + R\cos\Phi) \tag{4.53}$$

$$\dot{\Theta} = Q\cos\Phi - R\sin\Phi \tag{4.54}$$

$$\dot{\Psi} = \frac{1}{\cos\Theta}(Q\sin\Phi + R\cos\Phi) \tag{4.55}$$

$$\dot{X}_E = Uc\Psi c\Theta + V(c\Psi s\Theta s\Phi - s\Psi c\Phi) + W(c\Psi s\Theta c\Phi + s\Psi s\Phi) \tag{4.56}$$

$$\dot{Y}_E = Us\Psi c\Theta + V(s\Psi s\Theta s\Phi + c\Psi c\Phi) + W(s\Psi s\Theta c\Phi - c\Psi s\Phi) \tag{4.57}$$

$$\dot{Z}_E = -Us\Theta + Vc\Theta s\Phi + Wc\Theta c\Phi \tag{4.58}$$

Since the Earth-fixed z-position axis points downwards, a prefered representation of equation (4.58) is:

$$\dot{h} = Us\Theta - Vc\Theta s\Phi - Wc\Theta c\Phi \tag{4.59}$$

where h is the *altitude* of the aircraft.

4.3.1 Wind or Stability-Axes Equations

For control design purposes, it is usually preferable to transform the force equations (4.47)-(4.49) to the wind-axes coordinate system. The derivatives of V_T, α

and β defined by (4.4)-(4.6), according to [2] become:

$$\dot{V}_T = \frac{1}{\mathrm{m}}(F_T \cos\alpha \cos\beta - \mathcal{D} + \mathrm{m}g_1) \tag{4.60}$$

$$\dot{\alpha} = Q - (P\cos\alpha + R\sin\alpha)\tan\beta - \frac{1}{\mathrm{m}V_T \cos\beta}(\mathcal{L} + F_T \sin\alpha - \mathrm{m}g_3) \tag{4.61}$$

$$\dot{\beta} = P\sin\alpha - R\cos\alpha + \frac{1}{\mathrm{m}V_T}(\mathcal{C} - F_T \cos\alpha \sin\beta + \mathrm{m}g_2) \tag{4.62}$$

where the drag force $\mathcal{D}$, the cross-wind force $\mathcal{C}$ and the lift force $\mathcal{L}$ are defined as:

$$\mathcal{D} = -X_A \cos\alpha\cos\beta - Y_A \sin\beta - Z_A \sin\alpha\cos\beta \tag{4.63}$$

$$\mathcal{C} = -X_A \cos\alpha\sin\beta + Y_A \cos\beta - Z_A \sin\alpha\sin\beta \tag{4.64}$$

$$\mathcal{L} = X_A \sin\alpha - Z_A \cos\alpha \tag{4.65}$$

The gravity components are:

$$g_1 = g(-\cos\alpha\cos\beta\sin\Theta + \sin\beta\sin\Phi\cos\Theta + \sin\alpha\cos\beta\cos\Phi\cos\Theta) \tag{4.66}$$

$$g_2 = g(\cos\alpha\sin\beta\sin\Theta + \cos\beta\sin\Phi\cos\Theta - \sin\alpha\sin\beta\cos\Phi\cos\Theta) \tag{4.67}$$

$$g_3 = g(\sin\alpha\sin\Theta + \cos\alpha\cos\Phi\cos\Theta) \tag{4.68}$$

4.4 Decoupling of the Nonlinear Equations

In aircraft control applications, it is a usual practice to decouple the equations of motion into longitudinal and lateral modes. The key assumption is that the length of the aircraft's fuselage is much greater than its width and height, and also that the longitudinal velocity is much larger than the vertical and transversal velocities [3]. Another motivation is that most aircrafts spend most of their time in a wings-level steady-state flight condition, and since the model of the 3-DOF motion in the NED vertical plane is much simpler than the complete 6-DOF model, it is reasonable to adapt and simplify the equations of motion for specific mode control purposes as much as possible.

4.4.1 Longitudinal Equations

To investigate the equations of motion under wings-level flight condition, and for small sideslip, the effects of the lateral states (β, Φ, P and R) are assumed negligible in the longitudinal channel. This assumption is also justified by the fact that, if the roll and yaw rates (P and R) are small, the pitching moment equation is not coupled to the rolling and yawing moment equations (by inspection of the moment equations (4.50)-(4.52)). The resulting pure longitudinal motion is described by the state vector $[h,\ V_T,\ \alpha,\ \Theta,\ Q]^T$ and the following equations:

$$\dot{h} = V_T \cos\alpha\sin\Theta - V_T \sin\alpha\cos\Theta \tag{4.69}$$

$$\dot{V}_T = \frac{1}{\mathrm{m}}(F_T\cos\alpha - \mathcal{D} - \mathrm{m}g\sin(\Theta - \alpha)) \tag{4.70}$$

$$\dot{\alpha} = Q - \frac{1}{\mathrm{m}V_T}(F_T\sin\alpha + \mathcal{L} - \mathrm{m}g\cos(\Theta - \alpha)) \tag{4.71}$$

$$\dot{\Theta} = Q \tag{4.72}$$

$$\dot{Q} = \frac{m}{I_y} \tag{4.73}$$

A common alternative model uses flight-path angle (γ) as a state variable instead of pitch attitude (Θ):

$$\dot{h} = V_T \cos\alpha \sin(\gamma + \alpha) - V_T \sin\alpha \cos(\gamma + \alpha) \tag{4.74}$$

$$\dot{V}_T = \frac{1}{\mathrm{m}}(F_T \cos\alpha - \mathcal{D} - \mathrm{m}g \sin\gamma) \tag{4.75}$$

$$\dot{\gamma} = \frac{1}{\mathrm{m}V_T}(F_T \sin\alpha + \mathcal{L} - \mathrm{m}g \cos\gamma) \tag{4.76}$$

$$\dot{\alpha} = Q - \dot{\gamma} \tag{4.77}$$

$$\dot{Q} = \frac{m}{I_y} \tag{4.78}$$

where

$$\gamma := \Theta - \alpha \tag{4.79}$$

The above longitudinal equations are used for a variety of purposes, from performance analysis to automatic control system design [2].

4.4.2 Lateral Equations

The aircraft's motion in the lateral mode consists of rolling, yawing, and sideslipping, while the angle of attack α, the speed V_T, and the pitch angle Θ remain constant. Assuming the constant longitudinal states α_0, V_{T0}, Θ_0, and neglecting Q, the lateral channel equations are obtained:

$$\dot{\beta} = P \sin\alpha_0 - R \cos\alpha_0 + \frac{1}{\mathrm{m}V_{T0}}(\mathcal{C} - F_{T0} \cos\alpha_0 \sin\beta + \mathrm{m}g_2) \tag{4.80}$$

$$\dot{P} = \frac{I_z}{I_x I_z - I_{xz}^2}\ell + \frac{I_{xz}}{I_x I_z - I_{xz}^2}n \tag{4.81}$$

$$\dot{R} = \frac{I_{xz}}{I_x I_z - I_{xz}^2}\ell + \frac{I_x}{I_x I_z - I_{xz}^2}n \tag{4.82}$$

$$\dot{\Phi} = P + R \tan\Theta_0 \cos\Phi \tag{4.83}$$

$$\dot{\Psi} = \frac{1}{\cos\Theta_0} R \cos\Phi \tag{4.84}$$

where the gravity component g_2 takes the form:

$$g_2 = g(\cos\alpha_0 \sin\beta \sin\Theta_0 + \cos\beta \sin\Phi \cos\Theta_0 - \sin\alpha_0 \sin\beta \cos\Phi \cos\Theta_0) \tag{4.85}$$

The dominating states in the lateral model are β, Φ, Ψ, P, and R.

4.5 Linear state-space model

For control purposes, a linearized version of of the equations of motion is usually desirable. Using equation (4.11) as starting point, a linearization about (ν_0, η_0, τ_0) can be obtained by assuming that $\dot{\nu}_0 = 0$. The result is:

$$\boldsymbol{C_{RB}}(\nu_0)\nu_0 + \boldsymbol{g}(\eta_0) = \tau_0, \tag{4.86}$$

and the linearized state-space model expressed in matrix form becomes [3]:

$$\boldsymbol{M_{RB}}\dot{\boldsymbol{\nu}} + \boldsymbol{N_{RB}}\boldsymbol{\nu} + \boldsymbol{G}\boldsymbol{\eta} = \boldsymbol{\tau} \tag{4.87}$$

The linearization is achieved by redefining the aircraft's state-space vectors as a sum of *norminal values* and *perturbations* as follows:

$$\boldsymbol{\tau} := \boldsymbol{\tau}_0 + \delta\boldsymbol{\tau} = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ L_0 \\ M_0 \\ N_0 \end{bmatrix} + \begin{bmatrix} \delta X \\ \delta Y \\ \delta Z \\ \delta L \\ \delta M \\ \delta N \end{bmatrix} \tag{4.88}$$

$$\boldsymbol{\nu} := \boldsymbol{\nu}_0 + \delta\boldsymbol{\nu} = \begin{bmatrix} U_0 \\ V_0 \\ W_0 \\ P_0 \\ Q_0 \\ R_0 \end{bmatrix} + \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} \tag{4.89}$$

The angles are also defined according to:

$$\begin{bmatrix} \Theta \\ \Phi \\ \Psi \end{bmatrix} := \begin{bmatrix} \Theta_0 \\ \Phi_0 \\ \Psi_0 \end{bmatrix} + \begin{bmatrix} \theta_0 \\ \phi_0 \\ \psi_0 \end{bmatrix} \tag{4.90}$$

The linearized state-space model will therefore consist of the states $u, v, w, p, q, r, \theta, \phi$ and ψ. Generally, lowercase variables are used for linear state representations.

Control inputs $\boldsymbol{u}$, and aerodynamic forces and moments that affect the flight dynamics can be modelled into the rigid-body kinetics (4.87) as:

$$\boldsymbol{\tau} = -\boldsymbol{M_F}\dot{\boldsymbol{\nu}} - \boldsymbol{N_F}\boldsymbol{\nu} - \boldsymbol{B}\boldsymbol{u} \tag{4.91}$$

M_F is the aerodynamic added mass, N_F represents the aerodynamic damping, and B is a matrix that describes the aircraft's actuator configuration and force/-moment coefficients. Linear theory has been assumed here in order to reduce the number of aerodynamic coefficients [3]. Inserting (4.91) into (4.87) results in:

$$(\boldsymbol{M_{RB}} + \boldsymbol{M_F})\dot{\boldsymbol{\nu}} + (\boldsymbol{N_{RB}} + \boldsymbol{N_F})\boldsymbol{\nu} + \boldsymbol{G}\boldsymbol{\eta} = \boldsymbol{B}\boldsymbol{u} \tag{4.92}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{N}\boldsymbol{\nu} + \boldsymbol{G}\boldsymbol{\eta} = \boldsymbol{B}\boldsymbol{u} \tag{4.93}$$

Consequently, the state-space model, considering only the BODY velocity vector $\boldsymbol{\nu} = [u, v, w, p, q, r]^T$, takes the form:

$$\dot{\boldsymbol{\nu}} = -\boldsymbol{M}^{-1}\boldsymbol{N}\boldsymbol{\nu} + \boldsymbol{M}^{-1}\boldsymbol{B}\boldsymbol{u} \tag{4.94}$$

and finally:

$$\dot{\boldsymbol{\nu}} = \mathbf{F}\boldsymbol{\nu} + \mathbf{G}\boldsymbol{u} \tag{4.95}$$

where $\mathbf{F} = -\boldsymbol{M}^{-1}\boldsymbol{N}$ and $\mathbf{G} = \boldsymbol{M}^{-1}\boldsymbol{B}$.

As an alternative, a state-space model using wind/stability axes can be obtained from the nonlinear equations listed in section 4.3, by assuming that α and β are small such that $\cos\alpha \approx 1$ and $\sin\beta \approx \beta$. The equations relating BODY and wind axes velocities (4.10) can then be written as:

$$\begin{aligned} U &= V_T \\ V &= V_T\beta \\ W &= V_T\alpha \end{aligned} \tag{4.96}$$

which yields:

$$\begin{aligned} U &= V_T \\ \beta &= \frac{V}{V_T} \\ \alpha &= \frac{W}{V_T} \end{aligned} \tag{4.97}$$

The state-space vector for describeing motion in 6 DOF can be chosen as:

$$\boldsymbol{x} = \begin{bmatrix} u & \beta & \alpha & p & q & r \end{bmatrix}^T \tag{4.98}$$

Further, the relationship between BODY velocity vector $\boldsymbol{\nu}$ and the state vector $\boldsymbol{x}$ can be written as:

$$\boldsymbol{\nu} = \boldsymbol{T}\boldsymbol{x} = diag\{1, V_T, V_T, 1, 1, 1\}\boldsymbol{x} \tag{4.99}$$

where $V_T > 0$. For constant total speed $V_T = U_0$,

$$V_T = 0 \tag{4.100}$$

$$\dot{\beta} = \frac{1}{V_T}\dot{v} \tag{4.101}$$

$$\dot{\alpha} = \frac{1}{V_T}\dot{w} \tag{4.102}$$

It becomes therefore possible to transform the body-fixed state-space model (4.95) to

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B_u}\boldsymbol{u} \tag{4.103}$$

where

$$\boldsymbol{A} = \boldsymbol{T}^{-1}\mathbf{F}\boldsymbol{T} \tag{4.104}$$

$$\boldsymbol{B_u} = \boldsymbol{T}^{-1}\mathbf{G} \tag{4.105}$$

If a modular structure is adopted, the torques or aerodynamic moments can be regarded as generalized or virtual control inputs produced by a control law, as a response to some pilot commands. Virtual control inputs $\boldsymbol{v}$ can therefore be introduced by reformulating (4.103) as:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B_v}(\boldsymbol{B}\boldsymbol{u}) \tag{4.106}$$

$$\boldsymbol{B_u} = \boldsymbol{B_v}\boldsymbol{B} \tag{4.107}$$

The virtual control input is defined as:

$$\boldsymbol{v} := \boldsymbol{B}\boldsymbol{u} \tag{4.108}$$

Equation (4.106) then becomes:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B_v}\boldsymbol{v} \tag{4.109}$$

4.6 Control Effectiveness

The B matrix in (4.108) describes the aircraft's actuator configuration and moment coefficients. In other words, B can be seen as the control effectiveness matrix, which decides the total control effort produced by the actuators. Rewriting (4.108) as $v = (CK)u$, introduces the actuator configuration matrix C and the force coefficient matrix K, according to reference [29].

Alternatively, the aerodynamic moments can be assumed to be affine in the controls [30], implying that:

$$M_v = Bu + c \tag{4.110}$$
$$\Rightarrow \boldsymbol{v} = M_v - c \tag{4.111}$$

where M_v represents the aerodynamic moment coefficients to be produced in roll, pitch and yaw. Using this alternative, the model parameters B and c can be computed by linearizing M_v at each sampling instant t around the current measurement vector, $x(t)$ and the previous control vector, $u(t-T)$. Where T is the samplings interval.

A convenient approach is based on the aerodynamic moment derivatives presented in section 4.2.4. For the control input vector $u = \begin{bmatrix}\delta_{a_1} & \delta_{a_2} & \delta_{e_1} & \delta_{e_2} & \delta_r & T_c\end{bmatrix}^T$, a simplified version of the aerodynamic moments can be formulated as follows:

$$\ell = \bar{q}SbC_\ell(\delta_{a_1}, \delta_{a_2}, \delta_{e_1}, \delta_{e_2}, p, r, \beta) \tag{4.112}$$
$$m = \bar{q}S\bar{c}C_m(\delta_{a_1}, \delta_{a_2}, \delta_{e_1}, \delta_{e_2}, q, \alpha) \tag{4.113}$$
$$n = \bar{q}SbC_n(\delta_{a_1}, \delta_{a_2}, \delta_{e_1}, \delta_{e_2}, \delta_r, p, r, \beta) \tag{4.114}$$

where

$$\begin{aligned} C_\ell &= C_{\ell_{\delta a1}}\delta_{a_1} + C_{\ell_{\delta a2}}\delta_{a_2} + C_{\ell_{\delta e1}}\delta_{e_1} + C_{\ell_{\delta e2}}\delta_{e_2} + C_{\ell_p}p \\ &\quad + C_{\ell_r}r + C_{\ell_\beta}\beta \end{aligned} \tag{4.115}$$
$$\begin{aligned} C_m &= C_{m_{\delta a1}}\delta_{a_1} + C_{m_{\delta a2}}\delta_{a_2} + C_{m_{\delta e1}}\delta_{e_1} + C_{m_{\delta e2}}\delta_{e_2} \\ &\quad + C_{m_q}q + C_{m_\alpha}\alpha \end{aligned} \tag{4.116}$$
$$\begin{aligned} C_n &= C_{n_{\delta a1}}\delta_{a_1} + C_{n_{\delta a2}}\delta_{a_2} + C_{n_{\delta e1}}\delta_{e_1} + C_{n_{\delta e2}}\delta_{e_2} + C_{n_{\delta r}}\delta_r \\ &\quad + C_{n_p}p + C_{n_r}r + C_{n_\beta}\beta \end{aligned} \tag{4.117}$$

It is obvious from the above equations that the aerodynamic coefficients depend on the elements of the state vector and the elements of the control input vector u. The coefficients depending on the states will therefore become part of the linear model's A matrix, and those depending on the control surface deflections will be part of the B_u matrix. Considering the model (4.109), the virtual control input

v in terms of aerodynamic coefficients (i.e. $v = [C_\ell, C_m, C_n]_u^T$) becomes:

$$v = \begin{bmatrix} C_{\ell_{\delta a1}}\delta_{a_1} + C_{\ell_{\delta a2}}\delta_{a_2} + C_{\ell_{\delta e1}}\delta_{e_1} + C_{\ell_{\delta e2}}\delta_{e_2} \\ C_{m_{\delta a1}}\delta_{a_1} + C_{m_{\delta a2}}\delta_{a_2} + C_{m_{\delta e1}}\delta_{e_1} + C_{m_{\delta e2}}\delta_{e_2} \\ C_{n_{\delta a1}}\delta_{a_1} + C_{n_{\delta a2}}\delta_{a_2} + C_{n_{\delta e1}}\delta_{e_1} + C_{n_{\delta e2}}\delta_{e_2} \end{bmatrix} \tag{4.118}$$

$$v = \underbrace{\begin{bmatrix} C_{\ell_{\delta a1}} & C_{\ell_{\delta a2}} & C_{\ell_{\delta e1}} & C_{\ell_{\delta e2}} & 0 & 0 \\ C_{m_{\delta a1}} & C_{m_{\delta a2}} & C_{m_{\delta e1}} & C_{m_{\delta e2}} & 0 & 0 \\ C_{n_{\delta a1}} & C_{n_{\delta a2}} & C_{n_{\delta e1}} & C_{n_{\delta e2}} & C_{n_{\delta r}} & 0 \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} \delta_{a_1} \\ \delta_{a_2} \\ \delta_{e_1} \\ \delta_{e_2} \\ \delta_r \\ T_c \end{bmatrix}}_{u} \tag{4.119}$$

where the *dimensionless derivatives* for the control input coefficients in roll pitch and yaw can be calculated using:

$$C_{\ell_{\delta_s}} = \frac{\partial C_\ell}{\partial \delta_s} \tag{4.120}$$

$$C_{m_{\delta_s}} = \frac{\partial C_m}{\partial \delta_s} \tag{4.121}$$

$$C_{n_{\delta_s}} = \frac{\partial C_n}{\partial \delta_s} \tag{4.122}$$

where δ_s represents any of the control surfaces which is applicable for the described moment.

4.7 Actuator dynamics

A further extension of the flight dynamics model is to include actuator dynamics which can be generally modelled as:

$$\dot{\delta} = g(\delta, \delta_c) \tag{4.123}$$

where δ_c is the commanded input and δ represents the actual actuator deflection. When first order actuator dynamics is assumed, the model can take the form:

$$\dot{\delta} = T^{-1}(\delta_c - \delta) \tag{4.124}$$

where T can be designed as a diagonal matrix with positive time constants: $T = diag\{T_1, T_2, ..., T_r\}$ [3]. However, since the actuator dynamics are often very fast compared to the remaining aircraft dynamics, the actuator model can be neglected by using the steady state approximation $\delta \approx \delta_c$ [30].

Chapter 5

Aircraft Stability and Control Characteristics

The dynamics of flight (treated in chapter 4) forms a solid base for the study of stability and control of aircrafts. The concept of equilibrium or steady state is also important in the definitions of stability and control. The equilibrium of an aircraft in flight describes a uniform motion condition. Basically, the equilibrium of the aircraft is stable if, when the aircraft is slightly disturbed in any of its degrees of freedom, it returns ultimately to its initial state. Instability in aircrafts has a rather complex nature in the sense that the aircraft may be stable with respect to one degree of freedom and unstable with respect to another. Two kinds of instability are of interest in aircraft dynamics. In the first kind, termed *static* instability, the aircraft departs continuously from its steady state condition, while the second kind, termed *dynamic* instability, is a more complicated phenomenon in which the aircraft oscillates about its steady state condition with ever-increasing amplitude [31].

The view on aircraft stability introduced above can be seen as *inherent stability*, since it is a property of the basic airframe with either fixed or free controls. An inherently unstable aircraft will therefore need a *stability augmentation system* (SAS) to provide *synthetic* stability. SAS also forms the base or foundation on which most of the existing modern autopilots are designed. The main reason is to ensure that the closed loop system is stable in its response to atmospheric disturbances as well as commands from a navigation system and the pilot. It is therefore obvious that controls play an important role in stabilizing an aircraft. At the same time, controls are responsible for both fixing the aircraft in a steady state condition and producing accelerated motion (or maneuvers). The complex task of aircraft control, therefore, involves stability investigations of the maneuvers and transition from one steady state to another.

The unsteady motions of an aircraft can be separated for convenience into *longitudinal* (or *symmetric*) motions and *lateral* (or *asymmetric*) motions (see section 4.4). This separation can also be made for both dynamic and static stability analysis. However, the results of greatest importance for static stability are those assosiated with the longitudinal analysis [31]. Thus the main focus of the following section (5.1) will be on longitudinal stability and control. Nevertheless, brief highlights will be given on static aspects of directional and rolling motions. Dynamic stability properties are also briefly discussed in section 5.2.

5.1 Static Stability

Static stability analysis is used to determine whether the aircraft will return to a steady state flight condition after being subject to a small atmospheric disturbance. In static analysis, rate-dependent effects are not considered, and it is usually performed for the special case of wing-level, nonturning flight [2].

5.1.1 Longitudinal Static Stability

Longitudinal static stability analysis considers the pitching moment that acts on the aircraft when its angle of attack is changed from a steady state condition. The pitching moment coefficient, C_m, for a fixed-wing aircraft depends on the angle of attack, center of gravity (CG) location, and the elevator deflection. To achieve longitudinal static stability (*positive pitch stiffness*), the sign of the stability derivative given by the slope of C_m change with respect to angle of attack,

$$C_{m_\alpha} = \frac{\partial C_m}{\partial \alpha} \tag{5.1}$$

must be *negative* [31],[2].

Positive pitch stiffness ensures a restoring moment if the aircraft is disturbed from its equilibrium state (trimmed flight) due to disturbances like wind gusts. At an equilibrium angle of attack, C_m must be zero (at a balanced state) and the pitching moment curve must be such that when the angle of attack is increased (while the speed remains unaltered) the aircraft will respond with a moment that tends to reduce (or restore) the angle of attack to its equilibrium value. It should, however, be noted that restriction to angle of attack disturbances when dealing with stability has limitations, since the aircraft may be unstable with respect to disturbances in speed. For instance, the aerodynamic characteristics of the aircraft may change with speed due to compressibility effects, structural distortion, or the influence of the propulsive system.

An important aspect of longitudinal stability and control deals with the position of the CG with respect to the aircraft's neutral point (NP). The neutral point (also known as the aircraft's *aerodynamic center*) is defined as the CG position for which C_{m_α} is zero, and it represents the boundary between positive and negative pitch stiffness. The difference between the CG position h and the NP position h_n is termed the *static margin*,

$$K_n = h_n - h \tag{5.2}$$

Positive pitch stiffness requires that $C_{m_\alpha} < 0$, implying that $h < h_n$ or $K_n > 0$. In other words the CG must be forward of the NP. The farther forward the CG the greater is the static margin K_n, hence the more stable the aircraft. Figure 5.1 illustrates the effect of CG location on C_m using a linear C_m vs. α relation.

Infact, the trim condition of the aircraft can be changed by moving CG, which changes the value of α at which $C_m = 0$. Moving CG forward reduces the trim α, and hence produces an increase in the trim speed. An inherent consequence is that, when the trim speed is reduced, C_{m_α} changes at the same time, resulting in a reduction of pitch stiffness and stability of the aircraft [31].

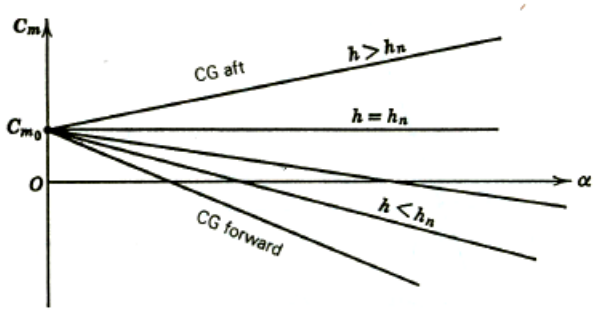


Figure 5.1: Effect of CG location on C_m curve [2]

In some cases, the neutral point tends to shift aft (i.e. CG forward in figure 5.1) when an aircraft moves from high subsonic speeds to supersonic speeds [1]. The result is a large increase in the static margin, which is followed by some undesirable consequences such as increased trim drag and reduced maneuverability. To minimize these penalties, some modern fighter aircraft (notably F-16) use a reduced, or negative, static margin at subsonic speeds. Since negative pitch stiffness normally leads to *dynamic instability* [2] in pitch, the fighter aircraft must rely on SAS for longitudinal stability.

5.1.2 Lateral Static Stability

The rotational stiffness in the longitudinal mode is about one axis only (the y-axis) and therefore serves as an important criterion for the dynamic behaviour of the aircraft. On the other hand, the rotational motion in the lateral mode takes place about two axes (x and z), and the moments assosiated with the rotations are cross-coupled. In addition, the gravity vector in normal flight lies in the aircraft plane of symmetry, eliminating CG position as a dominant parameter for the lateral characteristics. The simplicity of the rotational stiffness, depicting dynamic behavior of the aircraft, is lost when considering lateral motions. Therefore, a full lateral dynamic analysis is normally required.

Nevertheless, the requirement of positive *yaw stiffness* (also known as *weathercock stability*) is that

$$C_{n_\beta} = \frac{\partial C_n}{\partial \beta} \tag{5.3}$$

must be *positve* [31]. This implies that when the aircraft is at an angle of sideslip β relative to the flight path (i.e. along the *relative wind* x-axis, shown in figure 4.1), the yawing moment produced will tend to restore the aircraft to symmetric flight. Moreover, if the aircraft has positive yaw stiffness, and is truely symmetrical, it will attempt to maintain the sidelip angle at zero (a desirable control objective in most flight conditions [31]).

[1] Aerodynamic effects are classified according to *Mach number*, $M := V_T/a$, where a is the speed of sound at the ambient conditions. The Mach number ranges of interest in aerodynamics are $M < 1.0$ (subsonic speeds), $0.8 \leq M \leq 1.2$ (transonic speeds), $1.0 < M < 5.0$ (supersonic speeds), and $M \geq 5.0$ (hypersonic speeds) [2].

Lateral analysis also treats *roll stiffness*, but the results give very limited benefits to understanding the dynamic stability behavior of the aircraft. The answer to the question about whether the aircraft will return to wings-level attitude, or not, can only be provided by full dynamic analysis (see for example [31], [2] and [32] for details about this subject).

Static stability analysis provides the basic information for configuring the aircraft and evaluating its performance, and it is also a foundation for the dynamic analysis presented in the following section.

5.2 Dynamic Stability

Dynamic stability in all degrees of freedom can be determined from the eigenvalues of the linearized equations of motion. The eigenvalues λ can be found by solving the linear equation:

$$\det(\lambda \boldsymbol{I} - \boldsymbol{A}) = 0 \tag{5.4}$$

where $\boldsymbol{A}$ is the linearized system matrix (see section 4.5), and $\boldsymbol{I}$ is the identity matrix. The remainder of this section and the following subsections are based on the dynamic stability discussions in reference [32].

Basically, an aircraft is considered dynamically stable if the real parts of all its eigenvalues, λ_i, are negative. The aircraft will be dynamically unstable if it has zero [2] or positive real part of any complex eigenvalue.

5.2.1 Longitudinal Dynamic Stability

When longitudinal motions are considered, the result of expanding the determinant in (5.4) is a 4th degree polynomial in λ, and for most aircraft types, it takes the factorized form,

$$(\lambda^2 + 2\zeta_{ph}\omega_{ph}\lambda + \omega_{ph}^2)(\lambda^2 + 2\zeta_{sp}\omega_{sp}\lambda + \omega_{sp}^2) = 0 \tag{5.5}$$

where the two factors represent the aircraft's *phugoid* mode and *short period* mode, respectively.

The phugoid mode is observed as a long period oscillation with little damping. If the damping is negative, the phugoid mode is unstable and the oscillations increase with time. The phugoid mode is characterized by the natural frequency ω_{ph} and the relative damping ratio ζ_{ph}. The short period mode on the other hand is a fast mode characterized by the natural frequency ω_{sp}, and relatively well-damped with the damping ratio ζ_{sp}.

Supersonic aircraft may have very large and sufficiently negative value of the stability derivative M_u [3]. This results in the term ω_{ph}^2 in (5.5) becoming negative. Both roots (solutions) of the phugoid quadratic equation become real, where one is negative (stable/convergent) and the other positive (unstable/divergent). The

[2] A zero real part corresponds to a mode having simple harmonic motion, which, for practical flight situations, is considered unstable [32].

[3] M_u is the change in pitching moment caused by a change in forward speed.

unstable mode is referred to as the *tuck mode*, since the phenomenon is observed as a downward pointing nose (tucking under) as airspeed increases.

A modern fighter aircraft typically has its center of gravity CG located behind the neutral point NP. In this case, the stability derivative M_α [4] can take a value which will result in every root of the characteristic equation (5.5) being real. When the CG is moved further backwards, one of the real roots of the phugoid mode, and one of the real roots of the short period mode, become imaginary, forming a new complex pair. The complex pair is known as the *third oscillatory mode*, and it is the main influence upon the dynamic response of any automatic flight control system.

5.2.2 Lateral Dynamic Stability

The lateral characteristic equation obtained from (5.4) is of 5th degree, and it can be usually factorized into the following form:

$$\lambda(\lambda + e)(\lambda + f)(\lambda^2 + 2\zeta_D\omega_D\lambda + \omega_D^2) = 0 \tag{5.6}$$

where the simple term in λ corresponds to the heading mode ($\dot{\psi} = r$). The root $\lambda = 0$ (neutral stability) implies that the aircraft does not tend to restore its heading once it has been disturbed from its steady state. The term $(\lambda + e)$ corresponds to the *spiral convergence/divergence* mode, which is usually a very slow motion corresponding to a long term tendency either to maintain the wings level or to 'roll off' in a divergent spiral. The term $(\lambda + f)$ describes the *rolling subsidence mode*, which is a pure rolling motion. The quadratic term represents the *dutch roll* motion for which the damping ratio ζ_D is usually small, and the natural frequency ω_D depicts oscillatory motion.

[4] M_α is the change in pitching moment caused by a change in angle of attack α.

Chapter 6

Flight Control Systems and Faults

6.1 Automatic aircraft control

The desire of having automatic flight control can be attributed to the progress toward longer flight times, other than placing emphasis mainly on making the aircraft more controllable by the pilot. Infact, several factors including changes in aircraft mass and the need to reduce the area of aerodynamic surfaces (for lower drag at high speed), cause changes in natural modes of the aircraft, making some aircrafts not easily controllable by the pilot. In hypersonic aircraft control, for instance, a change in flight conditions can cause cause localized changes in airflow, which can lead to rapid temperature increases at some points on the surface of the aircraft. Manual control of such aircrafts is difficult or not feasible in most flight phases. The aircraft's trajectories can be carefully controlled by feedback comparison with a precomputed reference trajectory, or with real-time trajectory-prediction calculations. Feedback control can also be used to modify the aircraft dynamics when large changes in aircraft dynamics are experienced. It is interesting to note that a stable dynamic mode which is adequately damped for one flight condition may become unstable in another flight condition. Also, a slightly damped oscillatory mode may cause discomfort to passenger or make it difficult for the pilot to control the aircraft's trajectory precisely [2].

The above examples, among numerous others, show the obvious need of automatic control in modern aicrafts which are built for specific and varied performance targets, including speed, range, altitude, maneuverability and payload capacity. The design of modern flight control systems incorporates guidance and control, algorithms and simulations, numerical methods and digital implementation. In addition, flight control is advancing to many relatively new analytical techniques such as numerical optimization and multivariable control, adaptive techniques, and analysis of sensitivity and robustness to parameter variation.

To be able to implement automatic flight control, signals from rate gyros, accelerometers, air-data computers, and other data sources are processed by a flight control computer. The automatic control computer then generates augmentation signals to control effectors through electrical (and mostly hydraulic) means.

6.2 Augmentation systems and Autopilots

Automatic flight control systems consists of augmentation systems which either provide the pilot with a particular response to the control inputs or provide particular flight modes with adequate damping and natural frequencies. The former is termed *control augmentation* and the latter is known as *stability augmentation*. In addition to the augmentation systems, *autopilots* are usually implemented to provide "pilot relief" for the aircraft modes that are controllable by the pilot, and also other special functions. A brief description of classic autopilots that correspond to the NMPC autopilots designed in this project is included in appendix B. Autopilot applications are however broad and include pitch attitude hold control, altitude hold, speed/Mach hold, roll-angle hold, turn coordination, heading hold, automatic take-off and landing. Autopilots are normally designed to meet specifications on steady-state error and disturbance rejection, with less emphasis on dynamic response. Dynamic response, however, becomes a requirement when autopilots are coupled with guidance and navigation systems. A combination of these properties are incorporated in the design of the autopilots presented in this thesis.

6.3 Gain Scheduling

Control design using linearization approach has a basic limitation, which is the fact that the controller is guaranteed to work only in some neighborhood of a single operating (or equilibrium) point. To implement a linear control law on an aircraft, it must be gain-scheduled over the flight envelope where it will be used. *Gain scheduling* attempts to extend the validity of the linearization approach to a range of operating points. In its simplest form, gain scheduling involves linearization of the controlled system at several operating points, design of a linear feedback controller at each point (for example, finding optimal control gain for each one), and implementation of the resulting family of linear controllers as a single controller. This means that the gains most appropriate for the actual operating point of the aircraft will be used in the controller [2],[33].

The parameters of the gain-scheduled controller are typically varied by monitoring of some defined *scheduling variables* (either exogenous or endogenous signals with respect to the plant), by which the system model is parameterized. The most widely known gain scheduling in autopilot designs is that the autopilot loop gain is highly dependent on control surface effectiveness, due to altitude (or dynamic pressure) and Mach number (or speed) effects [34]. Dynamic pressure, Mach number, altitude, and angle of attack are therefore typical scheduling variables used in flight control systems.

The complete gain-scheduled autopilot design can be summarized as follows [2], [33], [34]: The nonlinear equations of motion are first linearized about selected operating points that capture key modes of the operation throughout the flight envelope (in other words, computing a linear parameter-varying (LPV) model of the nonlinear plant). Linear controllers are designed (as in section 6.2) to achieve the desired stability and performance requirements for the linearizations about the selected operating points. Traditionally, the designs are such that for each fixed value of the scheduling variables, the closed-loop system exhibits desirable

performance and stability. The parameters of the controller are then interpolated as functions of the gain scheduling variables. Finally, the gain-scheduled controller is implemented on the nonlinear system, and checks are made for nonlocal performance by simulating the nonlinear closed-loop model.

Further discussions, including detailed outline of gain scheduling techniques and their stability and robustness implications can be found in [34], [33] and [2].

6.4 Control surfaces and actuators

The control of the aircraft is achieved through control effectors. The engine primarily provides speed control while the rolling, pitching and yawing motions are mainly produced by ailerons, elevators and the rudder respectively. Figure 6.1 is an example of a conventional aircraft control surface configuration. Modern

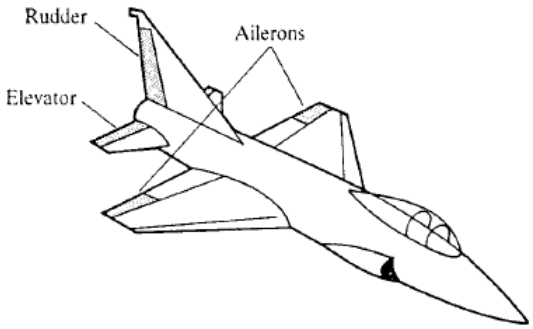


Figure 6.1: Control surfaces for a conventional aircraft [3]

aircraft typically have more control surfaces than shown in figure 6.1, and some are designed with specialized control surfaces and control configurations. The variation in configuration and number of control surfaces is usually motivated by performance and redundancy issues or requirements. Among the control surfaces

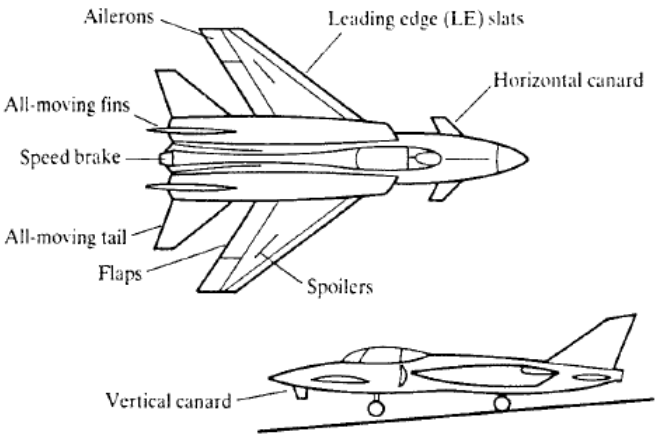


Figure 6.2: Control surfaces for a fighter aircraft [3]

not seen in figure 6.1 are canards, all-moving fins, flaps, spoilers, and combinations/merging of basic controls such as elevons, flaperons and stabilators. Figure

6.2 shows the location of some of these control surfaces on a fighter aircraft.

The control surface deflections commanded by the flight control system are realised either through electricservo driven actuation (fly-by-wire), hydraulicservo mechanism or a combination both. Figure 6.3 illustrates the orientation of the basic control surfaces. The aerodynamic forces and moments produced by the de-

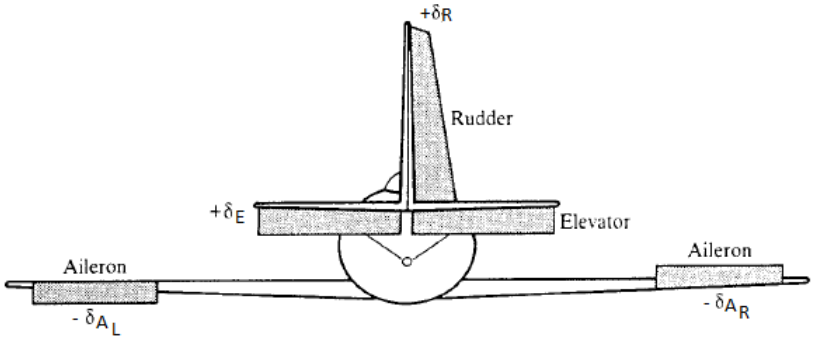


Figure 6.3: Positive orientation of the control surfaces

flection of the control surfaces occur basically because of divertion of airflow. The effects of control surface deflections on the rotational and translational motion of the aircraft can be either desirable or viewed as a disturbance. For instance, since the rudder is intended to provide directional control (yaw), a "cross-control" effect produced on the rolling moment of the aircraft becomes an unwanted effect [2]. Some of the control surfaces can however be used to produce desirable redundant effects. The differential deflection of the ailerons and spoilers produce yawing moments due to the difference in drag between the two sets. Also, roll control can be obtained in a number of different ways. The capability of several control surfaces effecting the same aerodynamic motion introduces *over actuation* in the aircraft control system. The effective utilization of this property in control system design has become a vital part of modern fault tolerant control.

6.5 Faults - actuator failures

Faults are undesirable events that can occur in different parts of the controlled system. In automatic aircraft control system environment faults can be classified as *actuator faults, sensor faults,* and *component faults*, according to reference [35].

Actuator faults present *partial* or *total* (complete) loss of control action. Complete actuator loss can occur, for instance, as a result of breakage, short circuits, or broken wiring, and the affected actuator can be 'jammed' or locked, producing no actuation regardless of the input applied to it. One of the worst kind of actuator faults is *actuator runaway* (or hard-over). Another kind of total actuator loss is *free-play* (*float-type* failure) or loss of actuator effectiveness. Partially failed actuators produce only a part of the normal actuation, and this can result from hydraulic or pneumatic leakage, increased resistance or a fall in the supply voltage. Due to their high prices and large size and mass, duplicating of the aircraft's actuators to increase fault tolerace is often not an option.

Sensor faults represent incorrect readings from the system's sensors. Sensor faults

can also be either *partial* (providing readings that are related to the measured signal) or *total* (from which on useful information can be retrieved). Due to the smaller sizes of sensors, they can be duplicated in the system to increase fault tolerance. Sensor faults are therefore not treated as part of this project.

Component faults on the other hand, are faults in the components of the aircraft itself. These faults represent changes in the physical parameters of the aircraft, for example, aerodynamic coefficients, damping constants, etc., usually caused by structural damage. Component faults are the most difficult to deal with, as they cover a very wide class of unexpected situations. This class of faults are considered out of scope of this project.

Reference [35] presents and discusses different mathematical representations of the above mentioned faults. For mathematical modeling purposes, faults can be classified as additive (representing component faults) and multiplicative (suitable for actuator and sensor faults). Faults can also be classified according to their time characteristics as *abrupt* (occuring instantaneously), *incipient* (slow parametric changes) and *intermittent* (appearing and disappearing repeatedly). The following section presents various methods used to detect and identify different kinds of fault scenarios.

6.6 Fault Detection and Diagnosis/Identification

Detailed information about fault-induced changes in the controlled system is normally required for a reconfigurable controller to be able to perform its tasks efficiently. A fault detection and identification (FDI) module, in a modern context will thus consist of algorithms that monitors system performance to detect the occurance of faults, and to determine their magnitudes.

Many techniques have been developed for detection and identification of different types of faults. These techniques can be generally grouped into data-based and model-based methods [1]. These two approaches can be further categorized into quantitative and qualitative methods. Figure 6.4 shows an overview of classification of FDI methods. Data-based methods detect faults by testing specific

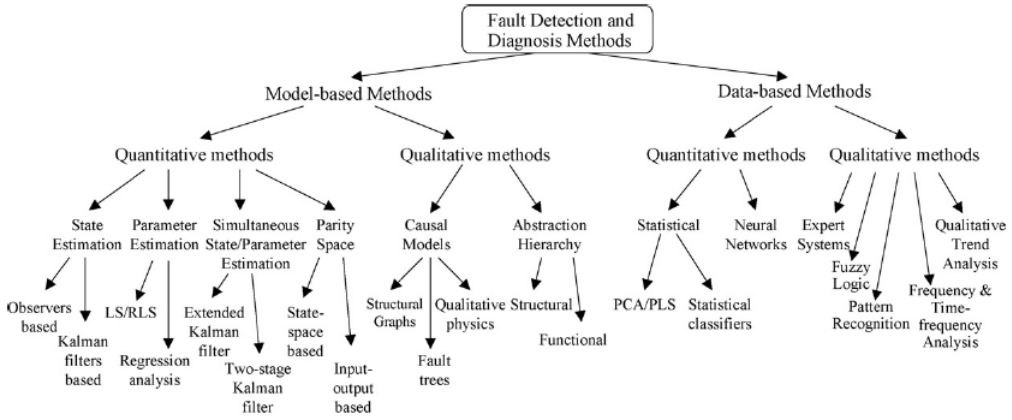


Figure 6.4: Classification of Fault Detection and Identification methods [1]

properties of measurement signals. Spectral analysis and bandpass filters are ex-

amples of several techniques used in data-based methods. Among data-based methods, neural networks (NN) and its advanced variants have become an active area of research due to their learning ability.

Model-based methods on the other hand have wider range of application. Since most of control techniques are model-based, fault tolerant controllers need to be designed based on mathematical model of the system being analyzed. Among quantitative model-based methods, state estimation schemes have been identified as most suitable for fault detection since they are inherently fast and cause a short time delay in the real-time decision-making process. However, parameter estimation based schemes become more desirable when the information from the state estimation based algorithms are not detailed enough for subsequent control system reconfiguration. This is because fault-induced changes in parameters or even system model need to be determined in some cases.

A very interesting idea for failure mode parameter estimation will be to employ active/persistent input excitation of dormant degrees of freedom in the controlled system. A scheme of this nature will have the advantage of not disturbing/influencing the main tasks of the automatic control system, and also "hidden" faults will be detected before an active control of the affected degree of freedom is needed. Fault detection and identification is, however, not treated as an active part of this project, and indept analysis of any potential FDI method, is defered to future studies.

Since the reconfigurable control strategy implemented in this project depends on the existence of an FDI system, a simple parameter identification model is adopted from which key parameters defining actuator performance are easily obtained. This type of FDI scheme is presented in [4], where each actuator is modelled as a single input, single output (SISO) system with rate limits and position limits as the key parameters to be identified. Figure 6.5 illustrates the actuator FDI system. Though simple, this FDI system can detect many different faults

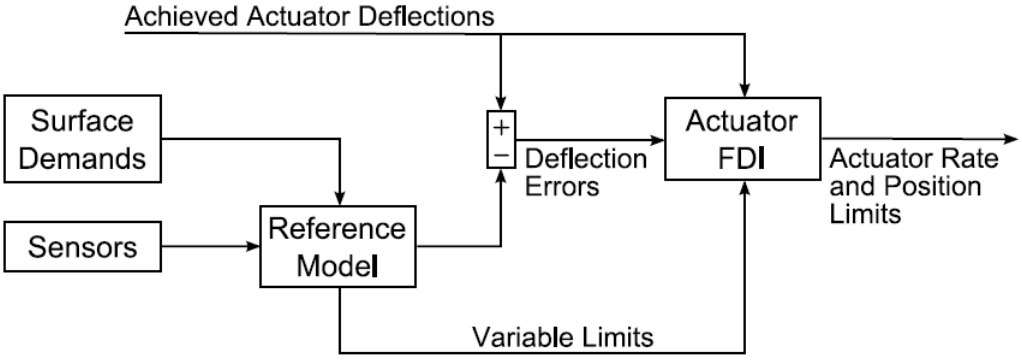


Figure 6.5: Illustration of Actuator FDI [4]

such as:

- Control restrictions caused by loss of hydraulic power or a physical restriction on the surface will be detected as a change in the upper and/or lower limits to new, non-equal values.

- Surface jams caused by total failure of a stepping actuator will be detected as a change of upper and/or lower position limits to new equal values.
- Reduced rate limits due to partial loss of actuation power will be detected as a new upper and/or lower rate limits.
- Surface runaway caused by an error in the signal driving the actuator or an internal malfunction will be initially detected as a change in upper and lower rate limits to the same value (being the rate at which the surface is "running" away). Once the actuator has saturated, the fault will change to the surface jam case.
- Disconnected or free-floating surface caused by physical damage (and possibly total loss of hydraulic power) will be detected as zero upper and lower rate limits (if the signal fed back to the actuator FDI system is surface deflection).

It is worth noting that a floating surface tends to have a greatly reduced aerodynamic effect on the aircraft dynamics. This type of fault could therefore be detected as an aerodynamic fault rather than an actuator fault [4].

Chapter 7

NMPC Autopilots

The classic autopilot designs presented in chapter 6.2 are based on *successive loop closure* guided by a good deal of intuition and experience that assists the selection of the control system structure. For instance, integral action is used to eliminate steady-state error, while standard compensator structures are used to approximate derivative action to stabilize the system. In addition, the effect of parameter variations are reduced by providing inner rate-feedback loops around the process model. Also, frequency-domain and well developed stability analysis tools (such as Bode and Nyquist plots) enable the visualization of how the system dynamics can be modified. However, the design procedure based on the classic one-loop-at-a-time approach become increasingly difficult when more loops are added, and does not guarantee success when the system dynamics are multivariable (that is, when there are multiple inputs, multiple outputs, or multiple feedback loops) [2]. The need for a modern and advanced multivariable control strategy that incorporates the aircraft's dynamics over it's entire flight envelope is well met by the inherent properties of NMPC (see chapter 3).

Stability augmentation systems, including automatic departure-/spin prevention and state-limitations are absolutely necessary strategies that must be implemented in order to achieve satisfactory flying qualities for fighter aircrafts. The autopilots presented earlier in chapter 6.2 will therefore rely on additional limiting strategies (discussed in [36]) for good performance. Emphasis can be placed on the fact that unstable systems can be augmented with stabilizing inner-loops before applying a model predictive control scheme. Practical reasons, including possible sampling rate reduction, are mentioned in [9] as additional advantages for employing a stabilzing inner-loop in an MPC framework. The sampling rate of the MPC scheme can be reduced since the inner-loop will be handling the high bandwidth disturbances and tracking requirements (with its smaller time implementation). It is again important to explore the numerous potentials of NMPC, to examine both stability and dynamic performance capabilities, when applied to systems with highly nonlilear and unstable modes. Moreover, the NMPC framework has inherent limitation strategies (such as constraints and set-point management) that can be explored for the fighter aircraft's stability and state-limitation requirements.

Maintaining the traditional approach in aircraft control system design structure presented in section 6.2 and appendix B, each autopilot can be replaced with its corresponding NMPC algorithm. This means the control variables and inputs will

be kept the same as described for each autopilot in appendix B. Additionally, the performance specification will be formulated in terms of a mathematically precise performance criterion as presented in chapter 3. Individual sampling times and horizons can in this case be employed to reflect the relative computational (or closed loop) speeds required for each autopilot dynamics. However, the extent of coupling that exists between the states of, for intance, the longitudinal equations of motion (see (4.69)-(4.73)), makes the separation into individual (or sub-autopilots) computationally demanding. For later reference, this approach of NMPC autopilot design is termed *design option 1*. Specifically, in design option 1, the longitudinal motion control is divided into an inner (fast) loop for pitch or flight path control and an outer (slow) loop for altitude control. Similarly, the lateral motion control consists of an inner (fast) roll (or turn coordination) controller and an outer (slow) heading controller.

To avoid recomputing the same equations in separate NMPC autopilots, two NMPC autopilots can be conveniently formulated, one controlling the aircraft's longitudinal dynamics while the other controls to the lateral mode. Besides, one of the remarkable properties of NMPC is its ability to handle large multivariable processes in an elegant and effective way. The main limitation of this strategy is that the flexibility of choice of loop rates (and horizon) for altitude h and heading Ψ control, which do not require high loop rates (or short horizon), become limited by their usually faster inner loops (i.e. pitch attitude Θ and roll angle Φ). This can result in aggresive control, but again, adequate choice of weighting matrices and the use of either set-points or constraints on the rate states (e.g. climb rate) are relatively simple ways of handling (otherwise) complicated system behavior in the model predictive control framework [14]. This NMPC autopilot design approach will also be refered to as *design option 2*. Design option 2 will therefore require the implementation of only two NMPC autopilots, whereas design option 1 suggests that four autopilots are implemented.

7.1 NMPC Longitudinal Motion Control

The longitudinal NMPC autopilot has as its main objective to follow time varying set-points in pitch Θ (or flight path γ) and altitude h, and at the same time serve as a longitudinal stability augmentation system. This complicated task can be formulated as a minimization of the following quadratic function:

$$J_{Lo}(x_{Lo}(t_k), u_{Lo}(t_k), t_k) = \sum_{k=0}^{N-1} \left(\|x_{Lo}(t_{k+1}) - r_{x_{Lo}}(t_{k+1})\|^2_{Q_{Lo}} + \|u_{Lo}(t_k) - r_{u_{Lo}}(t_k)\|^2_{R_{Lo}} \right) \tag{7.1}$$

subject to

$$x_{Lo}(t_{k+1}) = f_{Lo}(x_{Lo}(t_k), u_{Lo}(t_k), t_k), \qquad k = 0, ..., N-1 \tag{7.2}$$

$$x_{Lo}(t_0) = x_{Lo}(0), \qquad \text{given} \tag{7.3}$$

$$x_{min} \leq x_{Lo}(t_k) \leq x_{max}, \qquad t_k \in T_d \tag{7.4}$$

$$u_{min} \leq u_{Lo}(t_k) \leq u_{max}, \qquad t_k \in T_d \tag{7.5}$$

where all terms have the same interpretations as presented in chapter 3. Specifically, $x_{Lo} = [h,\ V_T,\ \alpha,\ \gamma,\ Q,\ \delta_e]^T$ represents the longitudinal state vector. All

the states in x_{Lo}, except the elevator actuator state, are included in the objective function (7.1). The corresponding setpoints are therefore $r_{x_{Lo}} = [r_h,\ r_{V_T} = V_{T0},\ r_\alpha,\ r_\gamma = 0,\ r_Q = 0]^T$. The chosen setpoints imply that straight level flight is the desired steady state condition for the aircraft, after a change in altitude r_h and a climb maneuver are accomplished. The last measured total speed V_{T0} is used as the speed setpoint, since a minimum change in speed is desired. r_α can be fixed for specific flight conditions or used as a reconfigurable aircraft stabilty limit (or flight envelope) reference. The control vector $u_{Lo} = [\delta c_e,\ T_c]^T$ consists of elevator deflection δc_e and thrust force $T_c \approx F_T$. The corresponding setpoint vector for the control inputs is $r_{u_{Lo}} = [\delta_{e0},\ T_{c0}]^T$, which represents the last measured values of the respective control effectors. The function f_{Lo} is a collection of the longitudinal equations of motion and elevator actuator dynamics, which take the following continuous form (when *design option 2* is used):

$$f_{Lo} = \begin{cases} \dot{h}(t) = V_T(t)(\cos\alpha(t)\sin(\gamma(t)+\alpha(t)) - \sin\alpha(t)\cos(\gamma(t)+\alpha(t))) \\ \dot{V}_T(t) = \dfrac{1}{\mathrm{m}}(F_T(t)\cos\alpha(t) - \mathcal{D}(\alpha(t),\delta_e(t)) - \mathrm{m}g\sin\gamma(t)) \\ \dot{\gamma}(t) = \dfrac{1}{\mathrm{m}V_T(t)}(F_T(t)\sin\alpha(t) + \mathcal{L}(\alpha(t),\delta_e(t)) - \mathrm{m}g\cos\gamma(t)) \\ \dot{\alpha}(t) = Q(t) - \dot{\gamma}(t) \\ \dot{Q}(t) = \dfrac{m(\alpha(t),\delta_e(t))}{I_y} \\ \dot{\delta}_e(t) = \dfrac{1}{T_e}(\delta c_e(t) - \delta_e(t)) \end{cases} \tag{7.6}$$

All the aircraft variables are explained in chapter 4, and the longitudinal equations are the same as those presented in section 4.4.1. The use of $\gamma = \Theta - \alpha$ instead of θ in the performance index J_{Lo} introduces an implicit control of both pitch attitude and angle of attack, and this proves to be a more effective way of altitude control (for the complex fighter aircraft dynamics examined in this work). Moreover, the detectability (or observability) criteria (3.20) suggests that, since the aircraft's longitudinal static stability highly depends on α, the aircraft's angle of attack needs to be included in the objective function in order to provide a form of artificial pitch stability for the aircraft. In fact, the incorporation of α feedback and limitations are necessary strategies to provide satisfactory flying qualities (see reference [36] for a NASA report on F-16 flight control).

7.2 NMPC Lateral Motion Control

Similar to the longitudinal NMPC autopilot formulation above, the lateral autopilot is designed to follow time varying set-points in roll Φ and heading Ψ. Turn coordination and directional control are also implicit in the NMPC lateral control objective. Lateral stability must also be maintained, but in view of the stability measures taken for the longitudinal mode, the stability augmentation of the lateral mode is less critical (in the flight envelope considered in this work). Moreover, the fighter aircraft configuration under study is statically stable in the lateral channel for a comparably wide range of the aircraft's angle of attack [36]. Nevertheless, an automatic departure-/spin prevention system effect and yaw stability augmentation can be achieved by including yaw-rate (R) feedback and limitations in the form of constraints.

The lateral motion control task can be formulated as a minimization of the following quadratic function:

$$J_{La}(x_{La}(t_k), u_{La}(t_k), t_k) = \sum_{k=0}^{N-1} \left(\|x_{La}(t_{k+1}) - r_{x_{La}}(t_{k+1})\|^2_{Q_{La}} + \|u_{La}(t_k) - r_{u_{La}}(t_k)\|^2_{R_{La}} \right) \tag{7.7}$$

subject to

$$x_{La}(t_{k+1}) = f_{La}(x_{La}(t_k), u_{La}(t_k), t_k), \qquad k = 0, ..., N-1 \tag{7.8}$$

$$x_{La}(t_0) = x_{La}(0), \qquad \text{given} \tag{7.9}$$

$$x_{min} \leq x_{La}(t_k) \leq x_{max}, \qquad t_k \in T_d \tag{7.10}$$

$$u_{min} \leq u_{La}(t_k) \leq u_{max}, \qquad t_k \in T_d \tag{7.11}$$

where all terms have the same interpretations as presented in chapter 3. In addition, $x_{La} = [\beta,\ \Phi,\ \Psi,\ P,\ R,\ \delta_a,\ \delta_r]^T$ represents the lateral state vector. All the states in x_{La}, except actuator states, are included in the objective function (7.7). The corresponding setpoints are therefore $r_{x_{La}} = [r_\beta = 0,\ r_\Phi = 0,\ r_\Psi,\ r_P = 0,\ r_R = 0]^T$. The chosen setpoints imply that wings-level flight is the desired steady state condition for the aircraft, after a change in heading r_Ψ and a turn maneuver are accomplished. The control vector $u_{La} = [\delta c_a,\ \delta c_r]^T$ consists of aileron deflection δc_a and rudder deflection δc_r. The corresponding setpoint vector for the control inputs is $r_{u_{La}} = [\delta_{a0},\ \delta_{r0}]^T$, which represents the last measured values of the respective actuator (or control surface) deflection. The function f_{La} is a collection of the lateral equations of motion and actuator dynamics, which takes the following continuous form (when *design option 2* is used):

$$f_{La} = \begin{cases} \dot{\beta}(t) = P(t)\sin\alpha_0 - R(t)\cos\alpha_0 \\ \qquad\qquad + \dfrac{1}{\mathrm{m}V_{T0}}(\mathcal{C}(\beta(t), \delta_a(t), \delta_r(t)) - F_{T0}\cos\alpha_0 \sin\beta(t) + \mathrm{m}g_2) \\ \dot{P}(t) = \dfrac{I_z}{I_x I_z - I_{xz}^2}\ell(\beta(t), \delta_a(t), \delta_r(t)) + \dfrac{I_{xz}}{I_x I_z - I_{xz}^2} n(\beta(t), \delta_a(t), \delta_r(t)) \\ \dot{R}(t) = \dfrac{I_{xz}}{I_x I_z - I_{xz}^2}\ell(\beta(t), \delta_a(t), \delta_r(t)) + \dfrac{I_x}{I_x I_z - I_{xz}^2} n(\beta(t), \delta_a(t), \delta_r(t)) \\ \dot{\Phi}(t) = P(t) + R(t)\tan\Theta_0 \cos\Phi(t) \\ \dot{\Psi}(t) = \dfrac{R(t)\cos\Phi(t)}{\cos\Theta_0} \\ \dot{\delta}_a(t) = \dfrac{1}{T_a}(\delta c_a(t) - \delta_a(t)) \\ \dot{\delta}_r(t) = \dfrac{1}{T_r}(\delta c_r(t) - \delta_r(t)) \end{cases} \tag{7.12}$$

All the aircraft variables are explained in chapter 4, and the lateral equations are the same as those presented in section 4.4.2. The gravity component g_2 is also the same as in equation (4.85).

In both longitudinal and lateral NMPC autopilots, adequate sampling times and

horizons must be selected to reflect the dominating (or slowest) time constants, and tuned such that the NMPC algorithm captures the correct system dynamics during the prediction model simulations. As indicated by the autopilot formulations, no additional stability measures in the form of terminal costs and constraints were necessary in the NMPC autopilots. The same applies to the use slack variables. It was mentioned in section 3.4.2 that the use of detailed first principles model is advantageous since the performance of the closed loop can be increased significantly without much tuning. This advantage proves to be significant, by reducing the tuning task to only horizon and weight (or penalty) adjustments. Specifics about tuning, reference trajectory generation, and other practical/implementation considerations are covered later in cahpter 8.

Chapter 8

NMPC Autopilot Implementations

8.1 The fighter aircraft test model: F-16

The F-16 fighter aircraft is the test aircraft for the NMPC autopilots designed and studied in this work. The mathematical model used in this work uses data from NASA-Langley wind tunnel tests on a subscale model of an F-16 aircraft [36]. The original aircraft data (referd to as *high fidelity* data) cover a very wide range of angle of attack α ($-20°$ to $90°$), and of sideslip angle β ($-30°$ to $30°$). Due to lack of sufficient pitching moment control for maneuvering at angles of attack beyond $25°$, the data (found in [2]) has been reduced to $-10° \leq \alpha \leq 45°$ range, and the β dependence is also approximated in some cases. The result is refered to as *low fidelity* aerodynamic data.

The F-16 model used as the *plant* in the simulations is a 6DOF completely nonlinear model based on the high fidelity data implementations of [37]. The aircraft model is written in C programming language and includes functions needed to interpolate aerodynamic data from look-up tables. The NMPC autopilot prediction models are also written in C and interfaced with MATLAB simulation environment using the ACADO toolkit (see section 8.3.1).

The aircraft equations of motion used for both plant simulations and control design are the nonlinear models presented earlier in section 4.3 and chapter 7. The specific coefficient equations used to sum the contributions to a given force or moment are based on the formulations presented in section 4.2 and can be found in appendix A. For simplicity and reduction of the required computational power, the low fidelity aerodynamic data tables were used in the NMPC autopilot prediction model calculations. It should, however, be noted that the approximate model constructed from the low fidelity data exhibits steady state flight trim conditions, and corresponding dynamic modes, that are close to those of a full high fidelity data model [2].

The geometric properties, dimensions, and limitations of the F-16 model are shown in tables 8.1, 8.2 and 8.3. The F-16 model data are given in English units, rather than SI units. English units are, therefore, used throughout this thesis and the implementation code-files to maintain consistency and easy refer-

ence to the data source. In additon to the control surfaces listed in table 8.2, the

Mass Properties		
	Weight(lbs):	$W = 20500$
	Moment of Inertia (slug-ft^2):	$J_{xx} = 9496$
		$J_{yy} = 55814$
		$J_{zz} = 63100$
		$J_{xz} = 982$
Wing Dimensions		
	Span:	$b = 30$ft
	Area:	$S = 300$ft^2
	mean aerodynamic chord:	$\bar{c} = 11.32$ft
Reference CG Location		$X_{CG} = 0.35\bar{c}$

Table 8.1: F-16 Geometric Properties

Controls	Deflection limit	Rate limit	Time constant
Elevator	$\pm 25.0°$	$60°/s$	0.0495 s lag
Aileron	$\pm 21.5°$	$80°/s$	0.0495 s lag
Rudder	$\pm 30.0°$	$120°/s$	0.0495 s lag

Table 8.2: F-16 Control Surface Actuator Models

Variable	Units	Min	Max
Altitude	ft	5000	40000
Angle of attack	deg	-10	45
Sideslip	deg	-30	30
Velocity	ft/s	300	900
Thrust	lbs	1000	19000

Table 8.3: A summary of the operational range for which the F-16 model is valid

F-16 has a leading-edge flap δ_{LEF} that is automatically controlled as a function of angle of attack α, static pressure p_s and dynamic pressure $\bar{q}$, according to the following transfer function [36]:

$$\delta_{LEF} = 1.38\frac{2s + 7.25}{s + 7.25}\alpha - 9.05\frac{\bar{q}}{p_s} + 1.45$$

The above automatic control strategy using the leading edge flap allows the aircraft to fly at high angles of attack. Since control at high angles of attack and the study of stall/post-stall characteristics are not considered in this work, the leading-edge flap control transfer function was not included in the internal model of the NMPC autopilot implementations. Further extension of the flight envelope, including angles of attack beyond 45°, will require the implementation of the δ_{LEF} transfer function.

All the actuators of the control surfaces are modeled as first order dynamics (see equation (4.124)) with time constants (which specify the bandwidths), deflection

limits and rate limits shown in table 8.2. A positive deflection of the control surfaces results in a decrease in the body-axis rates. Specifically, a positive aileron deflection gives a decrease in the roll rate, and requires that the right aileron deflect downwards and the left aileron deflect upwards. A positive elevator deflection causes a decrease in pitch rate, implying that the elevator is deflected downwards. A decrease in yaw rate will be achieved by a positive rudder deflection, which requires that the rudder is deflected to the right. A visual representation of the positive orientations for the control surfaces can be found in figure 6.3 in section 6.4.

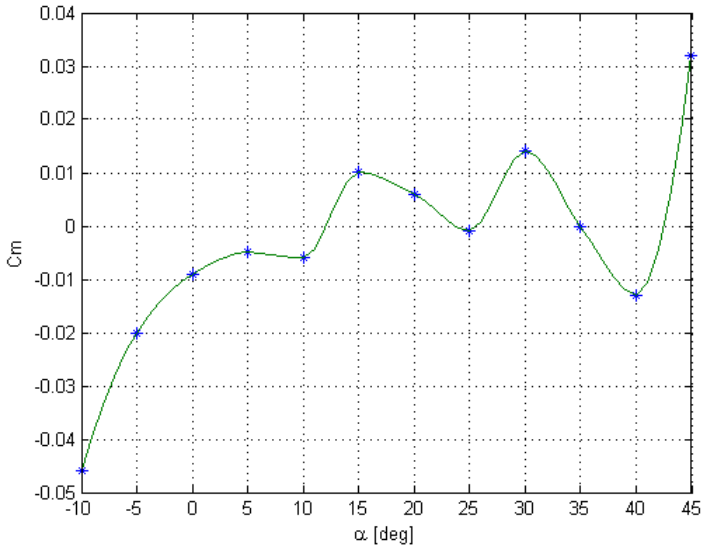
The F-16 is powered by an after-burning turbofan jet engine, which has a model that takes into account both gearing and engine power level lag. The thrust response is rather simplified by a first order dynamic lag model, where the lag time constant is a function of the current engine power level and the commanded power. Power level control plays an important role in any complete automatic flight control system, but in this work the control surfaces are considered as the main control effectors. The engine thrust force is therefore kept almost constant in all the test simulations presented later in this chapter.

8.2 Aircraft Stability verifications/study

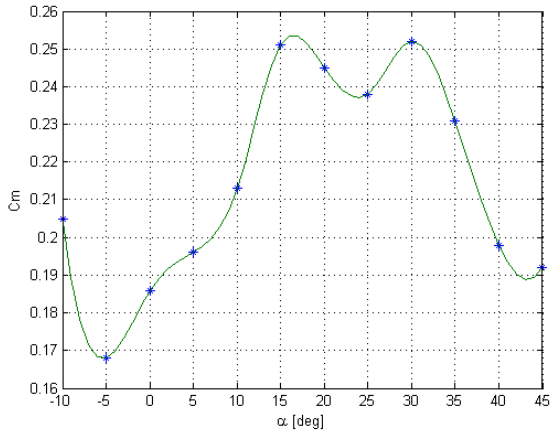
The F-16 is an inherently unstable aircraft. With the center of gravity (CG) at the reference location of $0.35\bar{c}$ and in steady wings-level straight flight, both stable and unstable equilibrium (balanced flight or trim conditions) can be obtained. Some of the trimmed conditions are achieved with saturated elevator control. In such conditions the aircraft may be balanced but unstable as well. The graphs of the pitching-moment coefficients about the CG versus angle of attack, as well as the aerodynamic data lookup-table from which they were generated are shown in figure 8.1 and table 8.4 respectively. The aerodynamic data represent body-axes dimensionless coefficients of the F-16 model at arbitrary values of the independent state variables (i.e. variables C_m does not depend on). The angle of attack range of the lookup-table 8.4 is from $-10°$ to $45°$, divided into $5°$ increments, and the elevator deflection is fixed at $-25°$, $-10°$, $0°$, $10°$ and $25°$ as indicated by the table variables. The interpolation algorithm used with the aerodynamic data table interpolates linearly between the data points.

$C_{m_{-25}}$	0.205,	0.168,	0.186,	0.196,	0.213,	0.251,
	0.245,	0.238,	0.252,	0.231,	0.198,	0.192
$C_{m_{-10}}$	0.081,	0.077,	0.107,	0.110,	0.110,	0.141,
	0.127,	0.119,	0.133,	0.108,	0.081,	0.093
C_{m_0}	-0.046,	-0.020,	-0.009,	-0.005,	-0.006,	0.010,
	0.006,	-0.001,	0.014,	0.000,	-0.013,	0.032
$C_{m_{10}}$	-0.174,	-0.145,	-0.121,	-0.127,	-0.129,	-0.102,
	-0.097,	-0.113,	-0.087,	-0.084,	-0.069,	-0.006
$C_{m_{25}}$	-0.259,	-0.202,	-0.184,	-0.193,	-0.199,	-0.150,
	-0.160,	-0.167,	-0.104,	-0.076,	-0.041,	-0.005

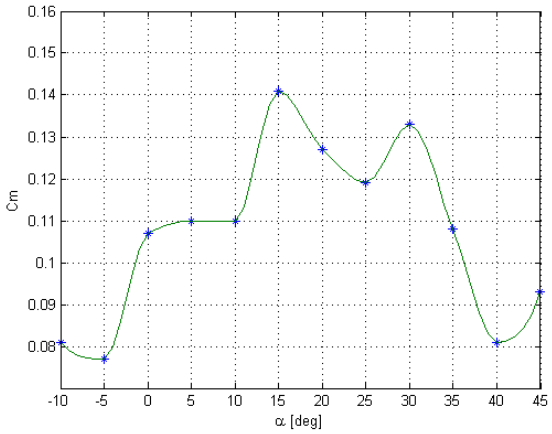
Table 8.4: F-16 C_m lookup-table for fixed elevator angles



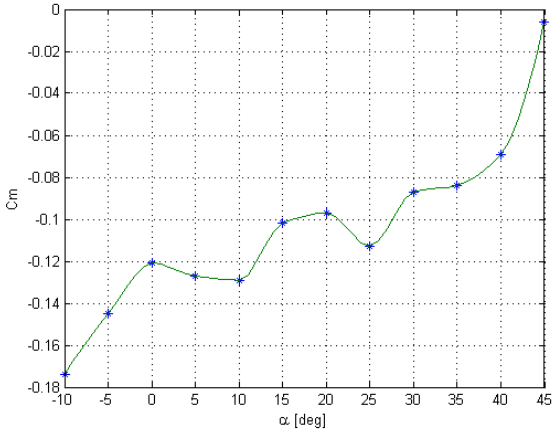
(a) C_{m_α} for $\delta_e = 0°$



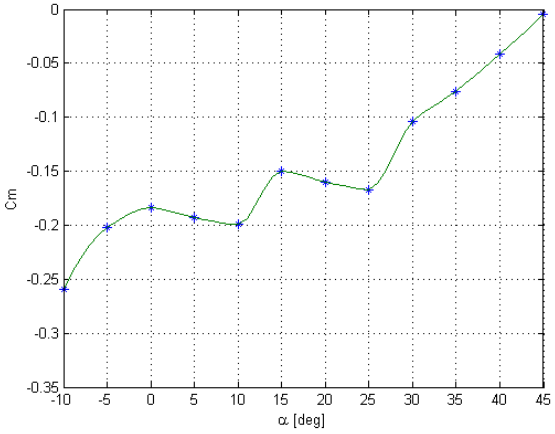
(b) C_{m_α} for $\delta_e = -25°$



(c) C_{m_α} for $\delta_e = -10°$



(d) C_{m_α} for $\delta_e = 10°$



(e) C_{m_α} for $\delta_e = 25°$

Figure 8.1: F-16 pitch moment coefficients for fixed elevator angles δ_e

From the pitching moment coefficient curves (figure 8.1), it can be observed that $\alpha = 15°$ is dominant as a transition point between static unstable and stable flight conditions. This point and the angle of attack range of $-5°$ and $25°$ were, therefore, chosen for further examination. The choice of the mentioned α-range is also because it reflects similar stability behavior as that desired for the tests to be performed in this project. To verify the stability behavior of the aircraft in the range under study the F-16 was first trimmed for steady-wing flight at $\alpha = 12.5°$ and then at $\alpha = 20°$. The trim routines used for this task trims the aircraft to desired altitude and velocity, while all other state-parameters are varied until level flight is achieved. The trim values for *negative* pitch stiffness (unstable α-range) verification test are shown in table 8.5.

thrust	3067.2634 lb
elevator	-3.9973 deg
aileron	0 deg
rudder	0 deg
alpha	12.4412 deg
velocity	350 ft/s
altitude	20000 ft

Table 8.5: Trim values for negative pitch stiffness test

8.2.1 Negative pitch stiffness test

During straight wing-level flight the F-16 was disturbed from its equilibrium attitude by increasing the angle of attack by 2° (for 5s) as shown in figure 8.2a. As expected, the angle of attack α (see figure 8.2d) continues to increase (rotating the aircraft further from its trimmed attitude) until it enters the stable α-range ($\alpha > 15°$) where the aircraft experiences a restoring moment. However, it can be observed in the α, velocity and altitude plots of figure 8.2 that the F-16 is unable to return to its initial equilibrium condition.

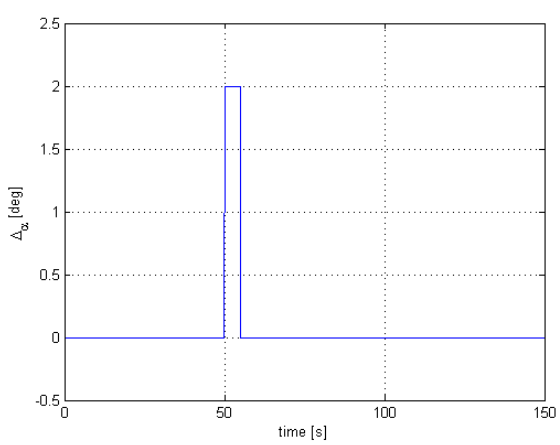
8.2.2 Positive pitch stiffness test

The trim values for positive pitch stiffness (stable α-range) verification test are shown in table 8.6. The α disturbance test was repeated for the angle of attack

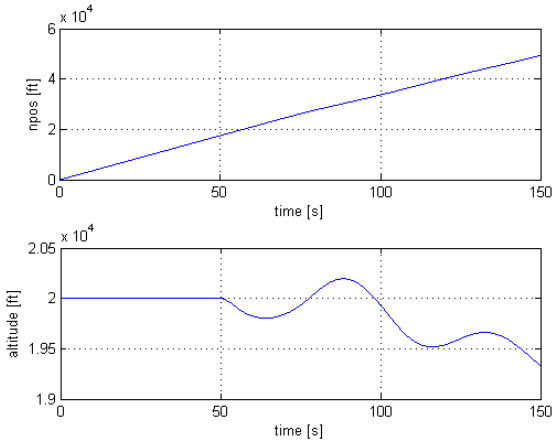
thrust	5247.322 lb
elevator	-6.0538 deg
aileron	0 deg
rudder	0 deg
alpha	20.2424 deg
velocity	300 ft/s
altitude	25000 ft

Table 8.6: Trim values for positive pitch stiffness test

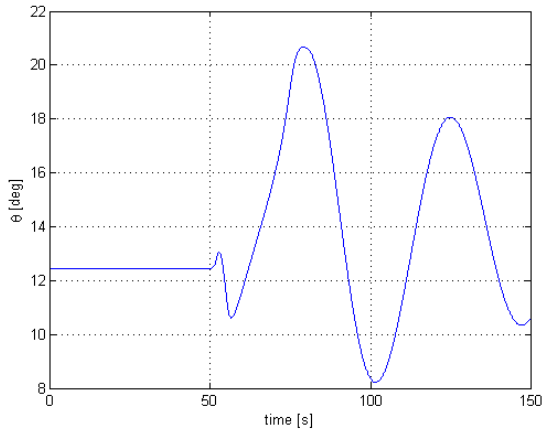
range which has positive pitch stiffness (statically stable). When the angle of attack disturbance was introduced, the aircraft immediately responded with a counter moment in the attempt of reducing the angle of attack to its equilibrium



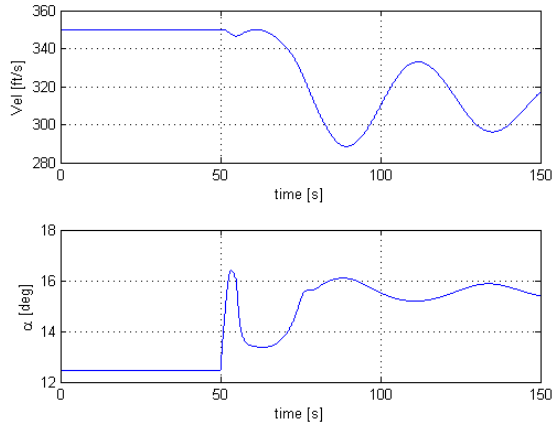
(a) Constant angle of attack disturbance



(b) North position and Altitude



(c) Pitch attitude response



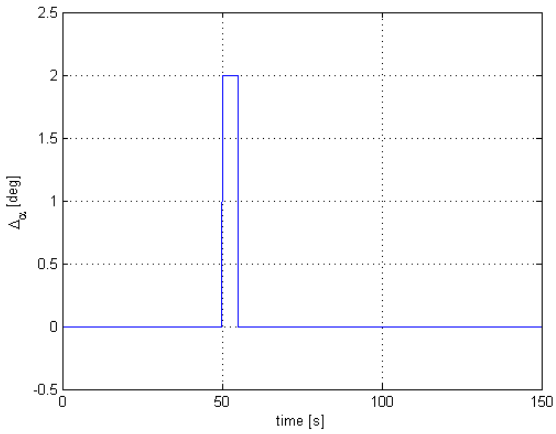
(d) Velocity and Angle of attack

Figure 8.2: F-16 *negative* pitch stiffness: static instability

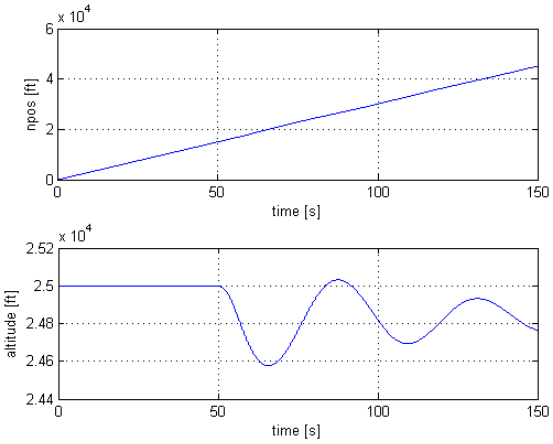
value. The α, velocity and pitch attitude plots in figure 8.3 show that the aircraft oscillates towards its initial equilibrium condition, as expected.

The preceding illustrations using the aerodynamic data show the demading control challenges at hand. Similar analysis can be performed using aerodynamic coefficient data for roll and yaw stiffness. However, as explained in chapter 5, the results of greatest importance for static stability are those assosiated with the longitudinal analysis. Analysis done in [36] show that the F-16 configuration under study is statically stable in the lateral channel for a comparably wide range of the aircraft's angle of attack. Maintaining lateral stability must also be included as a control objective, but again, the stability augmentation of the lateral mode is less critical (in the flight envelope considered in this work), when compared with the stability measures that are required for the longitudinal mode.

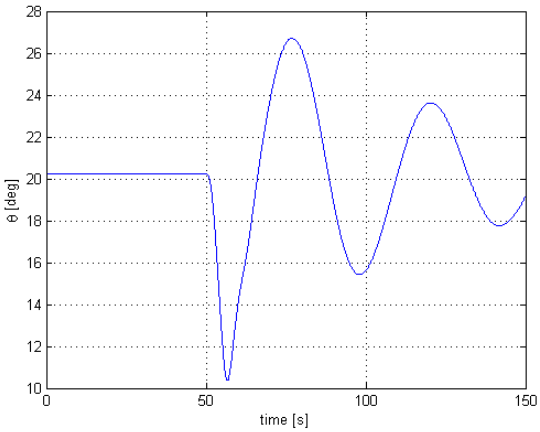
Preliminary study and discussion of the use of NMPC as both a stability and control augmentation system for the longitudinal dynamics are presented in following sections.



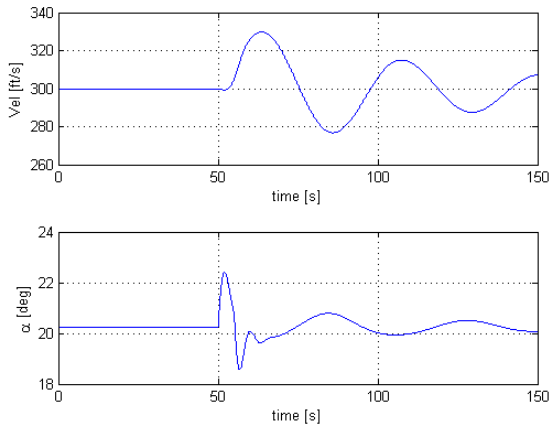
(a) Constant angle of attack disturbance



(b) North position and Altitude



(c) Pitch attitude response



(d) Velocity and Angle of attack

Figure 8.3: F-16 *positive* pitch stiffness: statically stable mode

8.3 NMPC and unstable flight mode control study

The longitudinal dynamics of the F-16 aircraft provides a good platform for studying the issues regarding stability, computational feasibility and performance of NMPC. The simulations and test results presented in the following sections cover important aspects of NMPC used in this work. Practical tuning tradeoffs and choice of constraints and set points or reference trajectories are also examined. In addition, particular attention is paid to the relationship between control input saturation, actuator bandwidth and controllability.

The optimization package used in solving the NMPC problems formulated in this work is ACADO Toolkit. Section 8.3.1 highlights the key features of ACADO and the optimizer parameters used in this work.

8.3.1 NMPC optimization solver: ACADO for MATLAB

ACADO Toolkit is an algorithm collection for automatic control and dynamic optimization, implemented as self-contained C++ code. ACADO provides a general framework for using a veriety of algorithms for direct optimal control, including Nonlinear Optimal Control, Multi-Objective Optimal Control, State and Parameter Estimation, Model Predictive Control, and Code Generation for Fast NMPC. A basic and key feature of ACADO is to provide solution to standard optimal control problems, which typically consist of a dynamic system with differential and/or algebraic states and control objectives. Several types of constraints, such as control and state bounds, terminal constraints, general nonlinear path constraints, and periodic boundary conditions can also be implemented in the ACADO software environment.

ACADO provides a MATLAB interface, which makes the ACADO algorithms and intergrators accessible from MATLAB. This feature is very convenient when MATLAB/Simulink is used as the main simulation environment as done in this project. The NMPC autopilots designed in this project were implemented in two parts. The first consisting of the optimization problem, composed in MATLAB using MATLAB syntax, and the second being the NMPC prediction model implemented as a C-code black-box. An alternative is to write C/C++ code for both parts as a MEX-function and compile it using MATLAB's built-in MEX-compiler. This flexibility provided by ACADO makes the coding of the nonlinear F-16 model, including aerodynamic coefficient calculations and table-lookup/interpolation routines, become a pleasant task.

The autopilot optimization problems are solved using ACADO's multiple-shooting SQP-type method combined with Runge-Kutta integrator (of order 4/5) for the state integration. Multiple-shooting is considered as a discretization-type parameter option in ACADO. In addition, the KKT [1] tolerance which is used for the convergence criterion of the SQP algorithm is set to $1e^{-6}$ (default value in ACADO). The default value of the maximum number of iterations (1000) is also maintained for all simulations and tests. The maximum number of iterations offers a means of specifying a trade off between result accuracy and computational time. Smaller values will reduce the execution time, but the result could be far from optimal. ACADO offers several other algorithmic and numerical optimizer options in the form of simple parameter settings. These settings are selected based on the background knowledge acquired from NMPC theory (see chapter 3) and numerical optimization. However, it is important to monitor the resulting trajectories and tune the optimizer paramters to suit the problem under study. Further details about the ACADO Toolkit can be found in [38], [39], and [40].

Scaling is also an important factor to consider in order to avoid numerical problems in the optimization problem solver. Numerical problems can arise as a result of the optimization problem becoming ill-conditioned, due to poor scaling. The variables in the optimization problem must be scaled such that the differences in tracking/control errors resulting during maneuvers are not too large. However, since the changes in the time-varing references considered in this work are not very large, and considering the autopilot design structure employed (i.e. separat-

[1] KKT refers to Karush-Kuhn-Tucker first order (necessary) optimality conditions.

ing the aircraft model into relatively well-conditioned sub-models, w.r.t. scaling), no prescaling of variables was necessary.

8.3.2 Dynamic Performace and Stability

A pitch attitude-hold autopilot is designed and examined as the main objective of the NMPC and unstable flight mode study. Each test is initialized with the aircraft at trimmed flight state, and perturbations are introduced to excite unstable modes and nonlinear dynamics in the closed loop control system. The trim routine implemented trims the aircraft to desired altitude and velocity, while all other state-parameters are varied until level flight is achieved. The trim values used as initial flight conditions in this study are shown in table 8.7.

Pitch angle Θ, pitch rate Q, and angle of attack α are the aircraft states consid-

thrust	2168.7066 lb
elevator	-2.7656 deg
aileron	0 deg
rudder	0 deg
alpha	5.5311 deg
velocity	500 ft/s
altitude	20000 ft

Table 8.7: Trim values for performance tests

ered through the implementation of their corresponding fully nonlinear equations of motion (4.71)-(4.73). The elevator angle δ_e (in degrees) is the main and only active/varying input, assuming a constant speed approximation (that is, a dedicated thrust control strategy is assumed). To incorporate actuator dynamics into the NMPC scheme, a first order actuator model with 20.2Hz bandwidth was implemented. A step response and Bode plot are shown in figure 8.4. The elevator angle is limited to $\pm 15^\circ$, and the elevator deflection rate is limited to $\pm 60^\circ/\text{s}$.

Unlike passenger aircrafts, which usually have pitch limits for passenger comfort, the F-16 pitch angle in this study is left unconstrained. Limits of -10° and $+45^\circ$ are, however, observed as a bound on angles of attack for which the F-16 model is valid.

Figure 8.5 shows the response of step changes in pitch set-point, governed by reference trajectories taking the form of smooth curves generated using the *pchip* [2] function in MATLAB. The *pchip* function is chosen over *spline* [3], since comparably less oscillation and no overshoots are obtained with *pchip*. The reference trajectory for a step change in pitch attitude from 0 to 10° is illustrated in figure 8.6.

The sampling interval used to obtain the results in figure 8.5 was $T_s = 0.05$s

[2] Piecewise Cubic Hermite Interpolating Polynomial. *pchip*(X,Y) provides the piecewise polynomial form of a certain shape-preserving piecewise cubic Hermite interpolant, to the values Y at the sites X.

[3] Cubic spline data interpolation. *spline*(X,Y) provides the piecewise polynomial form of the cubic spline interpolant to the data values Y at the data sites X.

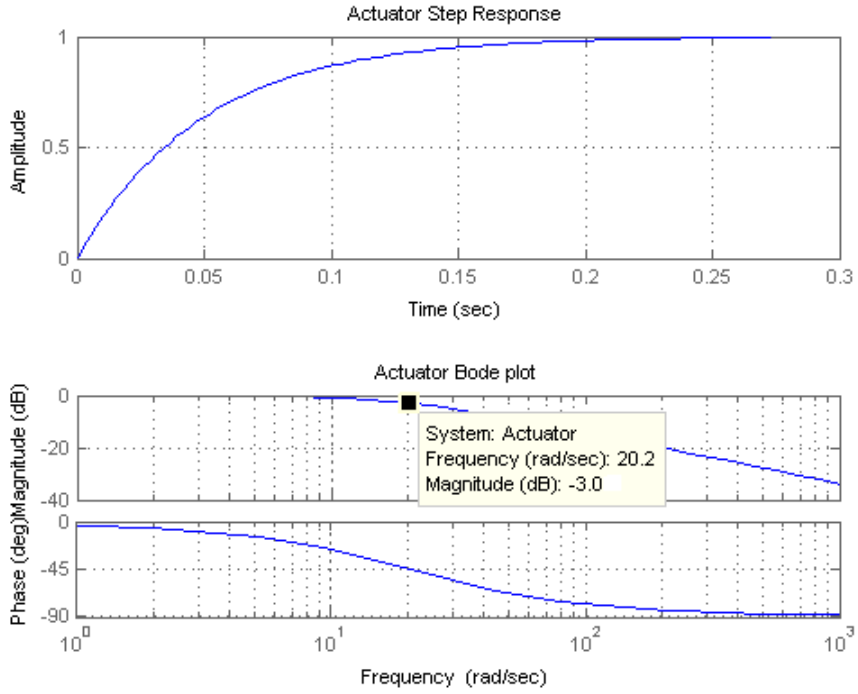
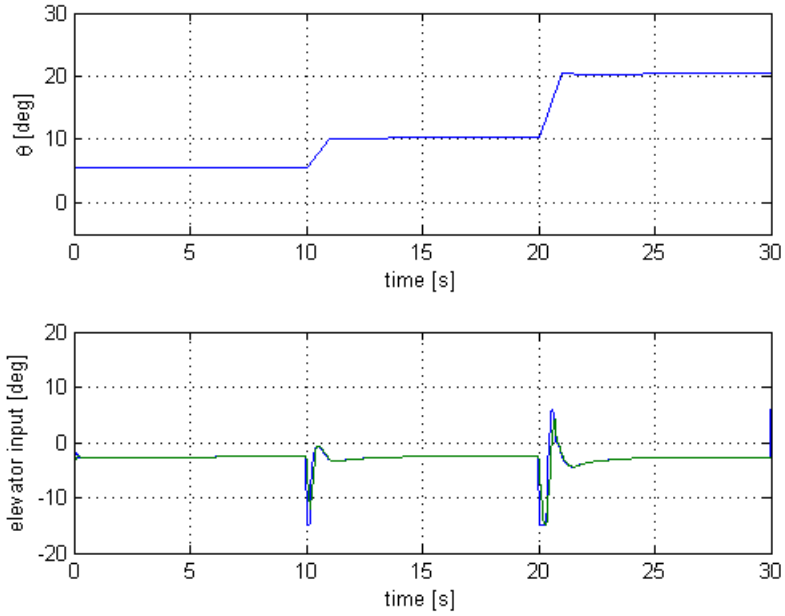


Figure 8.4: Control surface actuator system dynamics

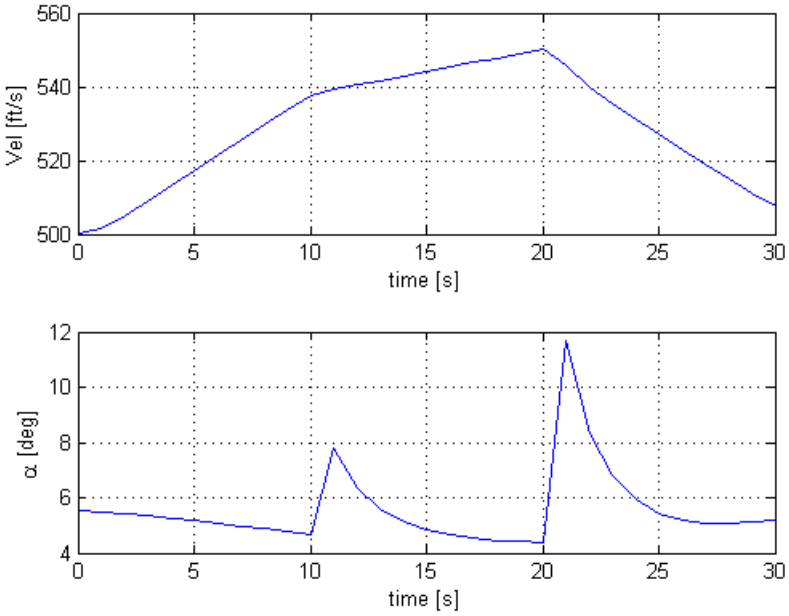
corresponding to 20Hz, and the prediction horizon and control horizon were kept equal as $H_p = 10$ (0.5s). The tracking errors were penalized over the whole prediction horizon with weights $diag([1000, 10, 1])$ corresponding to the state-vector $[\Theta,\ \alpha,\ Q]^T$, and 0.001 as the weight on δ_e.

It can be observed in figure 8.5 that none of the constraints were active between time points from 0s to 10s. The behavior of the resulting controller (in that time interval) can be easily compared to that of its linear time-invariant approximation. However, a change demand for the pitch angle and other manoeuver specifications, result in some constraints (particularly, elevator limits) becoming active, making the control action also nonlinear. The total speed profile (Vel [ft/s]) in figure 8.5b shows inaccuracies introduced into the prediction model when constant speed approximation is assumed. Implementing the speed state V_T equation, including thrust control T_c, will therefore be an appropriate alternative, especially for autonomous flight control strategies.

A strategy for including angle of attack in the overall control objective is necessary, for both stability and dynamic performance enhancement. The main reason is the direct dependence of the aircraft's pitching moment on angle of attack, and most importantly to correctly utilize the nonlinear relation between angle of attack and pitch moment coefficients. These measures are appropriate to enhance stability and performance as discussed in section 3.3. Moreover, it is desirable as an NMPC strategy to influence controllable states through minimization of the state-errors defined in the cost function of the optimization problem (see equation 3.20 in section 3.3).



(a) Pitch attitude and
elevator response



(b) Velocity and
Angle of attack

Figure 8.5: NMPC pitch-attitude hold autopilot response

Predictive pitch control without correct information about the angle of attack at several operating points in the flight envelope, results in satisfactory control only within stable α ranges for obvious reasons. Wrong predictions that result in an increase of the aircraft's angle of attack within an unstable range naturally drive the pitch angle away from its desired set point, inducing a disturbance which

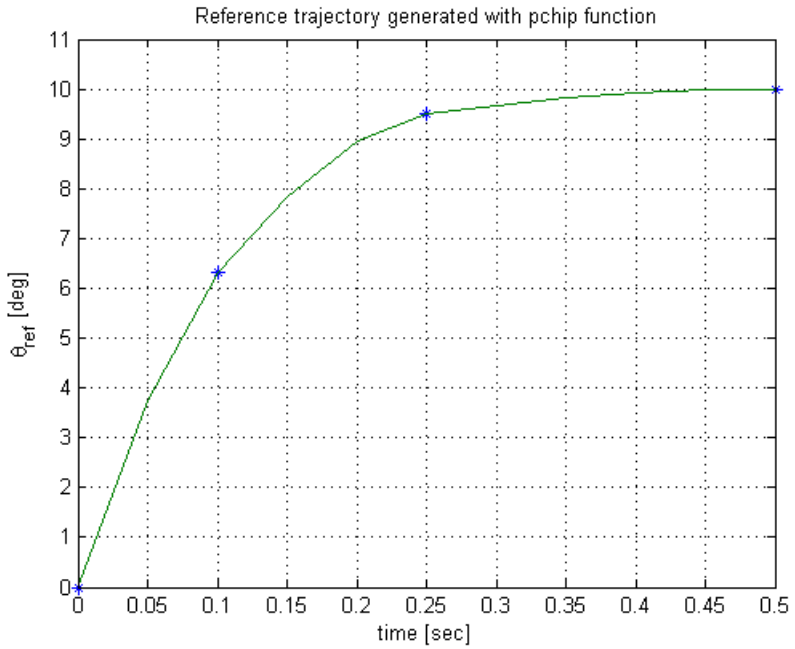


Figure 8.6: Pitch control reference trajectory

again needs compensation through controls. In worst cases the predictive control can end up destabilizing a well-trimmed flight condition.

The NMPC implementation tool (ACADO) used in this study facilitates the linking of C/C++ models to the NMPC formulation, making it possible to include interpolation routines for reading pitch moment coefficients based on angle of attack predictions and their corresponding elevator deflections. Other important parameters such as dynamic pressure can be allowed to vary to enhance the search of feasible solutions and optimal input predictions in the ACADO environment.

Observations from the NMPC pitch autopilot simulations confirm the fact that complicated behaviors can be obtained by relatively simple specifications (like constraints), without the need to introduce mode-switching logic explicitly [4].

The flexibility achieved through constraint specifications should, however, not be seen as a substitute for set-points. In the pitch control study, it was observed that introducing set points for angle of attack (as a secondary[5] objective) by gradually guiding the aircraft into stable α regions is a means of utilizing the flexibility of NMPC to enhance stability and robustness to disturbances. Operating in stable α range also comes with a drawback of reduced maneuverability, indicating the need of making trade-offs between stability and dynamic performance. Simula-

[4] The difference between the MPC approach (declarative programming) and mode-switching logic (procedural programming) is stated in [14] as follows:
Mode-switching logic specifies how the controller is to achieve the required behavior, whereas the use of constraints specifies what behavior it must achieve.

[5] Apart from the *primary* state, for instance Θ in a pitch-hold autopilot, all the other states in the cost function constitute the *secondary* objective. They are either allowed to be driven freely by any cahnge demand in the primary objective (i.e. weight/penalty set to zero) or given lower priority-weights (which are possibly state-dependent).

tion tests show that using angle of attack references as a stable mode seeking control strategy can increase the need of control efforts in tracking changes in the pitch reference trajectory, as stable α ranges increase the restoring tendencies [6] of the aircraft, and hence resisting control efforts. On the other hand maintaining a commanded pitch in an unstable α range demands relatively larger control efforts, especially in the presence of disturbances (such as wind gusts or turbulence).

8.3.3 NMPC Stability and Infeasibility issues

The only reason for considering NMPC stabilty issues in this work is the rather short horizons selected in the NMPC autopilot implementations. Considering the actuator time constant of 0.0495s in the pitch control loop, a horizon of 0.5s gives the optimizer very limited perspective of the aircraft's response. The tendency of generating control trajectories that are far from optimal becomes high, and any numerical problems that arise are very unlikely to be subdued within the short horizon. The risk of the optimization problem solver terminating with local solutions is also high.

When very demanding maneuvers are required, any wrong predictions easily result in the aircraft states rapidly departing from their desired reference trajectories and eventually voilating the F-16 model validity boundary constraints (beyond which the F-16 model is invalid). The model validity constraints in table 8.3 are rather 'hard' in nature, and allowing them to be voilated due to infeasibility issues, rather introduces numerical problems, that in worst cases stop the simulation.

Finding parameters that satisfy all the stability conditions of the general stabilizing NMPC design approach (see section 3.3.3), including a well-determined finite horizon, proved to be an extensive and time consuming task for the complete F-16 model. An alternative was to employ stability measures according to the stability preserving NMPC formulations in section 3.3.2, but due to the time-varying references intended to be used in this work, further extensions will be necessary. However, simple choices such as using a quadratic objective function, which has weighting matrices $Q = diag([1000, 10, 1])$ (for the 'runing' cost), $R = 0.001$ (for the elevator error), $P = diag([1000, 100, 10])$ (for the terminal cost), and no terminal constraint (i.e. $\Omega = \mathbb{R}^n$), tend to keep the system within its validity boundaries. Similar results were also ovbserved when the implementation strategy (3.50) in section 3.3.4 was used.

A rather interesting observation was later made when the complete NMPC autopilots for the longitudinal dynamics was implemented. The altitude autopilot which was implemented with a much longer horizon tend to generate well damped flight path reference trajectories that effectively keep the faster longitudinal state dynamics in check. In addition, introducing extra secondary objectives and constraints (such as climb rate for the longitudinal channel, and yaw rate and roll rate for the lateral channel) resulted in the generation of very reliable control trajectories. Because of these reasons, terminal costs and constraints were not

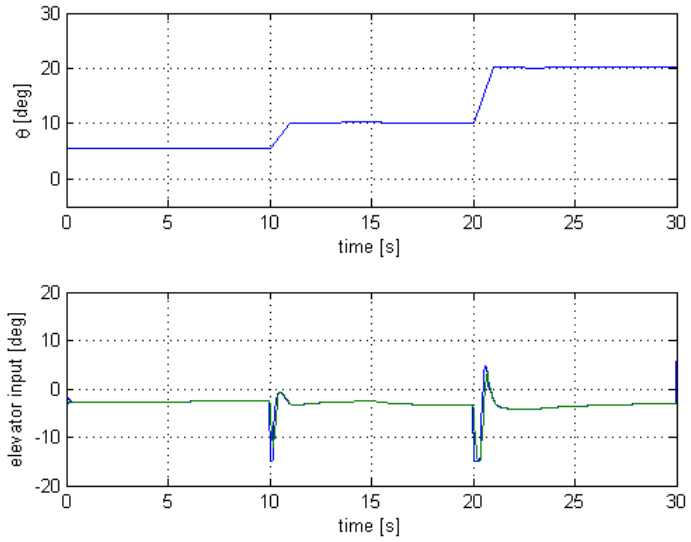
[6] Refer to static stability discussions in chapter 5.

necessary in the final NMPC autopilot implementations presented in section 8.4.

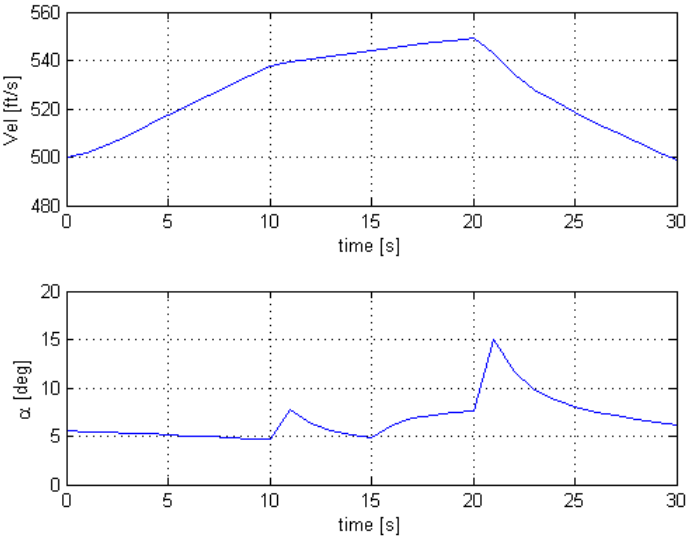
8.3.4 Disturbance rejection and tuning

Figure 8.7 shows the responses when wind gusts appear as a disturbance on the aircraft's angle of attack, causing a 2° increase and lasting for over 10s. The disturbance affected mainly unstable α range of the aircraft, but it can be observed from the plots that the resulting pitch control is very close to the results obtained for the disturbance-free simulation shown in figure 8.5. An obvious difference is noticed in the angle of attack profiles in figures 8.5b and 8.7b, caused by the simulated wind gust. A closer look at the elevator actions also confirms the expected increase in control efforts needed to accomplish the desired maneuver satisfactorily.

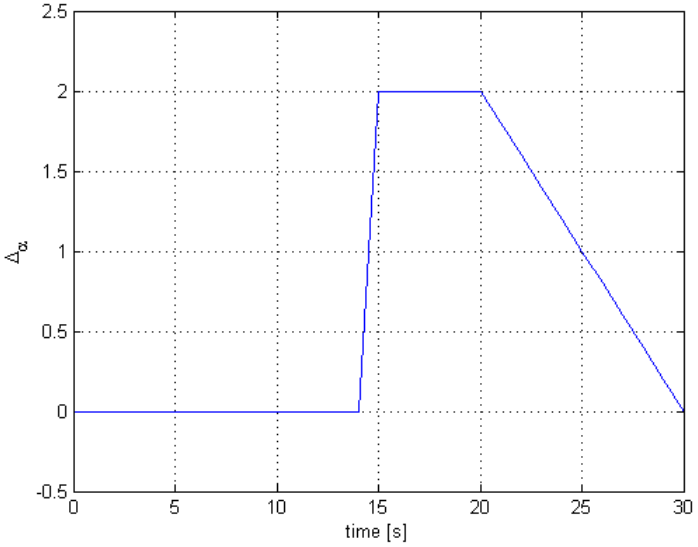
Introducing secondary objectives (for instance, specifying set points for pitch rate Q and α), may require changing tracking error weights during a maneuver, usually in favor of the primary objective when appropriate. The reconfigurable nature of NMPC comes in handy in this kind of situations. A good approach is to make the weights state-dependent, allowing the secondary objective states to vary freely within their limits until the primary error has been reduced to some small value. State-dependent weighting, in addition to a good understanding of the behavior of relative weighting, makes the NMPC tuning task much easier and relaxed. A typical (but not general) behavior is noticed when the primary objective (pitch) tracking error dominates the secondary objective (α and Q) errors during most transients, so that the secondary states depart from their set-points in order to allow the pitch error to be reduced. This behavior is a direct consequence of the relative sizes of the numerical values of the errors that arise. That is, the (weighted) pitch error becomes sufficiently larger than the other errors, making its contribution to the cost function most significant.



(a) Pitch control



(b) Velocity and α



(c) α disturbance profile

Figure 8.7: NMPC pitch autopilot response to α disturbances

8.4 NMPC Autopilot Simulation and Test-setup

The NMPC autopilots described in chapter 7 were implemented and tested in MATLAB/Simulink simulation environment. The MATLAB/Simulink modules consist of a Pilot/Control commands unit, a collection of the NMPC Autopilots as an Automatic Control System unit, and the F-16 Nonlinear Plant. Figure 8.8 shows the MATLAB/Simulink simulation setup. The Longitudinal and Lateral

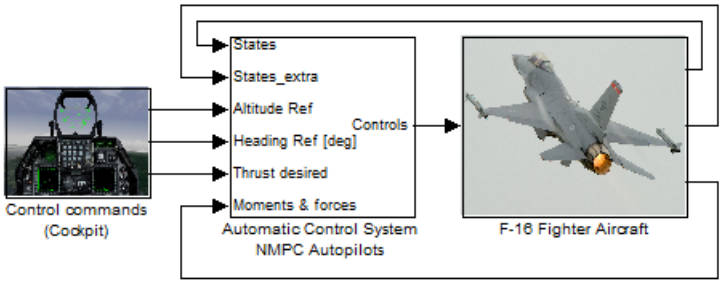


Figure 8.8: Reconfigurable NMPC autopilots, simulation setup

NMPC autopilots and their actuators are masked under the Automatic Control System block, shown in figure 8.9. The simulation details and selected test results

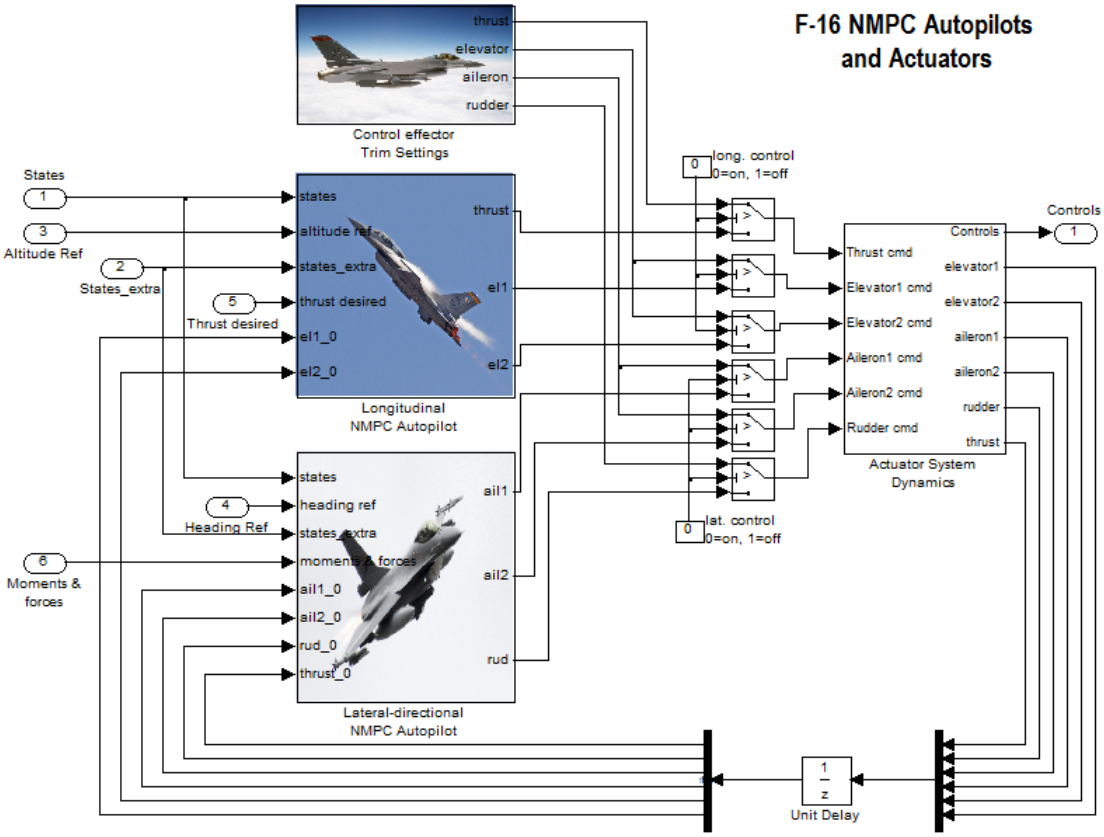


Figure 8.9: F-16 NMPC autopilots and Actuators

of the NMPC autopilots follow in sections 8.4.1 and 8.4.2. Actual F-16 perfor-

mance limit values were used to obtain the autopilot simulation results presented in this work.

8.4.1 Longitudinal motion control

An altitude-hold autopilot which relies on a flight path controller was implemented for maneuvers in the longitudinal channel, based on NMPC autopilot *design option 1* described in chapter 7. The altitude-hold and flight path control autopilots are shown in figure 8.10. The *design option 1* was preferable due to the extra

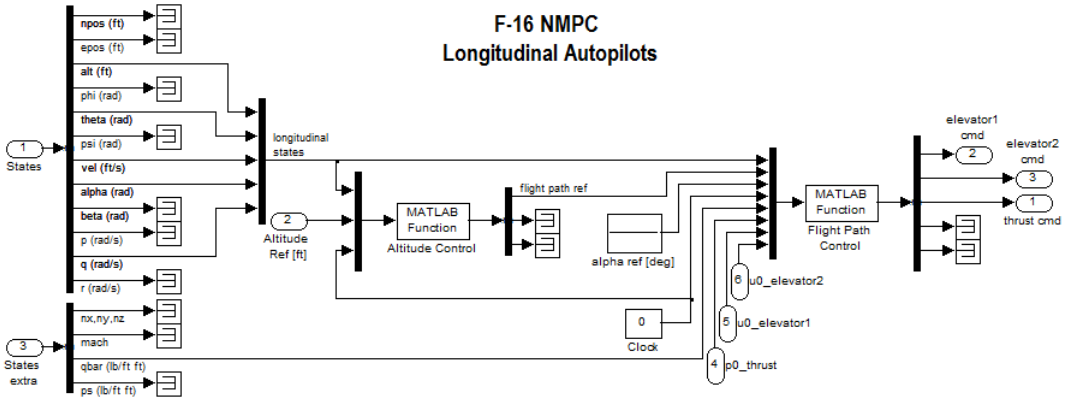


Figure 8.10: NMPC altitude-hold autopilot and flight path control implemented in simulink

flexibility in the choice of sampling time and horizon for the autopilots. In a classical autopilot design sense, the altitude control is seen as the (slower) outer loop while the flight path control is the (faster) inner loop. The NMPC tuning task also becomes less demanding and more intuitive, when *design option 1* is used. To reduce the computational burden of recomputing the same equations (including table-lookup interpolations, needed during optimization iterations) in separate NMPC autopilots, the measured states of the flight path NMPC problem were introduced as time-varying parameters in the altitude NMPC problem formulation (see design summary table 8.9).

In figure 8.10, the 'MATLAB function' blocks contain all the NMPC problem formulations implemented as a combination of MATLAB scripts and C-code (see appendix **??**). The ACADO for MATLAB NMPC formulations and design parameter values are presented in tables 8.8 and 8.9. The continuous time form of the prediction model was implemented directly using ACADO's continuous time formulation syntax. Refer to appendix A and **??** for aerodynamic coefficient calculation (or intermediate states) details that are not shown in the NMPC autopilot design summary tables. The initial values for states and parameters are the measured values from the last/previous sample time. As discussed in the NMPC preliminary studies in section 8.3.3, terminal costs and constraints were not necessary as stability measures in the autopilot designs presented in this section and all the sections that follow.

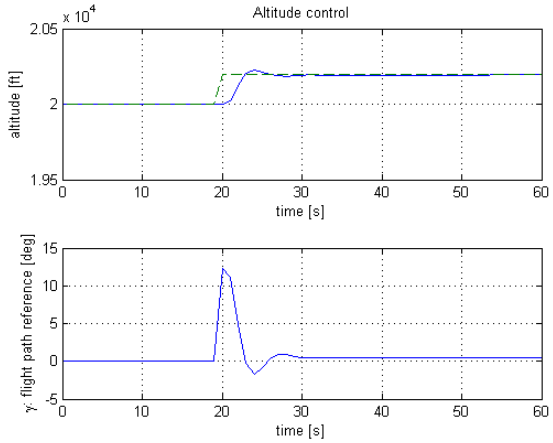
Climb maneuvers and steady flights at cruise altitude were used to test the altitude-hold and flight path autopilots. The results are shown in figure 8.11.

The results show pure longitudinal decoupled motion control. The lateral flight controls were kept at trim settings (see table 8.7) while the longitudinal flight maneuvers were performed. The test results in figure 8.11 can therefore be seen as a calm cruise of the F-16, and the NMPC autopilot outputs quickly settle to their optimal trajectories. The flight tests were initialized at trim-flight conditions, and it is interesting to see how the NMPC strategy quickly finds and maintains optimal trajectories for all states and controls in the $0 - 20$s time intervals of figure 8.11. The seemingly effortless control observed only confirms the reliability and effectiveness of predictive/optimal control when good initial values of the process are available.

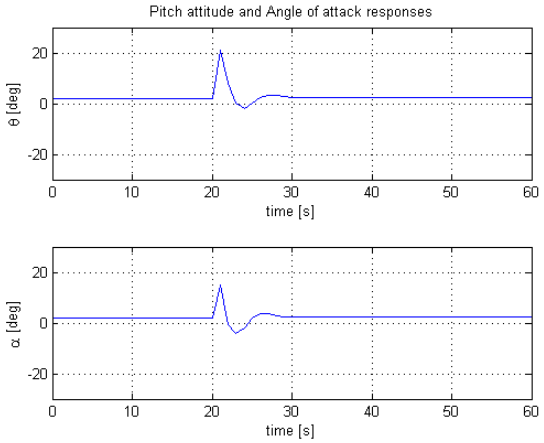
The flight path autopilot has an additional objective of maintaining low angle of attack (close to $4°$) as well as zero pitch rate. The corresponding weighting matrix values (in table 8.8) were therefore selected to ensure that flight path control is always prioritized. The dynamic performance observed in the pitch rate and angle of attack trajectories of figures 8.11b and 8.11c are considered satisfactory for the objectives of this work.

The demanded change in altitude was carried out smoothly, despite how close the angle of attack trajectory follows the pitch attitude of the F-16. Figure 8.11b is an evidence of the challenging task of separating the pitch and angle of attack trajectories to be able to perform rapid climb maneuvers of the F-16. From the pure pitch controller studied in section 8.3.2 it was observed that the angle of attack follows pitch very closely at the begining of each attitude change maneuver, but reduces gradually and takes about 10s to settle close to it's lower reference value (see figure 8.5b). Any climb maneuver that requires an optimal separation of pitch and angle of attack at the begining, followed by zero separation, under 10s is obviously not an easy task (especially in a classical autopilot design sense). Using pitch as the main climb maneuver objective and anlge of attack as a secondary stability enhancing objective, as suggested in section 8.3.2, posed a very extensive tuning task for the complete longitudinal autopilot implementation. Direct flight path control, through simultaneous (optimal) manipulation of both pitch and angle of attack proved to be more appropriate when altitude-hold is the primary objective of the longitudinal control augmentation system for the F-16.

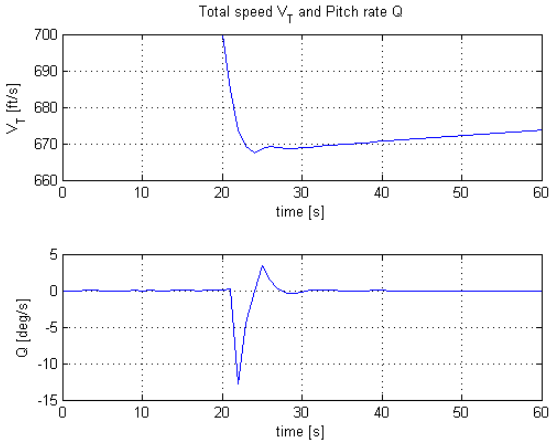
It is also important to note that the 5s horizon chosen for the altitude autopilot was a necessary trade-off between computational speed and dynamic performance of the F-16 (in view of the observed 10s settling time of α). Shorter horizons (less than 2s) result in aggressive control. Note the difference in the flight path command reference trajectories generated in the 5s horizon plot (figure 8.11a) and the 2s horizon plot (figure 8.12a). The slight overshoot observed in the altitude plot 8.11a is also avoided by increasing the horizon.



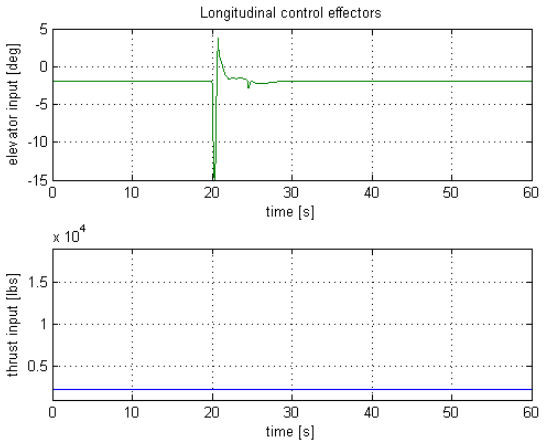
(a) Altitude and flight path control



(b) Pitch and Angle of attack



(c) Total speed and pitch rate



(d) Elevator input and thrust force

Figure 8.11: F-16 NMPC longitudinal motion control ($T_h = 5$s)

Differential states:

γ	flight path
Q	pitch rate
α	angle of attack
V_T	total speed
δ_e	elevator state
$\dot{\delta}_e$	elevator deflection rate

Controls:

δc_e	elevator command
T_c	(F_T) thrust command

Model Parameters:

$\bar{q}$	dynamic pressure

Optimization/prediction parameter values:

$T_s = 0.05$s	sample time
$T_h = 1.0$s	horizon (start: 0.0s)
$N = 20$	intervals in horizon

Diagonal weighting matrix values:

$Q_p(1,1) = 100.0$	h error weight
$Q_p(2,2) = 0.010$	Q error weight
$Q_p(3,3) = 0.100$	α error weight
$Q_p(4,4) = 0.001$	V_T error weight
$R_p(1,1) = 0.001$	δc_e error weight
$R_p(2,2) = 1.000$	T_c error weight

Objective function parameters:

$\{\gamma,\ Q,\ \alpha,\ V_T,\ \delta c_e,\ T_c\}$	states and controls
$\{r_\gamma,\ 0,\ 4^\circ,\ 700\text{ft/s},\ \delta_{e0},\ 2169\text{lb}\}$	setpoints

Differential equation (prediction model):

$$\dot{\gamma}(t) = \frac{1}{\mathrm{m}V_T(t)}(F_T(t)\sin\alpha(t) + \mathcal{L}(\alpha(t),\delta_e(t)) - \mathrm{m}g\cos\gamma(t))$$
$$\dot{Q}(t) = m(\alpha(t),\delta_e(t))/I_y$$
$$\dot{\alpha}(t) = Q(t) - \dot{\gamma}(t)$$
$$\dot{V}_T(t) = \frac{1}{\mathrm{m}}(F_T(t)\cos\alpha(t) - \mathcal{D}(\alpha(t),\delta_e(t)) - \mathrm{m}g\sin\gamma(t))$$
$$\dot{\delta}_e(t) = (\delta c_e(t) - \delta_e(t))/T_e$$

$[\gamma_0,\ Q_0,\ \alpha_0,\ V_{T0},\ \delta_{e0},\ \dot{\delta}_{e0}]$, $[\bar{q}_0]$	initial values (last measured values)

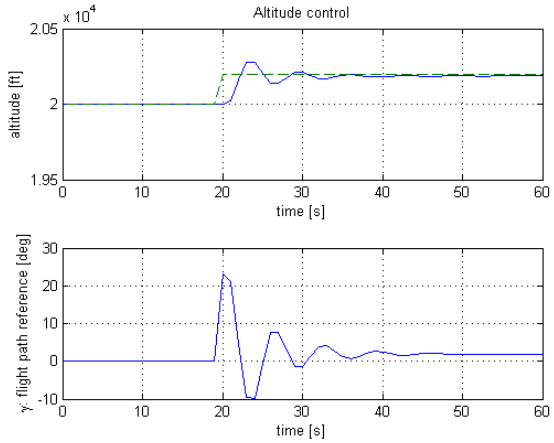
Constraints on states, parameters and controls:

$-10^\circ \le \alpha \le 45^\circ$	angle of attack
$900\text{ft/s} \le V_T \le 1000\text{ft/s}$	total speed
$-45^\circ \le \gamma \le 45^\circ$	flight path
$-15^\circ \le \delta c_e \le 15^\circ$	elevator deflection
$-60^\circ/\text{s} \le \dot{\delta} c_e \le 60^\circ/\text{s}$	elevator rate
$1000\text{lb} \le T_c \le 19000\text{lb}$	thrust force
$\bar{q} = \bar{q}_0$	dynamic pressure

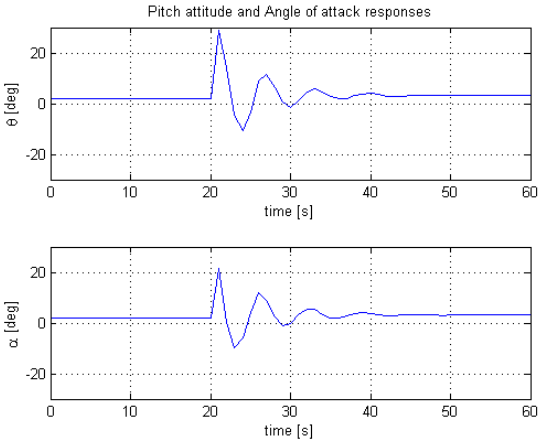
Table 8.8: NMPC flight path control design parameters and prediction model

Differential states		
	h	altitude
Controls		
	γ_c	flight path command
Model Parameters		
	α	angle of attack
	V_T	total speed
Optimization/prediction parameter values		
	$T_s = 0.5\text{s}$	sample time
	$T_h = 5.0\text{s}$	horizon (start: 0.0s)
	$N = 10$	intervals in horizon
Weighting matrix values		
	$Q_a(1,1) = 0.01$	h error weight
	$R_a(1,1) = 0.10$	γ_c error weight
Objective function parameters		
	$\{h,\ \gamma_c\}$	state and control
	$\{r_h,\ \gamma_0\}$	setpoints (γ_0: last measured value)
Differential equation (prediction model)		
	$\dot{h} = V_T \cos(\alpha)\sin(\gamma_c + \alpha) - V_T \sin(\alpha)\cos(\gamma_c + \alpha)$	
	$[h_0],\ [\alpha_0,\ V_{T0}]$	initial values (last measured values)
Constraints on states, parameters and controls		
	$-833.3\text{ft/s} \leq \dot{h} \leq 833.3\text{ft/s}$	climb rate
	$-25^\circ \leq \gamma_c \leq 25^\circ$	flight path command
	$\alpha = \alpha_0$	angle of attack
	$V_T = V_{T0}$	total speed

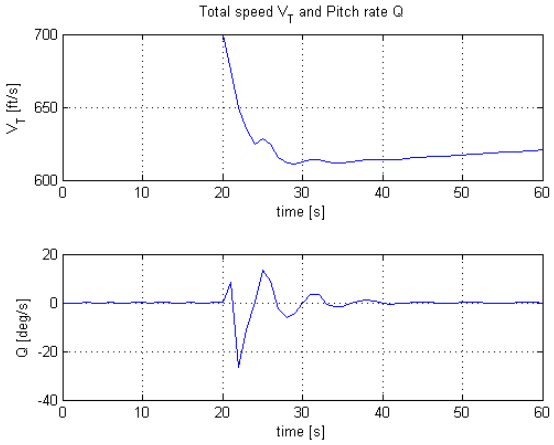
Table 8.9: NMPC altitude-hold (climb control) autopilot design parameters and prediction model



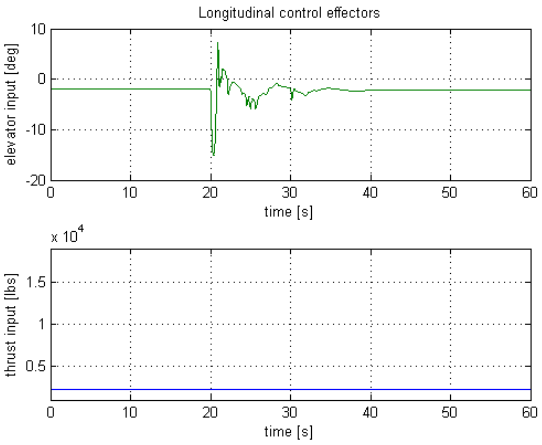
(a) Altitude and flight path control



(b) Pitch and Angle of attack



(c) Total speed and pitch rate



(d) Elevator input and thrust force

Figure 8.12: F-16 NMPC longitudinal motion control ($T_h = 2$s)

8.4.2 Lateral motion control

The NMPC lateral autopilots include a Heading hold controller which performs coordinated turns in conjunction with a roll-angle hold controller. Figure 8.13 show the simulink models of the Heading hold and roll-angle hold autopilots. Similar to

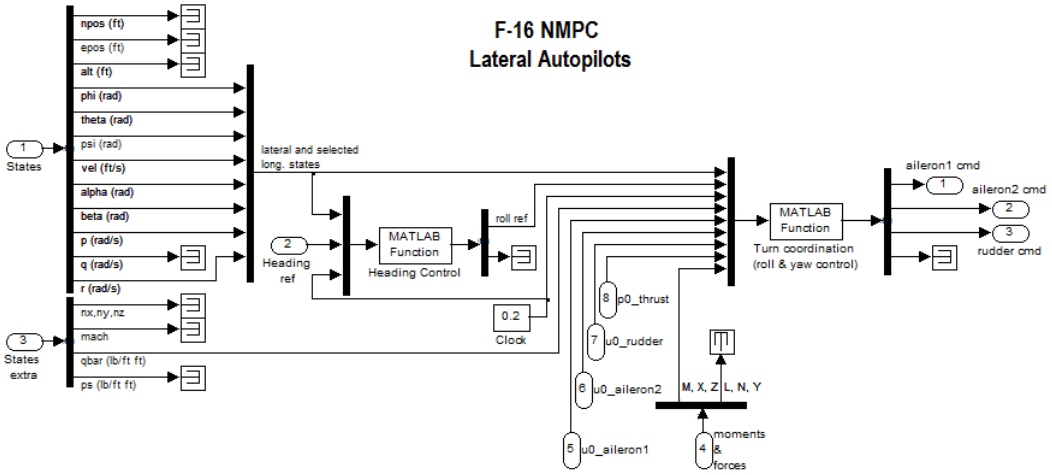


Figure 8.13: NMPC heading-hold autopilot and turn coordination implemented in simulink

the longitudinal motion control, *design option 1* was used for the lateral NMPC autopilot implementations. The measured states of the turn-coordination autopilot were introduced as time-varying parameters in the heading NMPC problem formulation (see design summary table 8.11). The 'MATLAB function' blocks in figure 8.13 contain the NMPC problem formulations implemented as a combination of MATLAB scripts and C-code (see appendix **??**). The ACADO for MATLAB NMPC formulations and design parameter values are presented in tables 8.10 and 8.11.

Coordinated turns and steady flights on preset headings were performed for testing the performance of the heading-hold and roll-angle hold autopilots. The results are shown in figure 8.14. The results show pure lateral decoupled motion control. The longitudinal flight controls were kept at trim settings (see table 8.7) while the lateral flight maneuvers were performed. The test flight began with a 20° heading demand and was stepped down to 10° after 30s. The demanded changes in heading were carried out smoothly, as can be observed in figure 8.14.

The turn coordination autopilot has an additional objective of maintaining zero sideslip as well as zero yaw rate and roll rate. The corresponding weighting matrix values (in table 8.10) were therefore selected to ensure that the roll attitude is always prioritized. The simple choice of weighting values works perfectly for the roll rate and yaw rate objectives (observed in figure 8.14c) since they are directly related to the change demands in roll and yaw attitudes (or moments). Sideslip on the other hand depends on speed control, and in addition, the direct effect of sideslip on yaw-dynamics, through yaw rate coupling effects, is not properly accounted for in the turn coordination strategy used. For these reasons further tuning will be necessary for sideslip if error-free control is required. Nevertheless,

keeping sideslip close to zero (observed in figure 8.14b) is considered satisfactory for the overall performance of the NMPC lateral autopilots.

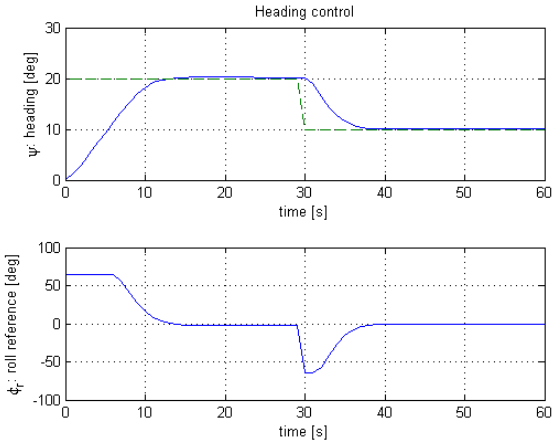
An important remark on the turn coordination strategy is the use of the non-linear version of the classical bank-to-turn equation (B.10) (also shown in table 8.11), instead of the original yaw-differential equation (4.84) which is part of the lateral aircraft model (7.12) presented in section 7.2. Equation (B.10) specifies the desired *bank-to-turn* behavior of the aircraft during a turn maneuver. That is, a roll angle ϕ different from zero will induce a yaw rate, which turns the aircraft. An imediate drawback anticipated from the use of the classic bank-to-turn strategy is the crippling of NMPC's ability to effectively coordinate and predict the effects of yaw rate and pitch angle on the turning behavior (described by the original yaw-equation). In addition, yaw rate R which is the direct coupling state variable between the yaw Ψ dynamics and the sideslip β dynamics will not be properly accounted for. The use of classical turn coordination strategy can therefore lead to 'suboptimal' results when the effects of sideslip becomes prominent in some maneuvers. However, the original yaw-differential equation was found to present some challenges, such as computational difficulties (leading to wrong control trajectory generation). This is possibly due to the optimization algorithm terminating with wrong local solutions, or due to numerical issues with the optimization problem solver.

Similar to observations made in the longitudnal control tests, horizons shorter than 2s result in very aggressive control. The 5s horizon chosen for the heading autopilot is also a trade-off between computational speed and dynamic performance of the F-16. The performance of the lateral NMPC autopilots can be further enhanced by influencing the rate of change of the state trajectories of the turn coordination autopilot. This can be done by introducing constraints on the yaw rate R, similar to the use of climb rate constraint on altitude control (in the longitudinal channel). A closer look at the roll Φ trajectory (in figure 8.14b) and its reference (in figure 8.14a) reveals that the roll controller tracks its reference well. The heading (or outer loop) controller can therefore be identified as an important deciding factor of the overall response of the turning maneuver accomplished.

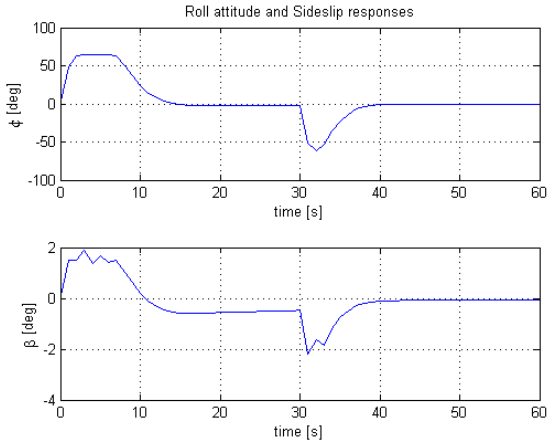
An aspect of the NMPC autopilots that offers a similar effect on the control performance as that obtained when rate constraints are applied is the internal reference trajectory [7]. The result of reducing the rate at which the reference trajectory approaches the setpoint trajectory (generated by the heading controller) can be seen in figure 8.15. The difference in reference trajectories used can also be seen in figure 8.16. The desired performance, in terms of how fast the reference trajectory approaches the setpoint trajectory, and the resulting effects on the state trajectories, can therefore be regarded as a tuning and design specification task. The influential roll played by the reference trajectory is very obvious when figures 8.14 and 8.15 are compared. The difference is a trade-off between heading control and rate of change in the aircraft lateral states. A very minimal

[7] The reference signal issued by a guidance system (or the pilot) is refered to as the *setpoint trajectory*. The NMPC *reference trajectory* defines an ideal trajectory along which the plant should return to the setpoint trajectory, for instance after a disturbance occurs (see section 3.1.2).

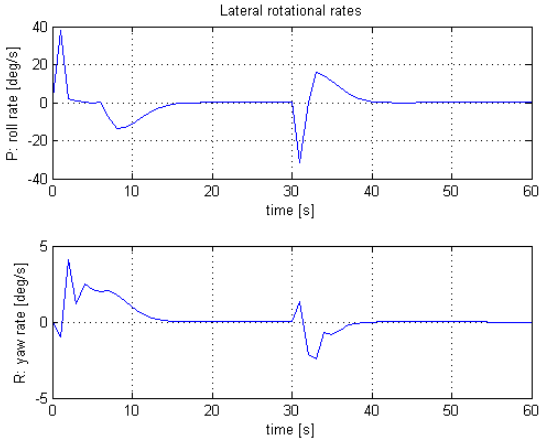
usage of the rudder is rocorded, and the limitations of sideslip β control and the bank-to-turn strategy no longer become an issue.



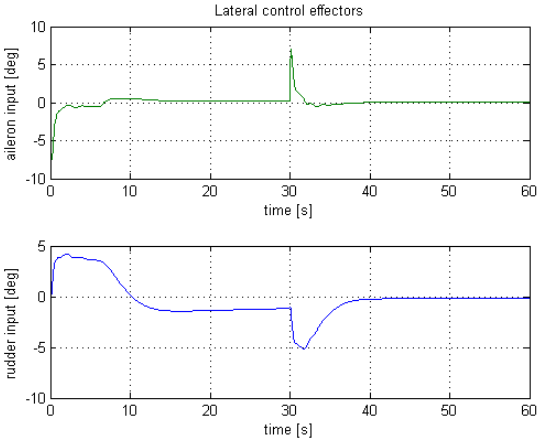
(a) Heading and bank-to-turn control



(b) Roll and sideslip

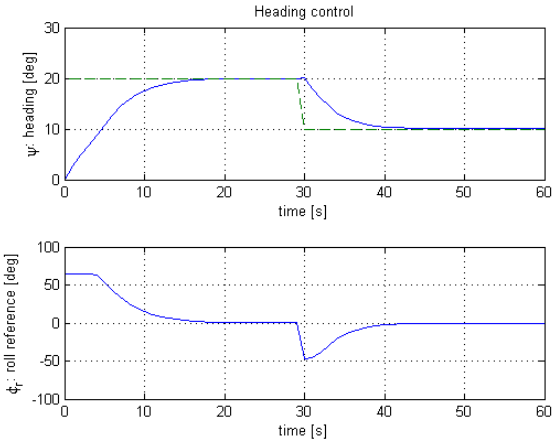


(c) Roll rate and yaw rate

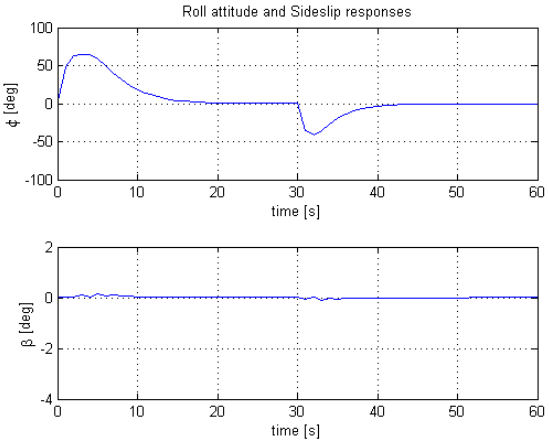


(d) Aileron and rudder inputs

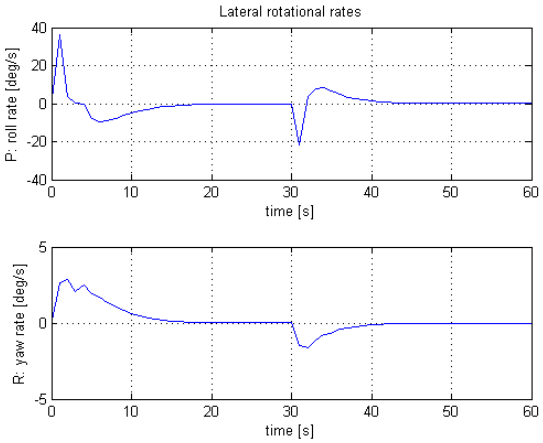
Figure 8.14: F-16 NMPC lateral motion control (*Reference trajectory* T_{ref1} implemented)



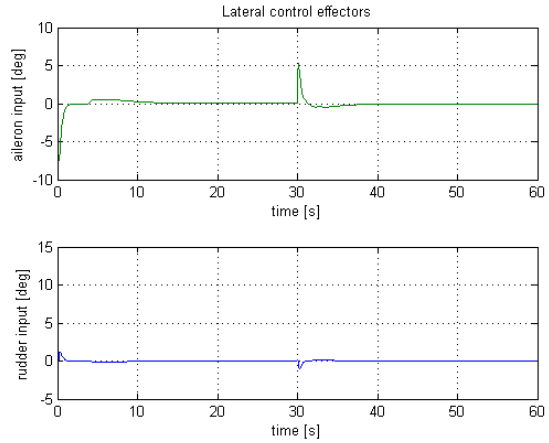
(a) Heading and bank-to-turn control



(b) Roll and sideslip



(c) Roll rate and yaw rate



(d) Aileron and rudder inputs

Figure 8.15: F-16 NMPC lateral motion control (*Reference trajectory* T_{ref2} implemented)

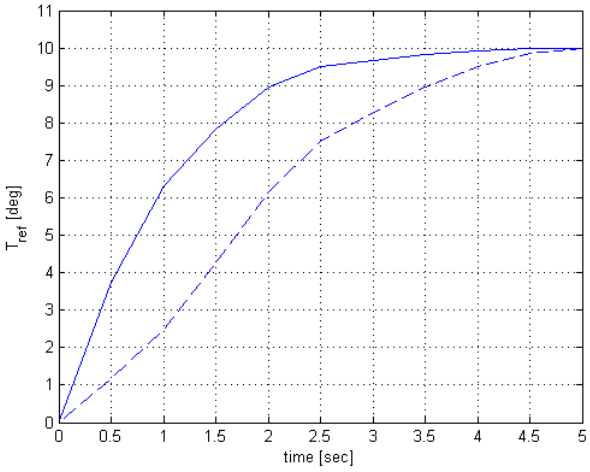


Figure 8.16: Reference trajectories compared. T_{ref1} (—), T_{ref2} (−−)

Differential states:	
β	sideslip
Φ	roll
P	roll rate
R	yaw rate
$\delta_a,\ \dot{\delta}_a$	aileron states
$\delta_r,\ \dot{\delta}_r$	rudder states
Controls:	
δc_a	aileron command
δc_r	rudder command
Model Parameters:	
$\bar{q}$	dynamic pressure
α	angle of attack
V_T	total speed
Θ	pitch
F_T	thrust force
$X_T,\ Z_T$	total force in X and Z
Optimization/prediction parameter values:	
$T_s = 0.05$s	sample time
$T_h = 1.0$s	horizon (start: 0.0s)
$N = 20$	intervals in horizon
Diagonal weighting matrix values:	
$Q_t(1,1) = 1.00$	β error weight
$Q_t(2,2) = 10.0$	Φ error weight
$Q_t(3,3) = 1.00$	P error weight
$Q_t(4,4) = 1.00$	R error weight
$R_t(1,1) = 0.05$	δc_a error weight
$R_t(2,2) = 0.10$	δc_r error weight
Objective function parameters:	
$\{\beta,\ \Phi,\ P,\ R,\ \delta c_a,\ \delta c_r\}$	states and controls
$\{0,\ r_\Phi,\ 0,\ 0,\ \delta_{a0},\ \delta_{r0}\}$	setpoints
Differential equation (prediction model):	
$\dot{\beta}(t) = P(t)\sin\alpha_0 - R(t)\cos\alpha_0 + \frac{1}{\mathrm{m}V_{T0}}(\mathcal{C}(\beta(t),\delta_a(t),\delta_r(t)) -$ $F_{T0}\cos\alpha_0\sin\beta(t) + \mathrm{m}g_2)$	
$\dot{\Phi}(t) = P(t) + R(t)\tan\Theta_0\cos\Phi(t)$	
$\dot{P}(t) = \frac{I_z}{I_xI_z - I_{xz}^2}\ell(\beta(t),\delta_a(t),\delta_r(t)) + \frac{I_{xz}}{I_xI_z - I_{xz}^2}n(\beta(t),\delta_a(t),\delta_r(t))$	
$\dot{R}(t) = \frac{I_{xz}}{I_xI_z - I_{xz}^2}\ell(\beta(t),\delta_a(t),\delta_r(t)) + \frac{I_x}{I_xI_z - I_{xz}^2}n(\beta(t),\delta_a(t),\delta_r(t))$	
$\dot{\delta}_a(t) = (\delta c_a(t) - \delta_a(t))/T_a, \quad \dot{\delta}_r(t) = (\delta c_r(t) - \delta_r(t))/T_r$	
$[\beta_0,\ Phi_0,\ P_0,\ R_0,\ \delta_{a0},\ \dot{\delta}_{a0},\ \delta_{r0},\ \dot{\delta}_{r0}]$	state initial values
$[\bar{q}_0,\ \alpha_0,\ V_{T0},\ \Theta_0,\ F_{T0},\ X_{T0},\ Z_{T0}]$	parameter initial values (last measured values)
Constraints on states, parameters and controls:	
$-308°/\mathrm{s} \le P \le 308°/\mathrm{s}$	roll rate
$-15° \le \delta c_a \le 15°$	aileron deflection
$-80°/\mathrm{s} \le \dot{\delta} c_a \le 80°/\mathrm{s}$	aileron rate
$-15° \le \delta c_r \le 15°$	rudder deflection
$-120°/\mathrm{s} \le \dot{\delta} c_r \le 120°/\mathrm{s}$	rudder rate
$\bar{q} = \bar{q}_0$	dynamic pressure
$\alpha = \alpha_0, \quad V_T = V_{T0}$	angle of attack, speed
$\Theta = \Theta_0, \quad F_T = F_{T0}$	pitch, thrust force
$X_T = X_{T0}, \quad Z_T = Z_{T0}$	total force in X and Z

Table 8.10: NMPC turn coordination autopilot parameters and prediction model

Differential states		
	Ψ	heading
Controls		
	ϕ_c	roll command
Model Parameters		
	V_T	total speed
Optimization/prediction parameter values		
	$T_s = 0.5$s	sample time
	$T_h = 5.0$s	horizon (start: 0.0s)
	$N = 10$	intervals in horizon
Weighting matrix values		
	$Q_h(1,1) = 10.10$	Ψ error weight
	$R_h(1,1) = 0.100$	ϕ_c error weight
Objective function parameters		
	$\{\Psi,\ \phi_c\}$	state and control
	$\{r_\Psi,\ \Phi_0\}$	setpoints
Differential equation (prediction model)		
	$\dot{\Psi} = g\sin(\phi_c)/V_T,$	$g = 32.17\text{ft/s}^2$
	$[\Psi_0]$, $[V_{T0}]$	initial values (last measured values)
Constraints on states, parameters and controls		
	$-65^\circ \leq \phi_c \leq 65^\circ$	roll command
	$V_T = V_{T0}$	total speed

Table 8.11: NMPC heading-hold autopilot parameters and prediction model

8.4.3 Aircraft coupled-motion control

A complete simulation of the F-16 using the decoupled autopilots posed some challenges due to the strong coupling effects of the longitudinal motion on the lateral channel, and vice versa. The coupling effects are most obvious when active control is desired simultaneously in the longitudinal and lateral channels. Handling such coupling effects in a classical linear autopilot design scheme can easily increase the complexity of the implementation and result analysis (considering nonlinearities that can be introduced into the closed loop system). In some cases it becomes necessary that both decoupled-motion autopilots affect all the available control surfaces (typically through a separate control allocator) to be able to function satisfactorily. When the complexity increases heavily, reliability of the resulting automatic control system usually becomes questionable.

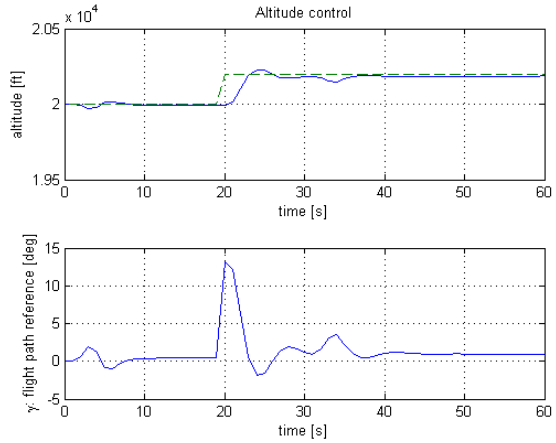
The F-16 coupled motion control challenge was, however, handled elegantly in the NMPC framework by treating the coupling effects of the longitudinal control as time-varying measured parameters (or disturbances) in the lateral NMPC formulation, instead of assuming they are 'constant' or 'zero'. The measured angle of attack α, total speed V_T, pitch Θ, total force in X and Y, and thrust force F_T were therefore fed to the turn-coordination autopilot prediction model at each sample instance. The results of the flight test maneuvers of the coupled control system are shown in figures 8.17 and 8.18. After making the lateral controller aware of the activities of the longitudinal controller, no further parameter tuning was necessary to achieve the results in the coupled-motion control tests.

The test maneuvers began with a 20° heading change demand to the heading and turn coordination autopilots, while the altitude autopilot was commanded to maintain 20000ft altitude for 20s. The heading change maneuver induced a 'bumpy' disturbance in the altitude, appearing before the 5s time mark in figure 8.11a. The observed disturbance is an indication of the limitations of the classic design strategy, which almost always involves controlling decoupled (and simplified) process/plant dynamics. The effect is similar to introducing perturbations beyond a linear controller's validity limit. It should be recalled that the pure (or decoupled) longitudinal dynamics was obtained based on a wings-level flight assumption (see section 4.4.1). However, the altitude autopilot detected the induced disturbance quickly and commanded corresponding flight path control actions (see changes in the reference signal in figure 8.17a), which resulted in the use of elevator actions to subdue the disturbance. These observations are made when comparing flight path reference and elevator actions, for the first 20s, in the pure longitudinal control plots and the coupled motion control plots (compare figure 8.11d with figure 8.17d, and figure 8.11a with figure 8.17a). It can also be seen that the corresponding heading change maneuvers of figure 8.18a are not as smooth as those of figure 8.14a.

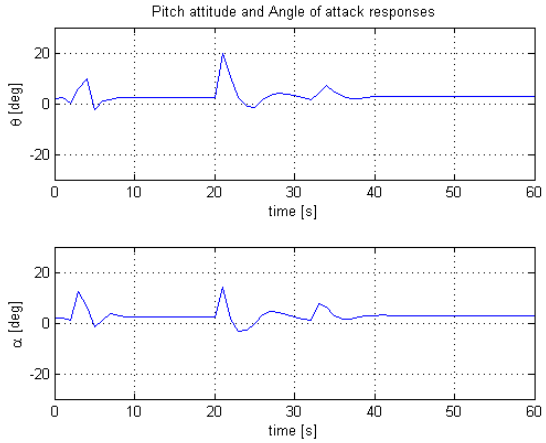
The F-16 coupled maneuvers were alternated after 20s, as was done in the pure decoupled motion control case. The altitude change demand introduced a similar 'bumpy' disturbance in the heading (figure 8.18a), when the control trajectories of the ailerons and rudder started settling to their optimal values (see the 20s - 30s interval of figure 8.18d). The effect of the rather 'sluggish' sideslip control strategy used (through the pure bank-to-turn implementation) became more ev-

ident in the coupled motion control results, observed in the size of sideslip (in figure 8.18b) and the slight deviation noticed in the heading control up to the 20s time mark (in figure 8.18a). Shortly after, a new 10° heading change was commanded while the climb maneuver and altitude-hold control were still in progress. The result is similar to the observations made earlier. The altitude settling time became longer, and the heading change maneuver was not as smooth as in the case of the pure lateral motion control maneuver.

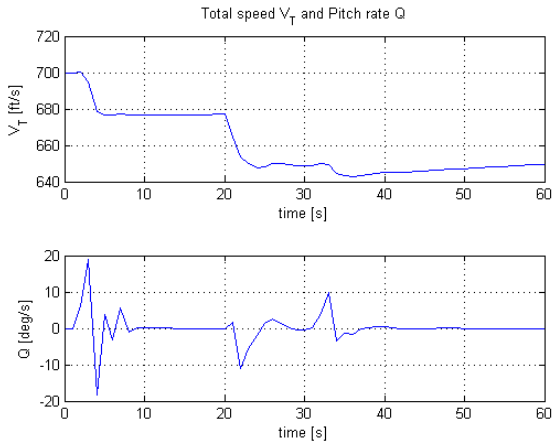
The need of adequate tuning of the reference trajectory (or the direct influence of rate-states through the use of constraints) can be illustrated even clearer by comparing the responses obtained when the two reference trajectories in figure 8.16 are implemented for the coupled motion control tests. Compare the trajectories of figures 8.17, 8.18, with those of figures 8.19, 8.20. It is obvious that the less aggressive lateral motion control makes its coupling effects on the longitudinal motion control almost unnoticeable.



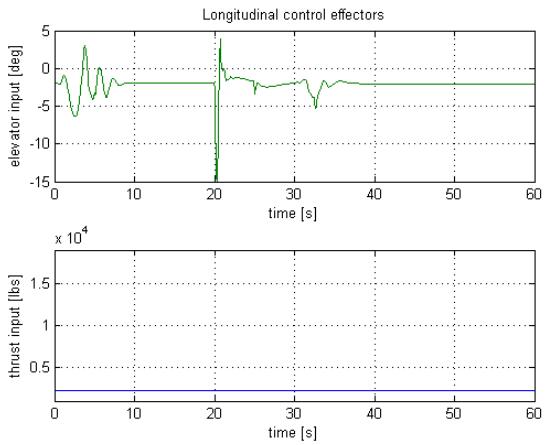
(a) Altitude and flight path control



(b) Pitch and Angle of attack

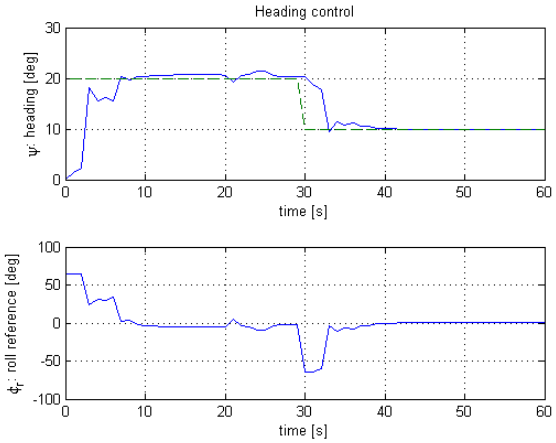


(c) Total speed and pitch rate

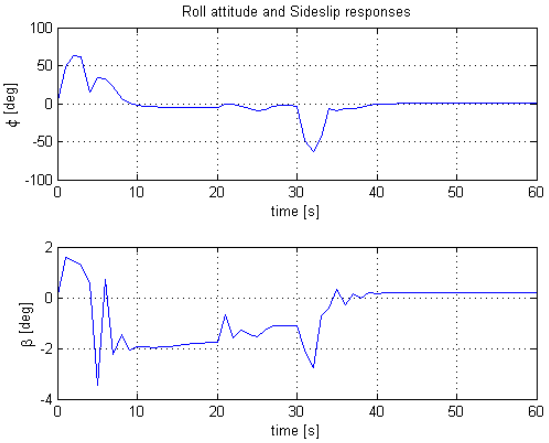


(d) Elevator input and thrust force

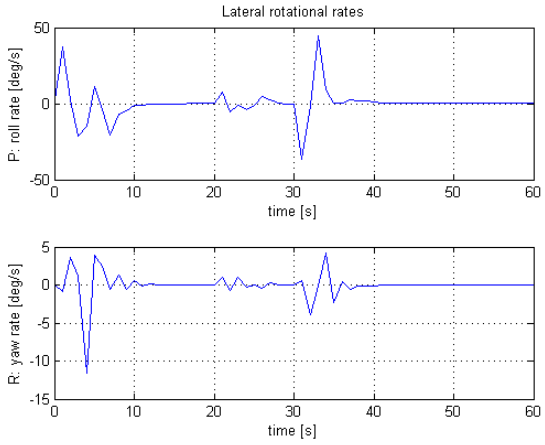
Figure 8.17: F-16 NMPC coupled motion control: Climb maneuvers (*Reference trajectory* T_{ref1} implemented)



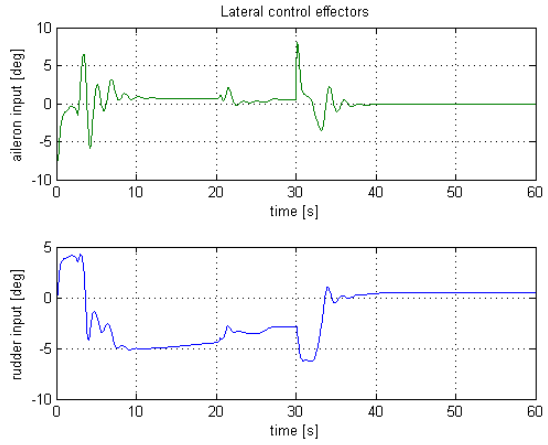
(a) Heading and bank-to-turn control



(b) Roll and sideslip

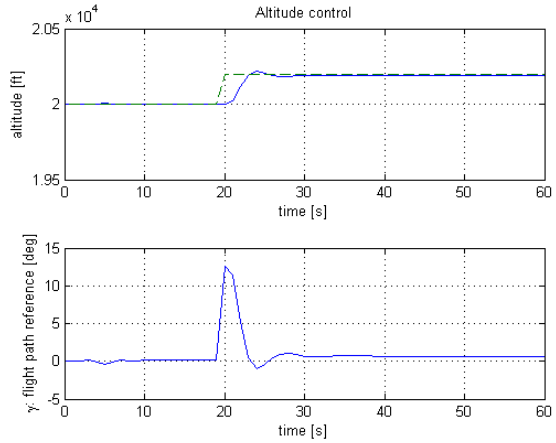


(c) Roll rate and yaw rate

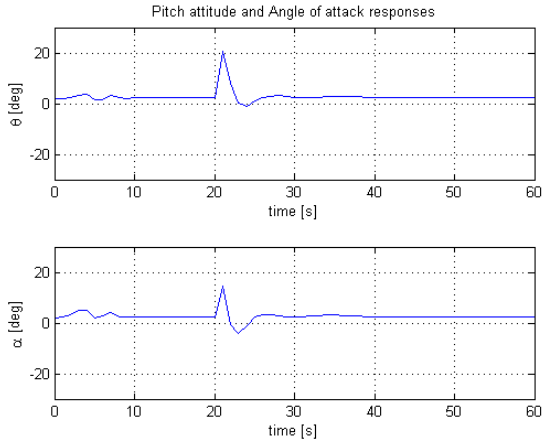


(d) Aileron and rudder inputs

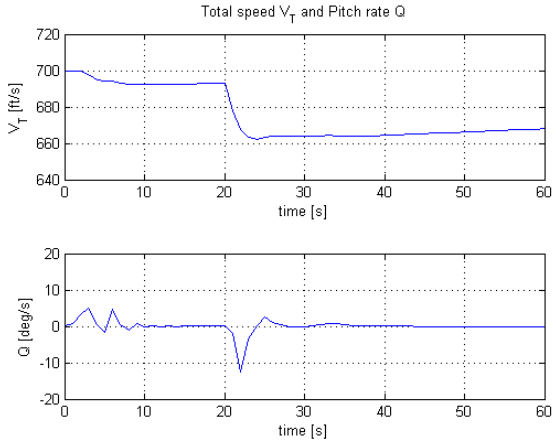
Figure 8.18: F-16 NMPC coupled motion control: Turn coordination (*Reference trajectory* T_{ref1} implemented)



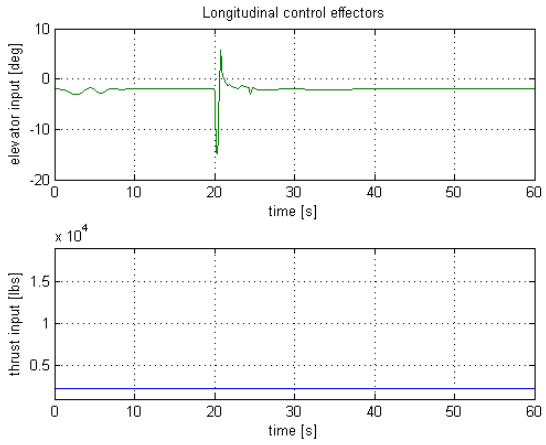
(a) Altitude and flight path control



(b) Pitch and Angle of attack

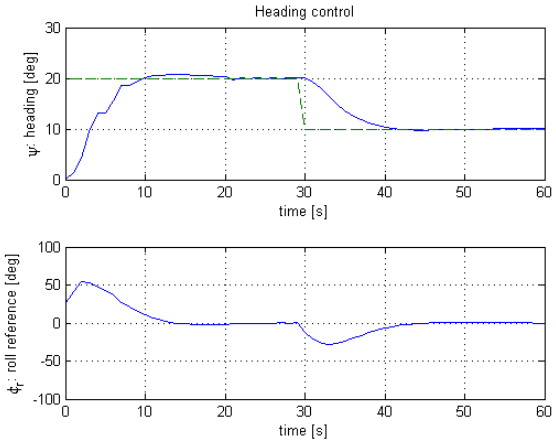


(c) Total speed and pitch rate

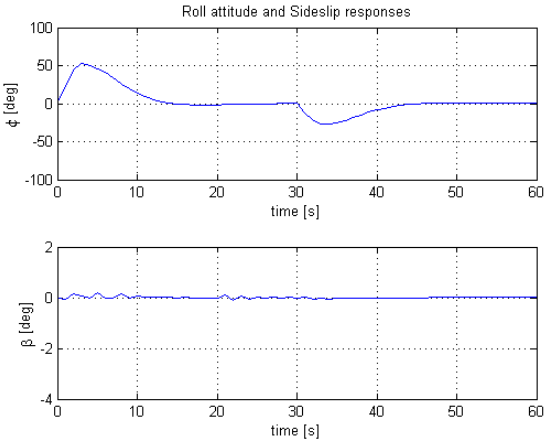


(d) Elevator input and thrust force

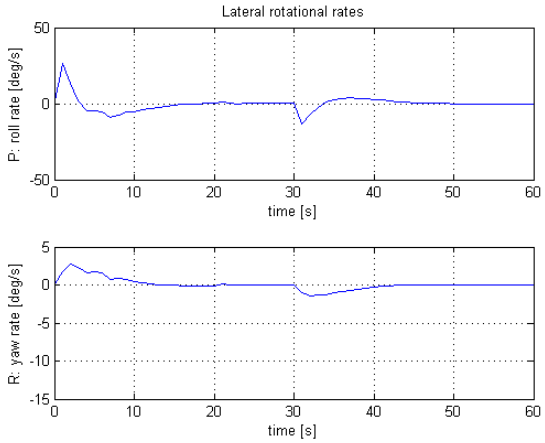
Figure 8.19: F-16 NMPC coupled motion control: Climb maneuvers (*Reference trajectory* T_{ref2} implemented)



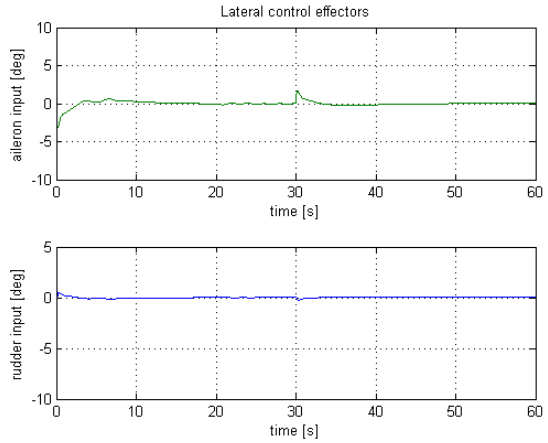
(a) Heading and bank-to-turn control



(b) Roll and sideslip



(c) Roll rate and yaw rate



(d) Aileron and rudder inputs

Figure 8.20: F-16 NMPC coupled motion control: Turn coordination (*Reference trajectory* T_{ref2} implemented)

8.5 NMPC Reconfigurability and Fault Tolerance

A primary objective of this thesis is to investigate the reconfigurability and fault tolerance of NMPC. The preceding implementation sections have touched on some inherent reconfigurability properties of NMPC, where, for instanse, time-varing parameters were easily introduced for cross-coupling effect adjustments in the lateral NMPC autopilots. Updating the prediction model through aerodynamic coefficient lookup-table interpolations can also be regarded as a reconfiguration strategy, which is easily incoorporated in the NMPC framework. This section therefore focuses on using control surface (or actuator) fault simulations to primarily verify the fault tolerant capabilities expected of NMPC as a reconfigurable controller (see section 2.2).

The simulation setup used assumes that a fault detection system is available, reduces the fault detection and identification (FDI) part of the implementation to only failure simulation and diagnostic reconfiguration. The control surface or actuator faults are described in section 6.5, and the FDI scheme on which the fault simulation is based is described in 6.6. The simulated faults were categorized into to main modes: Actuator Position failure mode and Rate failure mode. A position fault refers to any of the following conditions:

1. a significant change in the deflection limits or saturation.
2. *locked/jammed* actuator, including *actuator runaway* (or hard-over).

To simulate the faults mentioned above, the upper and lower limits for the faulty actuator are changed during a maneuver. The control surface is regarded as locked when the lower position limit is set equal to the upper limit.

The deflection rate failure also takes two forms, namely:

1. a significant change in actuator deflection rate.
2. *free-play/float-type* failure, or loss of actuator effectiveness.

The actuator is floating when both upper and lower deflection rate limits are set to zero, according to the FDI scheme described in section 6.6. Both locked actuator and a float-type failure render the control surface uncontrollable. However, a locked actuator introduces a significant disturbance (*adverse* yaw/roll/pitch) which needs dynamic compensation to restore stable flight conditions, and if possible perform flight maneuvers satisfactorily.

It is worth mentioning that the autopilot simulation results presented in the preceding sections were performed with 'harder' limitations on the control surfaces ($\pm 15^\circ$) than the normal F-16 actuator deflection ranges given in table 8.2. Only 70% aileron range, 60% elevator range, and 50% rudder working range were considered, and the situation can actually represent a multiple fault case. However, actuator saturation was not observed as a hinderance for the performance of the NMPC autopilots. The significant change in actuator deflection limit fault-type is therefore considered already examined. An early conclusion is therefore timely since significant change in actuator deflection induces only an insignificant effect on the NMPC autopilot's reaction. This is obviously due to NMPC inbuilt property of handling constraints.

The simulation results presented in the following sections cover mainly locked/-jammed actuators and float-type failures. The main objective is to stabilize the aircraft and attempt to perform turn and climb maneuvers as well as attaining straight level flight in the presence of control surface faults. The control surfaces (or actuators) involved in the F-16 reconfigurable fault tolerant control tests are the left aileron (a_1), the right aileron (a_2), the rudder (r), and the tail stabilator differential controls (i.e. left elevator (e_1) and right elevator (e_2)). The relationship between the use of symmetric and asymmetric control surface actuation was realized through the control input strategy: $\delta s = 1/2(\delta s_L + \delta s_R)$. Where δs_L and δs_R are the left and right control surface deflections, combined into a single control surface input δs. This strategy applies to aileron and elevator (or tail stabilator differential) control.

8.5.1 Aileron failures

The same maneuvers used earlier in the 'fault-free' simulations were repeated, but this time with the left aileron locked at $-15°$. The reaction of the F-16 to the jammed-actuator failure and the recovery actions of the reconfigurable NMPC autopilots are shown in figure 8.21.

It can be noted from figure 8.21f, that in order to perform the commanded coordinated turn, the right aileron (a_2) and the rudder were actively involved. The inherent reconfiguration property of NMPC comes to light nicely in the control plots, where corrective actions of the right aileron (a_1) are held very close to its upper limit ($15°$), and effectively supported by rudder actions. The result is a successful execusion of the commanded heading change maneuvers (see figure 8.21b), compared to the 'fault-free' maneuver in figure 8.18a.

The altitude-hold plot in figure 8.21a reveals corresponding disturbance from the aileron failure situation, but limited change in overall performance was recorded (compared to 'fault-free' results in figure 8.17a). This is a good indication of optimal utilization of attainable moments produced by all remaining functional control effectors. However, a small steady-state error is observed in the altitude control plot, whereas a much smaller error noticed in the corresponding plot, 8.17a, of the 'fault-free' coupled motion control case). The source of this error can be immediately identified as the decoupling assumptions made in section 4.4.1. The longitudinal control dynamics implemented entirely neglects the effects of sideslip β and roll Φ. The wings-level flight assumption's validity ceases when handling the aileron failure situation, since some roll angle (see figure 8.21d) is required to maintain the commanded heading. The sideslip is also not zero as assumed. Obviously, the extent of standard NMPC robustness to modling errors (or approximations) is a subject that requires further discussion.

The results presented in figure 8.21 (refered to as case *1*) assumes the existence of an FDI system that updates the turn coordination NMPC autopilot with the aileron's position, when stuck. It was interesting to find out what happens when NMPC has no knowledge about the failure. Another interesting remark is that no reconfiguration mechanism was implemented to achieve the results in figure 8.21. Further tests were therefore performed on the locked left aileron (a_2), where in

one case no FDI system was assumed (case *2*), and in another test (case *3*), the following simple reconfiguration mechanism was introduced (FDI assumed).

1. *for each actuator*, scan for locked/jammed actuator by checking for changes in lower and upper limits (worst case $\delta s_{min} = \delta s_{max}$)

2. if an actuator is locked, decrease the weight of all the 'healthy' actuators in the same channel (longitudinal or lateral), by a predefined factor.

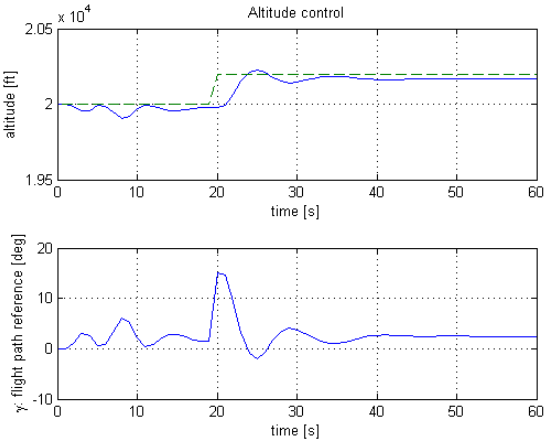
3. finally, reduce the weight on any secondary objectives that share the same remaining 'healthy' actuators (that are capable of achieving similar effects as the jammed actuator), by a predefined factor.

It can be noted that the above reconfiguration mechanism is similar to the strategy that was used to obtain good results for an optimal control allocator in [12] (see appendix D.3).

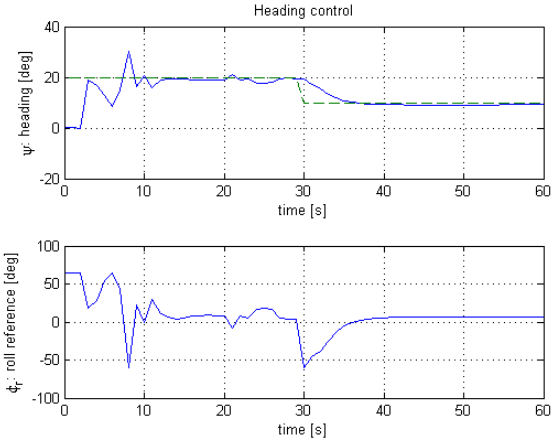
The results of the three test cases are compared in figure 8.22. Apart from differences noticed in the rudder input actions (figure 8.22f) and their accompainying sideslip trajectories (in figure 8.22d), the trajectories of Case *1* and Case *2* follow each other very closely. In Case *2*, the NMPC turn coordination controller quickly realizes through model simulations and input updates (as initial values) that the left aileron is not reponding to its commands. In other words, the controller perceives a situation similar to when it has driven the aileron to its limit, and starts to move other control surfaces. Without any knowledge of the failure case, the results of Case *2* can be considered as excellent performance, compared with those of Case *1* and Case *3*.

It can be emphasized that in Case *1*, the locked aileron fault was reported to the controller (by setting $\delta_{a1_{min}} = \delta_{a1_{max}} = -15°$) and, in addition, the weights (error penalties) were adjusted in Case *3*. The weights on keeping set-points for roll rate, yaw rate, and sideslip were reduced by a factor of 10, to further prioritize the use of roll angle to achieve the desired heading. The same reduction was made on the weight for the right aileron, with the intention of making it more active.

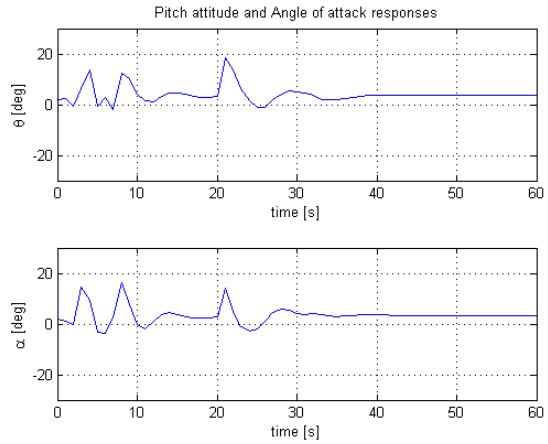
Another interesting result was obtained when the weight on keeping set-points for heading was reduced from 10 to 2.5. The intention was to relax the objective of fast control from a guidance system point of view. The results are shown in figure 8.23. The remarkable performance improvement recorded indicates that some change in the overall performance objective that reflects the fault situation may be necessary in the event of an extreme control surface failure.



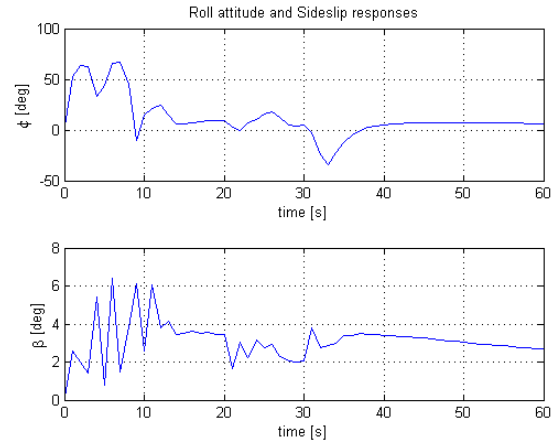
(a) Altitude and flight path control



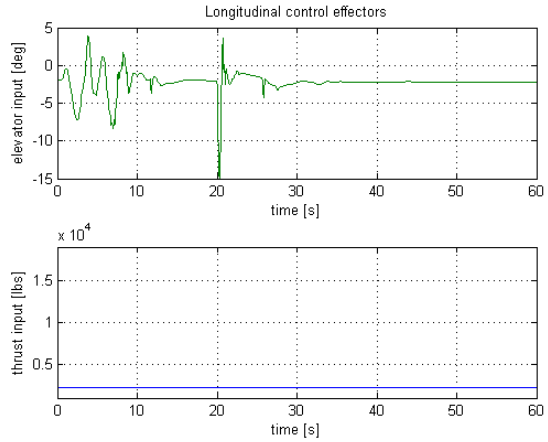
(b) Heading and bank-to-turn control



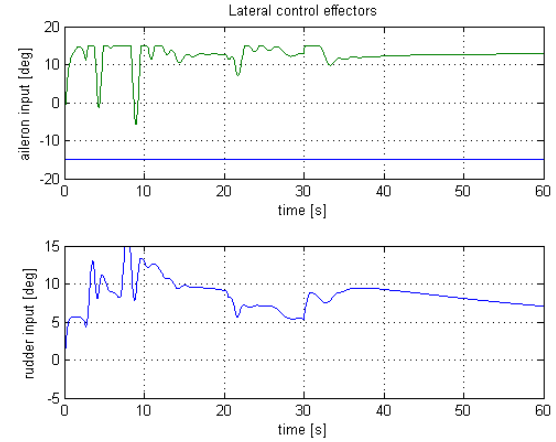
(c) Pitch and Angle of attack



(d) Roll and sideslip

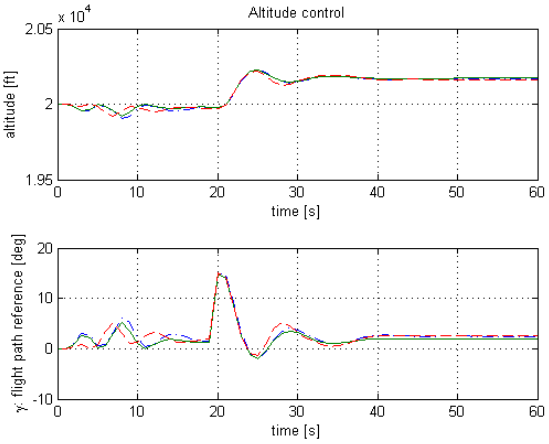


(e) Elevator input and thrust force

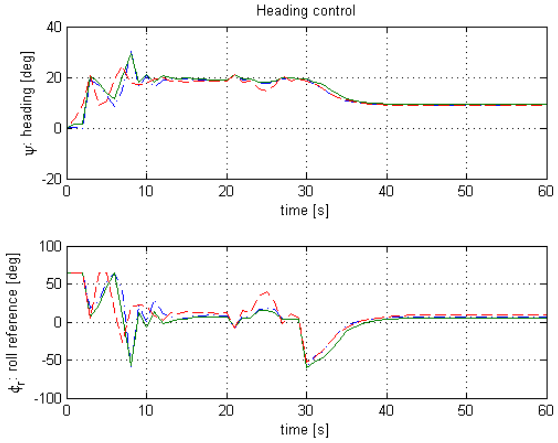


(f) Aileron and rudder inputs

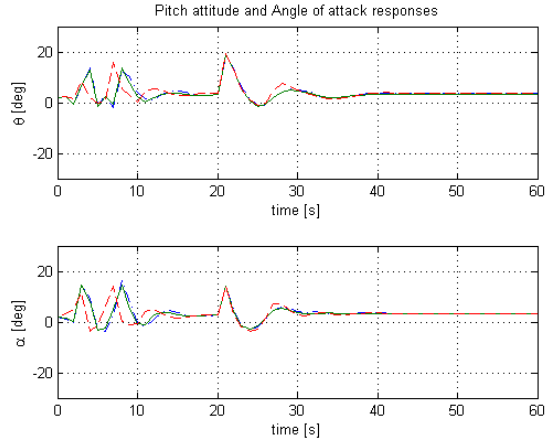
Figure 8.21: NMPC fault tolerance: F-16 left aileron (a_1) locked at $-15°$ (Case *1*)



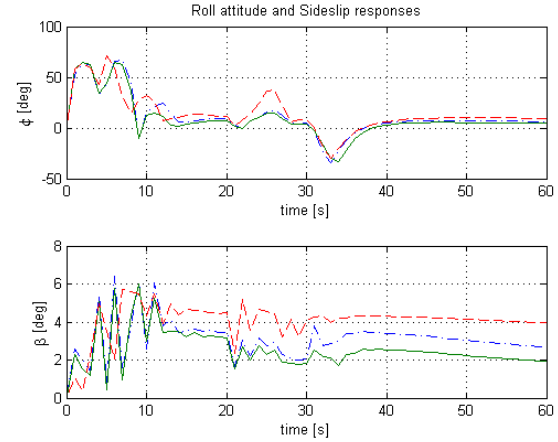
(a) Altitude and flight path control



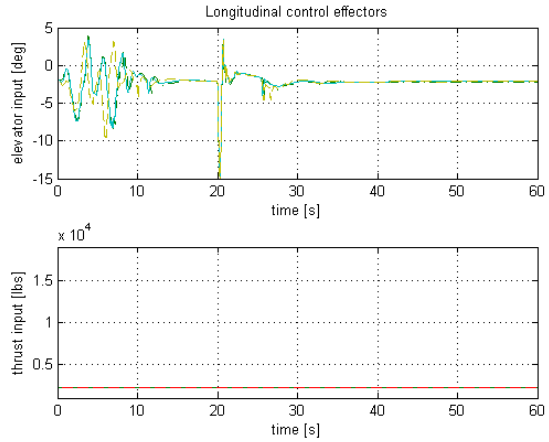
(b) Heading and bank-to-turn control



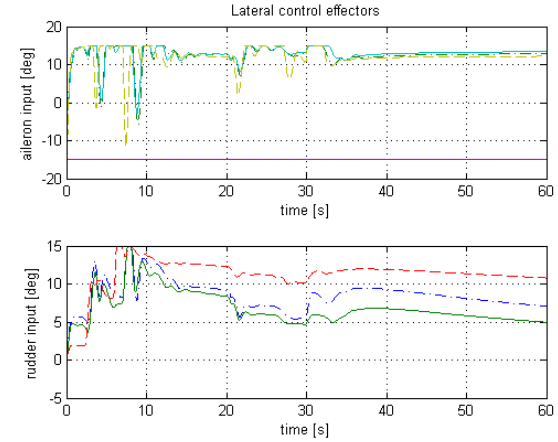
(c) Pitch and Angle of attack



(d) Roll and sideslip

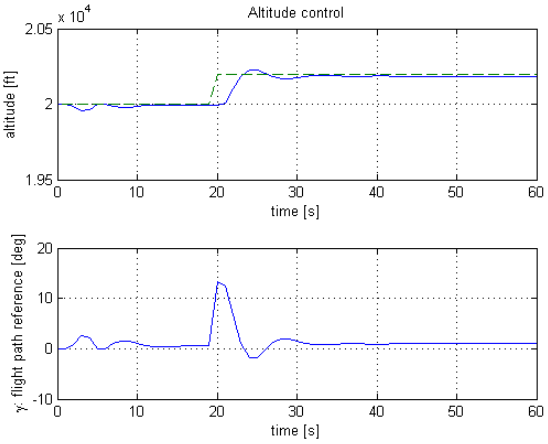


(e) Elevator input and thrust force

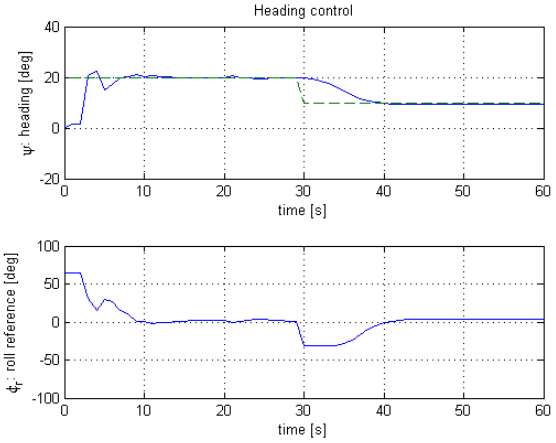


(f) Aileron and rudder inputs

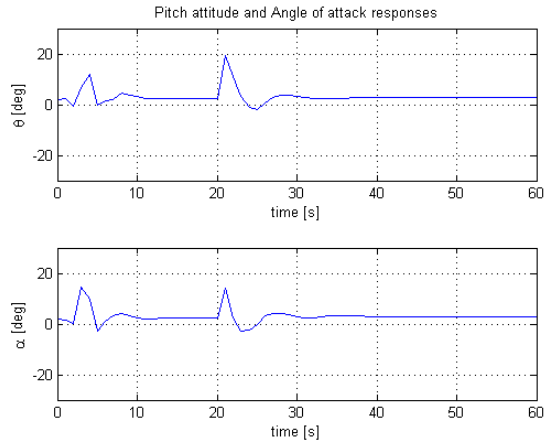
Figure 8.22: NMPC fault tolerance: F-16 left aileron (a_1) locked at $-15°$. Case *1* ($-\cdot$), Case *2* ($-$), Case *3* ($--$)



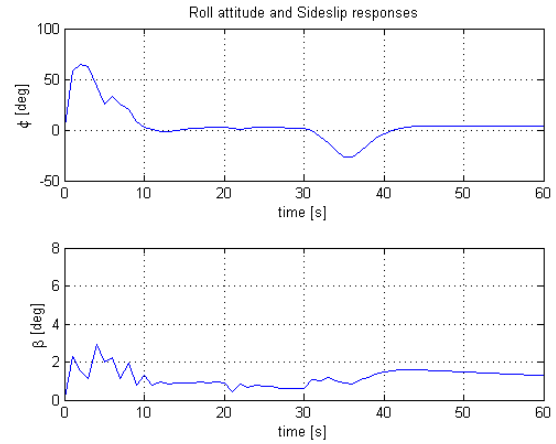
(a) Altitude and flight path control



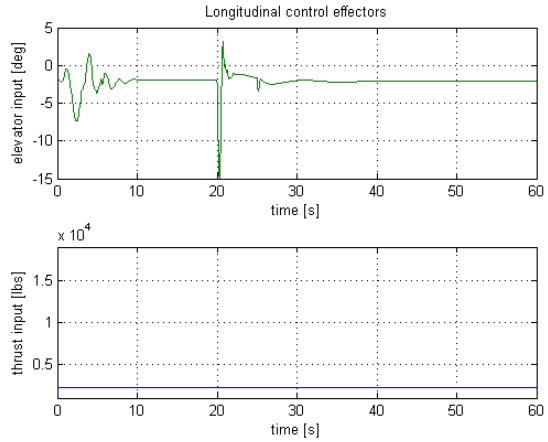
(b) Heading and bank-to-turn control



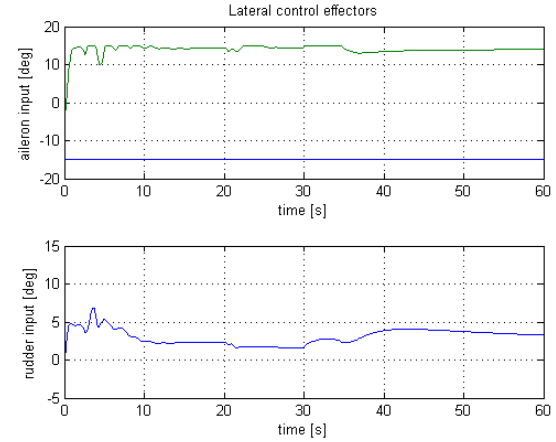
(c) Pitch and Angle of attack



(d) Roll and sideslip



(e) Elevator input and thrust force



(f) Aileron and rudder inputs

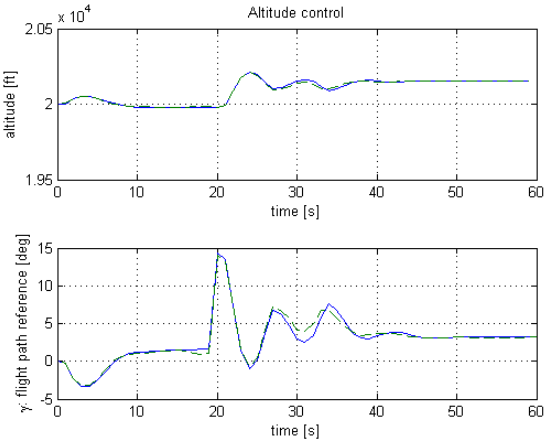
Figure 8.23: NMPC fault tolerance: F-16 left aileron (a_1) locked at $-15°$ (Case *2*) Relaxed heading control objective

8.5.2 Rudder failures

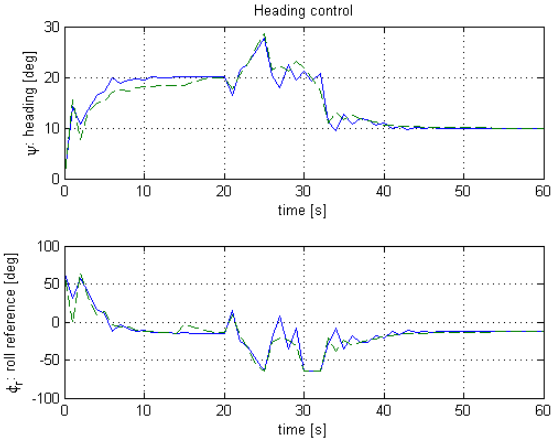
The rudder was also locked at $-15°$, and the heading change and climb maneuvers were repeated as before. The reaction of the F-16 to the jammed rudder situation and the recovery actions of the reconfigurable NMPC autopilots are shown in figure 8.24. Since no significant difference was noticed between Case *1* and Case *2* for the aileron hard-over test, only test Case *2* and Case *3* results are presented for the rudder failure.

The results of test Case *2* in figure 8.24, which were taken without any information about the rudder fault reported to the NMPC autopilots, and with no reconfiguration mechanism implemented, is rather impressive. The huge sideslip disturbance induced by the jammed rudder, which was later amplified by the climb maneuver (after 20s) did not prevent the heading and turn coordination autopilots from achieving the performance level shown in the plots of figure 8.24. Again, increasing the fault awareness of the NMPC autopilots through fault reporting and additional reconfiguration (as in Case *3*) did not yield any remarkable improvement in the responses observed.

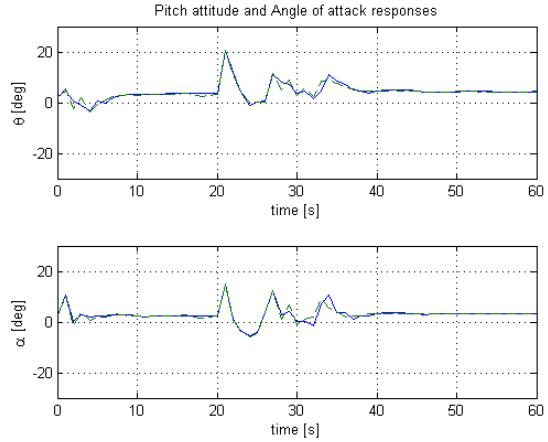
During the rudder failure tests, an intresting modification was made to the NMPC heading autopilot to illustrate some corrective measures that can be easily introduced to compensate for model approximations. It was pointed out earlier (in section 8.4.2) that the bank-to-turn dynamics used in the lateral autopilot design introduces some approximation errors into the complete aircraft dynamics. The effects of such errors become obvious when the sideslip magnitude becomes significant. Notice the steady-state error in the heading plot (10s - 20s interval) of figure 8.18a, and its corresponding sideslip magnitude in figure 8.18b. The heading autopilot was made aware of the sideslip disturbance β_d by correcting the yaw-angle prediction Ψ_p, using $\Psi = \Psi_p - \rho\beta_d$, where $0 \leq \rho \leq 1$ decides the extent of correction required. The Case *3* plot of figure 8.24b shows the effect of changing ρ from 0.5 to 0.25.



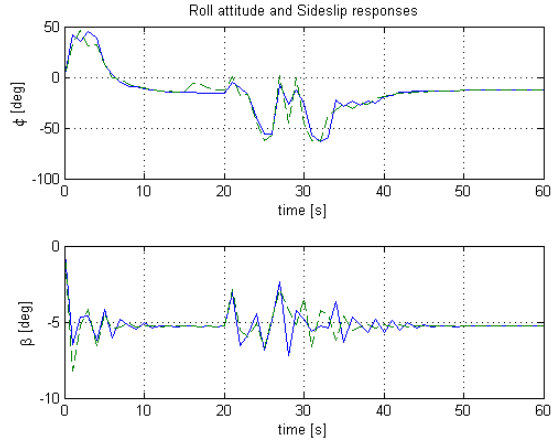
(a) Altitude and flight path control



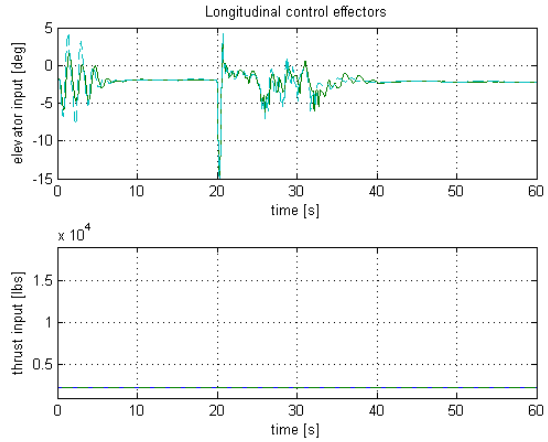
(b) Heading and bank-to-turn control



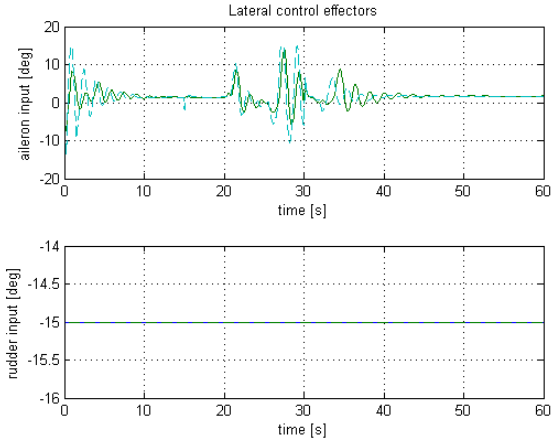
(c) Pitch and Angle of attack



(d) Roll and sideslip



(e) Elevator input and thrust force



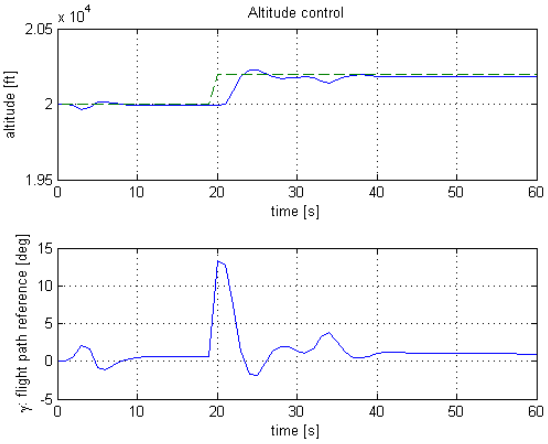
(f) Aileron and rudder inputs

Figure 8.24: NMPC fault tolerance: F-16 rudder locked at $-15°$. Case *2* $(-)$, Case *3* $(--)$

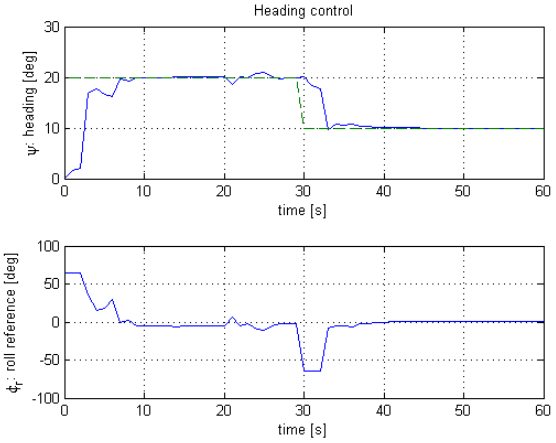
8.5.3 Elevator (or tail stabilator) failures

The free-floating type failure was tested on the right elevator (e_2), and the heading change and climb maneuvers were repeated. The left and right elevators on the F-16 are normally operated together within their upper and lower limits. They also serve as horizontal tail stabilators which accept differential deflection commands upto $\pm 5.38°$. The elevator failure test attempts to simulate a situation where the right elevator is broken, and moves freely without restricting the movement of the left elevator.

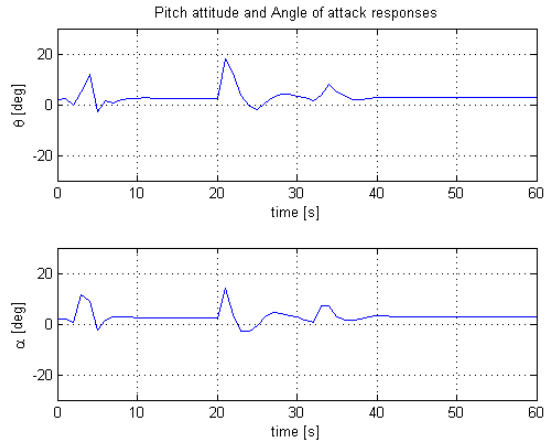
The plots in figure 8.25 show the results of Case *2*-type test where the F-16 relies only on the reconfigurable fault tolerance capabilities of NMPC. This test case is rather mild compared to the hard-over failure tests, since no significant disturbance was introduced by the 'floating' control surface. The results are almost identical to those obtained for the 'fault-free' condition.



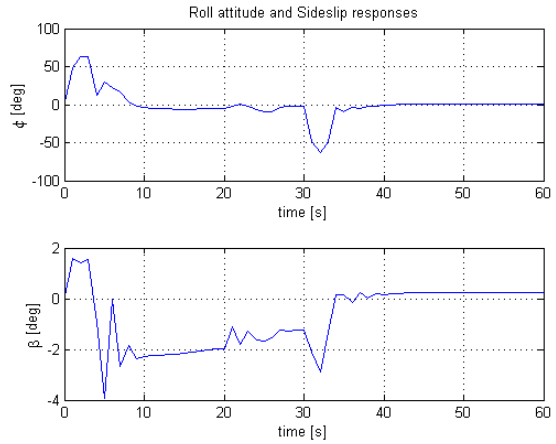
(a) Altitude and flight path control



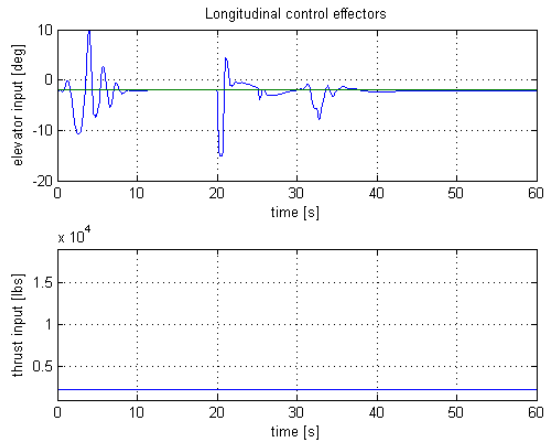
(b) Heading and bank-to-turn control



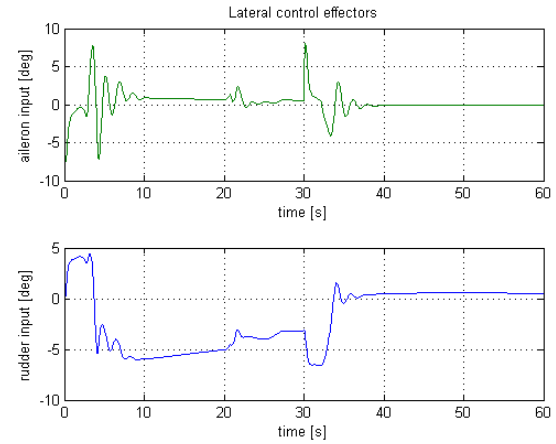
(c) Pitch and Angle of attack



(d) Roll and sideslip



(e) Elevator input and thrust force



(f) Aileron and rudder inputs

Figure 8.25: NMPC fault tolerance: F-16 left elevator (e_2) floating freely Case *2*

Chapter 9

Result analysis and discussion

The NMPC studies and autopilot results achieved in chapter 8 illustrate the elegance of combining multivariable, nonlinear, and constrained controller behavior into a well structured and powerful NMPC framework. In this chapter, the extensive study done on NMPC and its implementation issues are highlighted in a comparative discussion of expected performance and corresponding results. The discussions cover NMPC design choices made and their consequences on the dynamic performance of the overall automatic flight control system. The advantages and disadvantages of the implemented NMPC scheme are also discussed. Discussions on NMPC inherent reconfigurability, fault detection capabilities, and fault tolerance, are given attention.

It is important to emphasize that the results of this work are influenced by the design parameter choices made in a very limited computational power resource environment. The results are, therefore, mainly intended to illustrate the powerful potentials of NMPC and, at the same time, to serve as a platform for realistic discussions on the subject. For clarity and immediate result assessment, most of the implementation sections are supported with explanations and discussion of design options and result observations made. The discussions made in this chapter are therefore generalized, compared to the specific remarks and comparisons made for individual results in chapter 8.

9.1 Performance and Design Consequences

Several aspects of the NMPC performance have been touched through the studies done in this work. NMPC proves to excell in almost all its expected performance requirements. The coupled motion tests, and the measures taken to ensure satisfactory control can be seen as simple measures that enhance the inherent robustness of NMPC. It is comparable to handling known disturbances, by introducing the disturbance model as part of the NMPC prediction model. Any accurate information about the plant's behavior is always well utilized in the NMPC framework.

The extensive reliance on an internal model for accurate predictions and control, can in some cases become a weak point of the control scheme. Modeling errors, which usually occur due to approximations and underestimations, can limit the performance of the resulting controller, as observed through the use of a classical turn coordination dynamics instead of the aircraft's original yaw-dynamics. The use of detailed simulations will be adequate as an NMPC design aid, in order to

uncover any vulnerabilities introduced by model errors.

Any detected (or expected) model anomalies can normally be compensated for in a simple way, as was done for the decoupled autopilots. For instance, it can be seen that the nonzero sideslip contributes as a 'constant' drift disturbance in the heading controller, since sideslip is not directly accounted for in the bank-to-turn strategy. A possible compensation is to correct the turn coordination model predictions with the measured sideslip disturbance (similar to heading-course compensations, using sideslip angle), and using the corrected heading (yaw) angle in the objective function for error minimization. This compensation strategy was implemented in order to obtain the error-free results obtained in section 8.5.2 (in spite of the magnitude of sideslip induced by the rudder hard-over failure.) Nevertheless, the reconfigurability property of NMPC almost always triggers some 'corrective' actions (sometimes erratic) in attempt to compensate for behaviors that are not properly accounted for in neither the internal model nor constraint specifications. The result is normally a brief voilation of constraints, followed by a 'learning' phase, which can eventually lead to satisfactory performance. Careful observations of all state and, especially, control trajectories are needed to identify most modeling or design errors.

Producing local optimal solutions was mentioned as a possible source of conputational difficulties leading to unpredictable control activities, when the aircraft's nonlinear yaw-dynamics was used. This is a notable possible drawback of the NMPC scheme, but an implementation strategy, such as using the classic turn coordination dynamics as an initial trajectory generator for the original yaw-dynamics, can be suggested in this case as a possibly viable 'warm start' strategy.

The performances recorded for both pure decoupled motion (longitudinal/lateral) control and coupled motion control can be compared to the performances of the NMPC and Linear Parameter Varying (LPV) controllers in [10]. The main results are included in appendix C for easy reference. The results show that the NMPC scheme yields a better performance compared to the LPV control approach. It is, therefore, reasonable to anticipate that an autopilot based on a gain scheduling strategy (for example using gain scheduled PID control) will yield similar results as the LPV controller. The results of [10], which were obtained for the prestabilized F-16 aircraft, give an indication that the performances recorded in this work (especially, the 'fault free' test cases) can be considered as satisfactory. Apart from using an LPV controller for prestabilizing the F-16 aircraft, the control objective in [10] is to regulate state perturbations to a set, whereas the corresponding inner loop controllers in this work have an additional objective of tracking time-varying references from their outer loop controllers. Further, the results obtained on NMPC fault tolerance rely on the performance objectives selected in this work.

9.1.1 NMPC tuning and real-time considerations

Another very desirable property of NMPC was observed during the simulation and test process. The tuning task became much less demanding, after the dynamics of the system was well studied and analysed through simulations (mostly by simple step response tests). When a good starting point for sampling frequency and horizon length are determed (usually by considering the largest time constants),

the remaining tuning task reduces to finding adequate weight ratios for the state and controls (not forgetting variable scaling). The tuning task also appears to be less demanding when a detailed nonlinear model is used.

Despite the complex nature of a detailed nonlinear aircraft model, the tuning process becomes very predictable, and even more intuitive when the dynamics (fast/slow modes) are well grouped as subsystems. Splitting the autopilots into an inner (faster) loop and an outer (slower) loop prevented the need to compute the whole aircraft model for the selected (5s) horizon using the smallest required sampling time (0.05s). In addition to the intuitive tuning advantage, using subsystems can be regarded as advantageous when operating with limited computational power. A more elegant formulation of NMPC which supports the separation of fast dynamics from slow ones will be preferable - for instance, through state-dependent variable step-size (or horizon intervals) specifications, similar to existing input (or manipulated variable) blocking[1] strategies.

The choice of horizon and its consequences was illustrated in figures 8.11 and 8.12. Attempts to operate with shorter horizons are normally made due to real-time computational limitations, and this results mostly into stability issues that require modifications to the standard NMPC formulation (see section 3.3). The studies made in this work confirm the fact that longer horizon lengths allow the NMPC algorithm to acquire richer knowledge about the system behaviour to a given input. The result is a better decision made on optimality for every control trajectory generated. When the horizon is too short for the NMPC algorithm to properly observe the response of the aircraft to a given control input, the risk of generating a sequence of consistent large control inputs becomes high. Longer horizons (greater than 2s) on the other hand lead to the generation of less aggressive control trajectories and, as observed in figure 8.11, the state trajectories became smoother and more optimal. In addition, NMPC stability did not become an issue since, for instance, optimization over a longer horizon reduces the contribution of the terminal cost on the overall cost being minimized. This is also due to the fact that the terminal cost serves as an approximation that is greater than the value of the truncated integral, and reducing the contribution of the terminal cost yields a total cost that is closer to the optimal.

In view of the promising results obtained, and the preceding discussions, real-time considerations can be named as the main challenge of implementing the NMPC scheme. Solving the nonlinear programming problems formulated in this work places a significantly large computational burden on the implementation platform, but real-time implementation is not considered as a primary objective for this work. Nevertheless, simulation times up to 60s were recorded for a 1s simulation time on a computer with Intel® Core™ i3 CPU, 2.27GHz, running a 64-bit Windows 7 operating system. The simulation time was recorded for the complete NMPC autopilot simulation setup. That is, four NMPC autopilots and the full nonlinear F-16 model (or plant) running in a MATLAB/Simulink environment. It can be however emphasized that, using fast NMPC solutions or hardware with adequate computational capacity (with respect to the control application),

[1] Non-uniform intervals between control decisions, which allow control adjustments to be made throughout the predicted transient period without having too many decision variables in the optimization problem.

real-time implementation of the NMPC autopilots is possible.

Avoiding infeasibility and numerical problems was considered as an important part of the autopilot optimization problem formulations. Both physical input constraints and operational constraints of a real F-16 aircraft were used in the problem formulations. In addition, the F-16 model validity limitations were also incorporated in the problem formulation to ensure feasibility. Constraint relaxation through the use of slack variables was not necessary in this work. Careful design and simulations led to the complete skipping of any conservative stability enforcing terminal constraints, and therefore avoiding any accompanying feasibility issues. The use of last measured states and control input feedback as initial values (or warm start) for each optimization run is also a way of enhancing both problem feasibility and convergence of the optimization algorithms. Initialization with the last measured values is justified by the fact that the sampling intervals used are short compared to the dynamics of the F-16 (plant) model, and therefore making the nonlinear problem at one sampling instant closely related to the nonlinear problem at the previous sampling instant.

A known source of numerical problems is scaling (see section 3.4.2). It was mentioned in section 8.3.1 that prescaling of the model variables, in order to limit the relative tracking/control error differences, was not necessary in this work. However, it is worth mentioning that the importance of mentaining a well conditioned optimization problem (w.r.t. scaling) was experienced when some simulations were aborted due to wrong unit (i.e. degrees instead of radians) interpretations for reference trajectories. The situation in that case became similar to posing a poorly scaled problem to the optimization solver. It can also be noted that the choice of 0.01 weight on the altitude error (instead of 1) is intended to keep the weighted tracking error small compared to the error values of the flight path control reference generated.

9.2 Inherent Fault Tolerance

Fault tolerance of NMPC has been shown, through autopilot implementation results, to be achievable through an effective combination of inherent fault detection capabilities and reconfiguration. Observations made in the results of section 8.5.1 indicate a clear tendency of NMPC to effectively utilize its limited resources, and attempt to accomplish the set objectives, even with no knowledge of the fault. In cases where the failure affects the capabilities of the aircraft, it is possible to change the objectives, or the constraints, or both, accordingly. An external reconfiguration mechanism is therefore only a means of re-adjusting the control objectives, by using priority shifting strategies.

It is possible to introduce changes on-line mainly because the control signal is recomputed at each time step by solving an optimization problem, allowing changes to be made to the problem formulation. It may however not be an easy task to re-tune the NMPC on-line if a failure occurs, since an appropriate change may not be known for an arbitrary failure situation. It can be further argued that in cases where fault detection is not available or the available FDI system is not reliable, it will be a good alternative to allow the NMPC to utilize its inherent robustness.

In comparison to earlier linear MPC results (in [11]), indicating the inherent fault detection property of predictive control, a clear response improvement was obtained through changing of weights (or error penalties). NMPC shows further superiority on this subject, since results of section 8.5.1 show that only insignificant results were obtained in the attempt to improve the results obtained by only NMPC (i.e. without external FDI). NMPC (with no failure information) is capable of achieving good responses in the failure cases in this study. This is can be due to the detailed information of process dynamics used (compared to linear MPC).

When a classical autopilot is implemented, a way of achieving reconfigurability is to augment the system with an Optimal Control Allocator (OCA). This approach has been demonstrated in an earlier project [12](and in reference [41]) to be a very effective control strategy that enhances fault tolerance in existing aircraft control systems. However, OCA as a reconfigurable controller depends entirely on both an external FDI system and some kind of reconfiguration mechanism to achieve its results, since it has no inherent feedback. In the classical autopilot-OCA framework, all available actuators are usually combined into one module (i.e. all control effectiveness coefficients must be available in one unit) for effective control re-distribution to be possible. Results in this thesis show that good results can be obtained in the NMPC framework even when the separate NMPC autopilots do not have direct access to all the control surfaces through one unit.

An important factor in reconfigurable fault tolerant control is the extent of control effector redundancy present on the plant. For the F-16, only a minimal number of control surfaces have been considered in the work. Further flexibility/redundancy can be achieved, for instance in the lateral channel, by incorporating differential tail deflections as control inputs in addition to the ailerons and rudder. This additional degree of freedom can be naturally utilized by NMPC when aileron and rudder failures occur simultaneusly. Even in the absence of an explicit differential tail control implementation, NMPC inherent reconfiguration capability can result in automatic asymmetric control of the stabilators to battle out induced disturbances of high proportions experience in the longitudinal channel.

Another interesting observation made is that significant change in actuator (or control surface) position limits and deflection rate limits have very minimal effect on NMPC performance, since actuator constraints are handled effectively as part of NMPC's primary objectives.

9.3 Control design complexity

On handling fast and unstable system dynamics, the preliminary studies in section 8.3 and the performance of the complete NMPC autopilots indicate that NMPC is a promising control scheme that can replace traditional low level controllers. The main task is to find the appropriate control objectives and constraints that handle the stability augmentation requirements of the complete automatic control system.

Specifically, the NASA technical paper [36] that touches on F-16 stability augmentation specifies strategies necessary for angle of attack limiting and prevention of departures resulting from excessive pilot rudder usage at high angles of attack. Furthermore, an automatic departure-/spin prevention system, which uses yaw-rate feedback to drive the roll-control surfaces to oppose any yaw-rate buildup, was suggested. The classical approach used in the NASA paper depends on switching logic for engaging/disengaging several stability and control augmentation subsystems (or modes). Implementing all the suggested F-16 stability and control specifications, in a classical way, leads to control structure/design complexity of the magnitude shown in appendix B.5 (taken from [36]).

NMPC strategy, on the contrary, allows the same specifications to be formulated in precise mathematical objectives and constraints, with inbiult reconfiguration capabilities. As an example, the limiting strategy used to prevent angle of attack depature in the NMPC preliminary studies (in section 8.3) was to use state dependent weight (or penalty) adjustments in conjuction with the angle of attack constraints imposed on the optimization problem. Another example is to incorporate the command gradient curves, shown in figure B.11 (see appendix B.5), into the specification of the reference trajectories of the NMPC autopilots.

Chapter 10

Conclusion and Further Work

It has been shown from the results of this work that NMPC can be used to control highly nonlinear, multivariable, and unstable systems. In brief, NMPC simplifies control design for very complex systems, and can offer a complete control design package, including conrol augmentation, stability augmentation, control effector redistribution, inherent robustness (to some extent), and an overall effective fault tolerant control system. These desirable properties of NMPC make it a good choice for high performance and autonomous flight control.

Unlike ad hoc heuristics based methods for fault tolerant control design, NMPC has inherent reconfigurable control capabilities due to effective use of an internal plant model for prediction, an online optimization with input and state constraints, penalties (or weights) on tracking/control errors, and horizons that can be adjusted to account for resquirement changes imposed by failure conditions. The performances recorded for NMPC inherent fault detection and reconfigurability, compared to its corresponding 'fault-free' flight control performances, make NMPC superior to most existing fault-tolerant control schemes.

This work also highlights the fact that, performance of the overall fault tolerant system depends on many factors, such as the availability of the remaining 'healthy' actuators, the reconfiguration mechanism used, and the type of control strategies adopted. The speed and accuracy of the fault detection and identification scheme is also a deciding factor on the success of the reconfiguration attempt. Further, the integration of all the modules in the overall reconfigurable fault tolerant control system is very vital for the aircraft to survive any extreme control effector damages. However, the results of this work show that the reliance of NMPC on external FDI modules, external reconfiguration mechanisms, and their effective coordination is minimal.

A notable advantage of NMPC, and model predictive control in general, is the possibility of influencing the performance of the controller by direct on-line reconfiguration. This property allows the introduction of new information at each time step, either from estimation, or some adaptation strategy. The robustness and fault tolerance of NMPC can, therefore, be further enhanced when accurate data of disturbances is available. The same applies to critical model updates when severe damages, other than actuator (or control surface) faults, occur.

In addition, the results show that very fast (possibly unstable) dynamics that

are usually handled by classical PID controllers are well handled by NMPC. At the same time, NMPC offers a well structured formulation and specification of stability augmentation requirements/strategies that are otherwise implemented as complex mode switching tasks. The advantages of using NMPC for controlling both inner (fast) loops and outer (slow) loops were also demonstrated by the implemented autopilots in this work. Using NMPC to control the outer loop (or slow dynamics) of the aircraft can be seen as performing a guidance task with a controller that is capable of generating both feasible and optimal trajectories for the tracking tasks of the fast dynamics NMPC autopilots.

One notable disadvantage of using constrained NMPC is the danger of running into infeasibility. Especially in the case of model/plant mismatch, hard constraints and large disturbances that are not accounted for in the NMPC formulation. In this case, some effective measures have been suggested and shown to enhance the performance and overall stability of the control system. In particular, extending the horizon to the maximum computationally possible limit is advisable, since it gives the controller more time to satisfy the constraints.

Further work and remarks:
The implementation scope of this work was limited to the studies of the limitations associated with classical control design strategies (in terms of simplifications/assumptions considered in decoupling the aircraft dynamics) and the associated consequences faced when the coupled motion effects become significant. Even though NMPC handles coupling effects well, further investigations can be performed, where the fully coupled nonlinear model of the F-16 is implemented as the internal model of one NMPC autopilot (which incorporates all the objectives of the four NMPC autopilots implemented in this work). This will reveal (and also confirm) the advantages and the challenges that are associated with the use of an accurate (or high fidelity) model (of highly nonlinear, unstable, and multivariable plant) in the NMPC framework.

A classical autopilot of similar detailed design considerations was not available for direct performance comparisons in order to show the performance extents of using the NMPC design schemes explored in this work. Besides the numerous illustrated advantages, including the inherent flexibility and reconfiguration of NMPC, a measure of performance compared to a well established (and tested) benchmark will be an ultimate project objective. The work load of designing and implementing a classical/conventional controller that captures the significant nonlinearities and stability properties of the F-16 (at the same time, exhibiting comparable design and performance objectives), proved to be very extensive. In view of the limited time frame for the master's project work, emphasis was therefore placed on acquiring the high knowledge level required to successfully handle all the aspects of NMPC and aerodynamics involved in this work.

This work also highlighted on optimal reference trajectory generation as an important aspect of the NMPC scheme that determines the overall performance of the autopilots. It was also pointed out that similar response/behavior was observed for the use of rate constraints. A common result is a damping effect introduced into the closed loop system, which enhances the overall stability and reliability of the control system. It will be, therefore, interesting to investigate further on the

relation between the use of rate constraints to enhance (and possibly guarantee) stability in the NMPC framework. A related proposal is the performance/stability enhancing strategy introduced in section 3.3.4, which also offers an avenue for further scrutiny.

A project work that examines the real-time computational/implementation requirements and strategies for the NMPC autopilots can also be mentioned as an interesting proposal for further work.

Using the detailed model of the F-16 aircraft as the target plant made it possible to encounter most of the challenging NMPC implementation issues highlighted in this work. The F-16 model (including the available high/low fidelity data) is, therefore, recommended as a good platform for studying control designs that aim at handling highly nonlinear and unstable system dynamics.

Bibliography

[1] Youmin Zhang and Jin Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, (32):229–252, 2008.

[2] Brian L. Stevens and Frank L. Lewis. *Aircraft Control and Simulation*, chapter 1 to 5, pages 1–508. Wiley, 2003.

[3] Thor I. Fossen. Mathematical models for control of aircraft and satellites, 2011.

[4] Nicholas Swain and Shadhanan Manickavasagar. *Fault Tolerant Flight Control, A Benchmark Challenge*, chapter 14, pages 399–422. SpringerLink, 2010.

[5] Marc Steinberg. A historical overview of research in reconfigurable flight control. In *Proceedings of IMechE, Part G: Journal of Aerospace Engineering*, pages 263–275. IMechE, 2005.

[6] J.M. Maciejowski and C.N. Jones. Mpc fault-tolerant flight control case study: Flight 1862. In *Proceedings of the 5th IFAC symposium on fault detection, supervision and safety for technical processes*, pages 121–126. IFAC, June 2003.

[7] Colin N. Jones and Pembroke College. Reconfigurable flight control, first year report. Technical report, Department of Engineering, University of Cambridge, March 2005.

[8] Thomas Lombaerts, Hafid Smaili, and Jan Breeman. *Fault Tolerant Flight Control, A Benchmark Challenge*, chapter 1, pages 3–4. SpringerLink, 2010.

[9] T. Keviczky and G.J. Balas. Receding horizon control of f-16 aircraft: A comparative study. *Control Engineering Practice*, 14(9):1023–1033, 2006.

[10] Raktim Bhattacharya, Gary J. Balas, M. Alpay Kaya, and Andy Packard. Nonlinear receding horizon control of f-16 aircraft. *Journal of Guidance, Control, and Dynamics*, 25(5):924–931, 2002.

[11] J.M. Maciejowski. *Predictive Control: with constraints*, chapter 10 and appendix A, pages 276–312. Pearson and Prentice Hall, 2002.

[12] D. Kwame Kufoalor. Reconfigurable fault tolerant control of aircraft: Master specialization report. Technical report, Department of Engineering Cybernetics & NTNU, December 2011.

[13] M.M. Kale and A.J. Chipperfield. Reconfigurable flight control strategies using model predictive control. In *Proceedings of the 2002 IEEE International Symposium on Inteligent Control*, pages 43–48, Vancouver, Canada, October 2002. IEEE.

[14] J.M. Maciejowski. *Predictive Control: with constraints*, chapter 1 to 3, pages 1–104. Pearson and Prentice Hall, 2002.

[15] G.F. Franklin, J.D. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Addison-Wesley, 3rd edition, 1994.

[16] Lars Imsland. Introduction to model predictive control. Technical report, Department of Engineering Cybernetics & NTNU, May 2007.

[17] Tor Arne Johansen. Introduction to nonlinear model predictive control and moving horizon estimation. In *Selected Topics on Constrained and Nonlinear Control*, chapter 1, pages 1–29. STU/NTNU, 2011. ISBN 978-80-968627-4-0.

[18] S.J. Qin and T.A. Badgwell. An overview of nonlinear linear model predictive control applications. In *Nonlinear Predictive Control*, pages 369–392. Springer-Verlag, 2000.

[19] Jorge Norcedal and Stephen J. Wright. *Numerical Optimization*, chapter 18, pages 529–560. Springer, 2006.

[20] Frank Allgöwer, Rolf Findeisen, and Zoltan K. Nagy. Nonlinear model predictive control: From theory to application. *J. Chin. Inst. Chem. Engrs*, 35(3):299–315, 2004.

[21] Moritz Diehl, H. George Bock, Johannes P. Schlöder, Rolf Findeisen, Zoltan K. Nagy, and Frank Allgöwer. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *Journal of Process Control*, 12(4):577–585, 2002.

[22] H. Chen and Frank Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1217, 1998.

[23] Fernando A.C.C. Fontes. A general framework to design stabilizing nonlinear model predictive controllers. *Systems and Control Letters*, 42:127–143, 2001.

[24] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

[25] P.O.M. Scokaert, D.Q. Mayne, and J.B. Rawlings. Suboptimal model predictive control (feasibility implies stability). *IEEE Transactions On Automatic Control*, 44(3):648–654, 1999.

[26] L. Magni, G. De Nicolao, L. Magnani, and R. Scattolini. A stabilizing model-based predictive control algorithm for nonlinear systems. *Automatica*, 37:1351–1362, 2001.

[27] Jorge Norcedal and Stephen J. Wright. *Numerical Optimization*, chapter 2, pages 10–27. Springer, 2006.

[28] Guillaume Ducard, Hans P. Geering, and Emil Dumitrescu. Efficient control allocation for fault tolerant embedded systems on small autonomous aircraft. In *1st IEEE Symposium on industrial Embedded systems*. IEEE, 2006.

[29] Thor I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*, chapter 12, pages 398–415. Wiley, 2011.

[30] Ola Härkegård. Dynamic control allocation using constrained quadratic programming. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, Monterey-California, 2002. AIAA.

[31] Bernard Etkin and Lloyd Duff Reid. *Dynamics of Flight: stability and control*, chapter 1 to 3, pages 1–92. Wiley, third edition, 1996.

[32] Donald McLean. *Automatic Flight Control Systems*, chapter 3, pages 63–101. Prentice Hall, 1990.

[33] Hassan K. Khalil. *Nonlinear Systems*, chapter 12, pages 469–504. Prentice Hall, third edition, 2002.

[34] Wilson J. Rugh and Jeff S. Shamma. Research on gain scheduling. *Automatica*, 36:1401–1425, 2000.

[35] Michel Verhaegen, Stoyan Kanev, Reouane Hallouzi, Colin Jones, Jan Maciejowski, and Hafid Smaili. *Fault Tolerant Flight Control, A Benchmark Challenge*, chapter 2, pages 49–50. SpringerLink, 2010.

[36] Luat T. Nguyen, Marilyn E. Ogburn, William P. Gilbert, Kemper S. Kibler, Philip W. Brown, and Perry L. Deal. Simulator study of stall/post-stall characteristics of a fighter airplane with relaxed longitudinal static stability. Technical Report 1538, NASA Scientific and Technical Information Branch, Langley Research Center, Hampton, Virginia, December 1979.

[37] Richard S. Russell. Nonlinear f-16 simulation using simulink and matlab. Technical report, University of Minnesota, June 2003. Available at `http://www.aem.umn.edu/people/faculty/balas/darpa_sec/SEC.Software.html`.

[38] B. Houska, H.J. Ferreau, and M. Diehl. ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization. *Optimal Control Applications and Methods*, 32(3):298–312, 2011.

[39] B. Houska and H.J. Ferreau. ACADO Toolkit User's Manual. `http://www.acadotoolkit.org`, 2009–2011.

[40] D. Ariens, B. Houska, and H.J. Ferreau. Acado for matlab user's manual. `http://www.acadotoolkit.org`, 2010–2011.

[41] Ola Härkegård. *Backstepping and Control Allocation with Applications to Flight Control*. Linköping studies in science and technology. thesis no 820, Division of Automatic Control, Department of Electrical Engineering, Linköping University, SE–581 83 Linköping, Sweden, May 2003.

[42] Jorge Norcedal and Stephen J. Wright. *Numerical Optimization*, chapter 16, pages 448–492. Springer, 2006.

[43] Marc Bodson. Evaluation of optimal methods for control allocation. *Journal of Guidance, Control, and Dynamics*, 25(4):703–711, 2002.

[44] W. Yang, M.N. Hammoudi, G. Herrmann, M. Lowenberg, and X. Chen. Two-state dynamic gain scheduling control applied to an f16 aircraft model. *International Journal of Non-linear Mechanics*, pages 1401–1425, 2011.

[45] Johannes Tjønnås and Tor Arne Johansen. Optimizing adaptive control allocation with actuator dynamics. New Orleans, 2007. CDC.

[46] Alessandro and Emanuele Garone. Adaptive control allocation for fault tolerant overactuated automated vehicles. In *UAV Design Process/Design Criteria for Structures*, pages 3.2–1–3.2–15, France, 2007. RTO-MP-AVT-145.

[47] Ola Härkegård. Efficient active set algorithms for solving constrained least squares problem in aircraft control allocation. Technical Report LiTH-ISY-R-2426, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, May 2002. Available at `http://www.control.isy.liu.se`.

Appendix A

F-16 Aerodynamic Coefficients

The various aerodynamic contributions to a given force or moment coefficient are summed, according to [36], as follows.
For the X-axis force coefficient C_{X_T}:

$$\begin{aligned} C_{X_T} &= C_X(\alpha,\beta,\delta_e) + \delta C_{X_{LEF}}(1-\frac{\delta_{LEF}}{25}) \\ &+ \frac{q\bar{c}}{2V_T}\left[C_{X_q}(\alpha) + \delta C_{X_{q_{LEF}}(\alpha)}(1-\frac{\delta_{LEF}}{25})\right] \end{aligned}$$

where

$$\delta C_{X_{LEF}} = C_{X_{LEF}}(\alpha,\beta) - C_X(\alpha,\beta,\delta_e=0)$$

For the Y-axis force coefficient C_{Y_T}:

$$\begin{aligned} C_{Y_T} &= C_Y(\alpha,\beta) + \delta C_{Y_{LEF}}(1-\frac{\delta_{LEF}}{25}) \\ &+ \left[\delta C_{Y_{\delta_a}} + \delta C_{Y_{\delta_{a_{LEF}}}}(1-\frac{\delta_{LEF}}{25})\right]\left(\frac{\delta_a}{20}\right) \\ &+ \delta C_{Y_{\delta_r}}\left(\frac{\delta_r}{30}\right) + \frac{rb}{2V_T}\left[C_{Y_r}(\alpha) + \delta C_{Y_{r_{LEF}}}(\alpha)(1-\frac{\delta_{LEF}}{25})\right] \\ &+ \frac{pb}{2V_T}\left[C_{Y_p}(\alpha) + \delta C_{Y_{p_{LEF}}(\alpha)}(1-\frac{\delta_{LEF}}{25})\right] \end{aligned}$$

where

$$\begin{aligned} \delta C_{Y_{LEF}} &= C_{Y_{LEF}}(\alpha,\beta) - C_Y(\alpha,\beta) \\ \delta C_{Y_{\delta_a}} &= C_{Y_{\delta_a}}(\alpha,\beta) - C_Y(\alpha,\beta) \\ \delta C_{Y_{\delta_{a_{LEF}}}} &= C_{Y_{\delta_{a_{LEF}}}}(\alpha,\beta) - C_{Y_{LEF}}(\alpha,\beta) - \delta C_{Y_{\delta_a}} \\ \delta C_{Y_{\delta_r}} &= C_{Y_{\delta_r}}(\alpha,\beta) - C_Y(\alpha,\beta) \end{aligned}$$

For the Z-axis force coefficient C_{Z_T}:

$$\begin{aligned} C_{Z_T} &= C_Z(\alpha,\beta,\delta_e) + \delta C_{Z_{LEF}}(1-\frac{\delta_{LEF}}{25}) \\ &+ \frac{q\bar{c}}{2V_T}\left[C_{Z_q}(\alpha) + \delta C_{Z_{q_{LEF}}(\alpha)}(1-\frac{\delta_{LEF}}{25})\right] \end{aligned}$$

where

$$\delta C_{Z_{LEF}} = C_{Z_{LEF}}(\alpha,\beta) - C_Z(\alpha,\beta,\delta_e=0)$$

For the rolling-moment coefficient C_{ℓ_T}:

$$\begin{aligned} C_{\ell_T} &= C_\ell(\alpha,\beta,\delta_e) + \delta C_{\ell_{LEF}}(1-\frac{\delta_{LEF}}{25}) \\ &+ \left[\delta C_{\ell_{\delta_a}} + \delta C_{\ell_{\delta_{a_{LEF}}}}(1-\frac{\delta_{LEF}}{25})\right]\left(\frac{\delta_a}{20}\right) \\ &+ \delta C_{\ell_{\delta_r}}\left(\frac{\delta_r}{30}\right) + \frac{rb}{2V_T}\left[C_{\ell_r}(\alpha) + \delta C_{\ell_{r_{LEF}}}(\alpha)(1-\frac{\delta_{LEF}}{25})\right] \\ &+ \frac{pb}{2V_T}\left[C_{\ell_p}(\alpha) + \delta C_{\ell_{p_{LEF}(\alpha)}}\left(1-\frac{\delta_{LEF}}{25}\right)\right] + \delta C_{\ell_\beta}(\alpha)\beta \end{aligned}$$

where

$$\begin{aligned} \delta C_{\ell_{LEF}} &= C_{\ell_{LEF}}(\alpha,\beta) - C_\ell(\alpha,\beta,\delta_e=0) \\ \delta C_{\ell_{\delta_a}} &= C_{\ell_{\delta_a}}(\alpha,\beta) - C_\ell(\alpha,\beta,\delta_e=0) \\ \delta C_{\ell_{\delta_{a_{LEF}}}} &= C_{\ell_{\delta_{a_{LEF}}}}(\alpha,\beta) - C_{\ell_{LEF}}(\alpha,\beta) - \delta C_{\ell_{\delta_a}} \\ \delta C_{\ell_{\delta_r}} &= C_{\ell_{\delta_r}}(\alpha,\beta) - C_\ell(\alpha,\beta,\delta_e=0) \end{aligned}$$

For the pitching-moment coefficient C_{m_T}:

$$\begin{aligned} C_{m_T} &= C_m(\alpha,\beta,\delta_e)\eta_{\delta_e}(\delta_e) + C_{Z_T}[x_{cg_{ref}} - x_{cg}] + \delta C_{m_{LEF}}(1-\frac{\delta_{LEF}}{25}) \\ &+ \delta C_{m_{sb}}(\alpha)\left(\frac{\delta_{sb}}{60}\right) + \frac{q\bar{c}}{2V_T}\left[C_{m_q}(\alpha) + \delta C_{m_{q_{LEF}(\alpha)}}(1-\frac{\delta_{LEF}}{25})\right] \\ &+ \delta C_m(\alpha) + \delta C_{m_{ds}}(\alpha,\delta_e) \end{aligned}$$

where

$$\delta C_{m_{LEF}} = C_{m_{LEF}}(\alpha,\beta) - C_m(\alpha,\beta,\delta_e=0)$$

For the yawing-moment coefficient C_{n_T}:

$$\begin{aligned} C_{n_T} &= C_n(\alpha,\beta,\delta_e) + \delta C_{n_{LEF}}(1-\frac{\delta_{LEF}}{25}) - C_{Y_T}\left[x_{cg_{ref}} - x_{cg}\right]\frac{\bar{c}}{b} \\ &+ \left[\delta C_{n_{\delta_a}} + \delta C_{n_{\delta_{a_{LEF}}}}(1-\frac{\delta_{LEF}}{25})\right]\left(\frac{\delta_a}{20}\right) \\ &+ \delta C_{n_{\delta_r}}\left(\frac{\delta_r}{30}\right) + \frac{rb}{2V_T}\left[C_{n_r}(\alpha) + \delta C_{n_{r_{LEF}}}(\alpha)(1-\frac{\delta_{LEF}}{25})\right] \\ &+ \frac{pb}{2V_T}\left[C_{n_p}(\alpha) + \delta C_{n_{p_{LEF}(\alpha)}}\left(1-\frac{\delta_{LEF}}{25}\right)\right] + \delta C_{n_\beta}(\alpha)\beta \end{aligned}$$

where

$$\begin{aligned} \delta C_{n_{LEF}} &= C_{n_{LEF}}(\alpha,\beta) - C_n(\alpha,\beta,\delta_e=0) \\ \delta C_{n_{\delta_a}} &= C_{n_{\delta_a}}(\alpha,\beta) - C_n(\alpha,\beta,\delta_e=0) \\ \delta C_{n_{\delta_{a_{LEF}}}} &= C_{n_{\delta_{a_{LEF}}}}(\alpha,\beta) - C_{n_{LEF}}(\alpha,\beta) - \delta C_{n_{\delta_a}} \\ \delta C_{n_{\delta_r}} &= C_{n_{\delta_r}}(\alpha,\beta) - C_n(\alpha,\beta,\delta_e=0) \end{aligned}$$

The low fidelity aerodynamic data does not include leading edge flap. Therefore, all terms multiplied by δ_{LEF} will be zero. The same applies to all higher terms in the total coefficient equations. The $\eta_{\delta_e}(\delta_e)$ must be set equal to 1 when the low fidelity data is used. Refer to [36] for any further details.

Appendix B

Classical Autopilot design

The following sections about autopilot design are based on references [2] and [3].

B.1 Pich-Attitude Hold Autopilot

The controlled variable for pitch control is θ, which is related to the flight-path defined as:

$$\gamma := \theta - \alpha \tag{B.1}$$

Figure B.1 shows a block diagram of an attitude-hold autopilot. The design

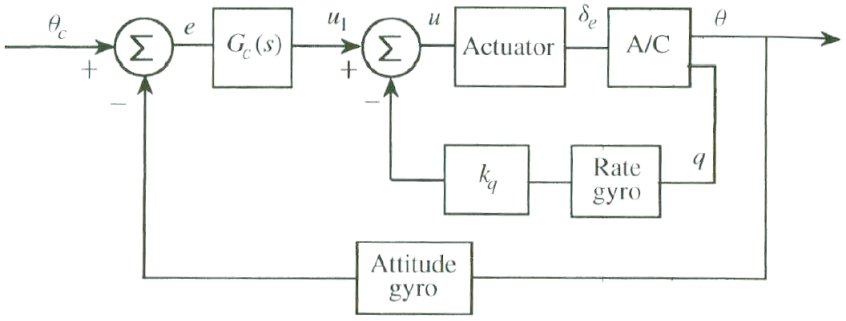


Figure B.1: A pitch-attitude autopilot [2]

includes a *dynamic compensator*, $G_c(s)$, which is necessary if a small steady-state error and good transient response are required. A PI (*Proportional Integral*) compensator can be used to ensure that steady-state pitch error is removed. Because of the very dynamic characteristics of the pitch-attitude hold autopilot, it is most suitable as an inner feedback loop for other autopilots, such as altitude hold.

B.2 Altitude Hold Autopilot

The altitude hold autopilot allows an aircraft to be held at a fixed altitude in an air-route corridor, serving as an important relief to the pilot. This autopilot can also include features that allow climb rate limiting. The feedback loops involved in altitude-hold control are shown in figure B.2. In addition to the compensator G_c, the effective lag of the pressure-altitude measurement is G_F is also included. G_c can be designed with the goal of achieving high loop gain, for good rejection of

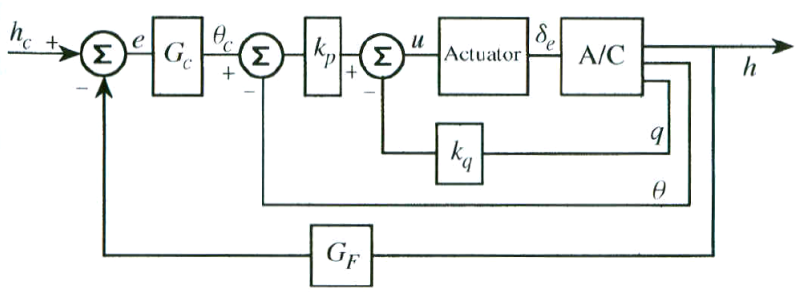


Figure B.2: An altitude-hold autopilot [2]

low-frequency altitude disturbances and small altitude error. It is also desirable that an altitude response is deadbeat and relatively slow for both energy efficiency and passenger comfort.

The equation for altitude control can be derived by considering the equation for vertical acceleration in the center of gravity expressed in NED coordinates, given in [3] as:

$$a_{z_CG} = \dot{W} + VP - QU - g\cos(\Theta)\cos(\Phi) \tag{B.2}$$

Assuming that $\dot{W}_0 = 0$, and using acceleration perturbation according to $a_{z_CG} = a_{z_0} + \delta a_z$, the following equilibrium condition is obtained

$$a_{z_0} = V_0P_0 - Q_0U_0 - g\cos(\Theta_0)\cos(\Phi_0) \tag{B.3}$$

Equation (B.2) can therefore be perturbed as:

$$a_{z_0}+\delta a_z = \dot{w}+(V_0+v)(P_0+p)-(Q_0+q)(U_0+u)-g\cos(\Theta_0+\phi)\cos(\Phi_0+\phi) \tag{B.4}$$

Further assumption that the altitude is changed by symmetric straight-line flight with horizontal wings, such that $V_0 = \Phi_0 = \Theta_0 = P_0 = Q_0 = 0$, gives:

$$a_{z_0} + \delta a_z = \dot{w} + vp - q(U_0 + u) - g\cos(\phi)\cos(\phi) \tag{B.5}$$

Neglecting 2nd-order terms, vp and uq, and subtracting the equilibrium condition (B.3) from (B.5) results in:

$$\delta a_z = \dot{w} - qU_0 \tag{B.6}$$

Differentiating the altitude twice with respect to time gives the relationship:

$$\ddot{h} = -\delta a_z = qU_0 - \dot{w} \tag{B.7}$$

Integrating (B.7) under the assumption that $\dot{h}(0) = U_0\theta(0) - w(0) = 0$, results in:

$$\dot{h} = U_0\theta - w \tag{B.8}$$

which finally yields the differential equation for altitude control:

$$\dot{h} = U_0\gamma \tag{B.9}$$

where γ is the flight path defined in (B.1) and $w = \alpha U_0$ has been used.

B.3 Roll-Angle Hold Autopilot

The roll-angle hold autopilot in a similar manner as the pitch attitude hold autopilot serve as an inner loop for other autopilots that allow the aircraft to fly on a fixed compass heading. The aircraft can however be steered in any direction by a single control if a means of varying the roll reference is provided. The roll-angle hold autopilot can also be used to hold the aircraft wings level, providing a pilot-relief function for long flights. Figure B.3 shows a block diagram of a roll-angle hold autopilot. Roll-rate gyro can be used to provide inner-loop damping for the

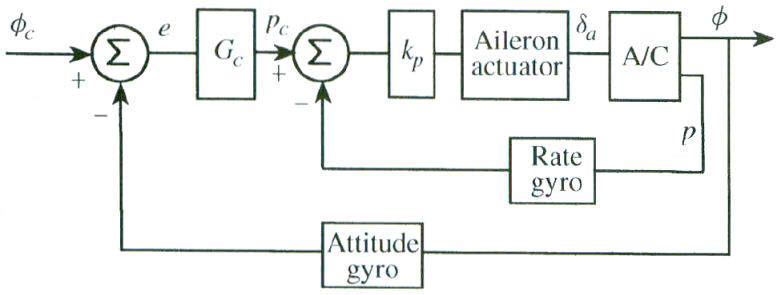


Figure B.3: A roll-angle control system [2]

autopilot, in addition to the compensator G_c placed in the roll-angle error path. There is usually no requirement for precise tracking of roll-angle commands, so a proportional gain for G_c can be used.

B.4 Turn Coordination and Heading-hold Autopilot

Turn coordination is desirable for several reasons among which is passenger comfort, minimizing undesirable aerodynamic loading of the aircraft structure, and enhancement of pilot effectiveness in high performance aircrafts. In coordinated turn, the aircraft maintains the same pitch and roll attitude with respect to the reference coordinate system, but the heading changes continuously at a constant rate. This implies that $\dot{\phi} = \dot{\theta} = 0$, and $\dot{\psi}$ is the turn rate. The acceleration in y-direction is also zero, and sideslip $\beta = \dot{\beta} = 0$. A similar exercise as done in the derivation of the altitude control equation is done to derive the equation for coordinated turn, given in [3] as:

$$\dot{\psi} = \frac{g}{U_0} \sin(\phi) \tag{B.10}$$

where it has been assumed that the aircraft is trimmed such that $\alpha_0 = W_0/U_0 = 0$ and that $V_T = V_0 =$constant. g is the gravitational acceleration. For small angles of ϕ, the equation for coordinated turn (B.10) becomes:

$$\dot{\psi} = \frac{g}{U_0} \phi \tag{B.11}$$

The above result states that a roll angle ϕ different from zero will induce a yaw rate which again turns the aircraft (*bank-to-turn*).

A collection of the above described autopilots can be used for automatic navigation, which is an important autopilot function for both military and civil aircraft.

One such combination is a *heading-hold* autopilot, which was implemented in this project for effective testing of flight control performance in different fault modes. The conventional method of implementing a heading-hold autopilot is to close an additional yaw-angle feedback loop around the roll-angle control system (including turn compensation). Figure B.4 shows the described arrangement. Moreover,

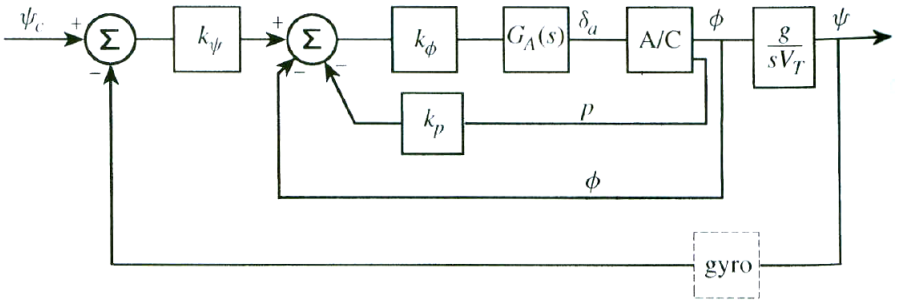


Figure B.4: A heading-hold control system [2]

it can be noted from the block diagram that the turn coordination equation (B.11) is used in the transfer function relating heading angle to roll angle.

B.5 Classical Control and Stability Augmentation system for F-16

The following figures were taken from [36], to illustrate the magnitude of design complexity involved in the implementation of a well functioning control and stability augmentation system for an F-16 fighter aircraft. Refer to [36] for detailed description of all aspects of the complete control system.

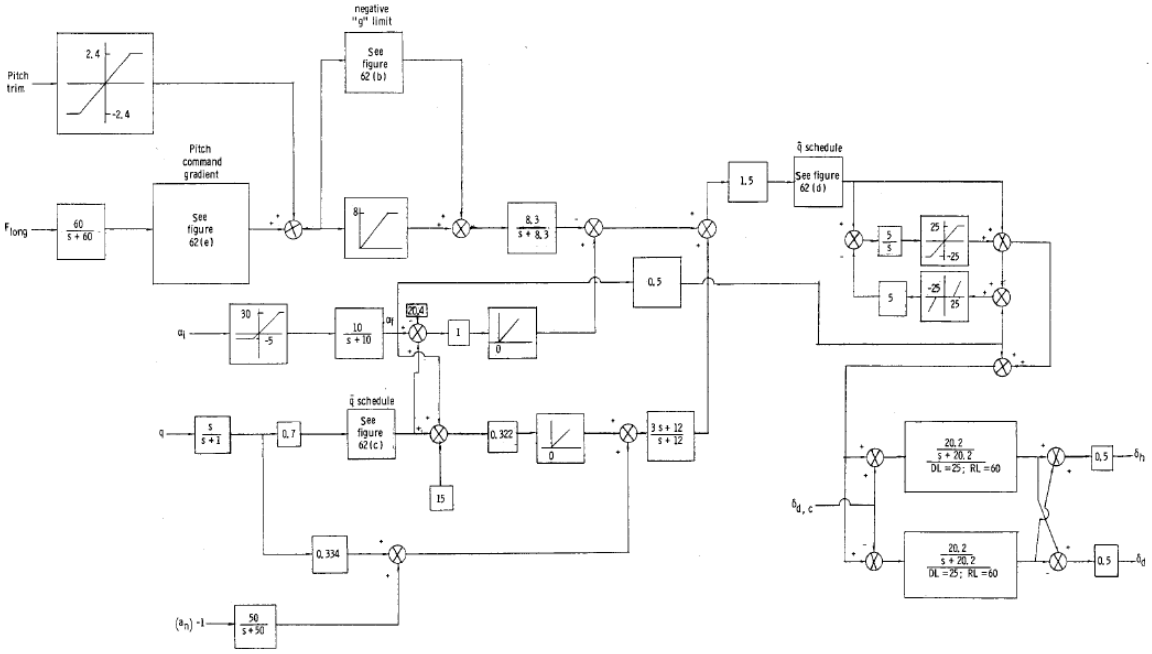


Figure B.5: Pitch control system (longitudinal) [36]

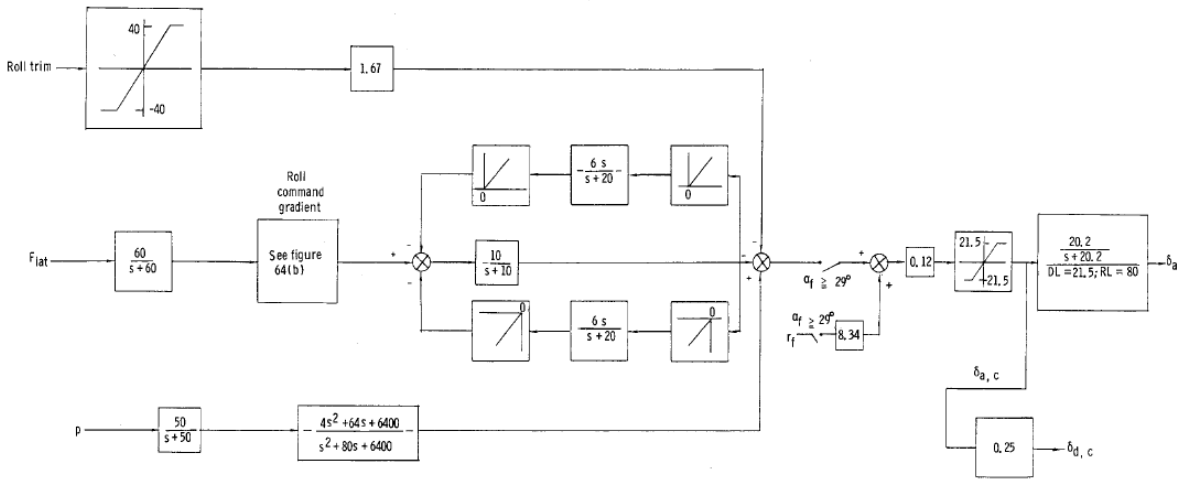


Figure B.6: Roll control system (lateral) [36]

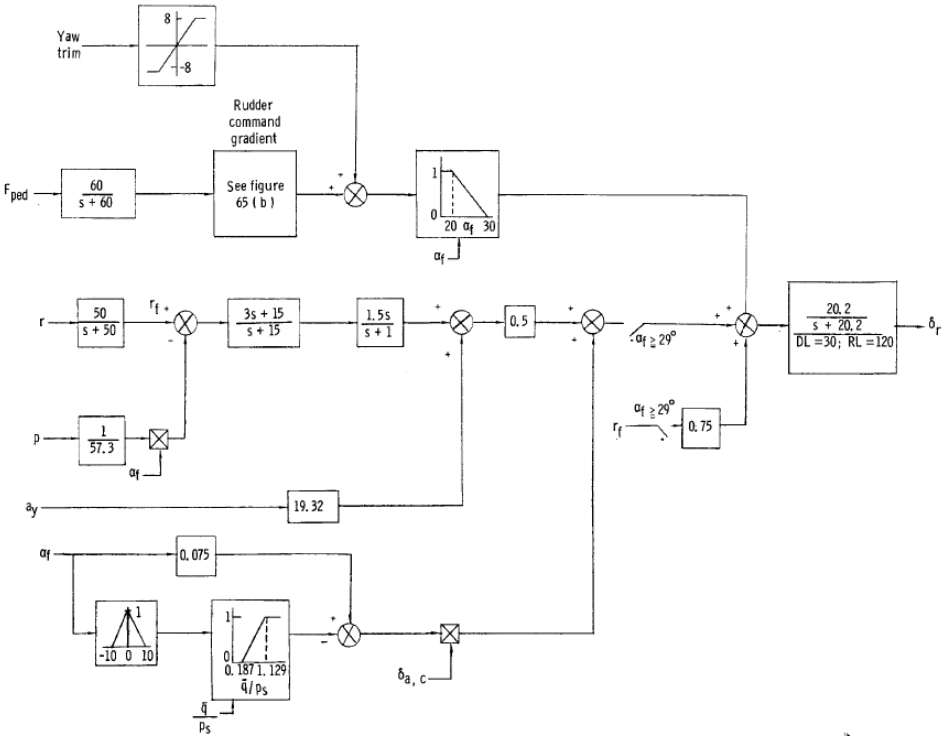
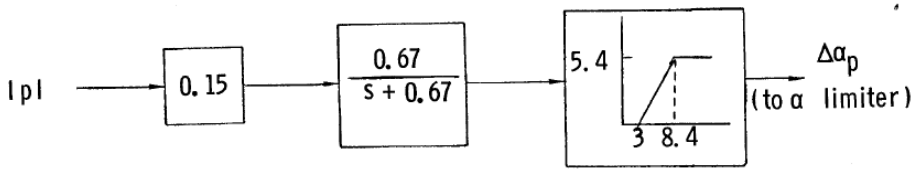
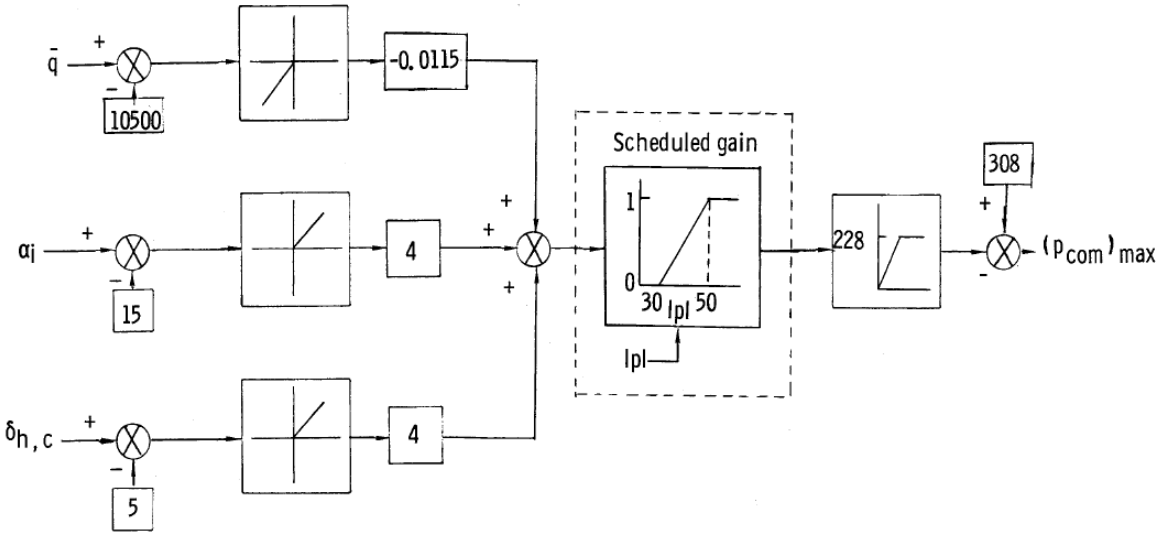


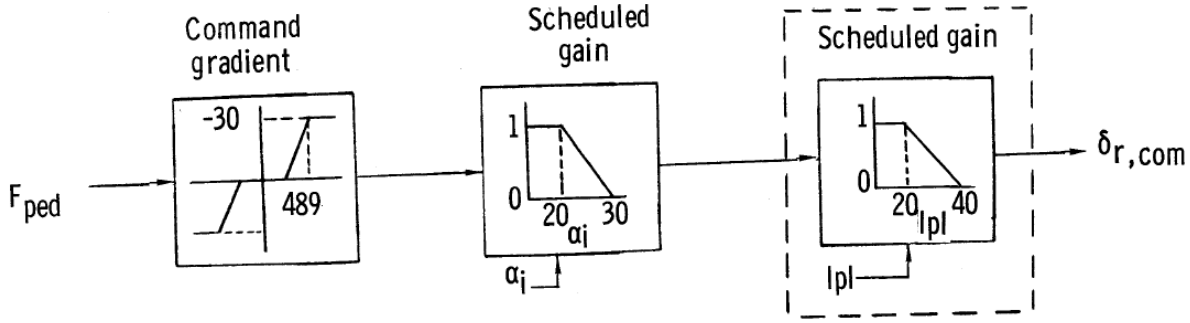
Figure B.7: Yaw control system (directional) [36]



(a) Pitch axis modification



(b) Roll axis modification



(c) Yaw axis modification

Figure B.8: Axes modifications [36]

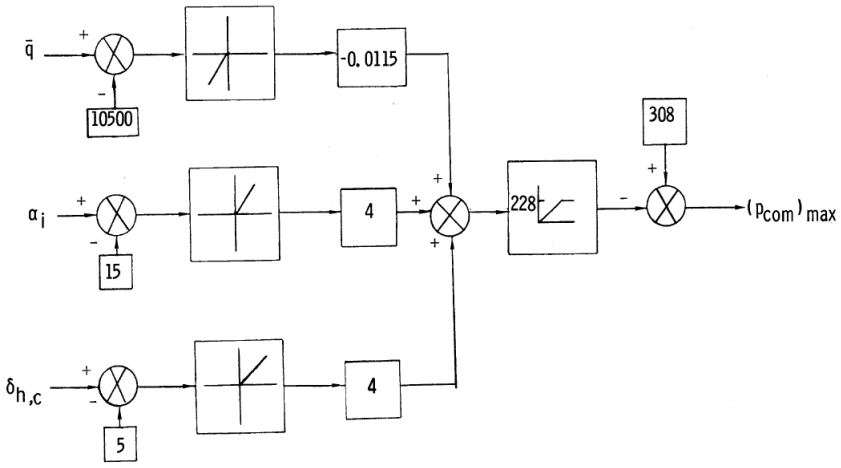
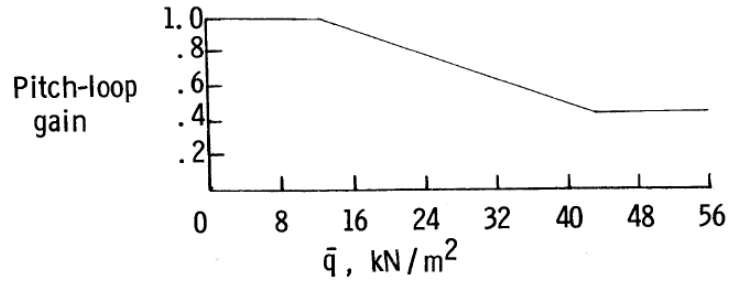
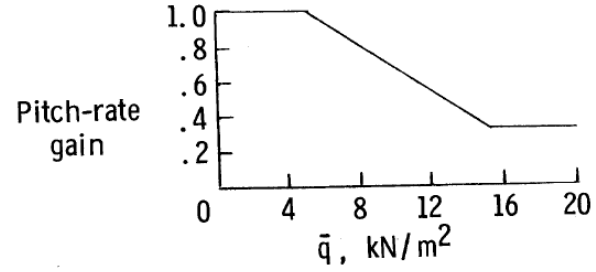


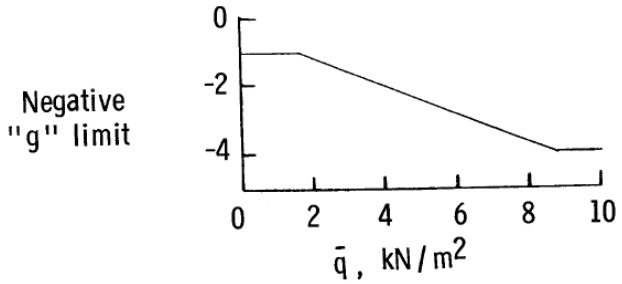
Figure B.9: Role rate limiting scheme [36]



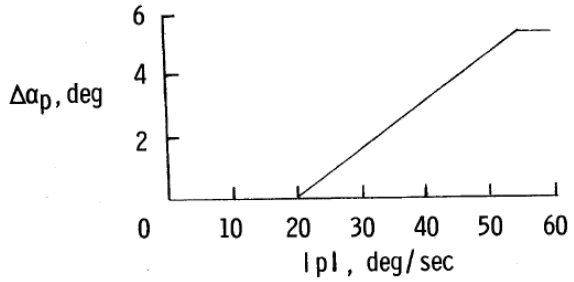
(a) Schedule of pitch-loop gain with $\bar{q}$



(b) Schedule of pitch-rate gain with $\bar{q}$

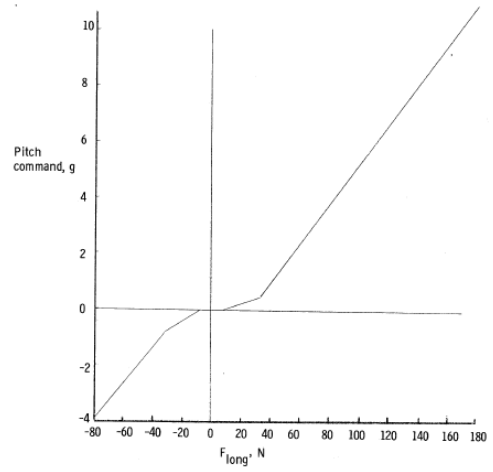


(c) Schedule of negative "g" with $\bar{q}$

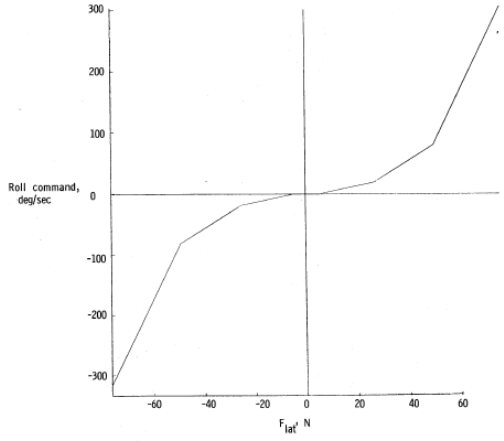


(d) Variation of δ_{α_p}

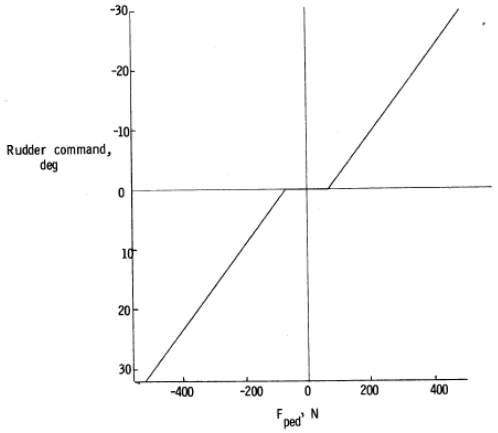
Figure B.10: Scheduling schemes [36]



(a) Pitch command gradient



(b) Roll command gradient



(c) Rudder command gradient

Figure B.11: Command gradients [36]

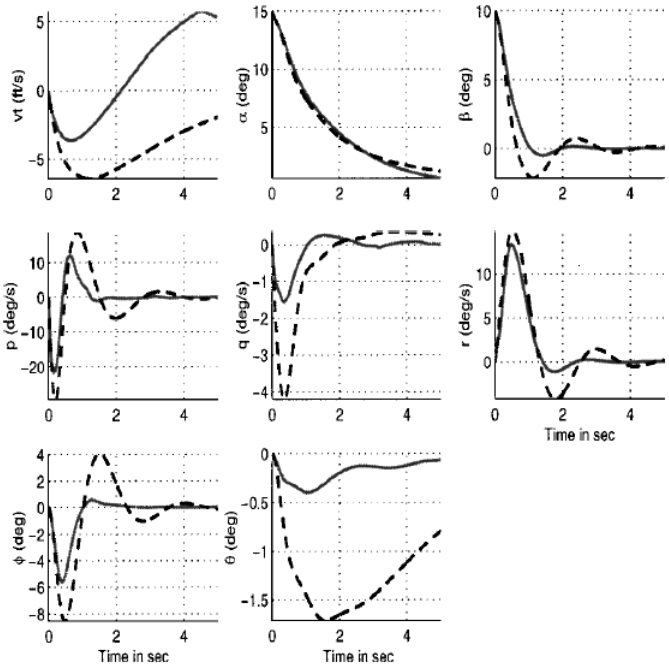
Appendix C

Relevant F-16 NMPC results

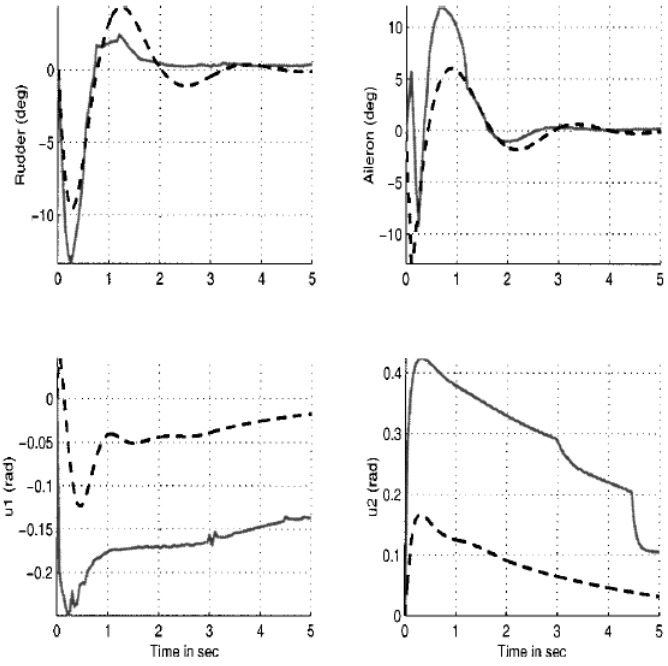
Some results from published work, [10], on Nonlinear Model Predictive Control (NMPC) for the same F-16 model used in this work are included here to present an alternative view on NMPC performance.

The NMPC problem investigated in [10] is a regulation-based problem, and the objective is to regulate state perturbations of the F-16 to a target set (in this case, the origin). Further, a Linear Parameter Varying (LPV) controller was used to obtain a Control Lyapunov Function (CLF), which was used as a terminal cost in the formulation of the optimization problem for NMPC. This approach allowed NMPC stability to be ensured for the short horizon lengths (0.1s and 1.0s) investigated in the paper. The LPV controller was also used for prestabilizing the F-16 and for comparing the performance of the NMPC scheme. A sampling time of 0.05s was used for the NMPC, while 0.01s was used for the LPV controller. The F-16 was trimmed at an altitude of 10000ft and 7° angle of attack. The control signals generated by the LPV and NMPC controllers are synthetic inputs. Refer to the paper for design and simulation details.

The performance of the NMPC scheme and that of the LPV controller are shown in figure C.1. The results show the response of the aircraft when 15° and 10° perturbations are applied to the angle of attack and sideslip, respectively.



(a) State perturbation trajectories



(b) Control perturbation trajectories

Figure C.1: Trajectories for 1.0s horizon length. NMPC (—), LPV (−−) [10]

Appendix D

Optimal CA as a reconfigurble controller

The multiple actuation system (*over actuation*) in modern aircrafts can be well managed through *Control Allocation* (CA). The control allocation problem in a flight control context can be defined as the determination of control effector deflections which generate a given set of desired moments specified by a pilot through the control stick or by a guidance system. In the case of actuator failures, it is desirable to reconfigure the control allocation scheme in order to make the best use of the remaining functional actuators. Among the existing control allocation algorithms, optimization methods based on linear programming or quadratic programming have been identified as having potential to be used for reconfigurable control [43]. The control allocation problem, formulated based on optimization, is presented in section D.1.

D.1 The Optimal Control Allocation Problem

The primary objective of a Control Allocator can be formulated as an optimazation problem where limits on actuator position and deflection rate are included as constraints. An optimal problem formulation is motivated by the fact that the exact solution of the Control Allocation problem might not exist, and that the solution might not be unique [43].

The virtual control input v presented in section 4.5 is restated here as:

$$v_d(t) = Bu(t) \tag{D.1}$$

Given the matrix B and a vector v_d of desired aerodynamic moments, the true control input u can be found such that

$$J = \|W_1(Bu(t) - v_d(t))\|_2^2 + \gamma\|W_2(u(t) - u_d(t))\|_2^2 \tag{D.2}$$

is minimized subject to:

$$u_{min} \leq u(t) \leq u_{max} \tag{D.3}$$

$$\rho_{down} \leq \rho(t) \leq \rho_{up} \tag{D.4}$$

where:

$$\rho(t) = \dot{u}(t) \tag{D.5}$$

The proposed optimization objective solves equation (D.1) for actuator position combinations that minimize 1) deviation from the desired moment and 2) control surface deflection. The objective can alternatively be a combination of the former (error minimization) and either the minimization of drag, wing load, radar signature or any other prefered optimal control objective. The vector $u_d(t)$ represents some prefered position or deflection of the actuators (for example, zero deflections). Penalizing the extent of control surface deflection in this proposal implicitely includes drag minimization as an objective. It is also worth noting that the combined objective problem can be solved faster than several individual minimization problems solved sequentially. An optimal problem with comparably better numerical properties is also achieved by using the proposed formulation [43].

Also, from a fault tolerant system perspective, when an actutor failure occures, the $u_p(t)$ vector can be generated from estimates of nominal actuator positions calculated from position and rate limit constraints corresponding to "healthy" aircraft conditions. Choosing $u_d(t)$ as the previous sampled position of the actuator yields a cost function which penalizes actuator rates as proposed in [30] for dynamic control distribution.

Error minimization is however a vital objective of the fault tolerent control scheme. Dynamically adjusting the design parameter $0 \leq \gamma \leq 1$ in equation (D.2) will give priority to error minimization in the event of control effector failure.

The primary Control Allocation problem (D.1) which is usually regarded as a collection of linear equality constraints is also included in the proposed cost function as part of the error minimization objective. W_1 and W_2 can be designed as diagonal weighting matrices whose (i,i)-entries specifiy the priority or importance of correction measures taken for the ith actuator u_i to reach its desired deflection point, or priority for corrections made to achieve desired moments in roll, pitch and yaw. Specifically, W_1 affects the prioritization among virtual control inputs when (D.1) can not be attained due to actuator constraints. W_2 on the other hand allows for actuator priortization, i.e., which actuators should be used primarily.

For digital implementation purposes, the actuator deflection rate (D.5) can be rewritten as:

$$\rho(t) = \dot{u}(t) = \frac{u(t) - u(t-T)}{T} \tag{D.6}$$

where T is the sampling interval. The deflection rate constraint can be reformulated into a time-varying position constraint by inserting (D.6) into (D.4):

$$\rho_{down} \leq \frac{u(t) - u(t-T)}{T} \leq \rho_{up} \tag{D.7}$$

$$\rho_{down} T \leq u(t) - u(t-T) \leq \rho_{up} T \tag{D.8}$$

$$\Rightarrow \rho_{down} T + u(t-T) \leq u(t) \leq \rho_{up} T + u(t-T) \tag{D.9}$$

The constraints in the Optimal Control Allocation problem can therefore be expressed as actuator position constraints at time t:

$$\max\{u_{min}, \rho_{down}T + u(t-T)\} \leq u(t) \leq \min\{u_{max}, \rho_{up}T + u(t-T)\} \quad \text{(D.10)}$$

The choice of using l_2-norm in the optimal control allocation objective (D.2) is motivated by the fact that the l_2-norm favors the use of effective control inputs [41]. Generally, an l_p-norm gives a measure of how 'good' a solution or an approximation is. The l_p-norm of a vector $u \in \mathbb{R}^m$ is defined as

$$\|u\|_p = \left(\sum_{i=1}^{m} \|u_i\|^p \right)^{1/p} \quad \text{for } 1 \leq p \leq \infty \quad \text{(D.11)}$$

Comparing the use of l_2-norm ($p = 2$) with l_1-norm ($p = 1$), for instance, shows clear advantages for the choice of l_2-norm. The l_2-norm distributes virtual control demand among all the control inputs, while the l_1-norm results in a solution that utilizes as few control inputs as possible to satisfy the virtual control demand. A unique solution to the control allocation objective can be achieved for an l_2-norm case if the weighting matrices W_1 and W_2 are nonsingular. Since the optimal problem formulation using l_2-norm entails the minimization of a strictly convex function over a convex set, solving (D.2) leads to a unique solution. The use of l_1-norm on the other hand does not always lead to a unique solution, even if the weighting matrices are nonsingular [41],[43].

D.2 Solving the Optimal CA problem

The optimization problem formulation (D.2) specifies the control allocation objective or the desired solution, but does not include how the solution is obtained. For l_2-optimal control allocation in general, a number of numerical methods are available, including *Pseudoinverse* methods, *fix-point* methods, *Ellipsoidal Constraints* method, and *active set* methods. In reference [41], active set methods have been identified as a viable choice for solving optimal control allocation problems. The main motivation is that approximation schemes such as pseudoinverse methods and the fix-point method are not guaranteed to find an optimal solution in a finite number of iterations, whereas active set methods can be shown to find the optimal solution in a finite number of iterations [41],[42]. The maximum number of iterations can also be set to reflect the computation time, since in each iteration of active set methods a feasible u^{i+1} is produced that yields a lower value of the objective function than the previous iterate, u^i. This special property makes active set methods more reliable and therefore a good and reasonable choice for a reconfigurable fault tolerant control scheme.

Reference [41] formulates the optimal control allocation problem in a similar form as (D.2), but uses rather large values of γ (as in $\gamma\|W_v(Bu - v)\|_p^p$) to emphasize that $Bu - v$ should be primarily minimized. Based on this formulation the control

allocation objective (D.2) can be rewritten as

$$\|W_1(Bu(t)-v_d(t))\|_2^2+\gamma\|W_2(u(t)-u_d(t))\|_2^2=\left\|\underbrace{\begin{pmatrix}W_1B\\ \gamma^{\frac{1}{2}}W_2\end{pmatrix}}_{A}u-\underbrace{\begin{pmatrix}W_1v_d\\ \gamma^{\frac{1}{2}}W_2u_d\end{pmatrix}}_{b}\right\|_2^2 \tag{D.12}$$

and the *weighted least squares* (WLS) problem to be solved becomes

$$\begin{aligned}u&=\arg\min_u\|Au-b\|\\ u_{min}&\leq u\leq u_{max}\end{aligned} \tag{D.13}$$

Equation (D.13) can be solved using proposed active set method algorithms in [41] or [42]

D.3 Reconfigurable Optimal CA implementation notes

An on-line parameter adjustment strategy is required to enable fault tolerance when optimal control allocation is used as a reconfigurable controller. A proposed and tested controller reconfiguration strategy [12] involves accumulating predefined *damage-weights* for each type of actuator, by running through the following simplified steps:

1. *for each actuator*: check for changes in lower and upper limits
2. if a limit change is detected: set damage-weighted count for that actuator (for calculation of new γ).
3. find percentage/fractional decrease in the limits with respect to the healthy state limits.
4. make corresponding percentage increase of the nominal W element for the failed actuator.
5. make corresponding decrease to W-elements of the remaining control surfaces (which are redundant effectors to the faulty control surface)
6. *after visiting each actuator*: decrease the value of γ using the accumulated *damage-weights* (reduces the importance/priority of control effort minimization)